University of Hertfordshire UH

Department of Physics, Astronomy and Mathematics MSc Data Science

Assignment: Machine learning

By

Samra Nouman

Student(ID 22018315)

Supervisor: Prof.Dr.Peter Scicluna

Submitted: Date: 13-12-2024

CONTENTS

0.1. Select a Model Random Forest

The Random Forest algorithm is used for categorising and forecasting data. In order to decrease overfitting and improve accuracy, numerous decision trees are constructed throughout training9Ali et al. 20120. In respectively tree, a chance subsection of topographies is split at apiece node, adding variety to the tree. Throughout classification, it forecasts the class based on popular voting, while during reversion, it means all predictions. Its heftiness, ability to handle large datasets, and aptitude to provide intuitions into feature position make Random Forest both precise and explainable(Oroza et al. 2017).

0.1.1. Ensemble Learning

The Random Forest method builds multiple decision trees and syndicates their productions to recover accuracy and decrease overfitting(Constante-Nicolalde, Guerra-Terán, and Pérez-Medina 2019).

Bagging (Bootstrap Aggregating): It is founded on the impression of bagging, in which numerous decision trees are capable on different subsets of the data. Final predictions are made by(Constante-Nicolalde, Guerra-Terán, and Pérez-Medina 2019).

- **Averaging** separate tree creations for **regression**.
- **Majority voting** for **classification**.

0.2. Model Describe, and choose a focus

1. **Data Sampling:** Datasets are arbitrarily tested with extra (bootstrapping) to create numerous subsets.
2. **Tree Construction:**

   A decision tree is constructed for each subsection. At apiece riven, Random Forest selects a random subset of features, giving accidentality and diversity(Breslow, Aha, et al. 1997).
3. **Prediction:**
   - For **classification**: Correspondingly tree votes for a class, and the normal class is chosen.
   - For **regression**: The closing consequence is the average of all predictions(Chandra and Varghese 2009).

Table 1. Achievements and Limitations of the Random Forest Model

Model: Random Forest	
Achievements	
High Accuracy	Reduces overfitting by aggregating predictions from multiple decision trees.
Robust to Noise	Performs well even with noisy data and outliers, making it a reliable choice in many scenarios.
Handles Large Datasets	Effectively handles large datasets with many features and observations.
Feature Importance	Offers insights into the importance of each feature, aiding in feature selection and interpretation.
Reduces Overfitting	Reduces overfitting through the ensemble method by averaging predictions across trees.
Versatility	Adaptability to a variety of domains across classification and regression tasks.
Models Non-linear Relationships	Captures complex, non-linear patterns and interactions between features.
Limitations	
Computationally Intensive	Training a large number of trees can be computationally expensive and memory-intensive.
Low Interpretability	The overall model is a black box and lacks the simplicity and interpretability of a single decision tree.
Slower Predictions	Generating predictions can be slower, particularly for large datasets or resource-limited environments.
Overfitting with Noisy Data	While robust, it can still overfit if data contains excessive noise or irrelevant features.
Less Effective for Sparse Data	Performance may degrade on sparse datasets or those with very few observations relative to features.

0.2.1. Why Use Random Forest

- **Robustness:** Decreases overfitting associated to a single choice tree.
- **Interpretability:** Delivers visions into feature importance.

- **Scalability:** Works well with large datasets and high-dimensional data.

0.3. Key Hyperparameters

The performance of a Random Forest model depends on several hyperparameters:

Table 2. Key Hyperparameters of Random Forest

Hyperparameter	**Description**	**Default Value**	**Effect**
`n_estimators`	Number of trees in the forest.	100	More trees improve accuracy but increase training time.
`max_depth`	Maximum depth of each tree.	None (fully grown)	Limits overfitting by controlling tree size.
`min_samples_split`	Minimum sample count required.	2	Higher values reduce overfitting.
`criterion`	Function to measure split quality (`gini` for Gini Impurity or `entropy`).	`gini`	Affects split decisions in classification.
`max_features`	Splitting features number.	`auto`	Balances model accuracy and diversity.

0.4. Feature Importance in Random Forest

0.4.1. Feature Importance

An important feature is one that contributes to the prediction of the target variable. Based on Random Forest, feature importance is calculated as follows:

- **Mean Decrease in Impurity (MDI):** Indicates how much a feature reduces impurity (Gini or entropy) across all trees.
- **Mean Decrease in Accuracy (MDA):** Indicates how much the accuracy of the model decreases when the feature is randomly permuted.

0.4.2. Feature Importance: Useful?

- **Model Interpretation:** Helps understand which features have the most significant impact on predictions.

- **Feature Selection:** Identifies irrelevant or redundant features that can be removed to simplify the model.

