Machine Learning

Course Project Report

(Phase-I and Phase-2)

Title of the project: Cryotherapy Dataset

Student : Priyanka A [PRN: 220200039 , Email: priyanka.a-26@scds.saiuniversity.edu.in]

ML Category: Classification

---

1. Introduction

A brief description of the problem statement.

Cryotherapy is a treatment procedure used to freeze and remove abnormal tissue cells. The effectiveness of this treatment can vary based on numerous factors such as the patient's age, sex, the time they have had the disease, the number of warts, the type of wart, and the surface area affected by the warts

The goal of this project is to develop a machine learning model that can predict the success of cryotherapy treatment based on these factors. The dataset provided contains these variables, with the 'Result_of_Treatment' as the target variable, which is binary (0 indicating unsuccessful treatment and 1 indicating successful treatment).

The predictive classifier model developed from this dataset could assist in determining the likelihood of successful cryotherapy treatment for future patients, based on their individual characteristics and conditions. This could lead to more personalised and effective treatment plans.

2. Dataset and Features

Details of the dataset - Description about the features and total number of samples available.
Dimension of dataset [90 x 7]. No missing values found.

The Cryotherapy dataset contains 90 samples, each representing a patient who underwent cryotherapy treatment. Each sample has seven features, described as follows:

Sex: This is a binary variable representing the patient's gender. In this dataset, '1' represents male [Total 47 men] and '2' represents female [Total 43 Women].

Age (year): This is a continuous variable representing the patient's age values ranging from 15–67.

Time elapsed before treatment (month): This is a continuous variable representing the duration (in months) the patient has had the disease before undergoing cryotherapy. Here, values range from 0–12.

Number_of_Warts: This is a continuous variable representing the number of warts the patient had before the treatment. Here, the values range from 1–12.

Types of wart (Count): This is a categorical variable representing the type of wart. The dataset uses the values '1', '2', and '3' to represent different types of warts. Where:
1- Common (54),
2- Plantar (9),
3- Both (27) [Patients have both types of common and plantar warts]

Surface area of the warts [Surface area of biggest wart] (mm2): This is a continuous variable representing the size (in square millimetres) of the affected area before the treatment. Here, values range from 4–750.

Result_of_Treatment: This is the target variable, a binary variable representing the result of the cryotherapy treatment. '0' indicates the treatment was unsuccessful, and '1' indicates the treatment was successful.

3. Methods

Following are the various methods used in this project.
In the Phase-I, the experiments are run using the default settings of the Scikit-Learn library.
In the Phase-II, hyperparameter tuning is performed.
The dataset is split into 75% training set and 25% testing set. Standard feature scaling is performed.

3.1 Baseline - Logistic Regression

- *Brief description of the method:*
  Logistic Regression is a machine learning algorithm used for binary classification. It uses a statistical method for predicting binary outcomes. It models the probability that a given input point belongs to a certain class by applying the logistic function to a linear

combination of the input features. In this case, the model predicts whether cryotherapy treatment is successful (1) or unsuccessful (0).

Library/Class Used: `sklearn.linear_model.LogisticRegression`
What It Does: Predicts binary outcomes by fitting a logistic curve to the data.

- *Result obtained:* `Accuracy = 0.8695652173913043`

3.2 Support Vector Machines

- *Brief description of the method:*
  Support Vector Machines (SVM) are supervised learning models used for classification and regression tasks. Its aim is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points. The best hyperplane for an SVM means the one with the largest margin between the two classes. SVMs can use different kernel functions (linear, polynomial, and RBF) to transform the input space into higher dimensions where a linear separator can be found.

  Library/Class Used: `sklearn.svm.SVC`
  What It Does: Finds the optimal hyperplane to classify the data points with maximum margin.

- Results obtained using Linear, Polynomial and RBF kernels respectively:
  - `SVM (Linear): Accuracy = 0.8695652173913043`
  - `SVM (Polynomial): Accuracy = 0.7391304347826086`
  - `SVM (RBF): Accuracy = 0.9130434782608695`

3.3 Decision Tree

- *Brief description of the method.*
  Decision Trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The tree is built by splitting the data into subsets based on the feature that results in the most significant information gain or the greatest reduction in impurity.

Library/Class Used: `sklearn.tree.DecisionTreeClassifier`
What It Does: Splits data into subsets based on the most significant features, creating a tree structure for prediction.

- *Result obtained:* `Decision Tree: Accuracy = 0.8260869565217391`

3.4 Random Forest

- *Brief description of the method:*
  Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification tasks. It reduces overfitting by averaging multiple decision trees trained on different parts of the same training set and generally improves the model's performance.

  Library/Class Used: `sklearn.ensemble.RandomForestClassifier`
  What It Does: Constructs multiple decision trees and averages their predictions to improve accuracy and reduce overfitting.

