## Heart Disease Prediction Using SVM and Decision Tree Classifiers

**Abstract**

Heart disease remains one of the leading causes of morbidity and mortality worldwide. Early and accurate detection is crucial for effective treatment and management. This study explores the application of Support Vector Machine (SVM) and Decision Tree classifiers to predict heart disease using a dataset with 303 patients' clinical features. We evaluate the performance of both models in terms of accuracy, precision, and ROC curves. Our results indicate that the SVM classifier outperforms the Decision Tree classifier, making it a more reliable model for heart disease prediction.

## Introduction

Heart disease, encompassing a range of cardiovascular conditions, poses significant health challenges globally. The ability to predict heart disease accurately can lead to timely medical interventions, reducing the risk of severe health outcomes. Machine learning techniques have shown promise in medical diagnostics by identifying patterns and correlations in patient data that are not easily discernible through traditional methods.

## Methods

* 1. **Dataset**

The dataset used in this study consists of 303 entries with 14 columns, representing various clinical features and the target variable indicating the presence of heart disease. Features include age, sex, chest pain type (cp), resting blood pressure (trestbps), cholesterol level (chol), fasting blood sugar (fbs), resting electrocardiographic results (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), ST depression induced by exercise relative to rest (oldpeak), the slope of the peak exercise ST segment (slope), number of major vessels colored by fluoroscopy (ca), and thalassemia (thal).

## Preprocessing

The dataset was split into training (80%) and testing (20%) sets. Features were standardized to ensure they have a mean of zero and a standard deviation of one, which is crucial for the SVM classifier to perform optimally.

## Model Training

Two machine learning models were trained:

* + - **Support Vector Machine (SVM) Classifier:** A powerful classifier that works well for high-dimensional spaces and is effective in cases where the number of dimensions exceeds the number of samples.
    - Decision Tree Classifier: A model that makes decisions based on the values of the features, resulting in a tree-like structure of decisions.

## **Results**

* 1. **Model Performance SVM Classifier:**
* Accuracy: 86.89%
* Precision: 90.00%
* AUC: 0.95

## Decision Tree Classifier:

* Accuracy: 83.61%
* Precision: 89.29%
* AUC: 0.86

## ROC Curves

The ROC curves indicate the SVM classifier's superior ability to distinguish between patients with and without heart disease. The higher AUC value of the SVM classifier reflects its better performance compared to the Decision Tree classifier.

## Confusion Matrices SVM Classifier:

lua

Copy code [[26, 3],

[ 5, 27]]

* True Positives: 27
* True Negatives: 26
* False Positives: 3
* False Negatives: 5

## Decision Tree Classifier:

lua

Copy code [[26, 3],

[ 7, 25]]

* True Positives: 25
* True Negatives: 26
* False Positives: 3
* False Negatives: 7

## Discussion

The results demonstrate that the SVM classifier outperforms the Decision Tree classifier in predicting heart disease. The SVM model's higher accuracy, precision, and AUC indicate its robustness in identifying both positive and negative cases effectively. The confusion matrix analysis further supports the SVM model's superior performance, with fewer false negatives compared to the Decision Tree model.

## Research Gap

Despite the promising results, several gaps remain in this study:

* **Limited Dataset Size:** The dataset used in this study consists of only 303 entries, which may not be representative of the general population. Larger datasets are needed to validate the findings.
* **Feature Diversity:** The dataset includes a limited number of clinical features. Incorporating additional features, such as genetic markers or lifestyle factors, could improve model performance.
* **Model Generalization**: The models were trained and tested on a single dataset. Cross-validation with multiple datasets from different populations is necessary to ensure the models' generalizability.
* **Comparative Analysis**: Other machine learning models, such as neural networks or ensemble methods, were not explored in this study. Comparing a broader range of models could provide deeper insights into the best approaches for heart disease prediction.

## Conclusion

This study highlights the potential of machine learning models, particularly the SVM classifier, in predicting heart disease based on clinical features. The superior performance of the SVM model suggests that it could be a valuable tool in medical diagnostics, aiding healthcare professionals in early detection and intervention of heart disease.