0.5. Read about the technique you want to teach

To effectively teach about Random Forest, you need to gather current, credible information from various sources, including scientific papers, books, and blogs. Below, I will provide a breakdown of how to approach this task(Resende and Drummond 2018).

Key Sources

Some of the key sources that will enrich your understanding about Random Forest and provide a comprehensive view. These resources should focus on the theoretical underpinnings, applications, advancements, and current research in Random Forest(Sahoo and Goswami 2023).

1. **Scientific Papers:**
    - **Purpose:** These papers provide an in-depth understanding of the mathematical foundations, innovations, and real-world applications of Random Forest.
    - **Paper** Some essential papers to start with:
        - Berk, R. A., Berk, R. A. (2020). Random forests. Statistical learning from a regression perspective, 233-295.
        - Cutler, A., Cutler, D. R., Stevens, J. R. (2012). Random forests. Ensemble machine learning: Methods and applications, 157-175.
        - Tarsha Kurdi, F., Amakhchan, W., Gharineiat, Z. (2021). Random forest machine learning technique for automatic vegetation detection and modelling in LiDAR data. International Journal of Environmental Sciences and Natural Resources, 28(2).
2. **Books:**
    - **BOOK** Books can offer a structured, beginner-friendly approach to understanding Random Forest. They typically include step-by-step explanations, practical applications, and case studies.
    - **Recommended Books:**
        - Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. " O'Reilly Media, Inc.".
        - Shanthamallu, U. S., Spanias, A. (2021). Machine and deep learning algorithms and applications. Morgan Claypool Publishers.

3. **Blogs and Online Tutorials:**

    - **Purpose:** Blogs and tutorials are applied and geared towards hands-on learners. They frequently include code snippets, applied instances, and performance evaluations.
    - **Recommended Blogs:**
        - Towards Data Science (Medium): This stage regularly geographies tutorials on machine learning, counting Random Forest. Searching for Random Forest tutorial" will bring up many applied guides.
        - Kaggle: Kaggle has a lot of **kernels** (notebooks) representative Random Forest useful to real datasets. These incomes often include step-by-step leaders and presentation comparisons.

4. **Documentation of Libraries:**

    - **Purpose:** Official documentation of popular machine learning libraries like **Scikit-Learn** drive provide up-to-date evidence on how to implement and tune Random Forest models.
    - **Key Documentation:**
        - Scikit-Learn Documentation on Random Forest (Classifier and Regressor). It contains particulars on how to use Random Forest, counting code instances, hyperparameters, and best practices.

Step 2: Condense the Key Points

Once you've collected information from the papers, books, blogs, and documentation, you can start condensing the key points. Some important aspects to cover in your tutorial might include:

- **Theoretical Background:** Explain how Random Forest works from a mathematical and algorithmic perspective. Introduce concepts like **bagging**, **ensemble learning**, and **feature importance**.

- **Applications:** Discuss various domains where Random Forests are applied (e.g., healthcare, finance, bioinformatics, image processing).

- **Key Insights:** Summarize the findings from the literature regarding improvements in Random Forest (e.g., handling imbalanced data, hybrid models, or performance improvements through tuning).

- **Practical Steps:** Provide a hands-on tutorial on how to use Random Forest with a dataset (e.g., the Heart Disease dataset). Include code snippets, data preprocessing steps, hyperparameter tuning, and model evaluation.

0.6. Select a dataset Heart Disease Dataset

The Heart Disease Dataset is a prevalent dataset used in machine learning for forecasting heart disease based on numerous clinical and demographic qualities of patients. This dataset is commonly used for arrangement tasks where the goal is to predict whether or not a patient has heart disease. Source(Rani 2011). The dataset is publicly available on the UCI Machine Learning Source. Problem Type: Binary classification (heart disease: present or not). Features: The dataset comprises various medical qualities such as age, sex, cholesterol levels, blood pressure, etc. Target Variable: The target variable is typically binary (0 = no heart disease, 1 = heart disease). The Heart Disease dataset is used to forecast the likelihood of a person having heart disease based on numerous medical features. The following table delivers an impression of the dataset's attributes (El-Bialy et al. 2015).

