Heart Disease Prediction – Machine Learning Project Documentation

Name: Devadevan S
Dataset: Heart_Disease_Prediction.csv
Platform: Google Colaboratory

---

1. Abstract

Heart disease is one of the leading causes of mortality worldwide. Early prediction and diagnosis can significantly reduce risks and improve patient outcomes. This project focuses on building a **Machine Learning–based Heart Disease Prediction system** using clinical and medical attributes. The objective is to predict whether a patient has heart disease based on diagnostic indicators such as age, cholesterol level, blood pressure, heart rate, and other health parameters. The project follows a complete ML pipeline including data preprocessing, model training, evaluation, and interpretation.

---

2. Problem Statement

To develop a supervised machine learning model that accurately predicts the **presence or absence of heart disease** in patients using historical medical data.

Business / Medical Value: - Helps doctors in early diagnosis - Reduces manual diagnostic errors - Supports clinical decision-making

Type of Problem: Supervised Classification
Target Variable: Presence of heart disease (0 = No disease, 1 = Disease)

---

3. Dataset Description

The dataset contains real-world clinical attributes used in heart disease diagnosis.

Each row: One patient record
Each column: A medical measurement or diagnostic indicator

Key Attributes:

- `age`: Age of the patient
- `sex`: Gender (1 = male, 0 = female)
- `cp`: Chest pain type
- `trestbps`: Resting blood pressure

- `chol`: Serum cholesterol
- `fbs`: Fasting blood sugar
- `restecg`: Resting ECG results
- `thalach`: Maximum heart rate achieved
- `exang`: Exercise-induced angina
- `oldpeak`: ST depression
- `target`: Heart disease presence

---

4. Data Understanding & Exploration

Steps performed: - Loaded dataset using Pandas - Checked shape, size, and data types - Identified missing values and handled them - Explored distributions using visualizations

```
print(df.shape)
print(df.info())
print(df.isnull().sum())
```

EDA helped identify patterns, correlations, and potential outliers.

---

5. Data Preprocessing

5.1 Missing Value Handling

- Numerical columns: Filled using **median**
- Dataset contained minimal or no missing values

5.2 Encoding Categorical Variables

- Converted categorical features into numeric form using **Label Encoding**

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['sex'] = le.fit_transform(df['sex'])
```

5.3 Feature Scaling

- Applied **StandardScaler** to normalize numerical features
- Improves performance of distance-based models

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

---

6. Train-Test Split

The dataset was split into training and testing sets to evaluate generalization performance.

- Training set: 70%
- Testing set: 30%

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
```

---

7. Model Building

Selected Algorithms:

- Logistic Regression
- Decision Tree
- Random Forest (Final Model)

Why Random Forest?

- Handles non-linear relationships
- Reduces overfitting using ensemble learning
- Provides feature importance
- Works well with medical datasets

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

---

8. Model Evaluation

The model was evaluated using standard classification metrics:

- **Accuracy** – Overall correctness
- **Precision** – Correct positive predictions
- **Recall** – Ability to detect disease cases
- **F1-Score** – Balance between precision and recall

```
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score
print(classification_report(y_test, y_pred))
```

Confusion Matrix

Visualized using Seaborn heatmap to analyze prediction errors.

---

9. Results & Interpretation

- Random Forest achieved high accuracy and stable performance
- Important features influencing prediction included:
  - Chest pain type
  - Maximum heart rate
  - Cholesterol level
  - Age

Feature importance analysis helped in understanding model decisions.

---

10. AI Logic & Explanation

- The model learns patterns from historical medical data
- Higher cholesterol and abnormal heart rate increase disease risk
- Ensemble learning improves prediction reliability

Possible Improvements:

- Use XGBoost or Gradient Boosting
- Apply SMOTE for class imbalance
- Add more patient history data

---

11. Deployment Readiness (Conceptual)

The trained model can be deployed using: - Flask / FastAPI - Input: Patient medical details - Output: Disease prediction (Yes/No)

---

12. Conclusion

This project successfully demonstrates the application of machine learning in healthcare for heart disease prediction. By following a structured ML pipeline—data preprocessing, model training, evaluation, and interpretation—the Random Forest model proved effective in identifying patients at risk of heart disease. Such predictive systems can support early diagnosis, reduce healthcare costs, and assist medical professionals in decision-making.

---

13. Tools & Technologies Used

- Python
- Pandas, NumPy
- Matplotlib, Seaborn
- Scikit-learn
- Google Colaboratory

---

14. Project Files

- Dataset: `Heart_Disease_Prediction.csv`
- Notebook: `project.ipynb`

End of Documentation