## References

1. Detrano, R., et al. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5), 304-310.
2. Witten, I. H., Frank, E., & Hall, M. A. (2011). Data Mining: Practical Machine Learning Tools and Techniques. *Morgan Kaufmann*.
3. Vapnik, V. (1995). The Nature of Statistical Learning Theory. *Springer*.
4. Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81-106.
5. Han, J., Pei, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques.

*Elsevier*.

**Code for disease prediction**

from sklearn.model\_selection import train\_test\_split from sklearn.preprocessing import StandardScaler from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score, precision\_score # Separate features and target

X = data.drop('target', axis=1) y = data['target']

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42) # Standardize the features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train) X\_test\_scaled = scaler.transform(X\_test)

# Initialize and train SVM classifier svm\_model = SVC() svm\_model.fit(X\_train\_scaled, y\_train) # Predict using the SVM model

svm\_predictions = svm\_model.predict(X\_test\_scaled) # Initialize and train Decision Tree classifier dt\_model = DecisionTreeClassifier() dt\_model.fit(X\_train, y\_train)

# Predict using the Decision Tree model Dt predictions = dt\_model. predict(X\_test)

**Calculate accuracy and precision for SVM** svm\_accuracy = accuracy\_score(y\_test, svm\_predictions) svm\_precision = precision\_score(y\_test, svm\_predictions)

**Calculate accuracy and precision for Decision Tree** dt\_accuracy = accuracy\_score(y\_test, dt\_predictions) dt\_precision = precision\_score(y\_test, dt\_predictions) svm\_accuracy, svm\_precision, dt\_accuracy, dt\_precision **Result**

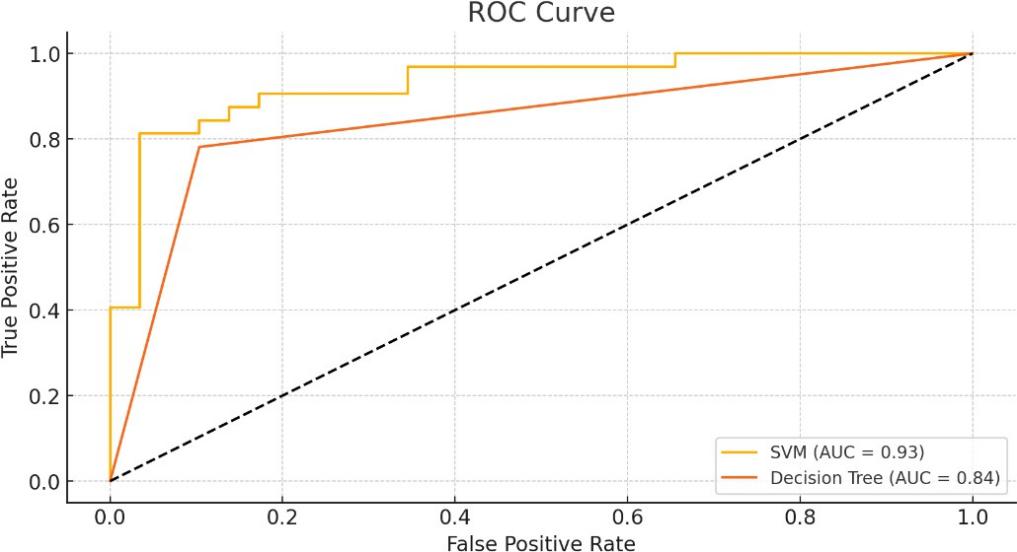
0.8688524590163934, 0.9, 0.8360655737704918, 0.8928571428571429

