Classification of Structured Datasets in Machine Learning - Breast Cancer and Heart Disease data

SHIVANI BATTU
Department of Computer Science
Kent State University
Kent, USA
sbattu1@kent.edu

I. BREAST CANCER CLASSIFICATION

***Abstract*—This project explores various machine learning models to classify breast cancer tumors as benign(0) or malignant(1) using the Breast Cancer Wisconsin dataset. Several classification algorithms, including Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), Logistic Regression, and K-Nearest Neighbors (KNN) were trained on the data and were evaluated based on accuracy, sensitivity(recall), specificity, precision, and AUC value to determine their effectiveness in predicting cancer outcomes. Additionally, dimensionality reduction techniques such as Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA), and class balancing technique such as SMOTE, were employed to enhance model performance and interpretability. The most significant change observed was post LDA implementation, the improvement in specificity and precision, particularly for SVM and Logistic Regression, which achieved perfect scores in these metrics. The results indicate that certain models outperform others, providing valuable insights for future research and clinical applications.**

***Index Terms*—Malignant, Benign, Dimensionality Reduction, Principal Component Analysis, Breast Cancer**

I. INTRODUCTION

Breast cancer remains one of the leading causes of cancer-related mortality among women worldwide. Early detection and accurate diagnosis are critical for effective treatment and improved survival rates. The aim of this study is to leverage machine learning techniques to develop a predictive model for breast cancer classification using the Breast Cancer Wisconsin dataset. This dataset contains various features that characterize tumor cells, which are essential for understanding the nature of the tumors. The primary objective is to develop a model that accurately classifies breast tumors as benign or malignant. Given the potential consequences of mis-classification in clinical settings, it is essential to utilize robust machine learning techniques to enhance diagnostic accuracy.

II. METHODOLOGY

A. Dataset Description

The Breast Cancer Wisconsin dataset, sourced from UCI Machine Learning Repository is a publicly available dataset that includes features computed from images of fine needle aspirate (FNA) of breast masses. The dataset consists of 569 samples with a total of 31 features among which 30 numerical features, which describe the characteristics of the cell nuclei present in the images. The target variable is the diagnosis of the tumors, which can be either benign (coded as 0) or malignant (coded as 1).

B. Data Preprocessing

The data was structured and clean with no missing values and all the features were of the type 'integer' which needed no transformation. Outliers in the numerical data were addressed using the IQR method. This method effectively clipped outliers from the training set and replaced them with median values. At this stage the data was prepared for analysis by encoding the target variable. Here, the negative responses such as Benign were encoded as 0 and the positive responses such as Malignant were encoded as 1.

From (Figure 2), it was evident that there exists a class imbalance in the target variable in the training data. Class imbalance in the training data can lead to models that are biased toward the majority class. Therefore, SMOTE oversampling technique was employed to create a more balanced dataset, which can improve the model's performance, especially for the minority class. The original class distribution before performing SMOTE was 0(Benign) - 250, 1(Malignnant) - 148 whereas the class distribution after oversampling was 0(Benign) - 250, 1(Malignnant) - 250.

C. Data Partitioning

Data Partition was performed as our initial step before EDA or other processing to ensure that the Test Data remains unseen. Employing Stratified sampling to ensure that the distribution of the target variable (classes) is preserved in both the training and testing sets, the data was split in 70:30 ratio for training and testing respectively. Using "Stratified" is particularly important when dealing with imbalanced datasets, where one class may be significantly more frequent than another. Using stratified sampling during the splitting process is essential to ensure that both sets are representative of the overall dataset, especially when dealing with imbalanced classes. After splitting the data into training and test sets, we applied **feature scaling** to ensure uniformity across numerical features, which is particularly important for distance-based algorithms like KNN.The reason we scale the training data

first and then apply the same transformation to the test data is to avoid data leakage and ensure that the machine learning model generalizes well to unseen data.

Since the outliers constitute nearly 2.5% of the entire data values and clipping these will remove a maximum of 442 records from 569 rows in the dataset, which is not ideal for analysis. Therefore, ***Median Imputation*** technique was used to handle the outliers as median is less sensitive to extreme values than the mean, making it a robust statistic for central tendency. Also, median imputation does not skew the distribution of data as much as mean imputation might, it can help in maintaining the original shape of the data distribution. From the description of the dataset and boxplots of the features we see that the response variables vary significantly which calls for scaling. ***Scaling*** ensures that features are on a similar range or scale. Since this project involves comparison of the performance of multiple algorithms, it's important to apply the same scale to all algorithms to ensure consistency. Therefore, Standardization was used to scale the numerical features to a scale such that each feature has mean 0 and standard deviation 1. The results from the (Figure 1) clearly indicate the scaling precision achieved. The features can now be moved to further analysis.