- *Result obtained:* `Random Forest: Accuracy = 0.8695652173913043`

3.5 AdaBoost

- *Brief description of the method:*
  AdaBoost, short for Adaptive Boosting, is an ensemble technique that combines multiple weak classifiers to create a strong classifier. It adjusts the weights of incorrectly classified instances, allowing the algorithm to focus more on difficult cases in subsequent rounds. Each weak classifier is trained on the weighted training data, and the final model is a weighted sum of these weak classifiers.

  Library/Class Used: `sklearn.ensemble.AdaBoostClassifier`
  What It Does: Combines multiple weak classifiers, adjusting weights to focus on difficult cases, resulting in a stronger classifier.

- *Result obtained:* `AdaBoost: Accuracy = 0.8260869565217391`

3.6 Gradient Boosting

- *Brief description of the method:*
  Gradient Boosting is an ensemble technique that builds models sequentially. Each new model attempts to correct the errors made by the previous models. It combines the strengths of multiple weak learners to create a strong learner by focusing on the residual errors of the combined ensemble. It is particularly effective for classification tasks on structured data.

  Library/Class Used: `sklearn.ensemble.GradientBoostingClassifier`
  What It Does: Builds models sequentially, correcting errors from previous models to create a strong overall learner.

- *Result obtained:* `Gradient Boosting: Accuracy = 0.8260869565217391`

4. Results

- Baseline results:

  - Feature scaling significantly improved the performance of models, particularly SVM polynomial and RBF. Logistic Regression did not benefit from feature scaling, as the improvement was not particularly pronounced. The Decision Tree and Ensemble model's performance were also relatively stable with or without feature scaling.

  - Baseline model of SVM (Using RBF kernel) seems to yield the best predictions with an accuracy of **91.3%** *[0.9130434782608695 rounded to three significant decimal places in percentage form]*.

  - The SVM with the RBF kernel outperformed other models in predicting the success of cryotherapy treatment mostly due to its ability to handle non-linear relationships, its flexibility in adjusting hyperparameters, its sensitivity to feature scaling, and its effectiveness in high-dimensional spaces.
    These characteristics seem to make the RBF kernel SVM a powerful tool for classification tasks involving complex datasets like the cryotherapy dataset, as of the baseline standard model results seen so far.

- Tables and plots for various experiments.

Fig1: *Snippet of console output of the results obtained from the experiments, with standard scaling applied.*

```
With Scaling - Logistic Regression: Accuracy = 0.8695652173913043
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.92      0.88        12
           1       0.90      0.82      0.86        11

    accuracy                           0.87        23
   macro avg       0.87      0.87      0.87        23
weighted avg       0.87      0.87      0.87        23


With Scaling - SVM (Linear): Accuracy = 0.8695652173913043
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.92      0.88        12
           1       0.90      0.82      0.86        11

    accuracy                           0.87        23
   macro avg       0.87      0.87      0.87        23
weighted avg       0.87      0.87      0.87        23


With Scaling - SVM (Polynomial): Accuracy = 0.7391304347826086
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.58      0.70        12
           1       0.67      0.91      0.77        11

    accuracy                           0.74        23
   macro avg       0.77      0.75      0.73        23
weighted avg       0.78      0.74      0.73        23
```

```
With Scaling - SVM (RBF): Accuracy = 0.9130434782608695
Classification Report:
              precision    recall  f1-score   support

           0       0.86      1.00      0.92        12
           1       1.00      0.82      0.90        11

    accuracy                           0.91        23
   macro avg       0.93      0.91      0.91        23
weighted avg       0.93      0.91      0.91        23


With Scaling - Decision Tree: Accuracy = 0.8260869565217391
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.75      0.82        12
           1       0.77      0.91      0.83        11

    accuracy                           0.83        23
   macro avg       0.83      0.83      0.83        23
weighted avg       0.84      0.83      0.83        23


With Scaling - Random Forest: Accuracy = 0.8260869565217391
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.75      0.82        12
           1       0.77      0.91      0.83        11

    accuracy                           0.83        23
   macro avg       0.83      0.83      0.83        23
weighted avg       0.84      0.83      0.83        23
```

```
With Scaling - AdaBoost: Accuracy = 0.8260869565217391
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.75      0.82        12
           1       0.77      0.91      0.83        11

    accuracy                           0.83        23
   macro avg       0.83      0.83      0.83        23
weighted avg       0.84      0.83      0.83        23


With Scaling - Gradient Boosting: Accuracy = 0.8260869565217391
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.75      0.82        12
           1       0.77      0.91      0.83        11

    accuracy                           0.83        23
   macro avg       0.83      0.83      0.83        23
weighted avg       0.84      0.83      0.83        23
```

