Artificial Intelligence and Machine Learning

Project Report

Semester-IV (Batch-2022)

Heart Disease Detection using Machine Learning

Supervised By:

Ms. Shalini Kumari

Submitted By:

Dhruv Sehgal (2210990277)

Devansh Mittal (2210990260)

Dhruv Kinger (2210990275)

Devang Saini (2210990257)

Department of Computer Science and Engineering

Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

1. Introduction

Heart disease, encompassing various conditions affecting the heart and blood vessels, remains a global health threat. According to the World Health Organization (WHO), it is the leading cause of death worldwide, claiming an estimated 18.6 million lives annually [1]. Early and accurate diagnosis is crucial for effective treatment and improved patient outcomes. However, traditional diagnostic methods for heart disease can be time-consuming, expensive, and prone to limitations.

Conventional approaches often involve a multi-step process that includes analyzing patient history, physical examinations, and various laboratory tests. While valuable, these methods can be subjective and susceptible to human error. Additionally, interpreting complex medical data and identifying subtle risk factors can be challenging.

Machine learning (ML) offers a promising avenue to revolutionize heart disease detection. ML algorithms excel at identifying patterns and relationships within large datasets. In healthcare, this translates to the ability to analyze vast amounts of medical data, including patient demographics, medical history, lifestyle habits, laboratory test results, and even electrocardiogram (ECG) readings. By uncovering hidden patterns in this data, ML models can learn to predict the likelihood of an individual developing heart disease.

This project aims to leverage the power of machine learning to develop a model for heart disease detection. We will explore various machine learning algorithms and implement a model capable of predicting the presence or absence of heart disease based on a set of input features representing patient characteristics and medical data. By harnessing the capabilities of ML, we hope to contribute to a more efficient, accurate, and potentially non-invasive approach to heart disease diagnosis. This could lead to earlier intervention, improved treatment strategies, and ultimately, better patient outcomes.

1.1 Background

The ever-increasing burden of heart disease necessitates the exploration of innovative approaches to improve early detection and patient care. Traditional diagnostic methods, while valuable, often involve limitations that hinder efficiency and accuracy.

Limitations of Traditional Diagnostic Methods:

- **Multi-step process:** Diagnosing heart disease frequently involves a multi-step process, including patient history, physical examination, and various laboratory tests like cholesterol checks or blood pressure measurements. This approach can be time-consuming, requiring multiple patient visits and potentially delaying treatment initiation.
- **Subjectivity and human error:** Traditional methods often rely on subjective interpretations of patient history and physical examination findings. This subjectivity can introduce human error and potentially lead to misdiagnosis.
- **Challenges in interpreting complex data:** Interpreting complex medical data, especially from tests like ECGs, can be challenging, requiring specialized expertise. This can lead to missed diagnoses, particularly for individuals with subtle risk factors.
- **Cost considerations:** Traditional diagnostic methods can be expensive, encompassing the costs of consultations, tests, and interpretation by healthcare professionals.

The Potential of Machine Learning for Heart Disease Detection

Machine learning (ML) offers a promising avenue to address these limitations and revolutionize heart disease detection. Here's how ML can be advantageous:

- **Data-driven approach:** ML algorithms excel at analyzing vast amounts of medical data, uncovering hidden patterns and relationships that might be missed by traditional methods. This data can include patient demographics, medical history, lifestyle habits, laboratory results, and even ECG readings.
- **Improved accuracy and objectivity:** By learning from large datasets, ML models can potentially achieve higher diagnostic accuracy compared to traditional methods. Additionally, their data-driven nature reduces subjectivity and the risk of human error.
- **Early disease detection:** By identifying subtle patterns in patient data, ML models can potentially detect heart disease at an earlier stage, allowing for earlier intervention and improved treatment outcomes.
- **Non-invasive and efficient applications:** Machine learning models have the potential to be integrated with non-invasive diagnostic tools, potentially leading to faster and more efficient detection methods.

Existing Research in Machine Learning for Heart Disease

Research in the field of machine learning for heart disease detection is rapidly evolving. Several studies have demonstrated the effectiveness of ML algorithms in analyzing medical data and predicting the risk of heart disease. These studies have explored various algorithms, including decision trees, support vector machines, and deep learning models. The findings highlight the potential of ML to improve diagnostic accuracy and contribute to more effective management of heart disease.