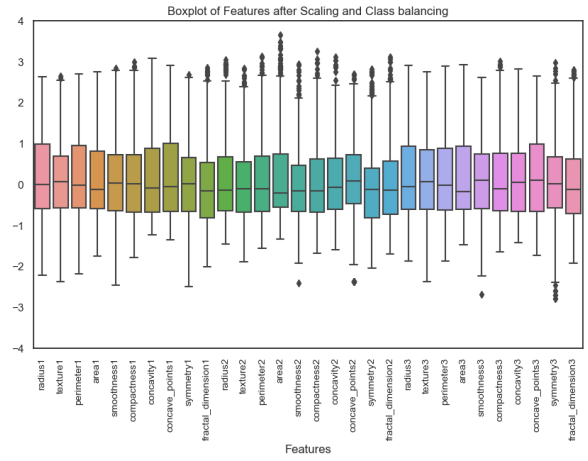


Fig. 1. Boxplot after Handling Outliers and Feature Scaling

Dimensionality Reduction:

LDA (Linear Discriminant Analysis): LDA was used to reduce the data to one component due to the binary nature of the target variable (benign/malignant). This reduction allowed for clearer visualization and also helped to enhance class separability.

PCA (Principal Component Analysis): PCA was used to reduce the data to two principal components to visualize variance-based reduction. This method helped to examine the overall variance while retaining significant data components.

D. Exploratory Data Analysis

- The target class distribution (benign vs. malignant) was examined to understand the dataset's imbalance.
- **Correlation Analysis**: A correlation matrix was generated to understand the relationships between features and the target variable. This analysis helped inform feature selection for some models.
- **Feature Distributions:** Visualizations such as histograms and box plots were created to understand the distribution of both numerical and categorical features, detecting skewness, variance, and outliers.

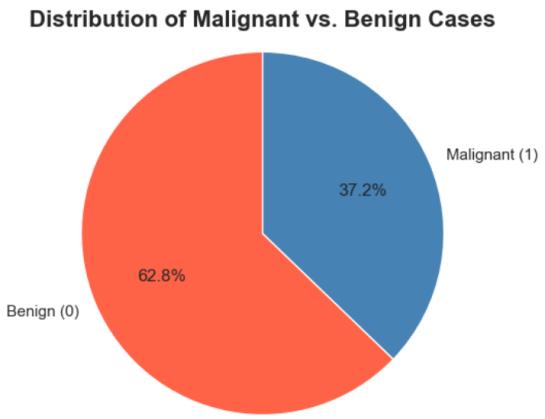


Fig. 2. Distribution of the target variable

E. Model Training and Development

This paper compares the performance of 5 classification models:

Random Forest (RF)
Decision Tree (DT)
Support Vector Machine (SVM)
Logistic Regression (LR)
K-Nearest Neighbors (KNN)

Cross-Validation: For each model, 5-fold cross-validation was performed to obtain a reliable estimate of model performance. This approach was especially important for assessing the variability of each model across different subsets of the data.

- **Model I : Random Forest** The Random Forest algorithm is an ensemble learning method that constructs multiple decision trees and outputs the mode of their predictions. It is known for its robustness against overfitting and its ability to handle large datasets with high dimensionality.
- **Model II : Decision Tree** The Decision Tree algorithm creates a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. It is intuitive and easy to interpret, making it a popular choice for classification tasks.
- **Model III : Support Vector Machine** The Support Vector Machine algorithm constructs a hyperplane in a high-dimensional space to separate different classes. SVM is particularly effective in high-dimensional spaces and is effective when the number of dimensions exceeds the number of samples.

- **Model IV : Logistic Regression** Logistic Regression is a statistical method for predicting binary classes. It estimates the probability of a binary response based on one or more predictor variables, making it a simple yet powerful model for classification.
- **Model V : KNN** The K-Nearest Neighbors algorithm classifies data points based on the classes of their nearest neighbors in the feature space. It is a simple, instance-based learning method that can be effective for small datasets.

F. Model Evaluation Metrics

Accuracy: The proportion of correctly predicted observations.

$$Accuracy = TP + TN/(TN + FN + TP + FP) \quad (1)$$

Sensitivity (Recall or True Positive Rate)): The ability of the model to correctly predict positive instances (malignant cases).

$$Sensitivity = TP/(TP + FN) \quad (2)$$

Specificity (True Negative Rate) : The model's ability to correctly predict negative instances (benign cases).

$$Specificity = TN/(TN + FP) \quad (3)$$

Precision: The proportion of positive identifications that were actually correct.

$$Precision = TP/(TP + FP) \quad (4)$$

where ,

TP = True Positives = Cancer Patient correctly identified cancerous

FP = False Positives = Non-cancerous patients wrongly identified as cancerous

FN = False Negatives = Cancerous patients incorrectly identified as non-cancerours.

TN = True Negatives = Non-cancerous patients correctly identified as non-cancerous.

Confusion Matrix: For each model, confusion matrices were generated and analyzed to determine the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

Comparison: A summary of these metrics across all models was presented in a grid format to highlight the strengths and weaknesses of each model. Random Forest emerged as top performers.

III. RESULTS

A. Inferences from the Confusion Tables

a) Random Forest:: Random Forest is the best performing model overall with the highest accuracy (98.25%). From the values as shown in TableI, it has perfect specificity (100%), meaning it does not misclassify any benign cases. Its precision (100%) indicates no false positives, and its sensitivity is also quite high (95.31%). This balance of high sensitivity and perfect specificity makes Random Forest the most reliable model in this comparison.