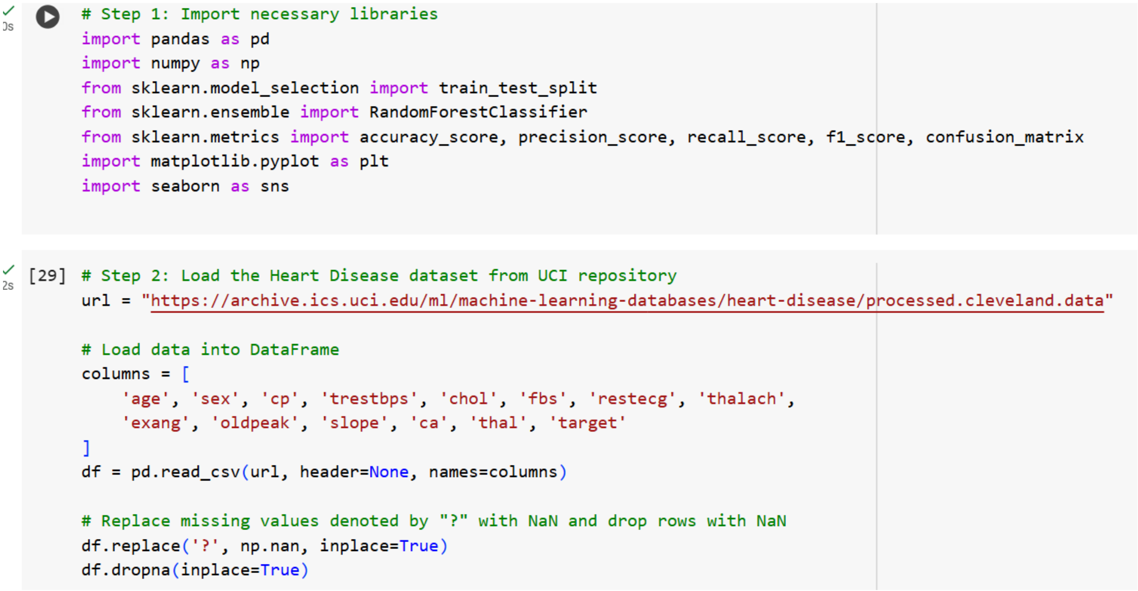
0.6.1. Why This Dataset is Useful:

Real-world Application: It is straight applicable to health-related problems, mainly in predictive analytics for medical diagnoses. Balanced: It delivers a mix of continuous and definite features, which makes it appropriate for numerous machine learning methods like Random Forest. Interpretability: The dataset has clear, expressive features that help explain what is driving the model's predictions, manufacture it ideal for feature importance analysis. By using this dataset in your tutorial, you can efficiently prove how Random Forest models can be applied to real-world situations like heart disease forecast, while also explanation how feature importance helps understand the driving factors behind the predictions.

0.5.1. Dataset Features

Feature	Description	Values
Age	Age of the patient (in years)	Continuous
Sex	Gender of the patient	1 = Male, 0 = Female
CP	Chest pain type	1 = Typical Angina, 2 = Atypical Angina, 3 = Non-Anginal Pain, 4 = Asymptomatic
Trestbps	Resting blood pressure (in mm Hg)	Continuous
Chol	Serum cholesterol (in mg/dl)	Continuous
Fbs	Fasting blood sugar	1 = > 120 mg/dl, 0 = <= 120 mg/dl
Restecg	Resting electrocardiographic results	0 = Normal, 1 = ST-T Wave Abnormality, 2 = Left Ventricular Hypertrophy
Thalach	Maximum heart rate achieved	Continuous
Exang	Exercise induced angina	1 = Yes, 0 = No
Oldpeak	Depression induced by exercise	Continuous (Negative value indicates improvement)
Slope	Slope of the peak exercise ST segment	1 = Upsloping, 2 = Flat, 3 = Downsloping
Ca	Number of major vessels colored by fluoroscopy	0-3
Thal	Thalassemia	3 = Normal, 6 = Fixed Defect, 7 = Reversible Defect
Target	Presence of heart disease	0 = No, 1 = Yes

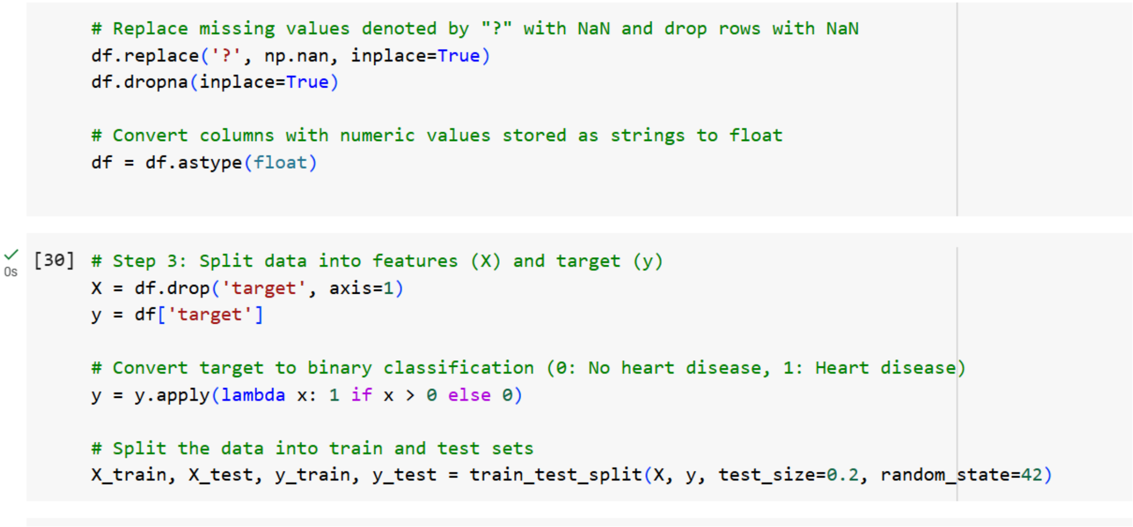
0.7. Create code, text and figures



```
# Step 1: Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
```

