HEART DISEASE PREDICTION

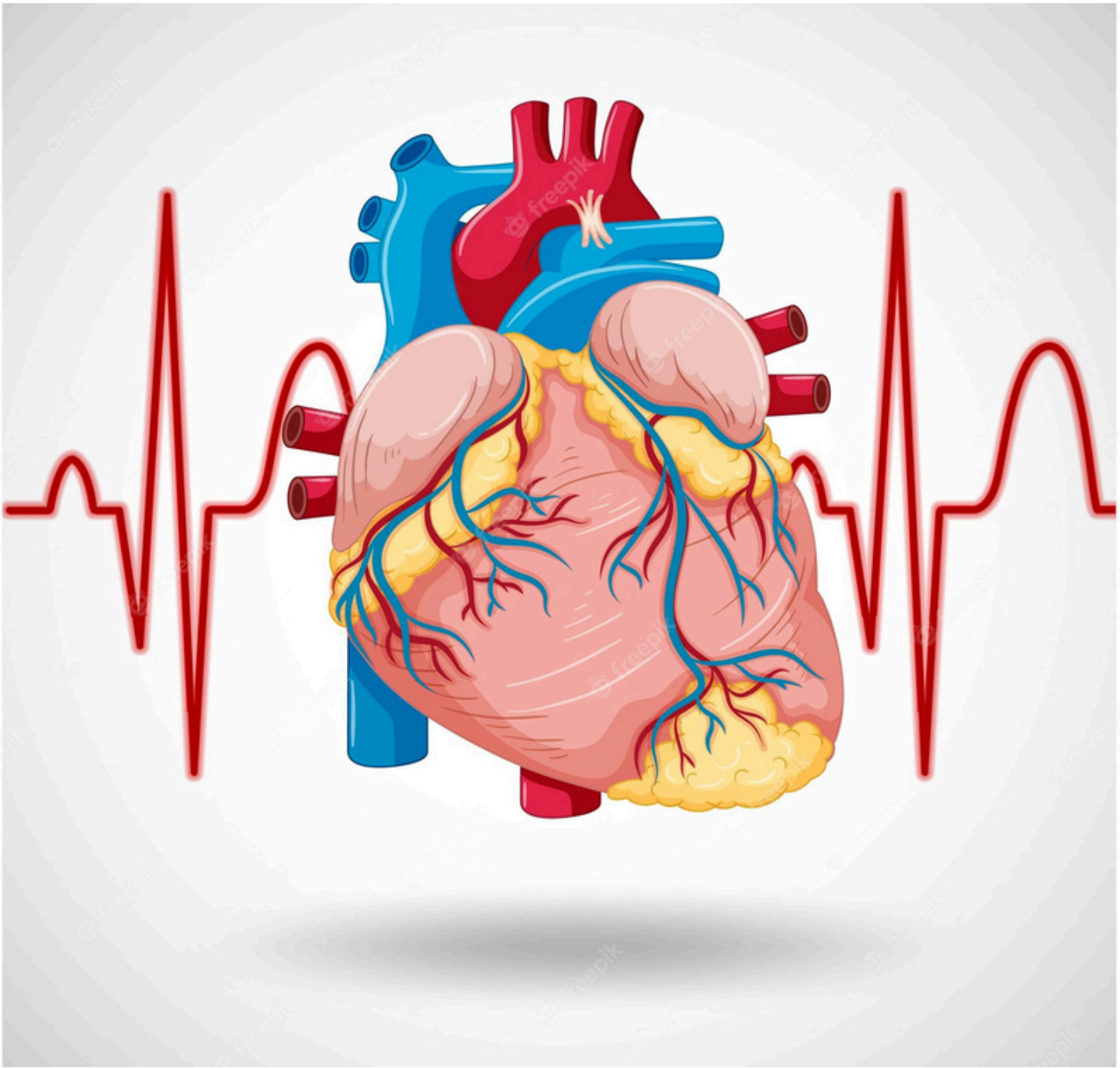
NAME:Rajyalakshmi Rajana

ROLL NO:2360530098

INSTITUTION: ADITYA WOMEN'S DEGREE COLLEGE, VISAKHAPATNAM

COURSE: BCA

SUBMISSION DATE:09-02-2025



Abstract:

This project aims to predict heart disease using machine learning techniques. The dataset is analysed and pre-processed, followed by the implementation of multiple classification algorithms, including K-Nearest Neighbours, Support Vector Machine, Decision Trees, and Random Forest. Performance evaluation metrics are used to compare the models, leading to a conclusion on the most effective approach. The results indicate that machine learning can significantly aid in early detection and diagnosis of heart disease. Heart disease remains one of the leading causes of mortality worldwide, with various factors contributing to its development, including genetic predisposition, lifestyle, and environmental influences. This project aims to explore the key risk factors of heart disease, its pathophysiology, and potential preventive measures through a comprehensive analysis of clinical data, case studies, and scientific literature.

Table of content

Introduction: -

Heart disease is one of the leading causes of mortality worldwide. Early prediction using machine learning can enhance diagnosis and treatment plans. This project leverages various classification techniques to predict heart disease based on patient data. This

project aims to build a Machine Learning (ML) model to predict whether a person is at risk of heart disease based on medical parameters such as age, blood pressure, cholesterol levels, and lifestyle habits.

Frameworks: -

- **Programming Language: python**
- **Libraries Used: Pandas,Numpy,Matplotlib,sklearn**

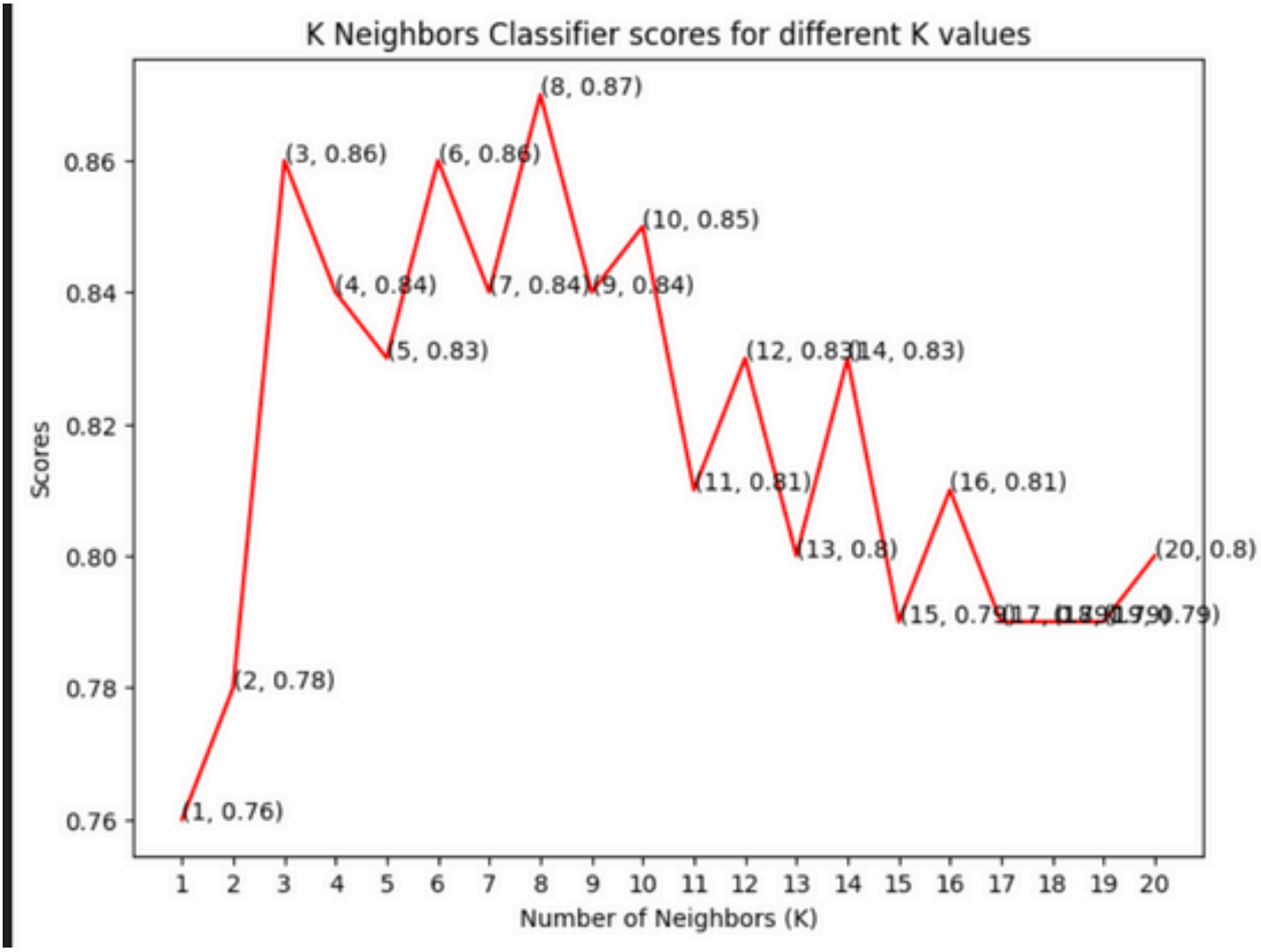
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