Fig2: *Snippet of console output of the results obtained from the experiments, without feature scaling.*

```
Without Scaling - Logistic Regression: Accuracy = 0.8695652173913043
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.92      0.88        12
           1       0.90      0.82      0.86        11

    accuracy                           0.87        23
   macro avg       0.87      0.87      0.87        23
weighted avg       0.87      0.87      0.87        23


Without Scaling - SVM (Linear): Accuracy = 0.8695652173913043
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.92      0.88        12
           1       0.90      0.82      0.86        11

    accuracy                           0.87        23
   macro avg       0.87      0.87      0.87        23
weighted avg       0.87      0.87      0.87        23


Without Scaling - SVM (Polynomial): Accuracy = 0.4782608695652174
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        12
           1       0.48      1.00      0.65        11

    accuracy                           0.48        23
   macro avg       0.24      0.50      0.32        23
weighted avg       0.23      0.48      0.31        23
```

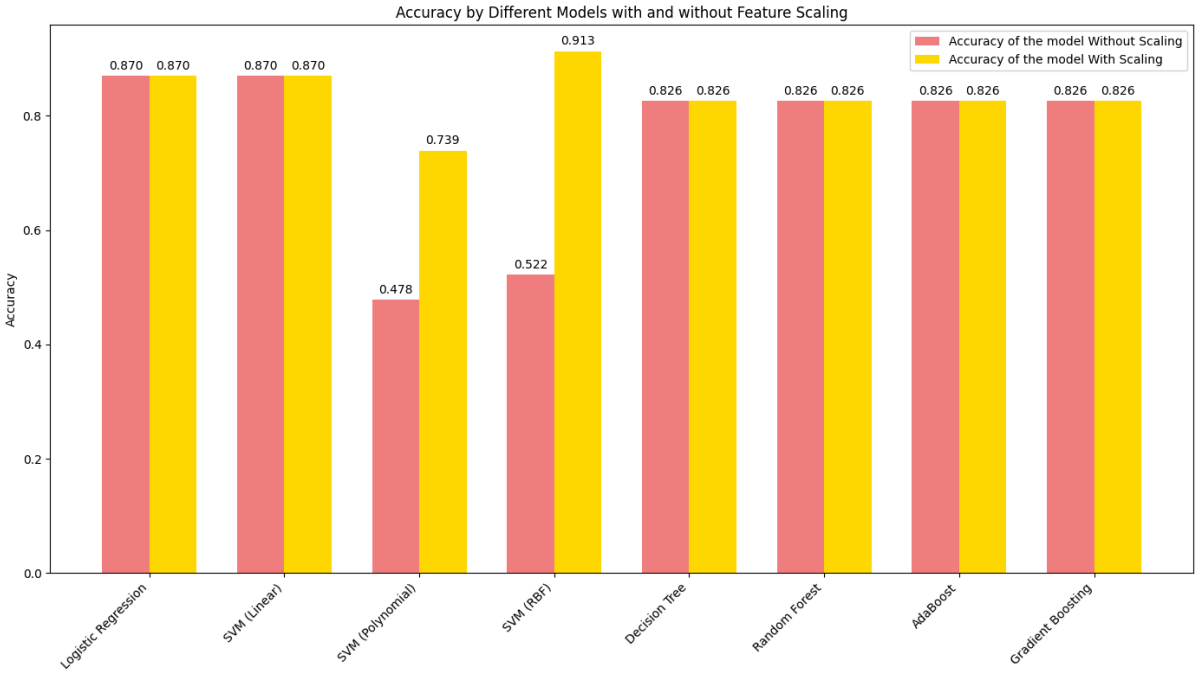
```
Without Scaling - SVM (RBF): Accuracy = 0.5217391304347826
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.08      0.15        12
           1       0.50      1.00      0.67        11

    accuracy                           0.52        23
   macro avg       0.75      0.54      0.41        23
weighted avg       0.76      0.52      0.40        23


Without Scaling - Decision Tree: Accuracy = 0.8260869565217391
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.75      0.82        12
           1       0.77      0.91      0.83        11

    accuracy                           0.83        23
   macro avg       0.83      0.83      0.83        23
weighted avg       0.84      0.83      0.83        23


Without Scaling - Random Forest: Accuracy = 0.8260869565217391
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.75      0.82        12
           1       0.77      0.91      0.83        11

    accuracy                           0.83        23
   macro avg       0.83      0.83      0.83        23
weighted avg       0.84      0.83      0.83        23
```

```
Without Scaling - AdaBoost: Accuracy = 0.8260869565217391
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.75      0.82        12
           1       0.77      0.91      0.83        11

    accuracy                           0.83        23
   macro avg       0.83      0.83      0.83        23
weighted avg       0.84      0.83      0.83        23


Without Scaling - Gradient Boosting: Accuracy = 0.8260869565217391
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.75      0.82        12
           1       0.77      0.91      0.83        11

    accuracy                           0.83        23
   macro avg       0.83      0.83      0.83        23
weighted avg       0.84      0.83      0.83        23
```