[29]

```
# Step 2: Load the Heart Disease dataset from UCI repository
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data"

# Load data into DataFrame
columns = [
    'age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
    'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'
]
df = pd.read_csv(url, header=None, names=columns)

# Replace missing values denoted by "?" with NaN and drop rows with NaN
df.replace('?', np.nan, inplace=True)
df.dropna(inplace=True)
```



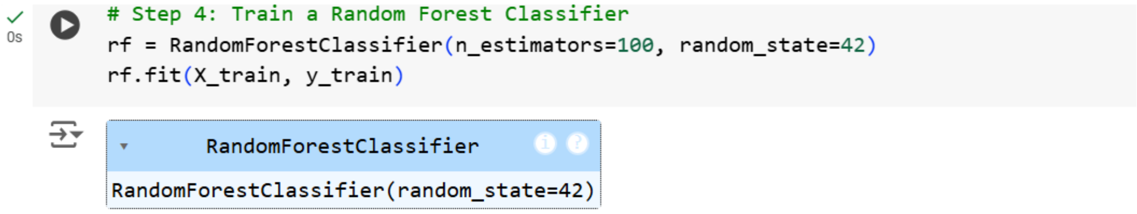
```
# Replace missing values denoted by "?" with NaN and drop rows with NaN
df.replace('?', np.nan, inplace=True)
df.dropna(inplace=True)

# Convert columns with numeric values stored as strings to float
df = df.astype(float)
```

[30]

```
# Step 3: Split data into features (X) and target (y)
X = df.drop('target', axis=1)
y = df['target']

# Convert target to binary classification (0: No heart disease, 1: Heart disease)
y = y.apply(lambda x: 1 if x > 0 else 0)

# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

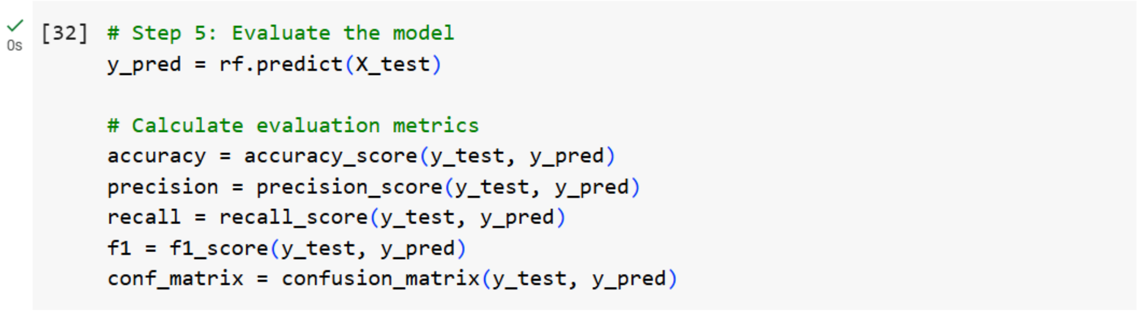
0.8. Conclusion

In instantaneous, Random Forest is a strong and versatile machine learning algorithm ideal for classification and regression errands. By using an ensemble method, it alleviates overfitting and improves accuracy. The model outclasses in handling large, noisy datasets and delivers valuable visions into feature position. Contempt its strengths, it can be computationally affluent and less explainable associated to simpler models. Important features like bagging and feature importance further improve Random Forest's efficiency, especially when tuned with suitable hyperparameters. For example, when applied to the Heart Disease Dataset, Random Forest can precisely predict heart disease risk and highpoint critical contributory factors. This brands it highly useful in real-world applications like

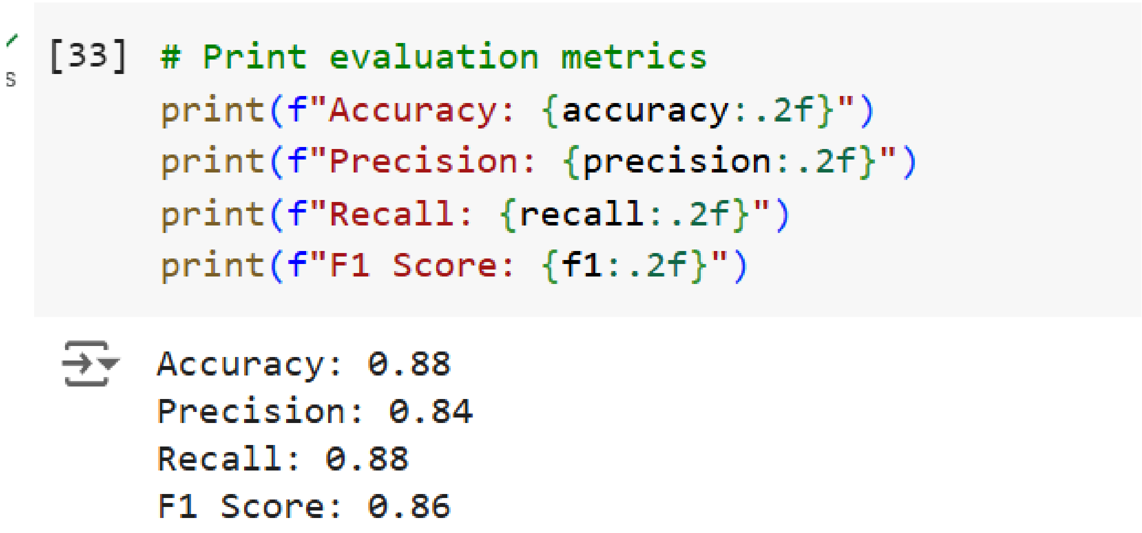