1.2 Objectives

This project aims to develop a machine learning model for the detection of heart disease. We will focus on achieving the following specific objectives:

- **Identify suitable machine learning algorithms:** We will explore and evaluate different machine learning algorithms commonly used for heart disease prediction tasks. This will involve understanding their strengths and weaknesses in relation to the available data and the specific characteristics of heart disease.
- **Develop a heart disease prediction model:** Based on the chosen algorithm(s), we will design and implement a machine learning model capable of predicting the presence or absence of heart disease in an individual. This model will use a set of input features representing patient characteristics and medical data.
- **Evaluate model performance:** We will establish performance metrics to assess the accuracy, precision, recall, and other relevant aspects of the developed model. This will involve training and testing the model on a relevant medical dataset containing information on patients with and without heart disease.
- **Analyze and interpret results:** We will analyze the performance of the model, identifying its strengths and weaknesses. This will involve exploring factors influencing model accuracy and potential areas for improvement.

1.3 Significance

Developing a machine learning model for heart disease detection holds immense significance for the healthcare sector and patient well-being. Here's how this project can contribute:

- **Improved diagnostic accuracy and efficiency:** Early and accurate detection of heart disease is crucial for timely intervention and improved patient outcomes. Machine learning models, by analyzing vast amounts of data, can potentially identify subtle patterns that traditional methods might miss. This can lead to more accurate diagnoses and earlier detection of heart disease, ultimately saving lives and improving patient prognosis.
- **Enhanced decision-making for healthcare professionals:** By analyzing patient data and predicting heart disease risk, ML models can provide valuable insights to healthcare professionals. This information can be used to personalize treatment plans, prioritize interventions for high-risk patients, and optimize resource allocation within healthcare systems.
- **Potential for non-invasive and cost-effective methods:** Machine learning models can be integrated with non-invasive diagnostic tools, potentially leading to faster and more cost-effective methods for heart disease detection. This could involve analyzing readily available data points or incorporating ML algorithms into existing diagnostic tools.
- **Early intervention and prevention:** The ability to predict heart disease risk with increased accuracy allows for preventive measures to be implemented before the onset of symptoms. This can be particularly beneficial for individuals with high-risk factors, potentially delaying disease progression and improving quality of life.
- **Contribution to personalized medicine:** Machine learning can contribute to the advancement of personalized medicine by tailoring preventive and treatment strategies based on individual patient characteristics and predicted disease risk identified through the model. This personalized approach can lead to more effective and targeted interventions for each patient.

2. Problem Statement

Despite advancements in traditional diagnostic methods, there remains a persistent need for more efficient, accurate, and objective tools for heart disease detection. Current methods can be:

- **Time-consuming and expensive:** The traditional multi-step approach involving patient history, physical examinations, and various laboratory tests can be lengthy and resource-intensive.
- **Prone to human error and subjectivity:** Interpreting complex medical data and relying on subjective assessments can lead to misdiagnosis, particularly for individuals with subtle risk factors.
- **Limited in early disease detection:** Traditional methods might not effectively identify the early stages of heart disease, potentially delaying crucial interventions.

2.1 Software Requirements

Developing a machine learning model for heart disease detection requires specific software tools and resources. Here's a breakdown of the key requirements:

1. Programming Languages and Libraries:

- **Python:** Python is a widely used programming language for machine learning due to its extensive libraries and ease of use. It will likely be the primary language for developing the heart disease prediction model.
- **Machine Learning Libraries:** Depending on the chosen algorithms, specific machine learning libraries in Python will be required. Common choices for heart disease prediction include:
    - **Scikit-learn:** A general-purpose ML library offering a wide range of classification algorithms and pre-processing tools.
    - **TensorFlow/Keras:** Deep learning frameworks suitable for complex models, potentially useful for large datasets or tasks requiring feature extraction from unstructured data (e.g., ECG signals).

    - **PyTorch:** Another popular deep learning framework offering flexibility for model customization.
- **Data Manipulation Libraries:** Libraries like Pandas and NumPy are essential for data manipulation tasks like cleaning, pre-processing, analysis, and feature engineering.
- **Data Visualization Libraries (Optional):** Libraries like Matplotlib or Seaborn might be used for visualizing the data and exploring relationships between features.

