# Heart Disease Prediction

**Name:** Shrusti Chintawar
**Roll No:** 09
**Course:** DMBI CA-2
**Date:** April 16, 2025

## 1. Abstract

This project applies a machine learning approach to predict the likelihood of heart disease using the Random Forest Classifier. With the UCI Heart Disease dataset, various features such as chest pain type, cholesterol levels, and maximum heart rate are analyzed. Exploratory Data Analysis (EDA) is performed, followed by model training and evaluation. Results are visualized with relevant graphs including a confusion matrix and feature importance chart.

## 2. Introduction

Heart disease remains a top health concern globally. Early prediction can reduce risk through early interventions. Machine learning offers automated and accurate prediction models. This project utilizes a Random Forest Classifier due to its robustness and interpretability.

## 3. Objective

- Predict heart disease presence using a machine learning model.
- Visualize feature influence on disease.
- Evaluate classifier performance with suitable metrics.

## 4. Dataset Description

**Source:** UCI Repository / Kaggle
**Features:**

- age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal **Target:** target (1 = heart disease, 0 = no disease)

---

## 5. Tools and Libraries Used

- Python (Colab)
- Libraries: pandas, numpy, matplotlib, seaborn, sklearn

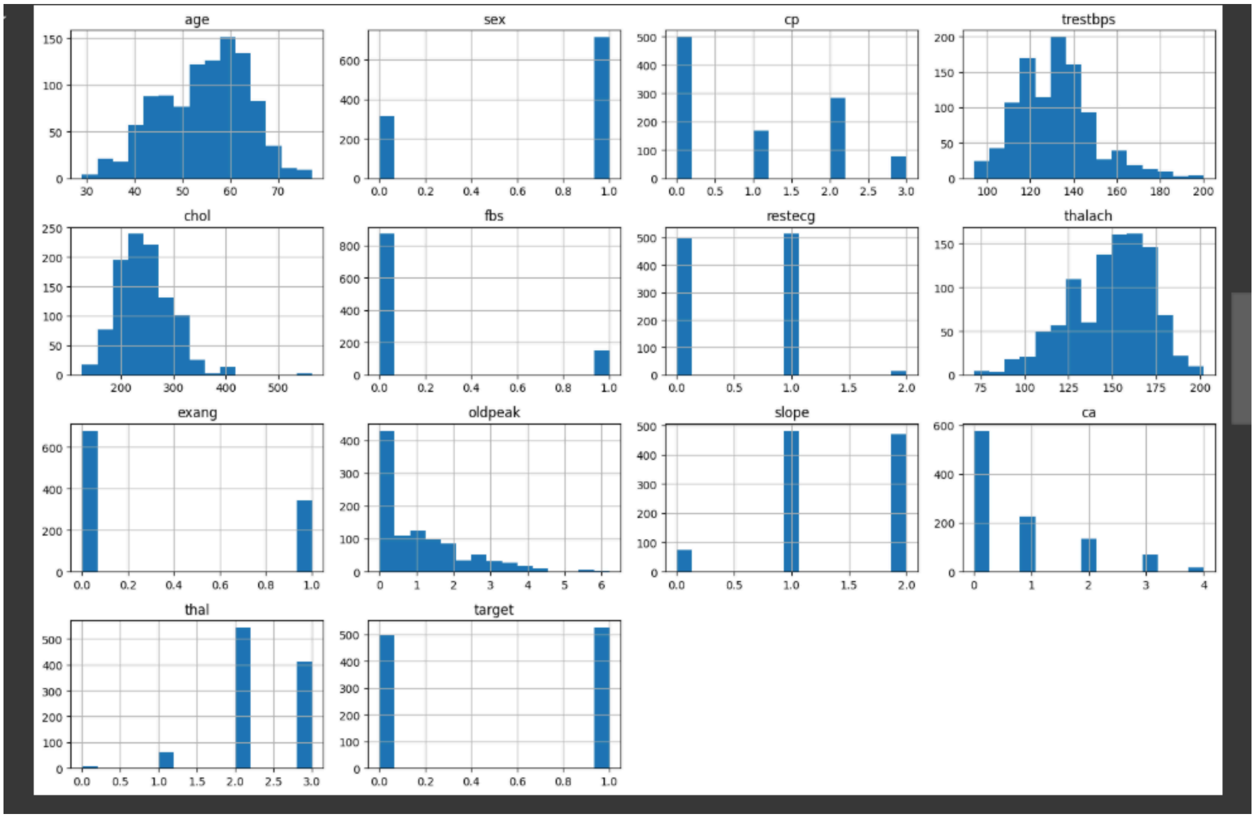
---

## 6. Exploratory Data Analysis (EDA)

- **Missing Values:** None present.