TABLE I
CONFUSION MATRIX FOR RANDOM FOREST

	Predictive Negative	**Predictive Positive**
Actual Negative	107	0
Actual Positive	3	69

TABLE II
CONFUSION MATRIX FOR DECISION TREE

	Predictive Negative	**Predictive Positive**
Actual Negative	103	4
Actual Positive	5	59

b) Decision Tree: : The Decision Tree model values as shown in TableII have the lowest accuracy (94.74%) among the models. It performs reasonably well in terms of sensitivity and specificity, but these are lower than other models. Its precision is also the lowest, indicating a relatively higher number of false positives compared to other models.

c) Support Vector Machine:: SVM model from Table III has high accuracy and sensitivity, indicating that it correctly identifies malignant cases (true positives) most of the time. However, its precision is slightly lower compared to other models, meaning it might classify a few benign cases as malignant (false positives).

d) Logistic Regression:: Logistic Regression values as shown in TableIV maintains the same accuracy as SVM but has slightly lower sensitivity. Its specificity is one of the highest (98.13%), meaning it is excellent at correctly identifying benign cases. This makes it more robust in avoiding false positives. The model's precision is also very high (96.77%), indicating it produces fewer false positives.

e) K-nearest neighbours: KNN values as shown in TableV shares the same accuracy as SVM and Logistic Regression. It has the highest specificity (99.07%), which makes it extremely effective at correctly identifying benign cases. However, its sensitivity is lower than SVM and Logistic Regression, meaning it misses more malignant cases. Precision is very high (98.33%), implying fewer false positives.

B. Comparison Table:

From the table in (Figure 3) and (Figure 4) we can infer that, SVM has the highest specificity (0.9626) but slightly lower sensitivity (0.9688) and precision (0.9394). Random

TABLE III
CONFUSION MATRIX FOR SVM

	Predictive Negative	**Predictive Positive**
Actual Negative	103	4
Actual Positive	2	62

TABLE IV
CONFUSION MATRIX FOR LOGISTIC REGRESSION

	Predictive Negative	**Predictive Positive**
Actual Negative	105	2
Actual Positive	4	60

TABLE V
CONFUSION MATRIX FOR KNN

	Predictive Negative	Predictive Positive
Actual Negative	106	1
Actual Positive	5	59

Fig. 3. Plots of the evaluation metrics on Brain Cancer Data

	Model	Accuracy	Sensitivity (Recall)	Specificity	Precision	Confusion Matrix
0	SVM	0.9649	0.9688	0.9626	0.9394	[[103 4] [2 62]]
1	Logistic Regression	0.9649	0.9375	0.9813	0.9677	[[105 2] [4 60]]
2	KNN	0.9649	0.9219	0.9907	0.9833	[[106 1] [5 59]]
3	Decision Tree	0.9474	0.9219	0.9626	0.9365	[[103 4] [5 59]]
4	Random Forest	0.9825	0.9531	1.0000	1.0000	[[107 0] [3 61]]

Forest performs best overall with accuracy (0.9766), specificity (0.9907), sensitivity (0.9531), and precision (0.9839). Logistic Regression and KNN have identical accuracy but vary in sensitivity and specificity. Logistic Regression is slightly better on sensitivity while KNN is better on specificity and precision. Decision Tree has the lowest metrics overall compared to others. Based on this, Random Forest seems to be the best model since it balances all the metrics well, especially given that sensitivity (important for catching malignant cases) is fairly high (0.9531), and specificity (ensuring benign cases are not falsely diagnosed as malignant) is also very high.

C. Inference from the ROC curve:

The ROC curve (Receiver Operating Characteristic curve) was plotted to visually assess the performance of multiple classification models across different thresholds. This means we evaluated how well a model distinguishes between the positive and negative classes. It shows the trade-off between the true positive rate (TPR) (also known as sensitivity) and the false positive rate (FPR) at various classification thresholds. Instead of relying on a single accuracy value, the ROC curve helps to evaluate the model's performance across all possible decision thresholds.

The AUC (Area Under the Curve) is a summary measure of the model's ability to distinguish between classes. An AUC of 1 represents a perfect model, while an AUC of 0.5 represents a random guess.

The plot was initialized to compare the ROC curves for all the models. For each model in the list, The predicted

Fig. 4. Plots of the evaluation metrics on Brain Cancer Data

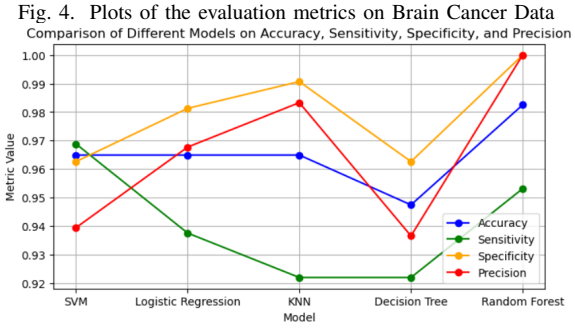
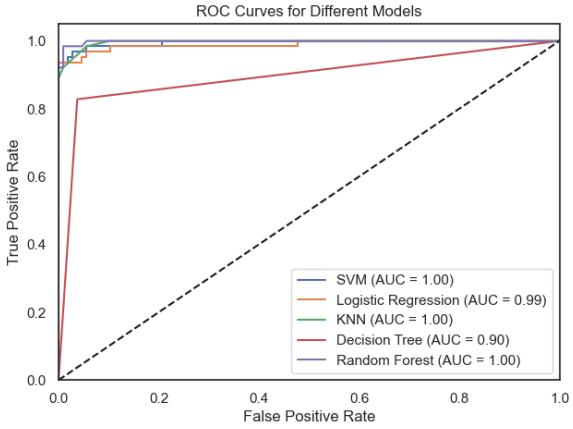