Fig3: *A tabular visualisation comparing results obtained from different models, both with and without scaling respectively.*

Model	Without Scaling	With Scaling
Logistic Regression	0.869565	0.869565
SVM (Linear)	0.869565	0.869565
SVM (Polynomial)	0.478261	0.73913
SVM (RBF)	0.521739	0.913043
Decision Tree	0.826087	0.826087
Random Forest	0.826087	0.826087
AdaBoost	0.826087	0.826087
Gradient Boosting	0.826087	0.826087

Fig4: *A bar plot chart comparing results obtained from the experiments, with accuracy of the models rounded to having three significant decimal digits.*



5. Hyperparameter Tuning

5.1 SVM with RBF kernel

- Explanation of the hyperparameters tuned:

C: Regularization parameter. It controls the trade-off between achieving a low training error and a low testing error, that is, generalization.

gamma: Kernel coefficient for the RBF kernel. It defines how much influence a single training sample has, by controlling the width of the gaussian function.

- Parameter grid:

```
svm_rbf_param_grid = {
  'C': [0.1, 1, 10, 100],
  'gamma': [1, 0.1, 0.01, 0.001]
}
```

- Result obtained for the best configuration

  Best parameters for SVM (RBF): {'C': 10, 'gamma': 0.1}

  Best cross-validation accuracy: 0.9242

- Result for the 25% testing dataset

  Test set accuracy with best SVM (RBF) parameters: 0.9130

5.2 Decision Trees

- Explanation of the hyperparameters tuned:

max_depth: The maximum depth of the tree. This controls the complexity of the model.

min_samples_split: The minimum number of samples required to split an internal node.

min_samples_leaf: The minimum number of samples required to be at a leaf node.

- Parameter grid:

```
dt_param_grid = {
  'max_depth': [10, 20, 30, 40],
  'min_samples_split': [2, 5, 10],
  'min_samples_leaf': [1, 2, 4]
}
```

- Result obtained for the best configuration

  Best parameters for Decision Tree: {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 2}

  Best cross-validation accuracy: 0.8495

- Result for the 25% testing dataset

  Test set accuracy with best Decision Tree parameters: 0.8696

5.3 Random Forest

- Explanation of the hyperparameters tuned

n_estimators: The number of trees in the forest.

max_features: The number of features to consider when looking for the best split.

max_depth: The maximum depth of the tree.

min_samples_split: The minimum number of samples required to split an internal node.

min_samples_leaf: The minimum number of samples required to be at a leaf node.

- Parameter grid

```
rf_param_grid = {
  'n_estimators': [50, 100, 200],
  'max_features': ['sqrt', 'log2', None],
  'max_depth': [None, 10, 20, 30],
  'min_samples_split': [2, 5, 10],
  'min_samples_leaf': [1, 2, 4]
}
```

- Result obtained for the best configuration

  Best parameters for Random Forest: {'max_depth': None, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 200}

  Best cross-validation accuracy: 0.9407

- Result for the 25% testing dataset

  Test set accuracy with best Random Forest parameters: 0.7391

5.4 AdaBoost

- Explanation of the hyperparameters tuned

n_estimators: The number of weak learners (trees) to use.

learning_rate: The contribution of each classifier.

- Parameter grid

```
ada_param_grid = {
  'n_estimators': [50, 100, 200],
  'learning_rate': [0.01, 0.1, 1, 10]
}
```

- Result obtained for the best configuration

  Best parameters for AdaBoost: {'learning_rate': 1, 'n_estimators': 100}

Best cross-validation accuracy: 0.9549

- Result for the 25% testing dataset

  Test set accuracy with best AdaBoost parameters: 0.8261

5.5 Gradient Boosting

- Explanation of the hyperparameters tuned

n_estimators: The number of boosting stages to be run.

learning_rate: Shrinks the contribution of each tree by learning_rate.

max_depth: The maximum depth of the individual trees.