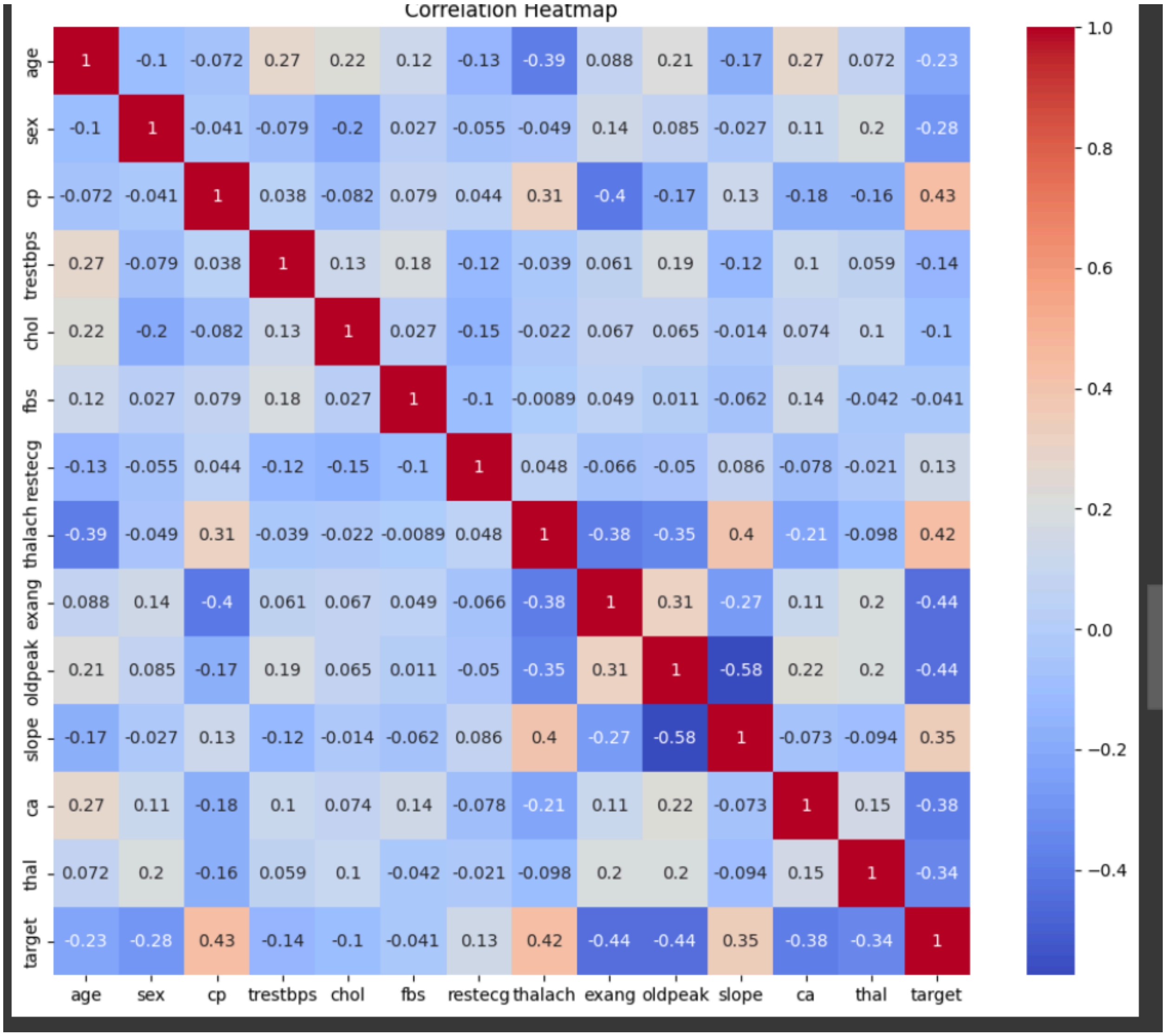
Exploratory Data Analysis is a critical step in understanding the structure and patterns within the dataset. It helps identify trends, spot anomalies, and test assumptions using visual and quantitative methods. In this project, EDA is used to assess the distribution of variables, check for class balance, and understand feature correlations.

