Navigating Heart Disease: Insights and Analysis for Improved Diagnosis

(Project Proposal)

Kazi Salith Ur Rahman
ID: 202291994

***Instructor**: Ebrahim Karami*
Department of Electrical and Computer Engineering
Faculty of Engineering and Applied Science
Memorial University of Newfoundland

Introduction:

Heart disease remains a leading cause of mortality worldwide, necessitating improved diagnostic approaches to combat its prevalence. Heart disease is considered as one of the major causes of death throughout the world. It cannot be easily predicted by the medical practitioners as it is a difficult task which demands expertise and higher knowledge for prediction [1].

"Navigating Heart Disease: Insights and Analysis for Improved Diagnosis" delves into the complexities of heart disease, offering a comprehensive analysis of current diagnostic methods. This exploration highlights advancements in medical technology, the role of early detection, and innovative strategies for accurate diagnosis. By integrating cutting-edge research and practical insights, this discussion aims to enhance understanding and foster better clinical outcomes, ultimately paving the way for more effective management and treatment of heart disease.

Project Description:

The project "Navigating Heart Disease: Insights and Analysis for Improved Diagnosis" aims to comprehensively analyze a heart disease dataset to identify key insights that can improve diagnostic accuracy and treatment strategies. Initially, the necessary libraries and dataset are imported to set up the analysis. The first and last few rows of the dataset are inspected to understand its structure and contents. The dataset's dimensions, including the number of rows andcolumns, are determined, and detailed information regarding data types and memory requirementsis gathered.

To ensure data integrity, any null values are identified and addressed, and duplicates are detected and removed. Summary statistics are generated to capture the central tendencies and variability within the dataset. A correlation matrix is drawn to visualize relationships between variables, helping to identify significant correlations.

The project delves into the prevalence of heart disease, gender distribution, age range, chest pain types, fasting blood sugar levels, resting blood pressure, and serum cholesterol levels. These variables are compared to understand how they relate to heart disease. Detailed plots of continuous variables are created to observe distributions and detect patterns. This thorough analysis aims to reveal critical insights, contributing to more effective diagnostic practices and better management of heart disease, ultimately enhancing patient outcome.

Data visualization:

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. It's an essential part of data analysis that helps in communicating complex data insights in a clear and visually appealing manner.

Importance of Data Visualization

1. **Simplifies Complex Data**: Converts complex datasets into simple, understandable visuals.
2. **Identifies Trends and Patterns**: Helps in spotting trends, correlations, and patterns that might go unnoticed in raw data.
3. **Facilitates Decision Making**: Supports data-driven decision-making by presenting data in a straightforward way.
4. **Enhances Communication**: Makes it easier to share and explain data findings with others, including stakeholders and non-technical audiences.

Types of Data Visualizations

1. **Bar Charts**: Useful for comparing quantities across different groups.
2. **Line Charts**: Ideal for showing trends over time.
3. **Histograms**: Display the distribution of a dataset.
4. **Scatter Plots**: Show relationships between two numerical variables.
5. **Heatmaps**: Use color to represent data values, useful for showing data density or intensity.
6. **Box Plots**: Summarize data distributions, showing medians and quartiles.
7. **Pie Charts**: Represent parts of a whole, useful for showing proportions.

Key Components of Effective Data Visualization

1. **Clarity**: The visualization should be easy to read and interpret.
2. **Accuracy**: Data should be represented accurately without misleading the audience.
3. **Relevance**: Choose the right type of visualization for the data and the message.
4. **Design**: Use appropriate colors, labels, and scales to enhance readability.

Tools for Data Visualization

1. **Matplotlib**: A widely used Python plotting library that offers a variety of chart types.
2. **Seaborn**: Built on top of Matplotlib, Seaborn provides a high-level interface for drawing attractive and informative statistical graphics.
3. **Tableau**: A powerful tool for creating interactive and shareable dashboards.
4. **Power BI**: A Microsoft tool that provides business analytics capabilities and data visualizations.
5. **D3.js**: A JavaScript library for producing dynamic, interactive data visualizations in web browsers.

