Tampere University of Applied Sciences

A Comparative Analysis of Machine Learning Techniques and Key Insights for Cardiovascular Disease Prediction

Jobayel Hossain

BACHELOR'S THESIS
May 2024

Software Engineering

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Software Engineering

HOSSAIN, JOBAYEL:
A Comparative Analysis of Machine Learning Techniques and Key Insights for Cardiovascular Disease Prediction

Bachelor's thesis 52 pages, appendices 1 pages
May 2024

Heart disease, or cardiovascular disease (CVD) is a prevalent and deadly condition globally. The purpose of this thesis was to develop and evaluate a robust system applying classification-based machine learning methods to predict heart disease using a comprehensive dataset of patient health information, including age, sex, chest pain type, blood pressure, cholesterol levels, and maximum heart rate. The goal was to improve the accuracy and reliability of heart disease predictions, thereby assisting healthcare providers.

The study was developed by significant patterns and correlations involving data preprocessing, exploratory data analysis, and visualization techniques. Classification algorithms such as Logistic Regression, Decision Tree, Support Vector Machine, and Gaussian Naive Bayes were implemented and achieved cross-validation accuracies of 86.62%, 80.51%, 86.77%, 86.48%, and 85.89%, respectively. A combined method using a Voting Classifier method that achieved a prediction accuracy of 87.83%. Feature importance analysis provided insights into key predictors of heart disease, aiding medical professionals in the decision-making process.

The findings suggest that standard models provide a robust framework for early detection and effective management of heart disease. Healthcare institutions can start using these models right away to improve diagnostic processes. Further research is required to integrate real-time data and expand the model to incorporate additional health metrics. Overall, this research underscores the transformative potential of advanced diagnostic models in healthcare, offering extensive opportunities for both medical professionals and patients in the management and treatment of heart disease.

Key words: cardiovascular disease, machine learning, heart disease prediction, predictive analytics, healthcare diagnostics

CONTENTS

ABBREVIATIONS AND TERMS

ACCURACY	A machine learning statistic called accuracy is calculated by dividing the total number of cases in the dataset by the number of properly predicted instances. An accuracy of 89% in this project means that 89% of the samples were correctly predicted (89 out of every 100), while 11% were incorrect.
AUC	Area Under the Curve, A binary classification model's overall performance is measured by this statistic metric, where higher values denote a stronger capacity to distinguish between classes.
BP	Blood Pressure.
CVD	Cardiovascular Diseases refers to heart diseases.
CNN	A type of neural network specifically designed to process and analyze visual data by using convolutional layers to learn spatial hierarchies of features automatically and adaptively.
DT	Decision Trees, a machine learning algorithm that makes predictions by dividing data into branches according to feature values, ultimately reaching a decision at the leaf nodes.
KNN	K Nearest Neighbors, a machine learning algorithm that predicts a result based on the most common outcomes among its closest data points.
LVH	Left Ventricular Hypertrophy, a disorder where the left ventricle of the heart's muscular wall thickens. LVH can result from high blood pressure or other heart conditions.
LR	Logistic Regression, A statistical method is used in machine learning to predict the probability of a binary outcome, such as true or false, depending on one or more input factors.
ML	Machine learning.

MSE	Mean Squared Error, a metric used in statistics and machine learning to assess model accuracy, is calculated by picking the squared differences between the expected and actual values by averaging them.
MAE	Mean Absolute Error, a metric for assessing model accuracy in predicting continuous values, is determined by averaging the absolute differences between the values that were predicted and those that were observed.
RNN	A type of neural network, which is a computational model based on the human brain, designed to recognize patterns in sequences of data by using loops to maintain context.
RMSE	Root Mean Square Error, a metric for measuring model accuracy in predicting continuous values, is determined by finding the sum of the squared difference between the expected and actual values and taking the square root of that difference.
ROC	Receiver Operating Characteristic, an illustrative diagram that evaluates a binary classification model's effectiveness by demonstrating the trade-off between the specificity (specificity) and sensitivity (sensitivity) of the true positive rate at various threshold values.
ST Slop	ST is a specific part of the electrocardiogram (ECG) waveform. ST Slop refers to the ST segment in an (ECG) which is part of the heart's electrical cycle. ST segment changes can indicate various cardiac conditions, including ischemia (a condition when a portion of the body's blood flow is decreased or limited).
SVM	A machine learning method called Support Vector Machine, that works by finding the best possible boundary, referred to as the hyperplane, that clearly divides data points into various types.