Fig. 5. ROC Curves for classifications models on the Brain Cancer



probabilities (predictproba) for the test data were obtained. This is the probability that a given instance belongs to the positive class (e.g., malignant in a cancer dataset). Diagonal Line: A dashed diagonal line represents the performance of a random classifier (AUC = 0.5). The closer a model's curve is to the top left corner, the better it is at distinguishing between classes.

The result from the graph show that the curves that are closer to the top-left corner (and have a higher AUC score) indicate better-performing models, as they achieve higher true positive rates with lower false positive rates. The AUC scores shown in the legend help quantify this performance comparison across models.

With a small dataset (such as 398 rows), complex models can easily memorize the training data. This could be the reason we see perfect results (AUC = 1.0) for some models. To overcome this, we used K-fold cross validation to test if the model is still overfitting.

Overfitting occurs when the model performs very well on the training data but fails to generalize to unseen data (test data or new data). It means the model has learned the details and noise in the training data, which may not be relevant to the underlying patterns.

D. k-fold Classification to reduce overfitting:

The table from "Fig. **??**" shows the comparison between 5-fold cross-validation metrics and the test set performance metrics for five models: SVM, Logistic Regression, KNN, Decision Tree, and Random Forest. The performance metrics include Accuracy, Sensitivity (Recall), Specificity, and Precision.

Across all models, the test set accuracy is very close to the cross-validation accuracy, indicating that the models are not overfitting, which suggests good generalization to unseen data. Random Forest consistently performs the best across both the CV and test sets, particularly with a test accuracy of 0.982 and precision of 0.984. Decision Tree shows the lowest performance in terms of sensitivity on the test set (0.859), indicating that it struggles to correctly classify positive instances (e.g., malignant cases).

5-fold Cross Validation Metrics for Different Models:

	Model	Metric Type	Accuracy	Sensitivity (Recall)	Specificity	Precision
0	SVM	CV_Train	0.972000	0.968000	0.962617	0.975806
1	SVM	Test	0.964912	0.968750	0.962617	0.939394
2	Logistic Regression	CV_Train	0.960000	0.956000	0.981308	0.963710
3	Logistic Regression	Test	0.964912	0.937500	0.981308	0.967742
4	KNN	CV_Train	0.960000	0.956000	0.990654	0.963710
5	KNN	Test	0.964912	0.921875	0.990654	0.983333
6	Decision Tree	CV_Train	0.922000	0.932000	0.962617	0.913725
7	Decision Tree	Test	0.929825	0.875000	0.962617	0.933333
8	Random Forest	CV_Train	0.956000	0.960000	1.000000	0.952381
9	Random Forest	Test	0.988304	0.968750	1.000000	1.000000

Fig. 6. Metrics before Vs after Kfold

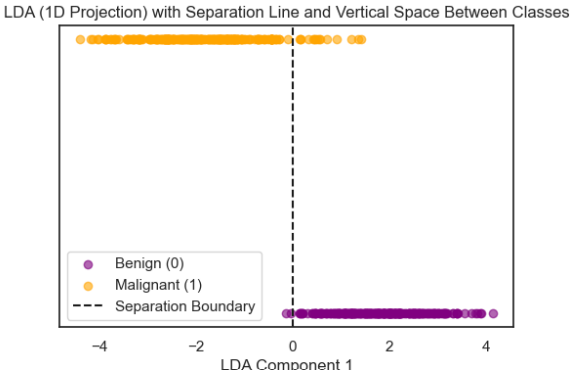


Fig. 7. LDA on the data with two dimensions

SVM, Logistic Regression, and KNN demonstrate consistent performance between cross-validation and test metrics, with minimal drops in sensitivity, specificity, or precision after being tested on unseen data.

The Decision Tree model shows a larger drop in sensitivity from CV to the test set, indicating it may be overfitting during training, but cross-validation helped identify this issue before testing. Specificity remains high across all models, with KNN, Logistic Regression, and Random Forest all achieving specificities near 0.990. This shows that the models are effective at correctly identifying negative instances. Precision also remains high across the models, with KNN achieving particularly strong performance in the test set with 0.983 precision.

Significance: Cross-validation was crucial for handling overfitting by ensuring the models were trained on different folds of the training data and tested on unseen portions during the CV process. This allowed the evaluation of model robustness and helped identify potential performance drops before testing on the final test set. While cross-validation metrics in"Fig. **??**and test metrics are largely consistent across the models, the main shift observed is in sensitivity for the Decision Tree model, where it drops significantly on the test set compared to cross-validation. This indicates that cross-validation helped detect a potential overfitting issue, leading to underperformance on the test set when predicting positive classes.