Examples

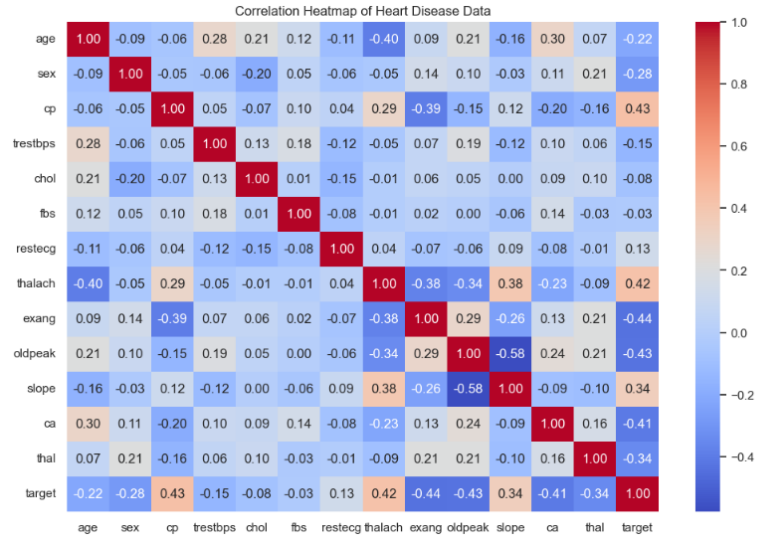
To visualize the heart disease dataset, we used various plots:

1. **Correlation Heatmap**: Shows the correlation between different features.
2. **Histograms**: Display the distribution of numerical features, colored by the target variable (presence or absence of heart disease).

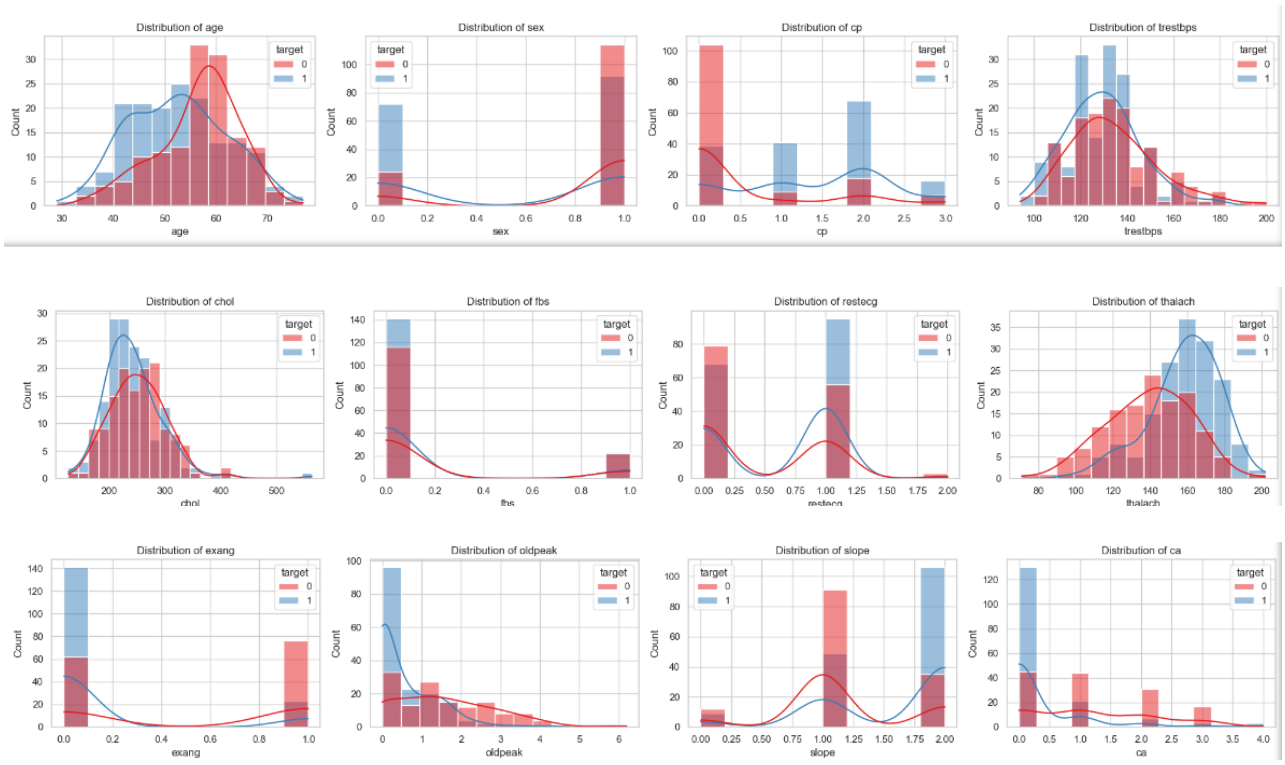
These visualizations help us understand the relationships between features, identify important variables, and see how they differ between individuals with and without heart disease.