- **Histograms:**



```
df.hist(bins=15, figsize=(15, 10))
```

- **Correlation Heatmap:**



| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | thal | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| age | 1 | -0.1 | -0.072 | 0.27 | 0.22 | 0.12 | -0.13 | -0.39 | 0.088 | 0.21 | -0.17 | 0.27 | 0.072 | -0.23 |
| sex | -0.1 | 1 | -0.041 | -0.079 | -0.2 | 0.027 | -0.055 | -0.049 | 0.14 | 0.085 | -0.027 | 0.11 | 0.2 | -0.28 |
| cp | -0.072 | -0.041 | 1 | 0.038 | -0.082 | 0.079 | 0.044 | 0.31 | -0.4 | -0.17 | 0.13 | -0.18 | -0.16 | 0.43 |
| trestbps | 0.27 | -0.079 | 0.038 | 1 | 0.13 | 0.18 | -0.12 | -0.039 | 0.061 | 0.19 | -0.12 | 0.1 | 0.059 | -0.14 |
| chol | 0.22 | -0.2 | -0.082 | 0.13 | 1 | 0.027 | -0.15 | -0.022 | 0.067 | 0.065 | -0.014 | 0.074 | 0.1 | -0.1 |
| fbs | 0.12 | 0.027 | 0.079 | 0.18 | 0.027 | 1 | -0.1 | -0.0089 | 0.049 | 0.011 | -0.062 | 0.14 | -0.042 | -0.041 |
| restecg | -0.13 | -0.055 | 0.044 | -0.12 | -0.15 | -0.1 | 1 | 0.048 | -0.066 | -0.05 | 0.086 | -0.078 | -0.021 | 0.13 |
| thalach | -0.39 | -0.049 | 0.31 | -0.039 | -0.022 | -0.0089 | 0.048 | 1 | -0.38 | -0.35 | 0.4 | -0.21 | -0.098 | 0.42 |
| exang | 0.088 | 0.14 | -0.4 | 0.061 | 0.067 | 0.049 | -0.066 | -0.38 | 1 | 0.31 | -0.27 | 0.11 | 0.2 | -0.44 |
| oldpeak | 0.21 | 0.085 | -0.17 | 0.19 | 0.065 | 0.011 | -0.05 | -0.35 | 0.31 | 1 | -0.58 | 0.22 | 0.2 | -0.44 |
| slope | -0.17 | -0.027 | 0.13 | -0.12 | -0.014 | -0.062 | 0.086 | 0.4 | -0.27 | -0.58 | 1 | -0.073 | -0.094 | 0.35 |
| ca | 0.27 | 0.11 | -0.18 | 0.1 | 0.074 | 0.14 | -0.078 | -0.21 | 0.11 | 0.22 | -0.073 | 1 | 0.15 | -0.38 |
| thal | 0.072 | 0.2 | -0.16 | 0.059 | 0.1 | -0.042 | -0.021 | -0.098 | 0.2 | 0.2 | -0.094 | 0.15 | 1 | -0.34 |
| target | -0.23 | -0.28 | 0.43 | -0.14 | -0.1 | -0.041 | 0.13 | 0.42 | -0.44 | -0.44 | 0.35 | -0.38 | -0.34 | 1 |