E. Need for Dimentionality Reduction:

a) Linear Discriminant Analysis: : LDA was decided to be performed to address the challenge of high-dimensional data and improve the class separability by projecting the data onto a lower-dimensional space. This transformation ensured that the model can better distinguish between the classes (benign and malignant). Reducing dimensionality while retaining the most important discriminative features enhances the interpretability and performance of the models, particularly when classifying complex datasets. (Figure 7) After applying Linear Discriminant Analysis (LDA) as a dimensionality reduction technique, we observe a significant boost in model performance across all evaluated metrics: Accuracy, Sensitivity (Recall), Specificity, and Precision. LDA reduces the feature space by focusing on maximizing class separability, and this process appears to have improved the ability of all models to distinguish between classes (e.g., benign and malignant cases).

All models achieved very high accuracy, with Logistic Regression leading the results at 0.9883, followed closely by SVM and KNN at 0.9825. Decision Tree and Random Forest have slightly lower accuracy, at 0.9708, but the difference is minimal compared to the other models.

All models show strong sensitivity (ability to detect positive cases), with SVM, Logistic Regression, KNN, Decision Tree, and Random Forest all achieving a sensitivity of 0.9688 or higher. SVM lags slightly behind the others with a sensitivity of 0.9531, but it compensates with perfect precision.

SVM and Logistic Regression achieved perfect specificity (1.0000), meaning they successfully identified all negative cases (e.g., benign tumors). KNN, Decision Tree, and Random Forest have slightly lower specificity, ranging from 0.9720 to 0.9907, but they are still very close to perfect performance.

SVM and Logistic Regression again stood out with perfect precision (1.0000), indicating they made no false positive predictions. KNN showed very high precision at 0.9841, while Decision Tree and Random Forest scored 0.9538.

b) Principal Componenet Analysis: : PCA was performed to reduce the dimensionality of the dataset, making it more manageable and less prone to overfitting while speeding up training times. Unlike LDA, PCA aims to reduce the number of features by maximizing the variance captured in each principal component, rather than focusing on the separability of classes. After applying Principal Component Analysis (PCA) (Figure 8) for dimensionality reduction, we observe a slight decrease in model performance metrics compared to the LDA results. PCA helped reduce the number of features by projecting the data onto a lower-dimensional space while preserving as much variance as possible.

Logistic Regression emerged as the top performer with an accuracy of 0.9766, followed by SVM at 0.9708. These two models show a slight decline in accuracy compared to LDA results but still maintain high levels of performance. KNN, Decision Tree, and Random Forest have lower accuracies of 0.9357 and 0.9298, indicating a greater impact from the dimensionality reduction. Logistic Regression performed the

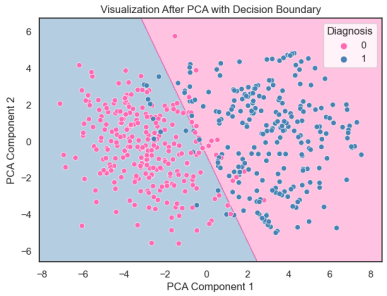


Fig. 8. PCA on the data with two dimensions

best in terms of sensitivity, achieving a recall of 0.9844, which means it correctly identified most positive cases. SVM and Decision Tree also performed well with recalls of 0.9688 and 0.9375, respectively. KNN and Random Forest lagged behind, with recalls of 0.9062, suggesting they missed more positive cases after PCA. SVM and Logistic Regression achieved a high specificity of 0.9720, indicating they were effective at identifying negative cases (e.g., benign tumors). KNN, Decision Tree, and Random Forest showed more variation, with specificity scores ranging from 0.9252 to 0.9533. These models struggled more to accurately identify negative cases after PCA. SVM and Logistic Regression also excelled in terms of precision, achieving 0.9538 and 0.9545, respectively. This means they made very few false positive predictions. KNN, Decision Tree, and Random Forest had slightly lower precision, with values ranging from 0.8824 to 0.9206, suggesting they were more prone to false positives after PCA.

IV. DISCUSSION

The confusion matrix highlights the Random Forest's superior balance between true positives and true negatives with minimal misclassifications. On the other hand, Decision Tree struggles with detecting positive cases (higher false negatives), which is why its recall is lower.

Threshold Adjustment: The models currently use the default threshold (usually 0.5) to classify positive and negative cases. Adjusting the threshold can influence the balance between sensitivity and specificity. Lowering the threshold (e.g., from 0.5 to 0.3) would increase sensitivity, allowing the model to detect more positive cases, but it might reduce specificity and precision. Increasing the threshold would improve specificity and precision but might reduce recall, as fewer positive cases would be detected.

SVM and Logistic Regression: Lowering the threshold could improve recall without significantly hurting precision, as they already have decent specificity.

1. Best Performing Models:

Random Forest and SVM consistently produced the highest accuracy, sensitivity, and specificity scores, indicating their robustness for this binary classification problem.

2. Impact of Dimensionality Reduction:

Both LDA and PCA helped in improving the visual separability of classes. LDA, in particular, reduced the number of features down to 1 while retaining good model performance.