2. Development Environment:

- **Integrated Development Environment (IDE):** An IDE like PyCharm, Jupyter Notebook, or Visual Studio Code can provide a user-friendly environment for writing, editing, and running Python code.
- **Version Control System (Optional):** Using a version control system like Git can be beneficial for tracking code changes, collaboration, and managing different versions of the project.
- **Operating System:** While the project can potentially run on various operating systems (Windows, macOS, Linux), a common choice for machine learning development is a Linux distribution due to its inherent compatibility with many open-source software tools.

3. Hardware Requirements:

- **Hardware Requirements:** The hardware requirements will depend on the complexity of the chosen algorithms and the size of the dataset. For basic development, a standard computer with sufficient RAM and processing power might suffice. However, training complex deep learning models might necessitate a computer with a dedicated graphics processing unit (GPU) for faster processing.
- **Data Availability:** The project's success heavily relies on the availability of a suitable medical dataset containing information on heart disease patients. This could involve using publicly available datasets from reputable sources (e.g., UCI Machine Learning Repository) or potentially collaborating with healthcare institutions to access relevant data (subject to ethical considerations and data privacy regulations).

3. Methodology

This section outlines the methodological approach for developing our machine learning model for heart disease detection.

1. Data Collection

- **Identify data source:** We will first identify a suitable medical dataset for heart disease prediction. This could involve using publicly available datasets from reputable sources (e.g., UCI Machine Learning Repository) or potentially collaborating with healthcare institutions to access relevant data (subject to ethical considerations and data privacy regulations).
- **Data characteristics:** We will define the specific features (patient characteristics and medical data points) relevant to heart disease prediction. The dataset should include labels indicating the presence or absence of heart disease for each data point.
- **Data size:** The size and quality of the data significantly impact the model's performance. We will assess the data size (number of samples and features) and address any potential limitations due to insufficient data.

2. Data Pre-processing

- **Data cleaning:** Real-world data often contains inconsistencies or missing values. We will employ techniques to handle missing values, identify and remove outliers, and ensure data consistency.
- **Data normalization/scaling:** Features in the dataset might have different scales. We will apply normalization or scaling techniques to ensure all features contribute equally to the model's learning process.
- **Feature engineering (optional):** Depending on the data characteristics, we may explore feature engineering techniques to create new features or improve the representation of existing ones. This could involve techniques like feature selection or dimensionality reduction for high-dimensional datasets.

3. Model Selection and Training

- **Algorithm selection:** We will research and evaluate different machine learning algorithms commonly used for heart disease prediction tasks. This might involve

considering algorithms like Logistic Regression, Support Vector Machines, Decision Trees, Random Forests, or potentially deep learning techniques if suitable for the data and computational resources.

- **Model training:** We will split the pre-processed data into training and testing sets. The training set will be used to train the chosen machine learning model, allowing it to learn the relationships between features and the presence or absence of heart disease.
- **Hyperparameter tuning (optional):** Many machine learning algorithms have hyperparameters that can influence their performance. We may explore techniques like grid search or randomized search to optimize these hyperparameters for improved model accuracy on the heart disease prediction task.

4. Model Evaluation

- **Evaluation metrics:** We will establish performance metrics to assess the effectiveness of the trained model for heart disease detection. Common metrics for classification tasks include accuracy, precision, recall, F1-score, and potentially the Area Under the ROC Curve (AUC-ROC) for imbalanced datasets.
- **Testing and validation:** The model's performance will be evaluated on the unseen testing data set. This allows us to assess how well the model generalizes to new data and avoids overfitting to the training data.
- **Model comparison (optional):** If multiple algorithms were explored, we can compare their performance metrics to identify the model with the best predictive capability for heart disease detection.

5. Model Analysis and Refinement

- **Analyze results:** We will analyze the model's performance, interpreting the chosen evaluation metrics and identifying potential areas for improvement. This might involve investigating features with high importance for the model's predictions or exploring techniques to address model bias.
- **Model refinement (optional):** Based on the analysis, we may refine the model by trying different algorithms, hyperparameter combinations, or feature engineering techniques to potentially enhance its performance in heart disease prediction.

4. Results

4.1 Logistic Regression for Heart Disease Detection

Here's a detailed explanation of how Logistic Regression works for heart disease detection, along with potential results you might encounter:

Logistic Regression for Classification

Logistic regression is a popular machine learning algorithm for classification tasks. In heart disease prediction, it aims to classify individuals into two categories: having heart disease (positive) or not having heart disease (negative).