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

---

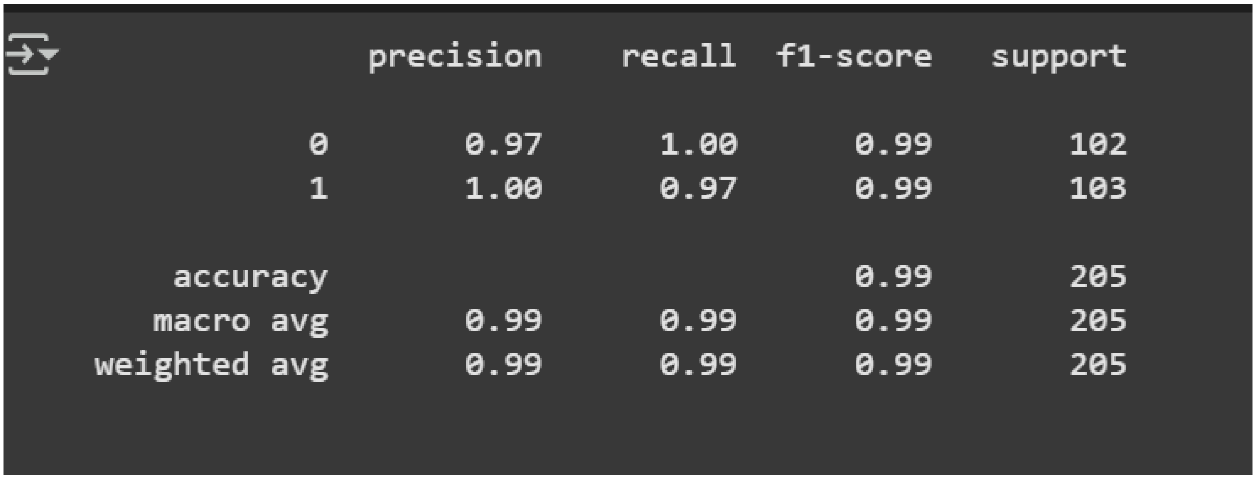
## 7. Data Preprocessing

Data preprocessing involves preparing the raw data for machine learning algorithms. This includes splitting the data into features and target variables, scaling the features to bring them to a common range, and dividing the dataset into training and testing sets. These steps ensure that the model learns effectively and performs accurately on unseen data.

.

## 8. Model Implementation

Model implementation refers to the process of training a machine learning algorithm on the preprocessed dataset. In this case, the Random Forest Classifier is chosen for its ability to handle high-dimensional data and avoid overfitting. The model is trained on the training dataset and then used to predict outcomes on the test dataset

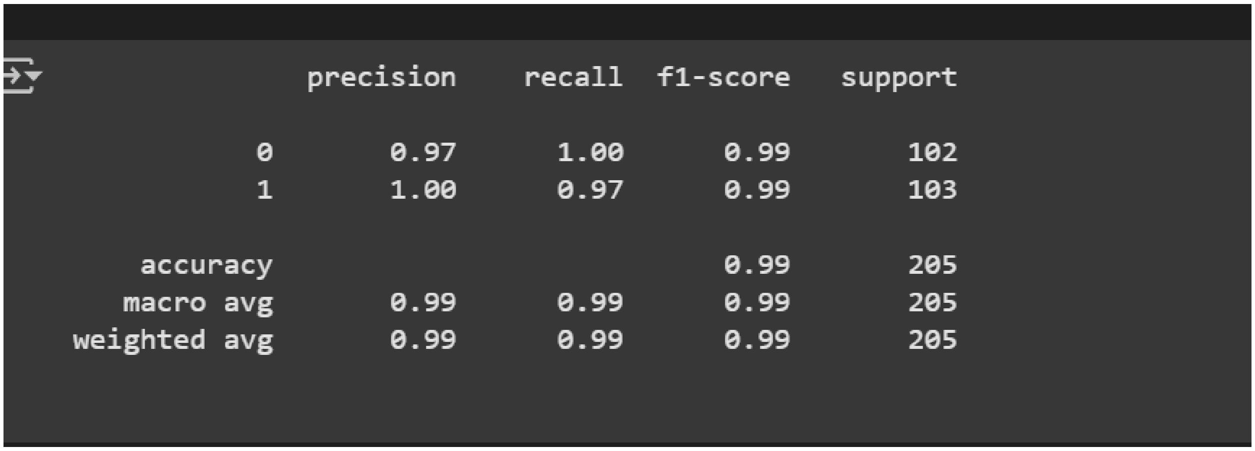


```
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       102
           1       1.00      0.97      0.99       103

    accuracy                           0.99       205
   macro avg       0.99      0.99      0.99       205
weighted avg       0.99      0.99      0.99       205
```