Machine learning algorithms:

KNeighborsClassifier:- The KNeighborsClassifier is a machine learning model used for classification tasks. It is part of the k-nearest neighbors (KNN) algorithm, which is a simple, yet effective, non-parametric, supervised learning method.

```
knn_scores = []
for k in range(1,21):
    knn_classifier = KNeighborsClassifier(n_neighbors = k)
    knn_classifier.fit(X_train, y_train)
    knn_scores.append(knn_classifier.score(X_test, y_test))
```

```
plt.plot([k for k in range(1, 21)], knn_scores, color = 'red')
for i in range(1,21):
    plt.text(i, knn_scores[i-1], (i, knn_scores[i-1]))
plt.xticks([i for i in range(1, 21)])
plt.xlabel('Number of Neighbors (K)')
plt.ylabel('Scores')
plt.title('K Neighbors Classifier scores for different K values')
```

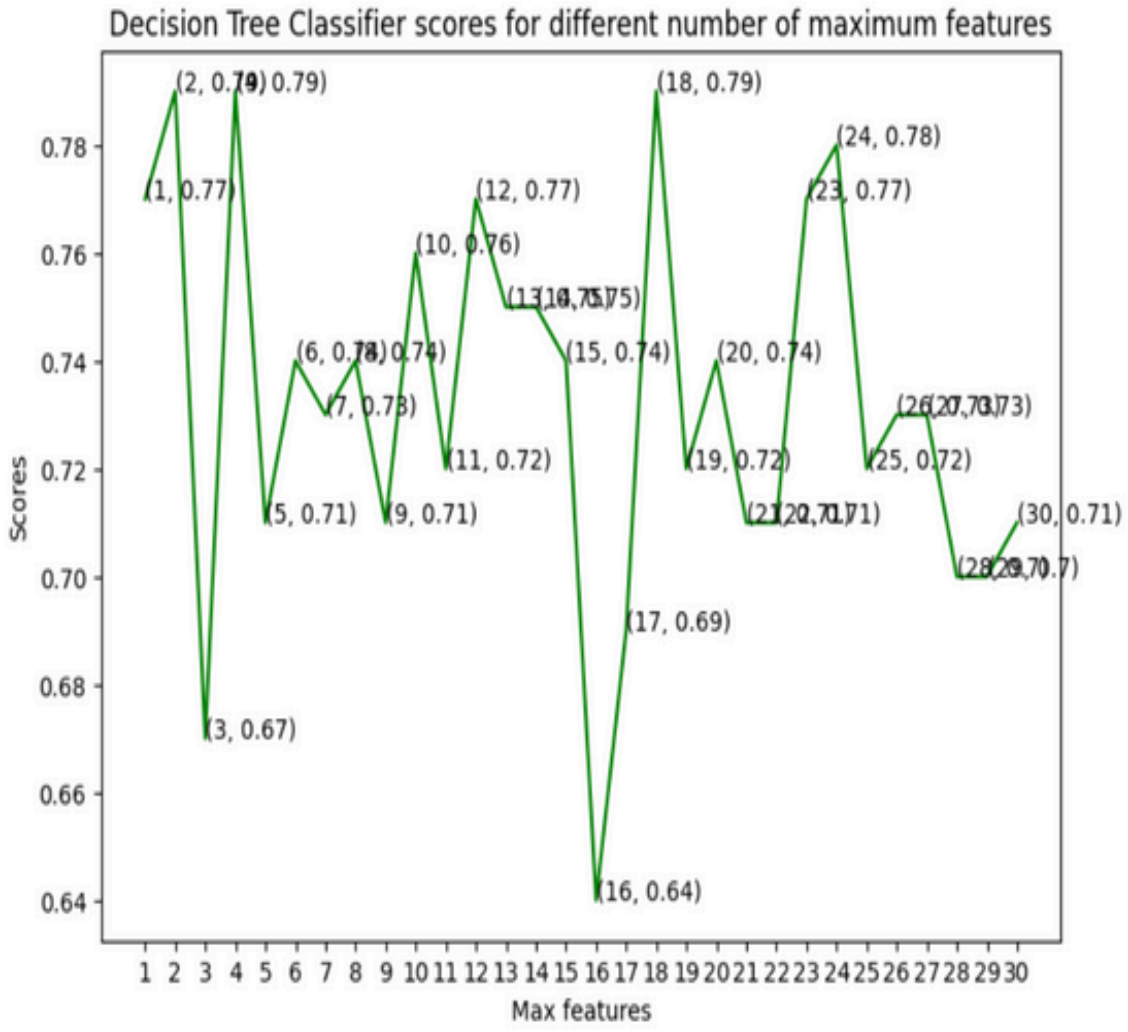


DecisionTreeClassifier: –

The DecisionTreeClassifier is a supervised machine learning algorithm used for classification tasks. It is based on a tree-like model of decisions, where data is split into branches based on feature values.

```
dt_scores = []
for i in range(1, len(X.columns) + 1):
    dt_classifier = DecisionTreeClassifier(max_features = i, random_state = 0)
    dt_classifier.fit(X_train, y_train)
    dt_scores.append(dt_classifier.score(X_test, y_test))
```

```
plt.plot([i for i in range(1, len(X.columns) + 1)], dt_scores, color = 'green')
for i in range(1, len(X.columns) + 1):
    plt.text(i, dt_scores[i-1], (i, dt_scores[i-1]))
plt.xticks([i for i in range(1, len(X.columns) + 1)])
plt.xlabel('Max features')
plt.ylabel('Scores')
plt.title('Decision Tree Classifier scores for different number of maximum features')
```