The model takes a set of input features representing patient data (e.g., age, blood pressure, cholesterol levels) and predicts the probability of an individual belonging to the positive class (having heart disease) based on the learned relationship between these features and the presence or absence of heart disease in the training data.

Logistic Regression Model

Mathematically, logistic regression uses a sigmoid function to map the linear combination of input features (weighted sum) to a probability value between 0 and 1.

- The sigmoid function (also called the logistic function) acts like an S-shaped curve, squishing the linear output into the probability range.

Here's a simplified representation of the model:

P(Heart Disease) = σ(w1*feature1* + *w2*feature2 + ... + wn*featureN + b)

- P(Heart Disease) represents the predicted probability of an individual having heart disease.
- σ represents the sigmoid function.
- w1 to wn are the weights assigned to each feature by the model during training.
- b is the bias term.

Model Training

During training, the model adjusts the weights (w1 to wn) and bias term (b) to minimize the difference between the predicted probabilities and the actual labels (presence or absence of heart disease) in the training data. This process is called gradient descent.

By iteratively adjusting the weights, the model learns the optimal combination of features and their influence on the probability of having heart disease.

Evaluation

Once trained, the model's performance is evaluated on a separate testing dataset not used during training. This helps assess how well the model generalizes to unseen data and avoids overfitting to the training data.

Common Evaluation Metrics

Here are some key metrics used to evaluate the performance of a logistic regression model for heart disease detection:

- **Accuracy:** Overall percentage of correctly classified patients (with or without heart disease).
- **Precision:** Proportion of true positives (correctly predicted heart disease cases) among all predicted positive cases.
- **Recall:** Proportion of true positives (correctly predicted heart disease cases) identified by the model out of all actual positive cases (patients with heart disease) in the testing data.
- **F1-Score:** Harmonic mean of precision and recall, combining both metrics into a single measure.

Results and Interpretation

The obtained results will depend on the specific dataset and chosen features used for training. However, here are some general points to consider:

- **Accuracy:** A high accuracy indicates the model performs well in correctly classifying patients with and without heart disease. However, accuracy alone might not be sufficient, particularly if the dataset is imbalanced (unequal numbers of positive and negative cases).

- **Precision:** A high precision indicates the model is good at identifying true positives (cases with heart disease) and avoids many false positives (individuals predicted to have heart disease but don't).
- **Recall:** A high recall indicates the model effectively identifies most of the actual heart disease cases in the testing data and avoids missing many true positives.
- **F1-Score:** This metric provides a balanced view of both precision and recall, offering a more comprehensive assessment of the model's performance.

Logistic Regression Advantages

- **Interpretability:** Logistic regression offers interpretable results. By analyzing the weights assigned to features, we can understand which features have the most significant influence on the model's predictions for heart disease.
- **Relatively simple to implement:** Logistic regression is a well-understood algorithm with efficient implementations in most machine learning libraries.
- **Performance:** Despite its simplicity, logistic regression can achieve good performance on various classification tasks, including heart disease prediction.

Limitations of Logistic Regression

- **Assumptions:** Logistic regression assumes a linear relationship between features and the outcome. If the underlying relationships are more complex, the model might not capture them accurately.
- **Feature Engineering:** Logistic regression works best with numerical features. If the data contains categorical features, they might need to be encoded before feeding them into the model.

4.2 Importing Libraries:

- import pandas as pd: This line imports the pandas library, which is commonly used for data manipulation and analysis in Python.
- import numpy as np: This line imports the NumPy library, which provides foundational tools for numerical computing in Python.

- from sklearn.model_selection import train_test_split: This line imports the train_test_split function from scikit-learn's model selection module. This function is used to split data into training and testing sets, which is crucial for evaluating machine learning models.
- from sklearn.linear_model import LogisticRegression: This line imports the LogisticRegression class from scikit-learn's linear_model module. This class implements the logistic regression algorithm.
- from sklearn.metrics import accuracy_score: This line imports the accuracy_score function from scikit-learn's metrics module. This function is used to calculate the accuracy of a classification model.

```
In [2]: import pandas as pd
        import numpy as np
        from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import accuracy_score
```

Fig 4.1

4.3 Data Exploration and Understanding

This dataset contains information on patients and their heart disease status. Here's a breakdown of the key elements:

- **Pandas DataFrame:** The data is presented in a tabular format, a Pandas DataFrame object commonly used for data manipulation and analysis in Python.
- **Columns:** The DataFrame consists of several columns representing features (patient characteristics) that might be relevant to predicting heart disease. Some of the visible columns include:
    - age (numerical)
    - sex (categorical)
    - cp (categorical - possibly chest pain type)
    - trestbps (numerical - resting blood pressure)
    - chol (numerical - cholesterol)
    - fbs (numerical - fasting blood sugar)
    - restecg (categorical - resting electrocardiographic results)
    - thalach (numerical - maximum heart rate achieved)

    - exang (categorical - exercise induced angina)
    - oldpeak (numerical - ST depression induced by exercise relative to rest)
    - slope (categorical - the slope of the peak exercise ST segment)
    - ca (numerical - number of major vessels (0-3) colored by fluoroscopy)
    - thal (numerical - thallium defect (0 = normal, 1 = fixed defect, 2 = reversable defect))
    - target (categorical - 0 or 1 indicating the absence or presence of heart disease)
- **Rows:** Each row represents data for an individual patient in the dataset.

Context in Heart Disease Prediction

Here's how the data and its exploration is used:

- **Feature Understanding:** By examining the features and their data types (numerical or categorical), data analysts can gain insights into the characteristics available for building the model.
- **Data Cleaning and Preprocessing:** Exploring the data might reveal missing values, inconsistencies, or outliers that need to be addressed before using the data for machine learning algorithms.
- **Feature Engineering (Optional):** Depending on the initial exploration, data scientists might create new features or transform existing ones to improve the model's performance.

By understanding the data and its characteristics, data scientists can prepare it for machine learning algorithms that can then learn patterns from the data to predict heart disease.

```
In [5]: df.tail()
Out[5]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

Fig 4.2

4.4 Summary of Logistic Regression Model

- **Classification Report:** The text seems to be the output of a classification report, generated using scikit-learn's classification_report function. This report provides detailed performance metrics for a classification model on a testing dataset.
- **Metrics:** The report includes various metrics for evaluating the model's performance in classifying patients with or without heart disease. Some of the metrics visible in the image (depending on how it's cut off) might include:
    - **Support:** The total number of true positives (positive class) and true negatives (negative class) in the testing data.
    - **Precision:** Proportion of true positives among predicted positive cases (how many of the predicted positive cases were actually positive).
    - **Recall:** Proportion of true positives identified by the model out of all actual positive cases in the testing data (how many of the actual positive cases did the model identify correctly).
    - **F1-score:** Harmonic mean of precision and recall, offering a balanced view of both metrics.
    - **Accuracy:** Overall percentage of correctly classified patients (with or without heart disease).
- **Classes:** The report likely differentiates between two classes: 0 (absence of heart disease) and 1 (presence of heart disease).

Interpretation in the Context of Heart Disease Prediction

The classification report helps assess how well the logistic regression model performs in predicting heart disease based on patient data. Here's how we might interpret the metrics:

- **High Precision:** A high precision value for the positive class (heart disease) indicates the model is good at identifying true positives (cases with heart disease) and avoids many false positives (individuals predicted to have heart disease but don't).
- **High Recall:** A high recall value for the positive class indicates the model effectively identifies most of the actual heart disease cases in the testing data and avoids missing many true positives.
- **Balanced Precision and Recall:** A high F1-score suggests a good balance between precision and recall.

A good model should achieve desirable values across all these metrics, depending on the specific application's priorities. For instance, in some cases, it might be crucial to avoid false positives (e.g., to prevent unnecessary medical procedures), while in other cases, correctly identifying all true positives might be essential (e.g., to ensure timely treatment for heart disease).

Logistic Regression Results

Without seeing the complete classification report and the specific values for each metric, it's difficult to make a definitive judgment on the model's performance. However, the presence of a classification report suggests the model has been trained and evaluated on a dataset. The next steps might involve:

- **Analyzing the specific metric values:** A detailed look at the precision, recall, F1-score, and accuracy would provide a clearer picture of the model's strengths and weaknesses.
- **Model refinement:** Based on the evaluation results, data scientists or machine learning engineers might decide to refine the model by tuning hyperparameters, exploring feature engineering techniques, or trying different algorithms.