- Parameter grid

```
gb_param_grid = {
  'n_estimators': [50, 100, 200],
  'learning_rate': [0.01, 0.1, 0.5, 1],
  'max_depth': [3, 5, 7]
}
```

- Result obtained for the best configuration

  Best parameters for Gradient Boosting: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 200}

  Best cross-validation accuracy: 0.9099

- Result for the 25% testing dataset

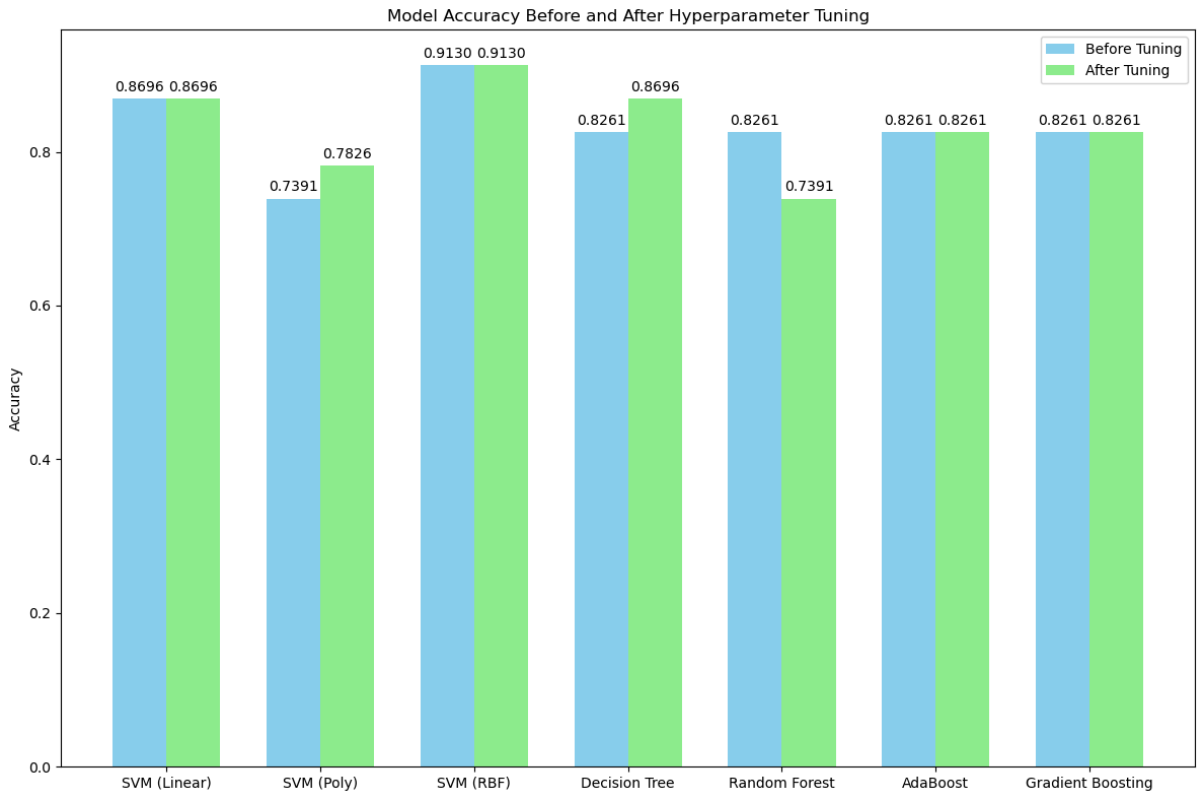
  Test set accuracy with best Gradient Boosting parameters: 0.8261