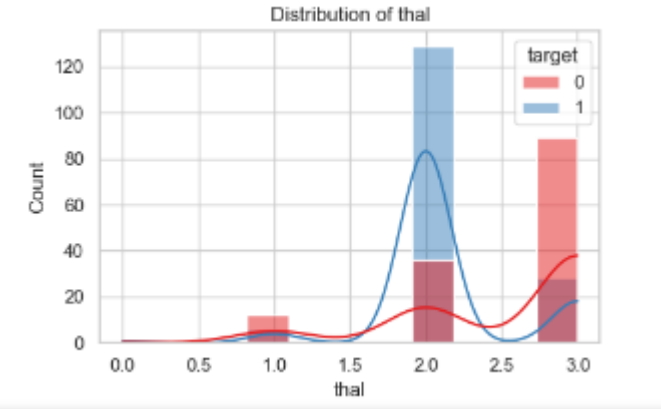
Visualization from my dataset:

Correlation Heatmap:



Distributions of Numerical Features by Target:





Algorithms used in dataset:

1. **Decision Tree:**

A Decision Tree is a supervised learning algorithm that is widely used for both classification and regression tasks. It mimics the human decision-making process by breaking down a complex decision-making problem into a series of simpler decisions, resulting in a tree-like model of decisions and their possible consequences.

Structure of a Decision Tree

A Decision Tree consists of three main components:

1. **Root Node**: The topmost node in a tree, representing the entire dataset, which is then split into two or more homogeneous sets.
2. **Internal Nodes**: Nodes where the data is split based on certain features.
3. **Leaf Nodes**: Terminal nodes that represent the outcome or final decision.

How Decision Trees Work:

1. **Splitting**: The process of dividing a node into two or more sub-nodes based on certain conditions. The goal is to find the feature and threshold that result in the most significant information gain or the largest reduction in impurity.
2. **Impurity Measures**: Decision Trees use impurity measures like Gini Impurity or Entropy (Information Gain) to determine the best split:
    - **Gini Impurity**: Measures the frequency at which any element of the dataset would be misclassified if it was randomly labeled.
    - **Entropy**: Measures the randomness in the information being processed; high entropy means the data is mixed, and low entropy indicates that the data is more homogeneous.
3. **Pruning**: The process of removing nodes to prevent the model from overfitting. Pruning can be done in two ways:
    - **Pre-Pruning**: Stopping the tree growth early based on certain conditions like maximum depth, minimum samples per split, etc.

- **Post-Pruning**: Allowing the tree to grow fully and then removing nodes that add little predictive power.