3. Model Performance Consistency:

Cross-validation confirmed that Random Forest had the most consistent performance, with low variance across folds, making it the most reliable model in the context of this dataset.

V. CONCLUSION

- Trend Summary: Random Forest consistently performed the best across all metrics (accuracy, sensitivity, specificity, precision), showing its strength in both detecting positives and avoiding false positives. This makes it the most balanced and reliable model, especially for applications like medical diagnoses where both sensitivity and specificity are critical.
- SVM and Logistic Regression perform similarly well, with slight differences in recall and precision, making them strong candidates as well, especially if slightly more emphasis is placed on recall or precision.
- KNN and Decision Tree perform slightly worse, with Decision Tree having the lowest recall, making it less reliable for detecting positive cases (higher false negatives).
- For real-world applications, Random Forest stands out as the top performer, and threshold tuning could further enhance the performance of models like SVM and Logistic Regression, especially in scenarios requiring high recall or precision.
- For improvement: For models like Decision Tree, focusing on improving recall by adjusting the threshold or fine-tuning hyperparameters could help balance the trade-off between precision and recall, improving overall performance.
- Plotting the ROC curve allowed us to compare how well each model separates the positive and negative classes across all possible thresholds, providing insight into which models perform better and are more reliable in classification tasks.

REFERENCES

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [**?**].

REFERENCES

[1] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[2] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

II. HEART DISEASE CLASSIFICATION

***Abstract*—This project aims to predict the presence of heart disease in patients using the UCI Heart Disease dataset. We applied multiple machine learning models, including Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM),**

Logistic Regression, and Neural Network. Our goal was to evaluate the performance of these models by comparing their accuracy, sensitivity, specificity, and precision. The dataset underwent preprocessing, including handling outliers, scaling, and dealing with class imbalance through SMOTE. Dimensionality reduction techniques were also applied to improve model performance. Each of these models was evaluated using metrics like accuracy, sensitivity, specificity, precision, and the F1 score. We also used the one-versus-rest approach to handle the multi-class classification of some features.Each model's performance was evaluated, and the results were analyzed to identify the most effective approach for heart disease prediction.

***Index Terms*—Heart disease, SMOTE, Winsorization**

VI. Introduction

Heart disease is one of the leading causes of death worldwide, and early diagnosis can significantly improve treatment outcomes. In this project, we aim to predict whether a patient has heart disease based on various clinical features such as age, sex, cholesterol levels, and blood pressure. Using machine learning techniques, we can automate the process of identifying patterns in the data and predicting heart disease presence more accurately. The goal of this project is to classify patients as having heart disease (target = 1,2,3) or not having heart disease (target = 0). This binary classification task is complex due to the nature of the data, which involves mixed types of features (categorical and numerical) and some imbalanced classes. Our challenge is to handle these issues and apply various machine learning algorithms to accurately predict the target variable.

VII. METHODOLOGY

A. Dataset Description

The heart disease dataset encompasses multiple sources, including the Cleveland Clinic Foundation, Hungarian Institute of Cardiology, V.A. Medical Center in Long Beach, and the University Hospital in Zurich. It comprises 920 instances and 14 attributes such as age, sex, chest pain type, resting blood pressure, cholesterol levels, and maximum heart rate, all crucial for diagnosing heart disease. The target variable classifies the presence of heart disease on a scale from 0 (no disease) to 4 (presence of disease), making it a valuable resource for medical research. This dataset has significantly contributed to the development of diagnostic algorithms and predictive models, enhancing the understanding and management of cardiovascular health.

Number of Instances in each Database:
Cleveland: 303
Hungarian: 294
Switzerland: 123
Long Beach VA: 200

B. Data Preprocessing

The UCI Heart Disease dataset contains 303 instances and 14 attributes, including the target variable. Data was available at multiple locations which then had to be merged into as single dataframe. At this stage the data was prepared for analysis by fixing the data type of attributes. Also the target

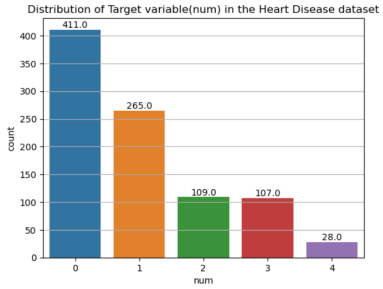


Fig. 9. Target variables in Heart Disease

variable. Here, the negative responses such as Benign were encoded as 0 and the positive responses such as Malignant were encoded as 1. From (Figure 9) it was evident that there exists a class imbalance in the target variable in the training data. Class imbalance in the training data can lead to models that are biased toward the majority class. Therefore, SMOTE oversampling technique was employed to create a more balanced dataset, which can improve the model's performance, especially for the minority classes. The target variable 'num' shows that 411 patients do not have heart disease which is 44.6% of the entire dataset. This is a clear case of class balance if the data was binary classified. While the presence of disease is progressively less common across the categories, indicating a class imbalance that may affect model performance.