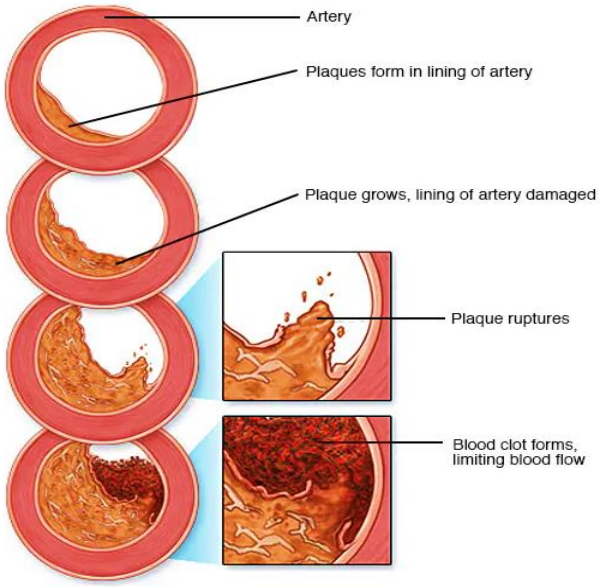
1 INTRODUCTION

The heart plays a vital role in pumping blood and supplying oxygen and nutrients to other organs. When the heart malfunctions, it affects overall health, and any disruption in this process can lead to various heart diseases. Around 17.9 million fatalities globally, or 31% of all deaths, are attributed to cardiovascular diseases (CVDs), many of which are preventable, according to the World Health Organization (WHO). These conditions can arise from infections, poor diet, lack of exercise, obesity, smoking, or congenital defects. Symptoms and severity can vary based on factors such as gender, with certain conditions presenting differently in men and women. Different types of cardiovascular diseases and their associated symptoms are mentioned here (Medicalnewstoday, A. 2023):

- **Valvular Heart Disease**: One of the cardiac valves may be damaged or defective, leading to issues such as internal bleeding.
- **Atherosclerotic Disease**: Plaque build-up in arteries; symptoms vary by gender, causing chest pain in males and discomfort in females.
- **Heart Arrhythmias**: Abnormal heartbeats that can lead to medical emergencies and sudden cardiac death.
- **Heart Infections**: Infections causing symptoms such as fever, fatigue, shortness of breath, and skin rashes.
- **Heart Defects**: Congenital conditions causing symptoms like cyanosis, swelling, and poor weight gain in infants.
- **Cardiomyopathy**: Thickening of the heart muscle, often symptomless in early stages, related to high blood pressure or aging.
- **Coronary Heart Disease**: Caused by unhealthy lifestyle factors, leading to damage to the heart and blood vessels.

Among these, Coronary Heart Disease (CHD) or Coronary Artery Disease (CAD) alone contribute to 7.5 million deaths per year, as illustrated in the next image. Misdiagnosis remains a significant challenge, with studies indicating that at least 16.1% of heart failure patients are incorrectly diagnosed. The detection and diagnosis of CHD typically involve imaging techniques like coronary calcium scans and CT Coronary Angiography (CTCA), which have sensitivity and specificity rates of 89% and 96%, respectively. Despite these advancements, there is a

growing need for more reliable diagnostic methods due to the subtle nature of some cardiac irregularities.



PICTURE 1. Causes of Coronary Artery Disease (Mayo-Clinic, A. 2022).

In the modern era, machine learning techniques have enhanced the accuracy of medical diagnoses by analyzing vast datasets, reducing the workload of medical professionals, and providing more precise results. ML algorithms are particularly useful in identifying non-linear relationships and minimizing errors between predicted and actual health outcomes. This thesis explores the application of various machine learning techniques to predict heart disease using a comprehensive dataset from the University of California, Irvine (UCI). By employing multiple ML algorithms and fine-tuning their parameters, this study aims to develop an effective predictive model for early detection and management of heart disease. The subsequent sections of this thesis detail related work, data description, applied techniques, and the results of the study.