2. **Random Forest:**

Random Forest is an ensemble learning method primarily used for classification and regression tasks. It was developed by Leo Breiman and Adele Cutler and builds upon the fundamental principles of Decision Trees. By aggregating the predictions of multiple Decision Trees, Random Forest enhances the accuracy and robustness of the model, effectively addressing some of the limitations inherent in individual Decision Trees.

Structure of Random Forest:

A Random Forest consists of numerous Decision Trees operating as an ensemble. Each tree in the forest is grown using a subset of the training data and a random selection of features. The final prediction is made by aggregating the predictions from all the individual trees (e.g., by majority vote for classification or averaging for regression).

How Random Forest Works:

1. **Bootstrap Aggregating (Bagging)**:
   - **Data Sampling**: From the original training dataset, multiple bootstrap samples (random samples with replacement) are generated. Each sample will be used to grow a Decision Tree.
   - **Feature Randomness**: At each node of a Decision Tree, a random subset of features is selected. This randomness helps in making the model less sensitive to any single feature and enhances the diversity among trees.
2. **Tree Construction**: Each Decision Tree is built independently using its bootstrap sample and the subset of features. The tree grows by splitting nodes based on the best feature from the randomly selected subset.
3. **Prediction Aggregation**:
   - **Classification**: For classification tasks, each tree in the forest votes for a class, and the class with the majority votes is chosen as the final prediction.
   - **Regression**: For regression tasks, the predictions from all the trees are averaged to produce the final prediction.

Terminologies

ACCURACY RATE OF CLASSIFICATION:
The Classification accuracy Rate is measured as exactly classified samples divided by the entire number of trials multiplied by 100. Exact classified sample is the sum of True-Negative and True-Positive (TP).
Accuracy Rate is measured as, $(TP + TN / total) \times 100$

PRECISION:
Following to the Confusion Matrix, Precision is the proposition between predicted yes samples and true-positive samples.

Precision calculates as, TP / (TP + FP)
Here, TP + FP = Predicted Yes which is followed.

RECALL:
Recall is quietly known as Sensibility. Following to the Confusion Matrix, Recall is the proportion between actual yes samples and true-positive samples.
Recall calculates as, TP / (TP + FN) Here,
TP + FN = Actual Yes.

F-SCORE:
F-Score is mainly called F-Measure or, F1-Score. The F-measure shall be given a more practicable measurement of a test calculation using both precision and recall. Whenever the result of F-measured becomes 1 that precises the perfection of both precision and recall.
F-Score calculates as, (2 × $Precision$ × $Recall$) / ($Precision$ + $Recall$)

Implementation:

Decision Tree (Cleaned data):

Report:

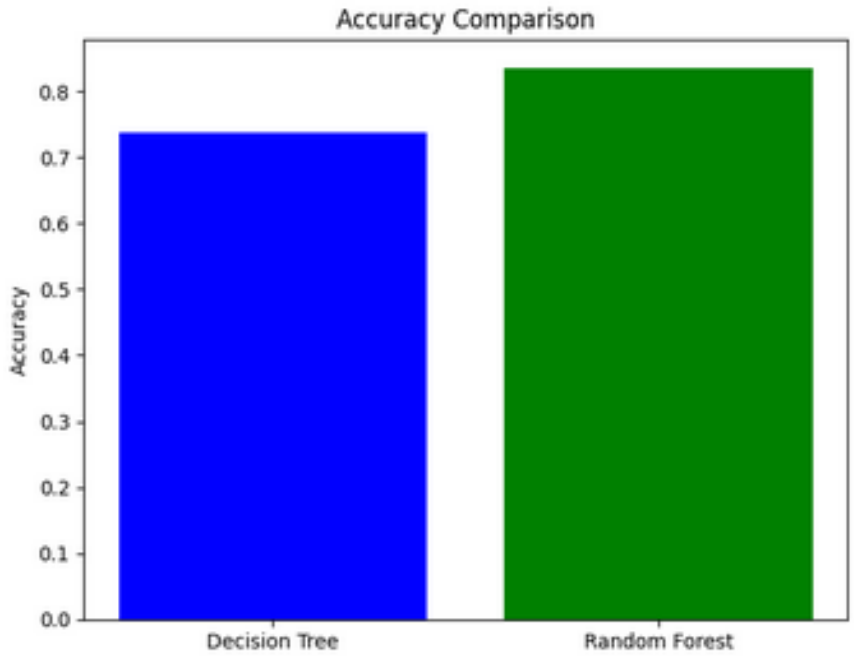
```
Accuracy: 0.7377
Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.78      0.76        32
           1       0.74      0.69      0.71        29

    accuracy                           0.74        61
   macro avg       0.74      0.74      0.74        61
weighted avg       0.74      0.74      0.74        61

Confusion Matrix:
[[25  7]
 [ 9 20]]
```