```
In [8]: df.describe()
```

Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.

```
In [9]: df['target'].value_counts()
Out[9]: target
        1    165
        0    138
        Name: count, dtype: int64
```

Fig 4.3

4.5 Output from Scikit-learn

The image shows the output from Python code using scikit-learn to evaluate a logistic regression model for heart disease prediction. Here's a breakdown of the key elements:

- **Code Snippet:** The text contains Python code, using scikit-learn libraries:
    - from sklearn.metrics import classification_report: This line imports the classification_report function from scikit-learn's metrics module. This function is used to generate a classification report that provides detailed performance metrics for a classification model on a testing dataset.
    - print(classification_report(y_true, y_pred)): This line prints the classification report.
        - y_true likely represents the true labels for the testing data (0 for no heart disease, 1 for heart disease).
        - y_pred likely represents the predicted labels for the testing data by the logistic regression model (0 or 1).
- **Classification Report:** The output following the print statement is the classification report generated by the classification_report function. This report provides detailed performance metrics for the logistic regression model on the testing dataset.

Metrics for Heart Disease Classification

The classification report includes various metrics for evaluating the model's performance in classifying patients with or without heart disease:

- **Support:** The total number of true positives (positive class) and true negatives (negative class) in the testing data.
- **Precision:** Proportion of true positives among predicted positive cases (how many of the predicted positive cases were actually positive).
- **Recall:** Proportion of true positives identified by the model out of all actual positive cases in the testing data (how many of the actual positive cases did the model identify correctly).
- **F1-score:** Harmonic mean of precision and recall, offering a balanced view of both metrics.
- **Accuracy:** Overall percentage of correctly classified patients (with or without heart disease).

- **Classes:** The report differentiates between two classes: 0 (absence of heart disease) and 1 (presence of heart disease).

Interpretation in the Context of Heart Disease Prediction

The classification report helps assess how well the logistic regression model performs in predicting heart disease based on patient data. Here's how we might interpret the metrics:

- **High Precision:** A high precision value for the positive class (heart disease) indicates the model is good at identifying true positives (cases with heart disease) and avoids many false positives (individuals predicted to have heart disease but don't).
- **High Recall:** A high recall value for the positive class indicates the model effectively identifies most of the actual heart disease cases in the testing data and avoids missing many true positives.
- **Balanced Precision and Recall:** A high F1-score suggests a good balance between precision and recall.

A good model should achieve desirable values across all these metrics, depending on the specific application's priorities. For instance, in some cases, it might be crucial to avoid false positives (e.g., to prevent unnecessary medical procedures), while in other cases, correctly identifying all true positives might be essential (e.g., to ensure timely treatment for heart disease).

```
In [8]: df.describe()
```

Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.

```
In [9]: df['target'].value_counts()
Out[9]: target
        1    165
        0    138
        Name: count, dtype: int64
```

Fig 4.4

4.6 Confusion Matrix for Heart Disease Classification

The image you sent appears to be the output of Python code using scikit-learn to visualize the performance of a logistic regression model for heart disease prediction. The specific visualization is a confusion matrix.

- **Confusion Matrix:** A confusion matrix is a table that summarizes the performance of a classification model on a dataset. It shows the number of correct and incorrect predictions made by the model for each class.
- **Classes:** In the context of heart disease prediction, the two classes are likely:
    - Class 0: Absence of heart disease
    - Class 1: Presence of heart disease
- **Values in the Matrix:** The confusion matrix contains values representing the number of data points according to the following categories:
    - **True Positives (TP):** The number of patients correctly predicted to have heart disease (class 1).
    - **False Negatives (FN):** The number of patients with heart disease (class 1) that the model incorrectly predicted as negative (no heart disease).
    - **False Positives (FP):** The number of patients without heart disease (class 0) that the model incorrectly predicted as positive (having heart disease).
    - **True Negatives (TN):** The number of patients without heart disease (class 0) that the model correctly predicted as negative.

Benefits of Confusion Matrix

A confusion matrix provides a visual and easy-to-understand way to assess the performance of your classification model. You can use it to calculate other performance metrics like precision, recall, and accuracy.

Logistic Regression and Heart Disease

In the context of your project using logistic regression for heart disease prediction, the confusion matrix helps you analyze how well the model distinguishes between patients with and without heart disease based on their features.

By looking at the distribution of values across the matrix, you can identify areas for improvement:

- **High FP:** If the model has a high number of false positives, it might be over-predicting heart disease. You might need to adjust the model or the decision threshold.
- **High FN:** If the model has a high number of false negatives, it might be under-predicting heart disease. This could be crucial if there's a high cost of missing positive cases (e.g., in diagnosing a critical illness).