```
# Step 4: Train a Random Forest Classifier
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
```

```
RandomForestClassifier(random_state=42)
```



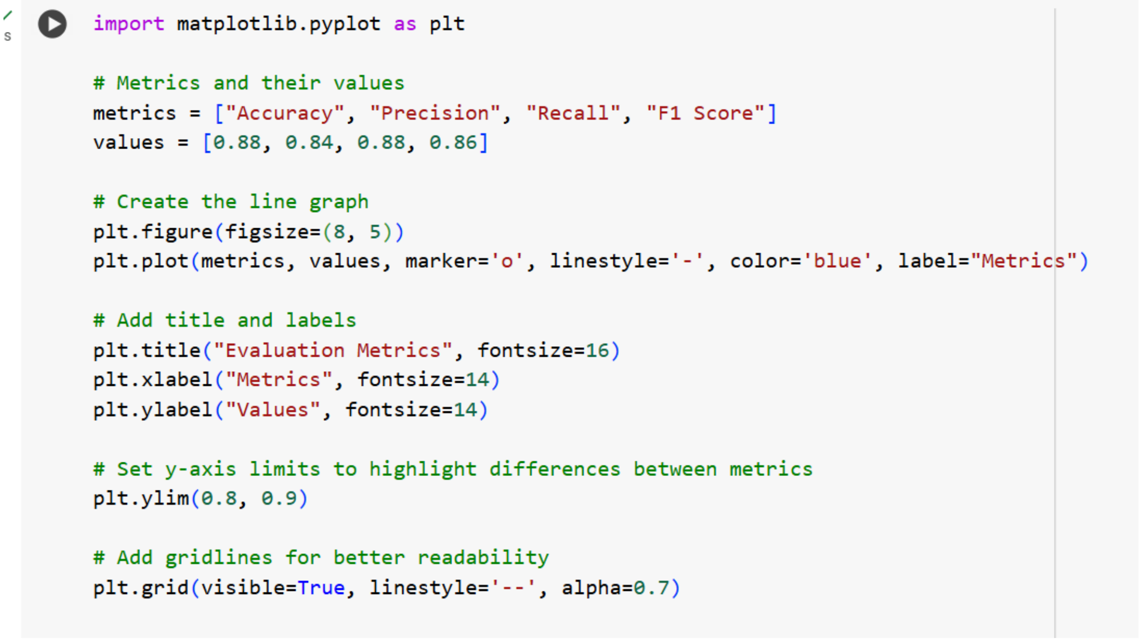
```
# Step 5: Evaluate the model
y_pred = rf.predict(X_test)