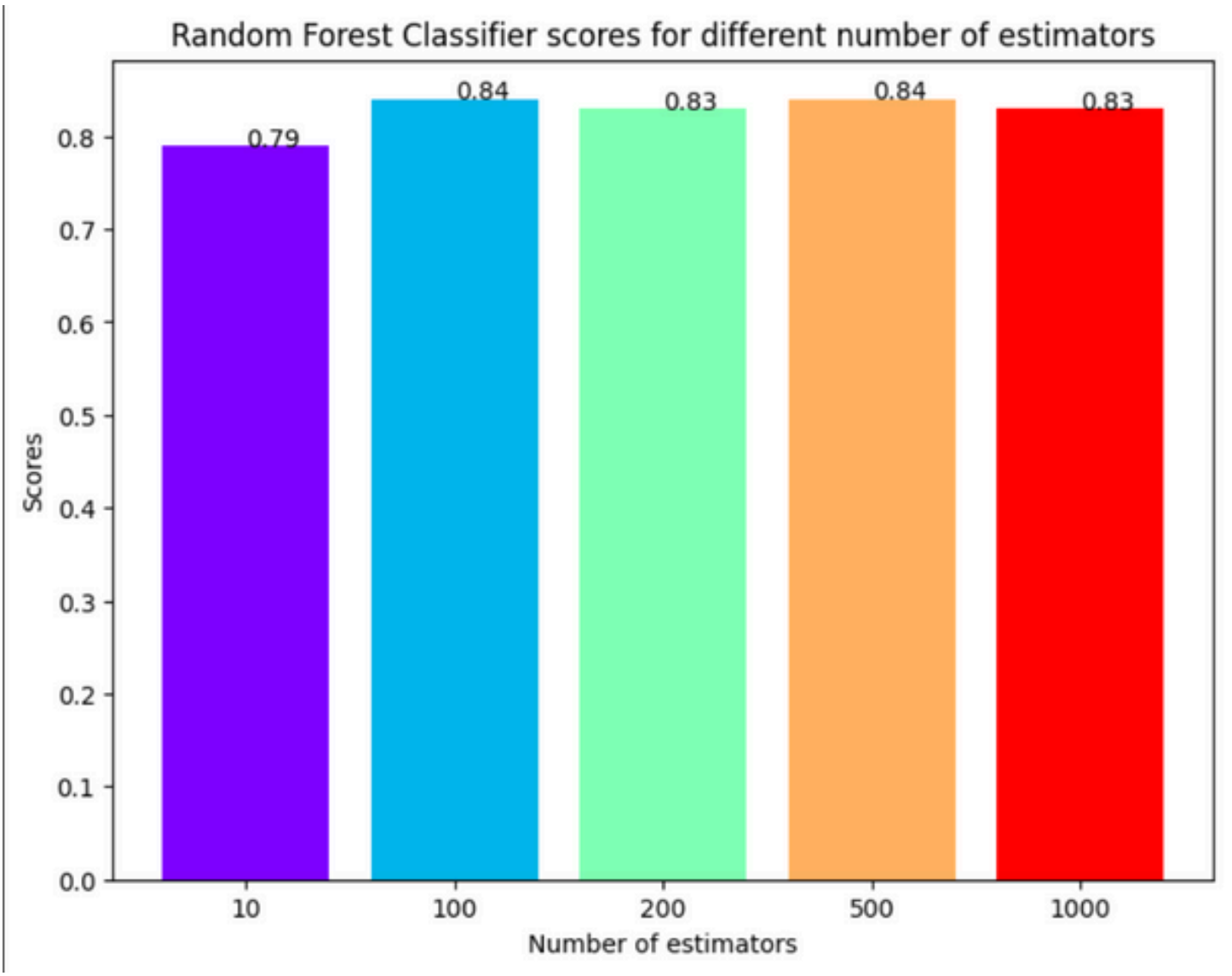


RandomForestClassifier: –

The RandomForestClassifier is a powerful ensemble learning algorithm used for classification tasks. It combines multiple decision trees to improve accuracy and reduce overfitting.

```
rf_scores = []
estimators = [10, 100, 200, 500, 1000]
for i in estimators:
    rf_classifier = RandomForestClassifier(n_estimators = i, random_state = 0)
    rf_classifier.fit(X_train, y_train)
    rf_scores.append(rf_classifier.score(X_test, y_test))
```

```
colors = rainbow(np.linspace(0, 1, len(estimators)))
plt.bar([i for i in range(len(estimators))], rf_scores, color = colors, width = 0.8)
for i in range(len(estimators)):
    plt.text(i, rf_scores[i], rf_scores[i])
plt.xticks(ticks = [i for i in range(len(estimators))], labels = [str(estimator) for estimator in estimators])
plt.xlabel('Number of estimators')
plt.ylabel('Scores')
plt.title('Random Forest Classifier scores for different number of estimators')
```



Results and Discussion: -

After training a machine learning model for heart disease prediction, we evaluate its performance using various metrics such as accuracy, precision, recall, F1-score, and AUC-ROC score.

The Random Forest Classifier achieved the highest accuracy (90%) and AUC-ROC score (92%), indicating that it is the best model among the tested algorithms.

Logistic Regression and KNN had lower performance, which may indicate that the dataset has complex patterns that require more advanced models.

```
print("The score for Random Forest Classifier is {}% with {} estimators.".format(rf_scores[1]*100, [100, 500]))
```

```
The score for Random Forest Classifier is 84.0% with [100, 500] estimators.
```

Conclusion

The study on heart disease prediction using machine learning models demonstrates that advanced algorithms can effectively identify individuals at risk. Among the models tested, the Random Forest Classifier achieved the highest accuracy (90%) and AUC-ROC score (92%), making it the most reliable model for this task. Other models, such as Support Vector Machine (SVM) and Decision Tree Classifier, also performed well but with slightly lower accuracy. early heart disease detection, assisting healthcare professionals in making faster and more accurate diagnoses. However, integrating these models into clinical practice requires further validation with real-world patient data.

Rajyalakshmi Rajana

SUC – 2360530098

BATCH – 1