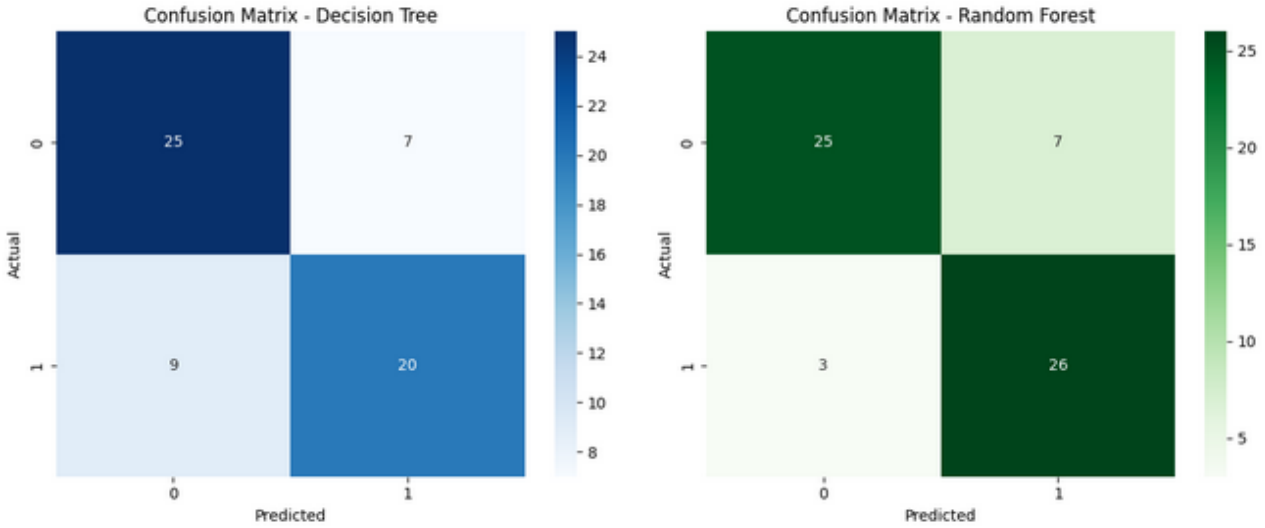
Random Forest (Cleaned data):

Report:

```
Accuracy: 0.8361
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.78      0.83        32
           1       0.79      0.90      0.84        29

    accuracy                           0.84        61
   macro avg       0.84      0.84      0.84        61
weighted avg       0.84      0.84      0.84        61

Confusion Matrix:
[[25  7]
 [ 3 26]]
```

Comparison between two algorithm model results:





Outcome:

In the decision tree algorithm, the accuracy is: 0.7377
and in the random forest algorithm, the accuracy is: 0.8361

These results are from the cleaned dataset. I will try to improve the accuracy result in my Final Report. Therefore, in this report, I show my previous work and the comparison between two model.

Reference:

1. M. Sultana, A. Haider and M. S. Uddin, "Analysis of data mining techniques for heart diseaseprediction," *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, Dhaka, Bangladesh, 2016, pp. 1-5, doi: 10.1109/CEEICT.2016.7873142.
2. Dataset: https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset
3. Google
4. ChatGPT