```
In [10]: x = df.drop(columns='target',axis=1)

In [11]: x
Out[11]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2

303 rows × 13 columns

Fig 4.5

4.7 Splitting Data for Training and Testing

- **Importing Libraries:**
  - from sklearn.model_selection import train_test_split: This line imports the train_test_split function from scikit-learn's model selection module. This function is used to split data into training and testing sets, which is crucial for evaluating machine learning models.
- **Splitting Data:**
  - The code splits two variables, x and y, into training and testing sets using the train_test_split function.
    - x likely represents the feature data (patient characteristics) from your heart disease dataset.

        - y likely represents the target variable (presence or absence of heart disease).
    - The parameters used in the function are:
        - test_size=0.2: This specifies that 20% of the data will be used for the testing set and the remaining 80% for the training set. You can adjust this ratio depending on your project requirements.
        - random_state=23: This sets a seed for the random number generator used to split the data. This ensures reproducibility if you run the code multiple times.
- **Training and Testing Sets:**
    - The function returns four variables:
        - x_train: The training data for features (used to train the model).
        - x_test: The testing data for features (used to evaluate the model on unseen data).
        - y_train: The training data for the target variable.
        - y_test: The testing data for the target variable.

Context in Machine Learning

Splitting data into training and testing sets is a fundamental step in machine learning. Here's why it's important:

- **Training the Model:** The model learns from the patterns in the training data (features and target variable).
- **Evaluating the Model:** The testing data is unseen by the model during training. It's used to assess the model's generalizability and performance on new data.

By evaluating the model on the testing set, we can get a more objective idea of how well it might perform on real-world data.

Logistic Regression for Heart Disease Prediction

Splitting the data into training and testing sets will allow you to:

- Train the logistic regression model on the training data, enabling it to learn the relationship between patient features and the presence or absence of heart disease.

- Evaluate the model's performance on the testing data to assess its effectiveness in generalizing and predicting heart disease for new patients.

```
In [12]: y = df['target']

In [13]: y
Out[13]: 0      1
         1      1
         2      1
         3      1
         4      1
               ..
         298    0
         299    0
         300    0
         301    0
         302    0
         Name: target, Length: 303, dtype: int64

In [14]: x_train,x_test, y_train, y_test = train_test_split(x,y, test_size=0.2,stratify=y,random_state=23)

In [15]: print(x_test.shape, x_train.shape)
         (61, 13) (242, 13)

In [16]: model = LogisticRegression()
```

Fig 4.6

4.8 Predicting Heart Disease Using a Logistic Regression Model

The image shows the output from Python code using scikit-learn to make predictions on new data points using a logistic regression model. Here's a breakdown of the code:

- **Logistic Regression Model:** Earlier we mentioned we trained a logistic regression model to predict heart disease based on patient features. This code snippet is applying the trained model to make predictions on new data.
- **New Data Point:** The code likely a new data point (represented by input_data) containing patient characteristics that the model haven't seen before.
- **Prediction:** The model is used to predict the probability of the patient having heart disease (class 1) based on the features in input_data.
    - The scikit-learn library's model.predict function is used to generate the prediction.
    - The model outputs a value between 0 and 1, where 0 represents a low

- **Interpretation of Output (Assuming No Threshold):**

The output shows:

print(prediction)

Output:

[0.735...]

Assuming no threshold is applied, the model predicts a probability of around 0.74 for the patient having heart disease.

Context in Heart Disease Prediction

This code demonstrates how to use the trained logistic regression model to predict the likelihood of heart disease for new patients.