2 LITERATURE REVIEW

2.1 Previous Studies on Heart Disease Prediction

A significant amount of research has been dedicated to identifying heart diseases using machine learning techniques over the years. Narendra Mohan, Vinod Jain, and Gauranshi Agrawal (2023) compared several machine learning models and found Logistic Regression to have the highest accuracy of 90.2%. Devansh Shah et al. (2018) reported the highest accuracy with K-Nearest Neighbors (KNN) using a dataset with 14 attributes, while Archana Singh et al. (2017) noted that KNN had an accuracy of 87%, followed by Decision Tree at 79%. Geetha S. and Santhana Krishnan J. (2019) achieved 91% accuracy with Decision Tree using data mining techniques.

Indrajani Sutedja (2021) explored machine learning and deep learning algorithms on a Kaggle dataset, with Support Vector Machines (SVM) achieving 88% accuracy and Recurrent Neural Networks (RNN) 90%. Pushkala V et al. (2019) achieved 91% accuracy with Naïve Bayes using a dataset of 303 samples and 14 features. Surai Shinde and Juan Carlos Martinez-Ovando (2020) developed a hybrid model combining RNN and Convolutional Neural Networks (CNN), enhancing prediction accuracy using heartbeat audio data from the PASCAL Classifying Heart Sounds Challenge and Kaggle. S. Musfiq Ali et al. (2022) utilized the Cleveland dataset, applying Gaussian Naïve Bayes (GNB) and Random Forest (RF), which achieved the highest level of accuracy of 91.2%.

S. Nithyavishnupriya et al. (2022) used a Deep Neural Network (DNN) combined with a chi-squared statistical model, improving prediction accuracy and addressing overfitting issues. Sonam Nikhar et al. (2019) focused on 19 attributes for Naive Bayes and Decision Tree classifiers, finding that Decision Tree performed better. Ravindra Yadav et al. (2020) applied various models, including Decision Tree, Naïve Bayes, Neural Network, Deep Learning, and SVM, achieving varied accuracies.

These studies collectively highlight the advancements and effectiveness of various machine learning techniques in heart disease prediction, emphasizing the importance of algorithm selection, feature optimization, and hybrid models in improving diagnostic accuracy and reliability.

2.2 Machine Learning in Healthcare

Machine learning (ML) has transformed the healthcare industry, driving notable advancements in diagnostics, treatment planning, patient management, and operational efficiency. By incorporating ML techniques into healthcare systems, medical practices have become more accurate and efficient, resulting in improved patient outcomes and optimized workflows.

ML algorithms have substantially enhanced diagnostic accuracy by processing extensive medical data to detect patterns and anomalies indicative of various diseases. Deep learning models, especially convolutional neural networks (CNNs), excel at interpreting complex medical images, assisting in the identification of conditions like cancer, diabetic retinopathy, and cardiovascular diseases. Research has shown that ML models can equal or even surpass the diagnostic capabilities of human experts in certain areas (Esteva et al., 2017; Litjens et al., 2017).

In terms of patient management, ML algorithms are employed to forecast disease progression and patient outcomes. Predictive models can identify high-risk patients who may benefit from early interventions, thereby preventing complications and minimizing hospital readmissions. For instance, ML models can predict the likelihood of sepsis in hospitalized patients by continuously monitoring vital signs and laboratory results, enabling timely interventions (Shashikumar et al., 2017).

Predictive modeling in healthcare involves using historical and real-time data to forecast future events, enhancing decision-making and patient care. Predictive models are employed in various healthcare applications, including predicting disease outbreaks, patient admissions, and chronic disease management. For in-

stance, models predicting hospital readmissions help healthcare providers implement preventive measures, improving patient outcomes and reducing costs (Rudin, 2019).

AI-powered tools like ChatGPT are significantly impacting healthcare by providing quick and reliable health information, enhancing patient engagement, supporting telemedicine, and assisting in mental health care. These tools offer immediate responses to health queries, remind patients about medications, and support healthcare providers with preliminary patient information and clinical decision-making. However, while beneficial, they cannot replace professional medical diagnosis and treatment. Their effectiveness lies in supplementing healthcare services and improving accessibility and efficiency (Hollis et al., 2015; Miner et al., 2016).

In summary, machine learning has the potential to greatly improve several elements of healthcare, including patient management, treatment planning, diagnosis, and operational effectiveness. The continued advancement and application of machine learning (ML) technologies in healthcare have great potential to improve patient outcomes and treatment quality.