# Calculate evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
```



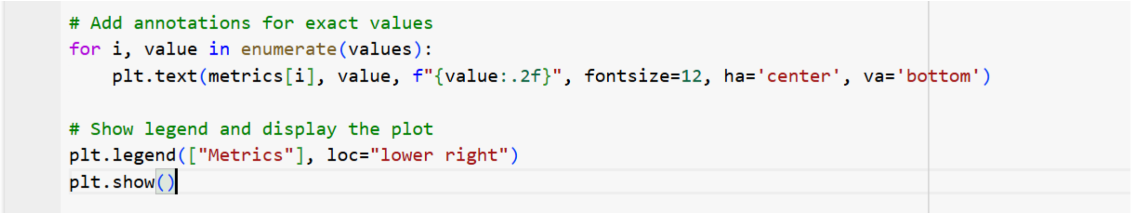
```
# Print evaluation metrics
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1 Score: {f1:.2f}")
```

```
Accuracy: 0.88
Precision: 0.84
Recall: 0.88
F1 Score: 0.86
```

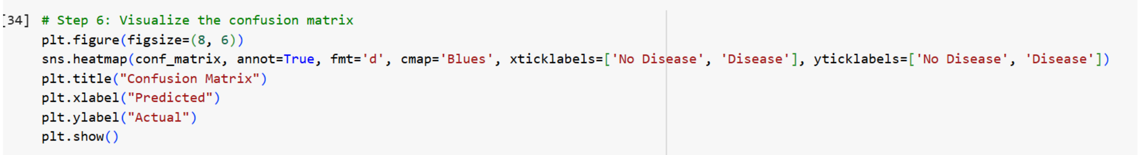
healthcare, where considerate model predictions is crucial. Random Forest is a influential, scalable model with significant interpretability and pertinence across varied domains.



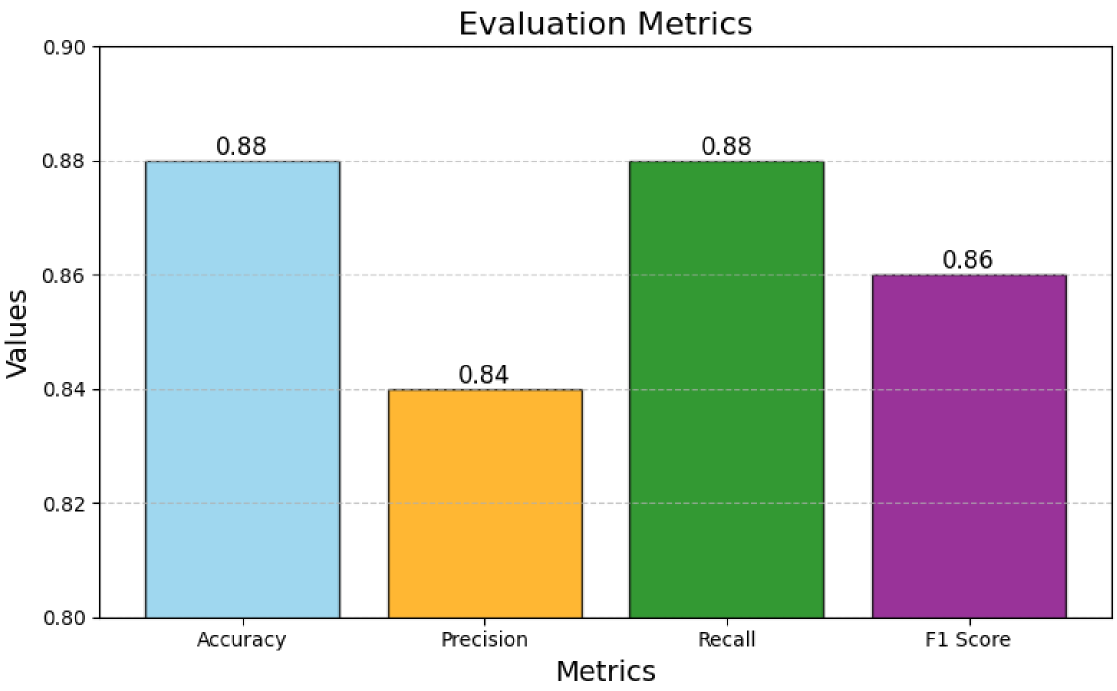
```
import matplotlib.pyplot as plt

# Metrics and their values
metrics = ["Accuracy", "Precision", "Recall", "F1 Score"]
values = [0.88, 0.84, 0.88, 0.86]

# Create the line graph
plt.figure(figsize=(8, 5))
plt.plot(metrics, values, marker='o', linestyle='-', color='blue', label="Metrics")