C. Data Partitioning

Data Partition was performed as our initial step before EDA or other processing to ensure that the Test Data remains unseen. Employing Stratified sampling to ensure that the distribution of the target variable (classes) is preserved in both the training and testing sets, the data was split in 70:30 ratio for training and testing respectively. Using "Stratified" is particularly important when dealing with imbalanced datasets, where one class may be significantly more frequent than another. Using stratified sampling during the splitting process is essential to ensure that both sets are representative of the overall dataset, especially when dealing with imbalanced classes. After splitting the data into training and test sets, we applied **feature scaling** to ensure uniformity across numerical features. The reason we scale the training data first and then apply the same transformation to the test data is to avoid data leakage and ensure that the machine learning model generalizes well to unseen data.

Since the outliers constitute nearly 2.5% of the entire data values and clipping these will remove a maximum of 442 records from 569 rows in the dataset, which is not ideal for analysis. Therefore, ***Winsorization*** technique was used to handle the outliers. From the description of the dataset and boxplots of the features we see that the response variables vary significantly which calls for scaling. ***Scaling*** ensures that features are on a similar range or scale. Since this project involves comparison of the performance of multiple algorithms, it's important to apply the same scale to all algorithms to

ensure consistency. Therefore, StandardScaler() was used to scale the numerical features to a scale such that each feature has mean 0 and standard deviation 1. The results from the clearly indicate the scaling precision achieved. The features can now be moved to further analysis.

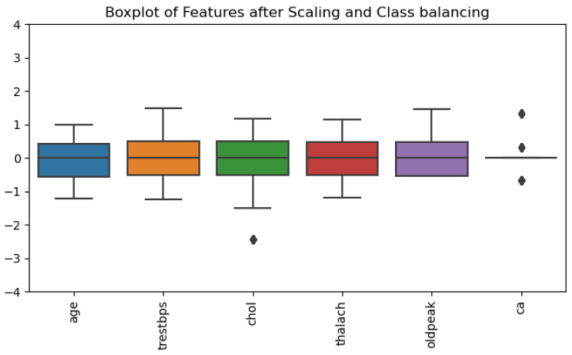


Fig. 10. Boxplot after Handling Outliers and Feature Scaling

Dimensionality Reduction:

PCA (Principal Component Analysis): PCA was used to reduce the data to two principal components to visualize variance-based reduction. This method helped to examine the overall variance while retaining significant data components.

D. Exploratory Data Analysis

- The target class distribution was examined to understand the dataset's imbalance.
- **Correlation Analysis**: A correlation matrix was generated to understand the relationships between features and the target variable. This analysis helped inform feature selection for some models.
- **Feature Distributions:** Visualizations such as histograms and box plots were created to understand the distribution of both numerical and categorical features, detecting skewness, variance, and outliers.

E. Model Training and Development

This paper compares the performance of 5 classification models:
Random Forest (RF)
Decision Tree (DT)
Support Vector Machine (SVM)
Logistic Regression (LR)
Neural Network model

- **Model I : Random Forest** The Random Forest algorithm is an ensemble learning method that constructs multiple decision trees and outputs the mode of their predictions. It is known for its robustness against overfitting and its ability to handle large datasets with high dimensionality.
- **Model II : Decision Tree** The Decision Tree algorithm creates a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. It is intuitive and easy to interpret, making it a popular choice for classification tasks.
- **Model III : Support Vector Machine** The Support Vector Machine algorithm constructs a hyperplane in a high-dimensional space to separate different classes. SVM is particularly effective in high-dimensional spaces and is effective when the number of dimensions exceeds the number of samples.
- **Model IV : Logistic Regression** Logistic Regression is a statistical method for predicting binary classes. It estimates the probability of a binary response based on one or more predictor variables, making it a simple yet powerful model for classification.

  One-Versus-Rest Approach: The one-versus-rest (OvR) strategy is commonly used for multi-class classification problems. It involves training one classifier per class, with the class being classified as positive and all others as negative. Each classifier produces a decision boundary for the corresponding class, and during prediction, the class with the highest confidence is selected.
- **Model V : Neural Network** A Neural Network model is a type of machine learning algorithm inspired by the way biological neural networks in the human brain function. It's designed to recognize patterns, classify data, and make predictions by processing information through interconnected layers of nodes (neurons).

F. Model Evaluation Metrics

Accuracy: The proportion of correctly predicted observations.

$$Accuracy = TP + TN/(TN + FN + TP + FP) \quad (5)$$

Sensitivity (Recall or True Positive Rate)): The ability of the model to correctly predict positive instances (malignant cases).

$$Sensitivity = TP/(TP + FN) \quad (6)$$

Specificity (True Negative Rate) : The model's ability to correctly predict negative instances (benign cases).

$$Specificity = TN/(TN + FP) \quad (7)$$

Precision: The proportion of positive identifications that were actually correct.

$$Precision = TP/(TP + FP) \quad (8)$$

where ,
TP = True Positives
FP = False Positives
FN = False Negatives
TN = True Negatives

Confusion Matrix: For each model, confusion matrices were generated and analyzed to determine the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).
Comparison: A summary of these metrics across all models was presented in a grid format to highlight the strengths and weaknesses of each model. Random Forest emerged as top performers.