2.3 Current Challenges and Opportunities

Implementing machine learning in healthcare involves challenges such as ensuring data quality and addressing privacy concerns. Electronic health records and wearable technology are two common sources of medical data, and medical imaging, resulting in inconsistencies and missing values. High-quality, standardized data are crucial for accurate ML models, but issues like inconsistent formats and incomplete records can hinder performance and lead to incorrect predictions (Jiang et al. 2017). Protecting patient privacy is another critical challenge, with regulations like HIPAA and GDPR requiring stringent data protection measures, adding complexity to ML applications (Rieke et al. 2020).

Additionally, the interpretability of ML models, especially deep learning algorithms, is a concern as they are often seen as "black boxes," making it difficult for healthcare providers to trust and understand their decision-making processes

(Topol, 2019). Integrating ML models into existing healthcare systems also poses challenges, requiring specialized hardware, continuous maintenance, and collaboration among healthcare providers, IT professionals, and ML experts (Shah et al. 2019).

Despite these challenges, machine learning (ML) offers significant opportunities in healthcare. Advanced algorithms can analyze large datasets, identifying patterns and correlations missed by clinicians, leading to earlier disease recognition and individualized treatment plans, thereby improving patient outcomes and reducing costs (Esteva et al. 2017). Predictive models can forecast disease outbreaks and patient admissions, enabling proactive healthcare (Shashikumar et al. 2017). Additionally, ML streamlines administrative tasks, optimizing staffing and inventory control, which enhances efficiency and reduces costs (Rudin, 2019).

In drug discovery, ML analyzes biological data to identify potential drug candidates and predict their effectiveness, accelerating development and reducing costs (Zhou et al. 2020). The rise of remote monitoring and telemedicine, especially during the COVID-19 pandemic, highlights machine learning’s role in managing patient care from a distance, making healthcare more accessible and efficient (Keesara et al. 2020). Continued research and innovation will further unlock ML's potential in transforming healthcare.

3 ENVIRONMENT AND TECHNOLOGIES

3.1 Python Programming Language

Python is a highly versatile and efficient programming language, widely acclaimed for its significant role in data science, machine learning, and artificial intelligence due to its readable syntax and extensive library support. Its simplicity and readability make it accessible to beginners while its extensive libraries and frameworks make it highly effective for complex tasks. In this project, several Python libraries were utilized:

- **Pandas**: Used for data manipulation and analysis, providing data structures like DataFrames. It simplifies data loading, cleaning, and transformation processes (Pandas Documentation, n.d.).
- **NumPy**: Facilitated numerical computations and matrix operations, which are essential for handling large datasets and performing mathematical operations efficiently (NumPy Documentation, n.d.).
- **Matplotlib and Seaborn**: Employed for data visualization to create graphs and plots. Matplotlib provides comprehensive 2D plotting, while Seaborn offers a higher-level interface to create visually appealing and informative statistical graphics (Matplotlib Documentation, n.d.; Seaborn Documentation, n.d.).

PICTURE 2. Popular Python Libraries for ML (Medium, A. 2020).

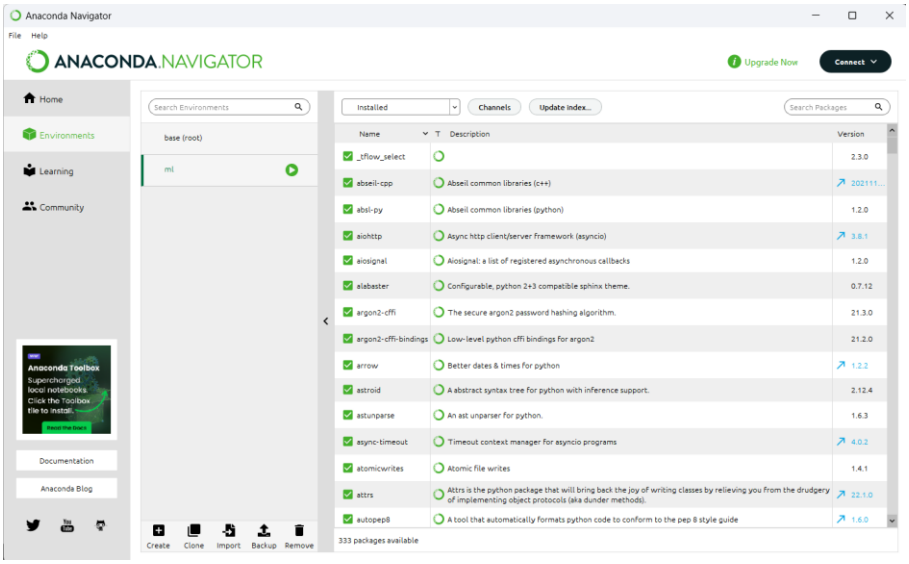
- **Scikit-learn**: Tools are provided for data preprocessing, model training, evaluation, and hyperparameter tuning. It encompasses a broad array of algorithms for classification, regression, clustering, and dimensionality reduction (Scikit-learn Documentation, n.d.).
- **TensorFlow and Keras**: Used for building and training deep learning models, TensorFlow is a robust library for numerical computation and machine learning. Keras, built on top of TensorFlow, provides an intuitive API for developing and evaluating deep learning models (TensorFlow Documentation, n.d.; Keras Documentation, n.d.).