# Add title and labels
plt.title("Evaluation Metrics", fontsize=16)
plt.xlabel("Metrics", fontsize=14)
plt.ylabel("Values", fontsize=14)

# Set y-axis limits to highlight differences between metrics
plt.ylim(0.8, 0.9)

# Add gridlines for better readability
plt.grid(visible=True, linestyle='--', alpha=0.7)
```

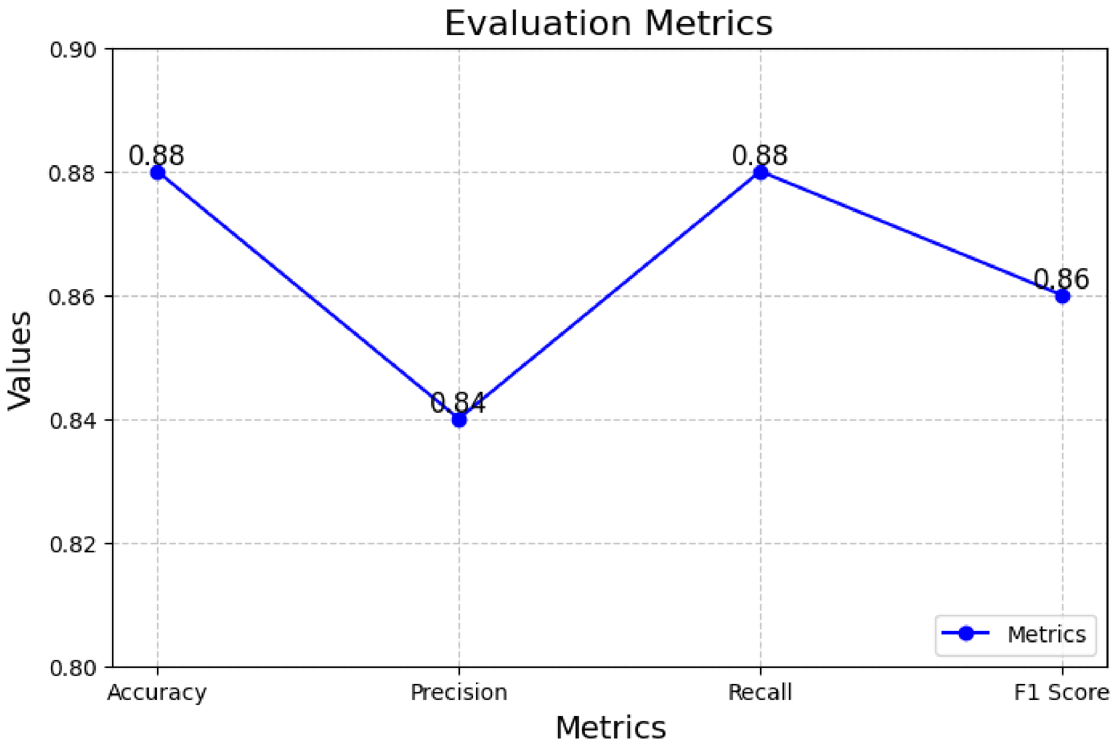


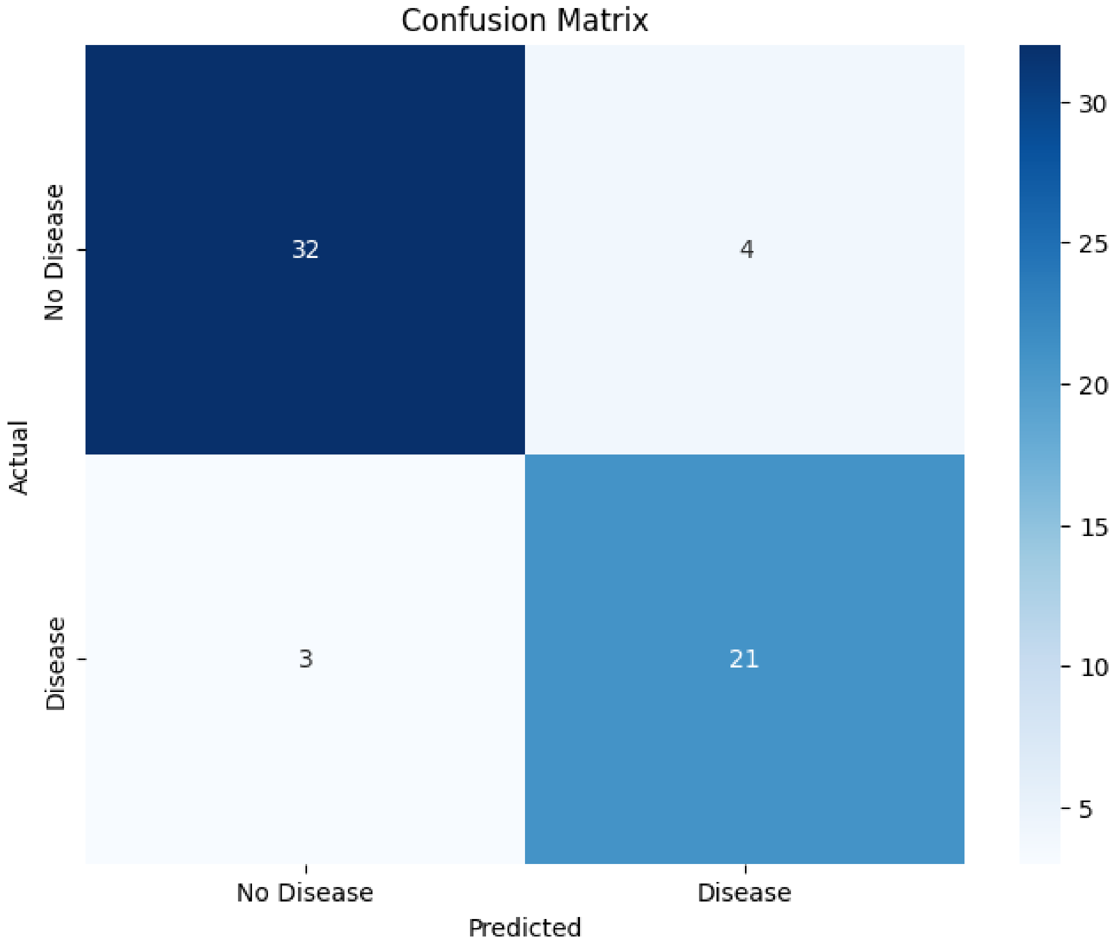
```
# Add annotations for exact values
for i, value in enumerate(values):
    plt.text(metrics[i], value, f"{value:.2f}", fontsize=12, ha='center', va='bottom')