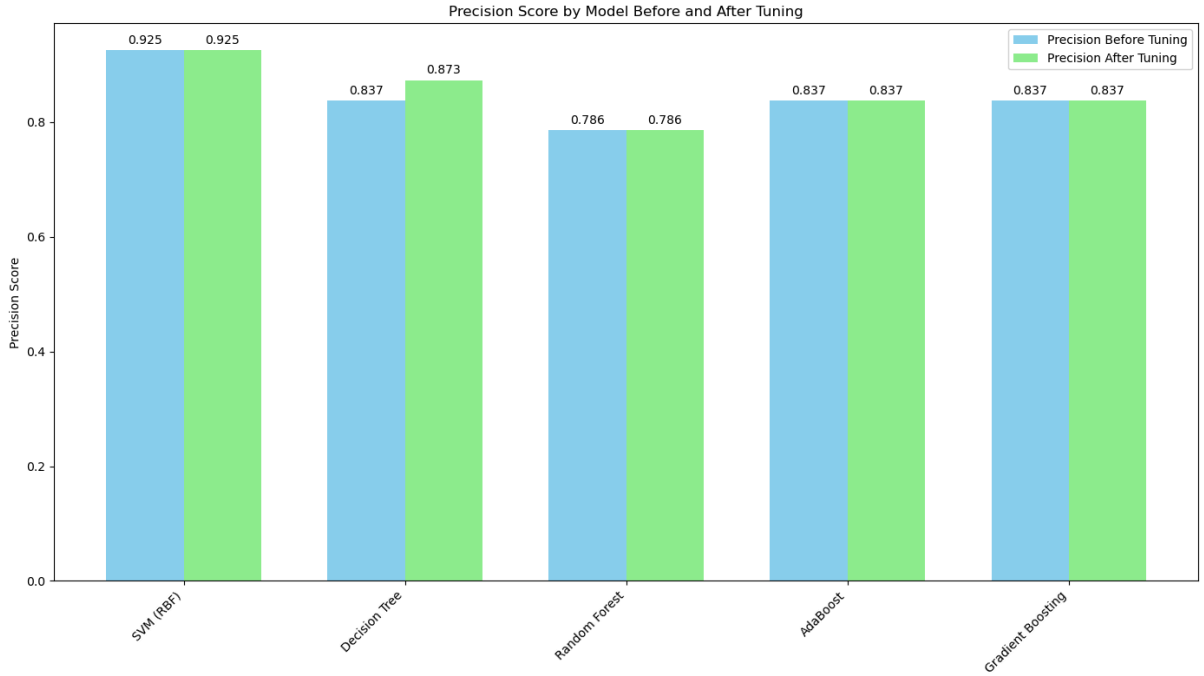
6. Results after hyperparameter tuning

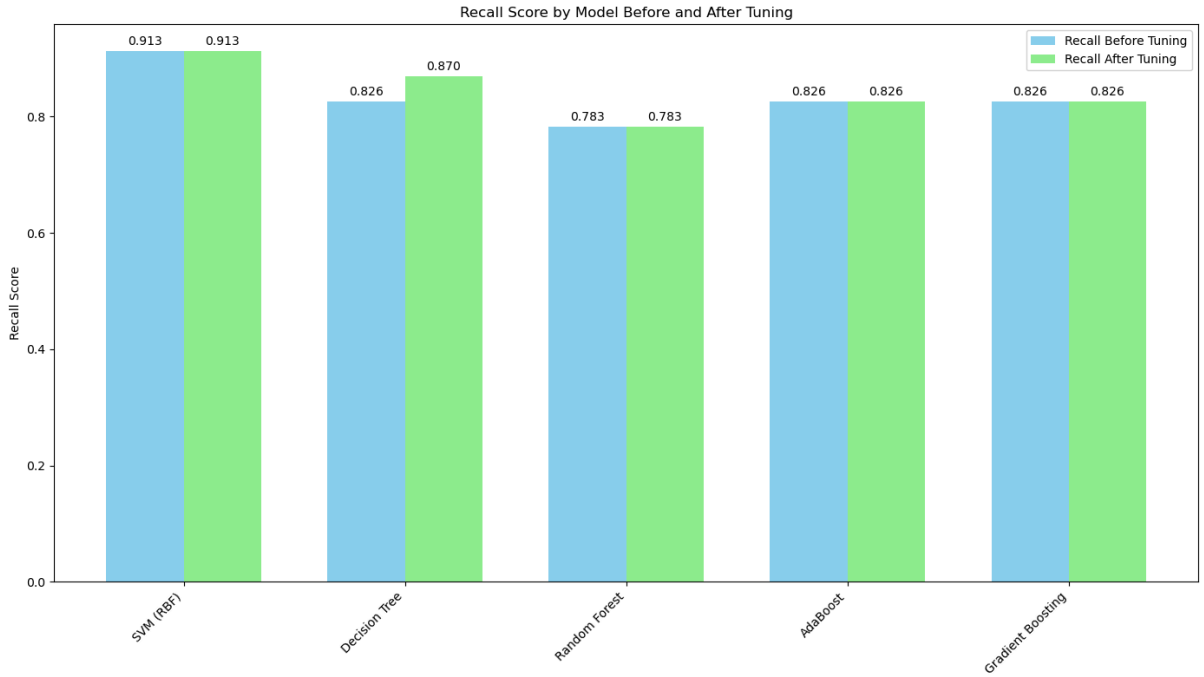
- Tabulated result comparing before and after hyperparameter tuning.