These libraries enabled efficient data handling, visualization, and the implementation of various machine learning algorithms, which were crucial for the project's success.

3.2 Anaconda Platform

Anaconda is one of the popular distributions that used widely for scientific computing of the Python and R programming languages. It streamlines package management and deployment, making it good choice for data science and machine learning projects. Key features of the Anaconda platform include:

- **Conda Package Manager**: Manages packages and dependencies, ensuring compatibility and ease of installation. It allows users to create isolated environments, which helps in managing project-specific dependencies without conflicts (Conda Documentation, n.d.).



PICTURE 3. Anaconda Environment and Packages for the project.

- **Integrated Development Environment (IDE)**: Anaconda comes with Jupyter Notebook and Spyder, which provide interactive coding environments that are especially useful for data analysis and visualization. Jupyter Notebook allows for the combination of code execution, text, and visualizations in a single document, making it a powerful tool for data science projects (Jupyter Documentation, n.d.).
- **Virtual Environments**: Allows the creation of isolated environments to manage different project dependencies, preventing conflicts between libraries. This feature is particularly beneficial when multiple projects requiring different versions of the same libraries are being managed. (Anaconda Documentation, n.d.).

Using Anaconda ensured that all necessary libraries were easily installed and maintained, and it provided a robust environment for developing and testing machine learning models.

3.3 Setting Up the Environment

Combining Python and Anaconda for this project involved several steps which are:

- **Installing Anaconda**: The first step was to download and install the Anaconda distribution, which included Python and the Conda package manager. This setup provided all necessary tools and libraries required for the project (Anaconda Documentation, n.d.).

- **Creating a Virtual Environment**: A new virtual environment was created using Conda to manage dependencies. This isolated environment ensured that the project's dependencies did not interfere with other projects. The following commands were used:
    - `conda create --name heart_disease_prediction python=3.8`
    - `conda activate heart_disease_prediction`

The first command initializes a new isolated Conda environment called "heart_disease_prediction" and installs Python version 3.8 in it, allowing to manage dependencies and packages separately from other projects.

The second command switches the current terminal session to this environment, setting it up as the active workspace where any subsequent commands will use the packages and settings specific to "heart_disease_prediction".

- **Installing Required Libraries**: Necessary libraries were installed within this environment using Conda and Pip. This step ensured that all dependencies were met and that the latest versions of the libraries were used:
    - `conda install pandas numpy matplotlib seaborn scikit-learn`
    - `pip install tensorflow keras`

  The first command installs the specified data analysis and machine learning libraries (Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn) within the active Conda environment, ensuring compatibility and dependency management.

  The second command installs the TensorFlow and Keras libraries using the pip package manager, typically for deep learning tasks, within the same environment or the system's Python environment if no specific Conda environment is active.

- **Developing the Project**: Jupyter Notebook was used for interactive development and documentation of the project. It allowed for real-time code execution and visualization, making it easier to debug and iterate on the machine learning models. The following command was used to launch Jupyter Notebook:
    - `conda install jupyter`
    - `jupyter notebook`

The first command installs Jupyter Notebook within the active Conda environment, providing an interactive web-based platform for creating and sharing documents that include live code, equations, visualizations, and narrative text.

The second command launches the Jupyter Notebook application, opening a web interface in your default browser where you can create and run Jupyter Notebooks.

The development work was carried out on Windows operating system, as the most familiar with this environment and used Linux commands on the terminal. This environment was chosen for its compatibility with the tools and libraries required for machine learning and data analysis.