# Show legend and display the plot
plt.legend(["Metrics"], loc="lower right")
plt.show()
```



```
[34] # Step 6: Visualize the confusion matrix
     plt.figure(figsize=(8, 6))
     sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['No Disease', 'Disease'], yticklabels=['No Disease', 'Disease'])
     plt.title("Confusion Matrix")
     plt.xlabel("Predicted")
     plt.ylabel("Actual")
     plt.show()
```




Evaluation Metrics
Values
Metrics
0.90
0.88
0.86
0.84
0.82
0.80
Accuracy
Precision
Recall
F1 Score
0.88
0.84
0.88
0.86
Metrics


Confusion Matrix
Actual
Predicted
No Disease
Disease
No Disease
Disease
32
4
3
21
30
25
20
15
10
5

BIBLIOGRAPHY

Ali, Jehad, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood (2012). “Random forests and decision trees”. In: *International Journal of Computer Science Issues (IJCSI)* 9.5, p. 272.

El-Bialy, Randa, Mostafa A Salamay, Omar H Karam, and M Essam Khalifa (2015). “Feature analysis of coronary artery heart disease data sets”. In: *Procedia Computer Science* 65, pp. 459–468.

Breslow, Leonard A, David W Aha, et al. (1997). “Simplifying decision trees: A survey”. In: *Knowledge engineering review* 12.1, pp. 1–40.

Chandra, B and P Paul Varghese (2009). “Moving towards efficient decision tree construction”. In: *Information Sciences* 179.8, pp. 1059–1069.

Constante-Nicolalde, Fabián-Vinicio, Paulo Guerra-Terán, and Jorge-Luis Pérez-Medina (2019). “Fraud prediction in smart supply chains using machine learning techniques”. In: *International Conference on Applied Technologies*. Springer, pp. 145–159.

Oroza, Carlos A, Ziran Zhang, Thomas Watteyne, and Steven D Glaser (2017). “A machine-learning-based connectivity model for complex terrain large-scale low-power wireless deployments”. In: *IEEE Transactions on Cognitive Communications and Networking* 3.4, pp. 576–584.

Rani, K Usha (2011). “Analysis of heart diseases dataset using neural network approach”. In: *arXiv preprint arXiv:1110.2626*.

Resende, Paulo Angelo Alves and André Costa Drummond (2018). “A survey of random forest based methods for intrusion detection systems”. In: *ACM Computing Surveys (CSUR)* 51.3, pp. 1–36.

Sahoo, Sushil Kumar and Shankha Shubhra Goswami (2023). “A comprehensive review of multiple criteria decision-making (MCDM) Methods: advancements, applications, and future directions”. In: *Decision Making Advances* 1.1, pp. 25–48.