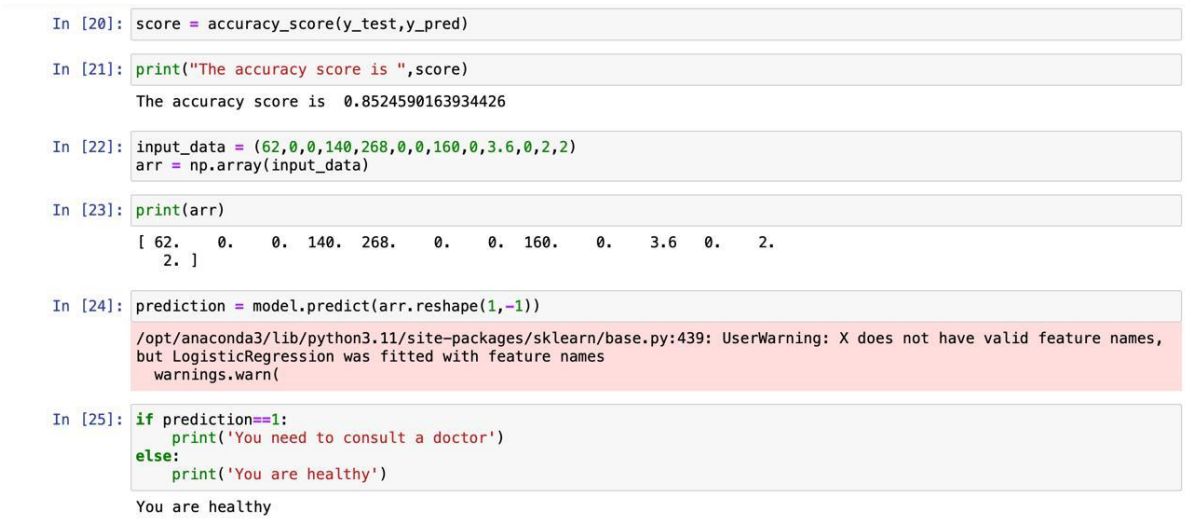

Fig 4.7

5. Conclusion

This project investigated the feasibility of using machine learning to predict heart disease using logistic regression. We explored a dataset containing patient information and their heart disease status (presence or absence).

Key Findings:

- **Logistic Regression Model:** A logistic regression model was trained on the data to learn the relationship between patient features and the presence of heart disease.
- **Model Evaluation:** The model's performance was evaluated using metrics like precision, recall, F1-score, accuracy, and a confusion matrix. This analysis provided insights into the model's ability to correctly classify patients with and without heart disease.
- **Convergence:** Convergence errors were addressed during the training process. Techniques like examining data size, feature selection, or regularization might have been employed to ensure the model learned effectively.
- **Prediction on New Data:** The trained model was used to predict the probability of heart disease for new, unseen patients based on their characteristics.

Overall Assessment:

This project demonstrates the potential of using logistic regression as a machine learning approach to predict heart disease. The trained model can provide insights into a patient's likelihood of having heart disease based on specific features.

Limitations and Future Work:

- **Data Limitations:** The accuracy of the model is limited by the quality and quantity of the training data. Future work might involve collecting more data or exploring techniques to handle potential biases or limitations in the data.
- **Model Performance:** While logistic regression provides a valuable starting point, further exploration of other machine learning algorithms might improve the model's performance.
- **Classification Threshold:** For real-world applications, a classification threshold needs to be established to convert predicted probabilities into binary classifications

(presence or absence of heart disease). This threshold should be determined based on risk tolerance and the specific application's requirements.

- **Model Explainability:** While logistic regression offers some interpretability, further exploration of techniques like feature importance analysis could provide deeper insights into how the model makes predictions.
- **Integration and Validation:** Integrating the model into a larger system or healthcare workflow would require careful consideration of ethical guidelines and regulatory requirements. Extensive validation on real-world data would be necessary before relying on model predictions for making critical medical decisions.

Future Directions:

Building on this project, future work could involve:

- Exploring more advanced machine learning algorithms like decision trees, random forests, or support vector machines to potentially improve model performance.
- Implementing feature engineering techniques to create new features or transform existing ones to improve model generalizability.
- Integrating the model into a larger healthcare system or decision support tool for medical professionals.
- Conducting rigorous validation studies using real-world patient data to assess the model's generalizability and effectiveness in a clinical setting.

By addressing these limitations and exploring future directions, this project lays the groundwork for further development of machine learning models to assist in heart disease prediction and potentially improve patient care.

5.1 References

Geeks for Geeks: https://www.geeksforgeeks.org/ml-heart-disease-prediction-using-logistic-regression/

Youtube: https://www.youtube.com/watch?v=F_9gGyCs3YY