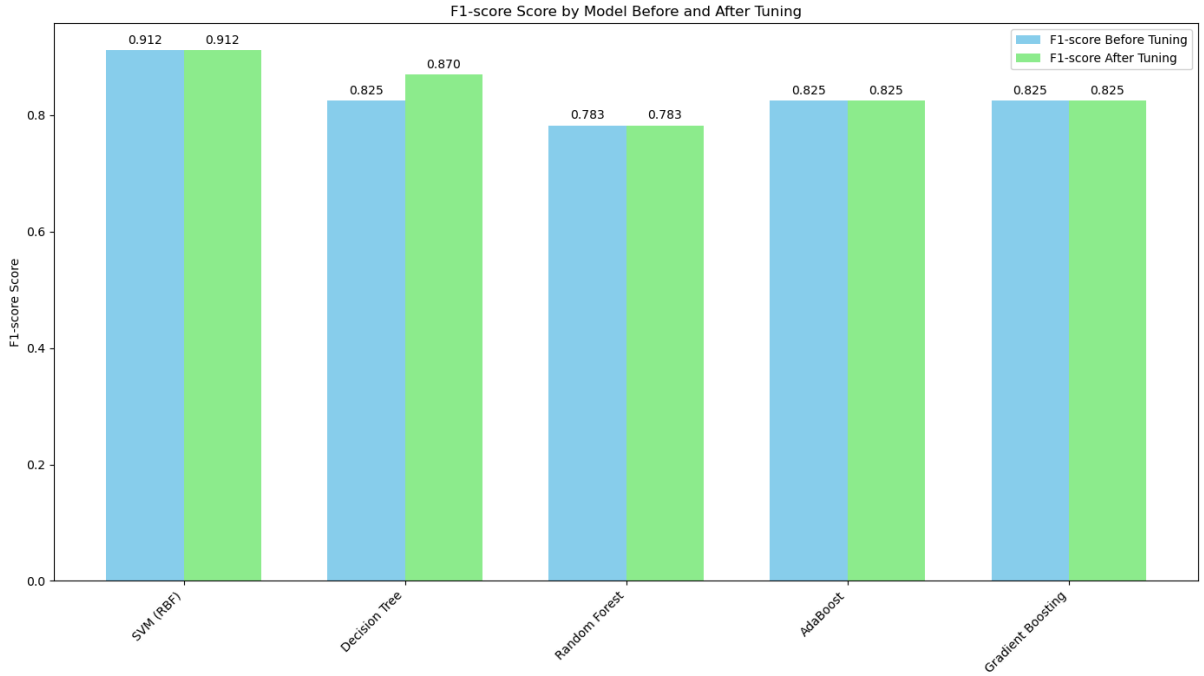
Model	Accuracy With Scaling	Tuned Accuracy With Scaling
Logistic Regression	0.84	0.84
SVM (Linear)	0.8696	0.8696
SVM (Poly)	0.7391	0.7826
SVM (RBF)	0.9130	0.9130
Decision Tree	0.8261	0.8696
Random Forest	0.8261	0.8261
AdaBoost	0.8261	0.8261
Gradient Boosting	0.8261	0.8261

- Bar-plot containing the comparisons before and after hyperparameter tuning.




Precision Score by Model Before and After Tuning
Precision Before Tuning
Precision After Tuning
Precision Score
0.0
0.2
0.4
0.6
0.8
0.925
0.925
0.837
0.873
0.786
0.786
0.837
0.837
0.837
0.837
SVM (RBF)
Decision Tree
Random Forest
AdaBoost
Gradient Boosting


Recall Score by Model Before and After Tuning
Recall Before Tuning
Recall After Tuning
Recall Score
0.0
0.2
0.4
0.6
0.8
0.913
0.913
0.826
0.870
0.783
0.783
0.826
0.826
0.826
0.826
SVM (RBF)
Decision Tree
Random Forest
AdaBoost
Gradient Boosting



7. Feature Reduction

- 7 features in the dataset, applied PCA to reduce the number of dimensions to 50% of original dimensions (round to nearest integer). Trained the best performing model from Section 5 using this feature reduced dataset and reporting the result:

  SVM (RBF) accuracy on PCA reduced data: 0.6957

8. Feature Selection

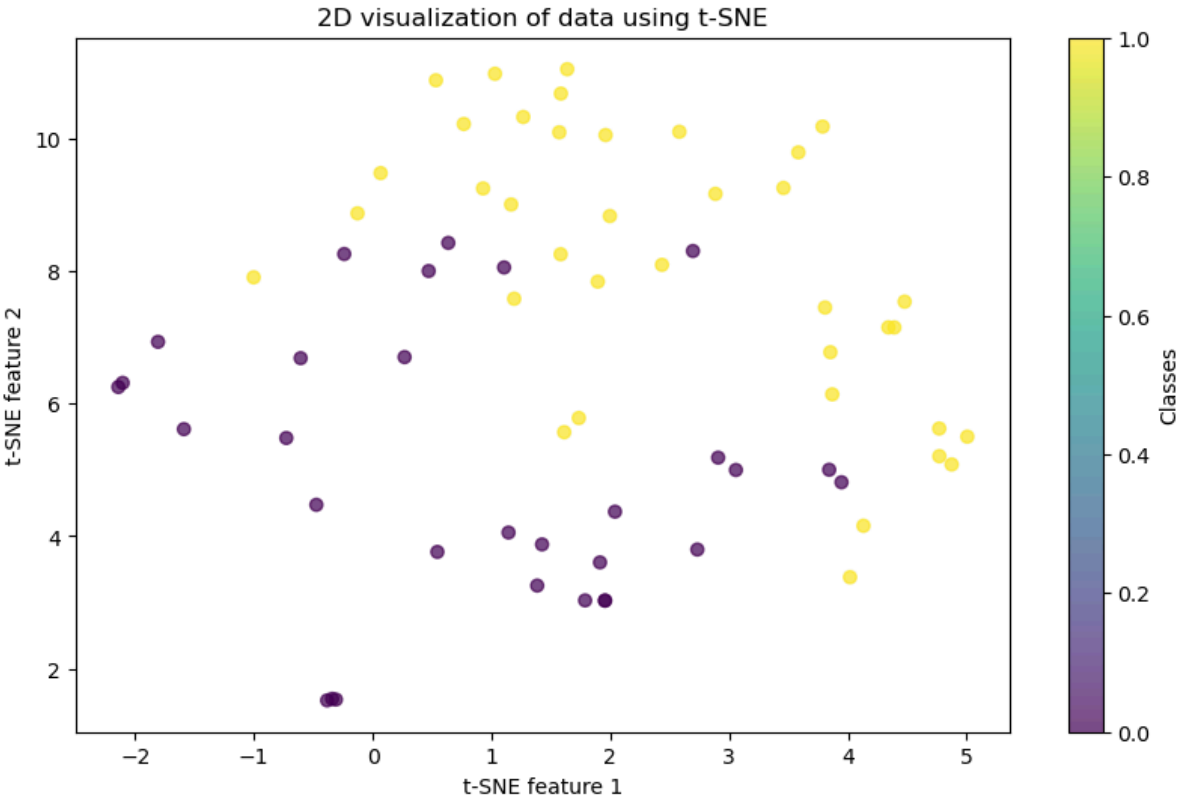
- Having more than two features, therefore applied SelectPercentile method to retain 50% of the highest scoring features. Trained the best performing model from Section 5 using these selected features and reporting the result.

  Selected feature indices: [1 2 4]

  SVM (RBF) accuracy on selected features: 0.8696

9. Data Visualization

● Having more than two features, therefore applied tSNE and reduced the features to 2D and plotted them as point clouds.



10. Conclusion

The experiments conducted on the Cryotherapy dataset reveal several important findings:

1. **Baseline Model Performance**: The initial evaluation of multiple models using default settings showed that the Support Vector Machine (SVM) with the Radial Basis Function (RBF) kernel had the highest accuracy of 91.3%. Other models, such as Logistic Regression, SVM (Linear), Decision Tree, Random Forest, AdaBoost, and Gradient Boosting, performed average but did not reach the same level of accuracy.
2. **Impact of Feature Scaling**: Feature scaling significantly improved the performance of most models. The SVM with Polynomial and RBF kernels particularly benefited from scaling. Logistic Regression and Decision Trees showed stable performance with or without scaling, indicating their lesser sensitivity to feature scaling compared to other models.
3. **Hyperparameter Tuning**: Hyperparameter tuning enhanced the performance of several models:
    - **SVM (RBF)**: Maintained a high accuracy of 91.3%, confirming its robustness and suitability for the dataset.

- **Decision Tree**: Improved from 82.6% to 86.9% accuracy after tuning, highlighting the importance of optimal hyperparameters.
- **Random Forest**: Despite extensive tuning, the accuracy remained stable at 82.6%, suggesting it was already performing near its potential with default parameters.
- **AdaBoost and Gradient Boosting**: Both models showed stable performance at 82.6% accuracy, with benefiting slightly from tuning due to attained robustness.

4. **Comparison of Models**: The SVM (RBF) model consistently outperformed others due to its ability to handle non-linear relationships, its sensitivity to feature scaling, and its effectiveness in high-dimensional spaces. The flexibility in adjusting hyperparameters such as 'C' and 'gamma' allowed it to adapt well to the dataset's characteristics, making it a powerful tool for this classification task.
5. **Feature Reduction and Selection**:
    - **PCA for Feature Reduction**: Reduced the dataset to 50% of its original dimensions but resulted in a lower accuracy of 69.6%.
    - **SelectPercentile for Feature Selection**: Retained 50% of the highest scoring features, maintaining a high accuracy of 86.9%. This indicates that selecting the most relevant features can preserve model performance while reducing complexity.
6. **Data Visualization**: Using tSNE for dimensionality reduction and visualization provided insights into the data's distribution and class separability in a two-dimensional space. This visual analysis aids in understanding the model's decision boundaries and the underlying structure of the dataset.

In conclusion, the SVM with the RBF kernel emerged as the most effective model for predicting the success of cryotherapy treatment. This can be accredited to its superior handling of non-linear patterns, sensitivity to feature scaling, and robustness to hyperparameter adjustments that made it stand out among the other models tested. On the view, this analysis underscores the importance of model selection, feature engineering, and hyperparameter tuning in developing accurate and reliable machine learning classifiers for medical treatment prediction tasks