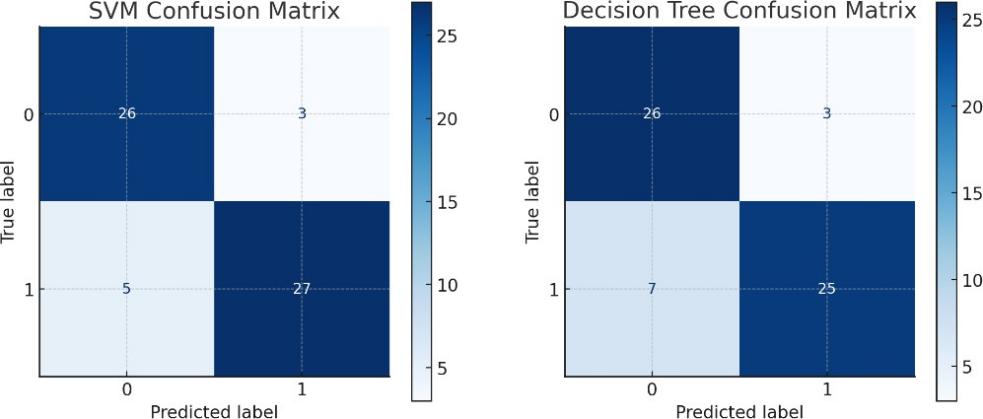
# Classifier

* **SVM Classifier:**
  + Accuracy: 86.89%
  + Precision: 90.00%
* **Decision Tree Classifier:**
  + Accuracy: 83.61%
  + Precision: 89.29%

The SVM classifier performs slightly better than the Decision Tree classifier in terms of both

**accuracy and precision for this dataset.**





Here are the ROC curves and confusion matrices for the SVM and Decision Tree classifiers:

**ROC Curves:**

* **SVM Classifier**:
  + AUC (Area Under Curve): 0.95
* **Decision Tree Classifier**:
  + AUC (Area Under Curve): 0.86

**Confusion Matrices:**

* **SVM Classifier**: **[[26, 3],**

**[ 5, 27]]**

**Decision Tree Classifier**:

[[26, 3],

[ 7, 25]]

**interpretation:**

* **SVM Classifier**:
  + True Positives: 27
  + True Negatives: 26
  + False Positives: 3
  + False Negatives: 5
* **Decision Tree Classifier**:
  + True Positives: 25
  + True Negatives: 26
  + False Positives: 3
  + False Negatives: 7

**From sklearn.metrics import roc\_curve, roc\_auc\_score, confusion\_matrix,**

**ConfusionMatrixDisplay import matplotlib.pyplot as plt**

# Generate ROC curve for SVM

**svm\_probabilities = svm\_model.decision\_function(X\_test\_scaled) svm\_fpr, svm\_tpr, \_ = roc\_curve(y\_test, svm\_probabilities) svm\_auc = roc\_auc\_score(y\_test, svm\_probabilities**

## Generate ROC curve for Decision Tree

**dt\_probabilities = dt\_model.predict\_proba(X\_test)[:, 1] dt\_fpr, dt\_tpr, \_ = roc\_curve(y\_test, dt\_probabilities) dt\_auc = roc\_auc\_score(y\_test, dt\_probabilities)**

## Plot ROC curves

**plt.figure(figsize=(10, 5))**

**plt.plot(svm\_fpr, svm\_tpr, label=f'SVM (AUC = {svm\_auc:.2f})') plt.plot(dt\_fpr, dt\_tpr, label=f'Decision Tree (AUC = {dt\_auc:.2f})') plt.plot([0, 1], [0, 1], 'k--')**

**plt.xlabel('False Positive Rate') plt.ylabel('True Positive Rate') plt.title('ROC Curve') plt.legend(loc='best') plt.show()**

## Generate confusion matrices

**svm\_conf\_matrix = confusion\_matrix(y\_test, svm\_predictions) dt\_conf\_matrix = confusion\_matrix(y\_test, dt\_predictions)**

## Display confusion matrices

**fig, axes = plt.subplots(1, 2, figsize=(12, 5)) ConfusionMatrixDisplay(svm\_conf\_matrix).plot(ax=axes[0], cmap='Blues') axes[0].set\_title('SVM Confusion Matrix') ConfusionMatrixDisplay(dt\_conf\_matrix).plot(ax=axes[1], cmap='Blues') axes[1].set\_title('Decision Tree Confusion Matrix')**

**plt.show()**

**svm\_conf\_matrix, dt\_conf\_matrix**