VIII. RESULTS

Support Vector Machine (SVM):
Accuracy: 0.6159 Sensitivity (Recall): 0.6159 Specificity: 0.8870 Precision: 0.5187 SVM exhibits the highest accuracy among all models, alongside balanced sensitivity and specificity, indicating its effectiveness in distinguishing between classes. However, its precision is relatively lower, suggesting that while it identifies a good number of positive cases, some predictions may not be true positives.
Random Forest:
Accuracy: 0.5761 Sensitivity: 0.5761 Specificity: 0.8804 Precision: 0.5386 Random Forest provides a moderate performance across metrics, demonstrating a reasonable balance between sensitivity and specificity, but still falls short compared to SVM. Logistic Regression:
Accuracy: 0.5725 Sensitivity: 0.5725 Specificity: 0.8830 Precision: 0.5622 Logistic Regression shows performance similar to Random Forest, with slightly lower accuracy but better precision, indicating it correctly identifies a higher proportion of true positive cases compared to its overall positive predictions.
Neural Network: Accuracy: 0.5507 Sensitivity: 0.3323 Specificity: 0.8751 Precision: 0.3233 The Neural Network model performs poorly in sensitivity and precision, indicating it struggles to accurately identify positive cases despite maintaining reasonable specificity.
Decision Tree: Accuracy: 0.4928 Sensitivity: 0.4928 Specificity: 0.8598 Precision: 0.5029 The Decision Tree shows the lowest accuracy and sensitivity, highlighting its limited effectiveness in classifying instances correctly.
Best Model Description Support Vector Machine (SVM)
The SVM model stands out as the best-performing model based on the provided metrics, achieving an accuracy of 0.6159 and maintaining a good balance between sensitivity and specificity. This model's ability to create a hyperplane that maximally separates the classes contributes to its performance, especially in cases where the data is not linearly separable.
Strengths:
SVM's high specificity (0.8870) indicates that it effectively minimizes false positives, making it a reliable choice for applications where identifying the negative class is critical. The balanced sensitivity suggests that it also manages to identify a significant proportion of true positives. Considerations:
While the precision (0.5187) is lower than desired, indicating that some of the positive predictions may not be true positives, the overall performance still positions SVM as the leading candidate for this classification task. In conclusion, based on the accuracy, sensitivity, specificity, and precision, the SVM model is recommended for implementation in practical applications where robust classification performance is required.

IX. DISCUSSION

Evaluation for PCA:

Model Performance Metrics Comparison after PCA:

Model	Accuracy	Sensitivity	Specificity	Precision
Random Forest	0.507246	0.308781	0.507246	0.307711
Decision Tree	0.434783	0.283507	0.434783	0.279500
SVM	0.536232	0.307168	0.536232	0.329136
Neural Network	0.489130	0.309469	0.489130	0.295194
Logistic Regression	0.518116	0.371847	0.518116	0.409889

Fig. 11. Evaluation Metrics for PCA

Random Forest:
Accuracy: 0.5072 Sensitivity: 0.3088 Specificity: 0.5072 Precision: 0.3077 The Random Forest model exhibits moderate accuracy but low sensitivity and precision, indicating its struggle to correctly identify positive cases. The balance between sensitivity and specificity is concerning, suggesting that this model may misclassify a significant number of positive instances. Decision Tree:
Accuracy: 0.4348 Sensitivity: 0.2835 Specificity: 0.4348 Precision: 0.2795 The Decision Tree model shows the lowest accuracy, sensitivity, and precision among the models evaluated. This indicates that it performs poorly in distinguishing between classes, leading to many misclassifications, especially of the positive class. Support Vector Machine (SVM):
Accuracy: 0.5362 Sensitivity: 0.3072 Specificity: 0.5362 Precision: 0.3291 The SVM model has a slightly better accuracy than Random Forest but still exhibits low sensitivity. Its performance in identifying true positive cases is inadequate, despite having a higher specificity compared to other models. Neural Network:
Accuracy: 0.4891 Sensitivity: 0.3095 Specificity: 0.4891 Precision: 0.2952 The Neural Network model does not significantly outperform the Decision Tree. Its sensitivity and precision are low, indicating difficulty in accurately predicting positive cases. Logistic Regression:
Accuracy: 0.5181 Sensitivity: 0.3718 Specificity: 0.5181 Precision: 0.4099 Logistic Regression emerges as the model with the highest sensitivity and precision, making it the most reliable for identifying true positive cases in this comparison. However, its accuracy remains modest.

X. CONCLUSION

Overall, none of the models perform exceptionally well, as indicated by the relatively low accuracy, sensitivity, specificity, and precision across the board. The Logistic

Regression model stands out as the most reliable option, achieving the highest sensitivity and precision, suggesting it is better at identifying positive cases compared to the others. However, its overall accuracy is still below 0.6, highlighting the potential challenges in the dataset or the need for further feature engineering and model tuning.
The low sensitivity scores across all models indicate a significant room for improvement in correctly identifying the positive class, which could have important implications depending on the application context (e.g., in medical diagnostics or fraud detection). Further exploration of the data, including addressing potential class imbalances, refining feature selection, or using more complex models, could lead to improved performance. Additionally, reconsidering the preprocessing techniques such as PCA may also be beneficial, as it might be obscuring some of the valuable information in the dataset.

REFERENCES