## 9. Evaluation Metrics

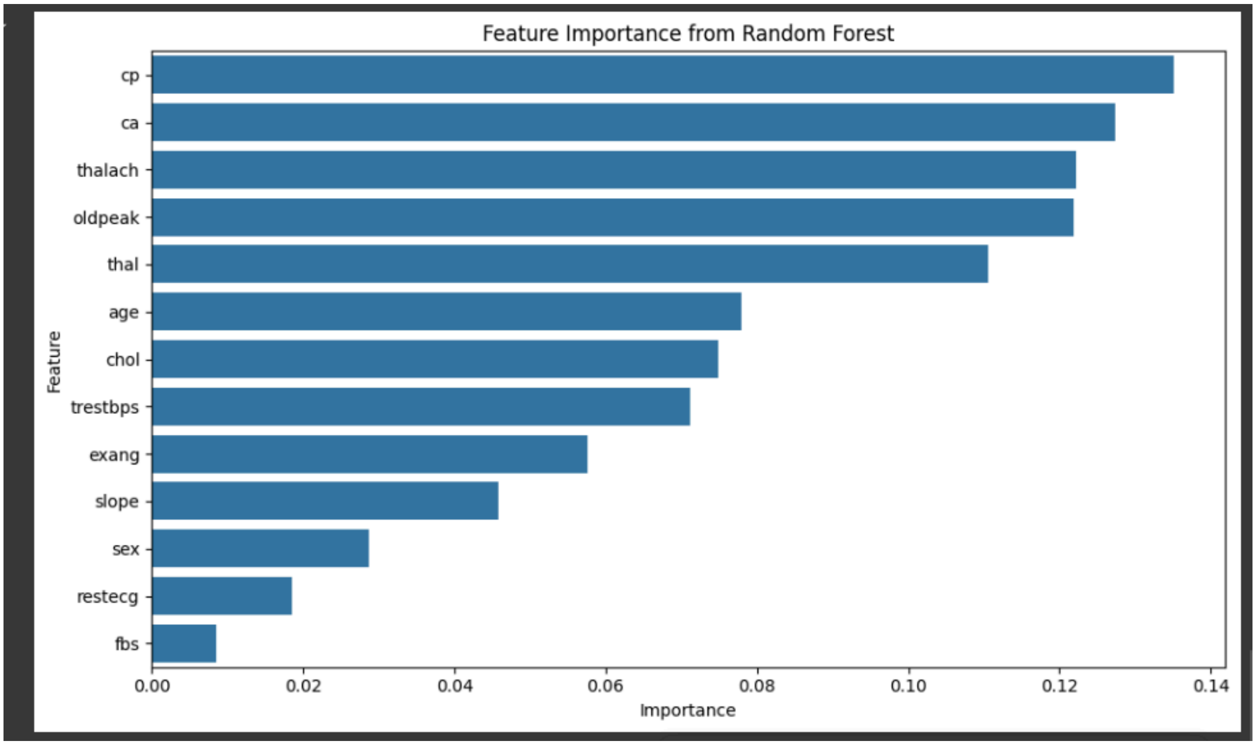
- **Classification Report:**

print(classification_report(y_test, y_pred))



```
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       102
           1       1.00      0.97      0.99       103

    accuracy                           0.99       205
   macro avg       0.99      0.99      0.99       205
weighted avg       0.99      0.99      0.99       205
```

- **Feature Importance Plot:**



---

## 10. Results and Insights

- Most significant features: `cp`, `thal`, `ca`
- High recall and precision indicate robust performance.
- Visualizations helped in understanding both data and model behavior.

---

## 11. Conclusion

The Random Forest classifier achieved effective and interpretable predictions for heart disease. With good accuracy and insightful feature importance, this model can support healthcare decisions.

---

## 12. Future Scope

- Use of grid search for hyperparameter tuning
- Ensemble comparisons (e.g., Gradient Boosting)
- Real-time integration with wearable data

---

## 13. References

- UCI Machine Learning Repository
- Kaggle: Heart Disease Dataset
- Scikit-learn Documentation