4 DATA DESCRIPTION AND PREPARATION

4.1 Data Collection

The dataset used in this project is mainly sourced from the UCI Machine Learning Repository and is titled "Heart Disease". This dataset is a comprehensive collection designed to facilitate the prediction and analysis of heart disease events using a variety of clinical and demographic factors collected at several locations worldwide. Previously separate datasets have been combined to create this new comprehensive dataset. After combining five separate heart datasets with eleven shared traits, this dataset is currently the largest heart disease dataset available for research. Its curation involved the usage of the following five datasets:

- Cleveland: 303 observations
- Hungarian: 294 observations
- Switzerland: 123 observations
- Long Beach VA: 200 observations
- Stalog (Heart): 270 observations

There was a total of 1,190 entries. After removing 272 duplicate observations, the merged dataset had 918 unique observations, which were used in the thesis project.

4.2 Data Description

Several factors that capture clinical and demographic data relevant to heart failure prediction make up the dataset. Numerous characteristics are covered, such as age, sex, resting blood pressure, cholesterol, echocardiography results, exercise routines, and more. The goal is to predict the existence of heart disease using binary values, where 0 indicates its absence and 1 indicates its presence. Categorical Columns are Sex, ChestPainType, RestingECG, ExerciseAngina and ST_Slope. The following are 11 variables that this thesis examined:

TABLE 1. Variable Descriptions of Data.

Variable Name	Description	Comments	Values (assumed)
Age	The patient's age in years	Numeric	Real
Sex	Patient's gender	Male, Female	1,0
ChestPainType	Type of chest pain experienced by the patient	TA = Typical Angina, ATA = Atypical Angina, NAP = Non-Anginal Pain, ASY = Asymptomatic	1,2,3,4
RestingBP	Resting blood pressure (measured in mm Hg on admission to the hospital)	Numeric	Real
Cholesterol	Serum cholesterol in mg/dl	Numeric	Real
FastingBS	Fasting blood sugar	FastingBS > 120 mg/dl, FastingBS <= 120 mg/dl	1,0
RestingECG	Resting electrocardiogram results	Normal, ST, LVH	0,1,2
MaxHR	Maximum heart rate achieved during a stress test	Numeric	Real
ExerciseAngina	Exercise-induced angina	Yes, No	1, 0
Oldpeak	ST depression induced by exercise relative to rest	Numeric	Real
ST_Slope	The slope of the peak exercise ST segment	Up, Flat, Down	0,1,2
HeartDisease	Presence of heart disease	Yes, No	1,0

	A	B	C	D	E	F	G	H	I	J	K	L
1	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
2	40	M	ATA	140	289	0	Normal	172	N	0	Up	0
3	49	F	NAP	160	180	0	Normal	156	N	1	Flat	1
4	37	M	ATA	130	283	0	ST	98	N	0	Up	0
5	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
6	54	M	NAP	150	195	0	Normal	122	N	0	Up	0
7	39	M	NAP	120	339	0	Normal	170	N	0	Up	0
8	45	F	ATA	130	237	0	Normal	170	N	0	Up	0
9	54	M	ATA	110	208	0	Normal	142	N	0	Up	0
10	37	M	ASY	140	207	0	Normal	130	Y	1.5	Flat	1
11	48	F	ATA	120	284	0	Normal	120	N	0	Up	0
12	37	F	NAP	130	211	0	Normal	142	N	0	Up	0

PICTURE 4. First Few Rows of the Dataset (Patients).

The dataset includes 918 patients, with 725 males and 193 females. This dataset offers a wide range of characteristics that are necessary for creating and assessing machine learning models that are intended to forecast cardiac disease. More reliable and broadly applicable models can be produced by merging several datasets into one large dataset with consistent properties.

4.3 Data Exploration and Visualization

To understand the characteristics of the dataset, a statistical summary was performed, which includes key metrics such as the mean, median, standard deviation, and ranges of the variables. Here's a summary of the main attributes:

TABLE 2. Statistical Summary of the Data.

Variable Name	Mean	Median	Std Dev	Min	Max
Age	53.51	54	9.08	29	77
RestingBP	132.4	130	18.5	0	200
Cholesterol	198.7	223	109.4	0	603
MaxHR	136.8	138	25.4	60	202
Oldpeak	0.89	0.8	1.1	0	6.2
HeartDisease	0.55	1	0.5	0	1

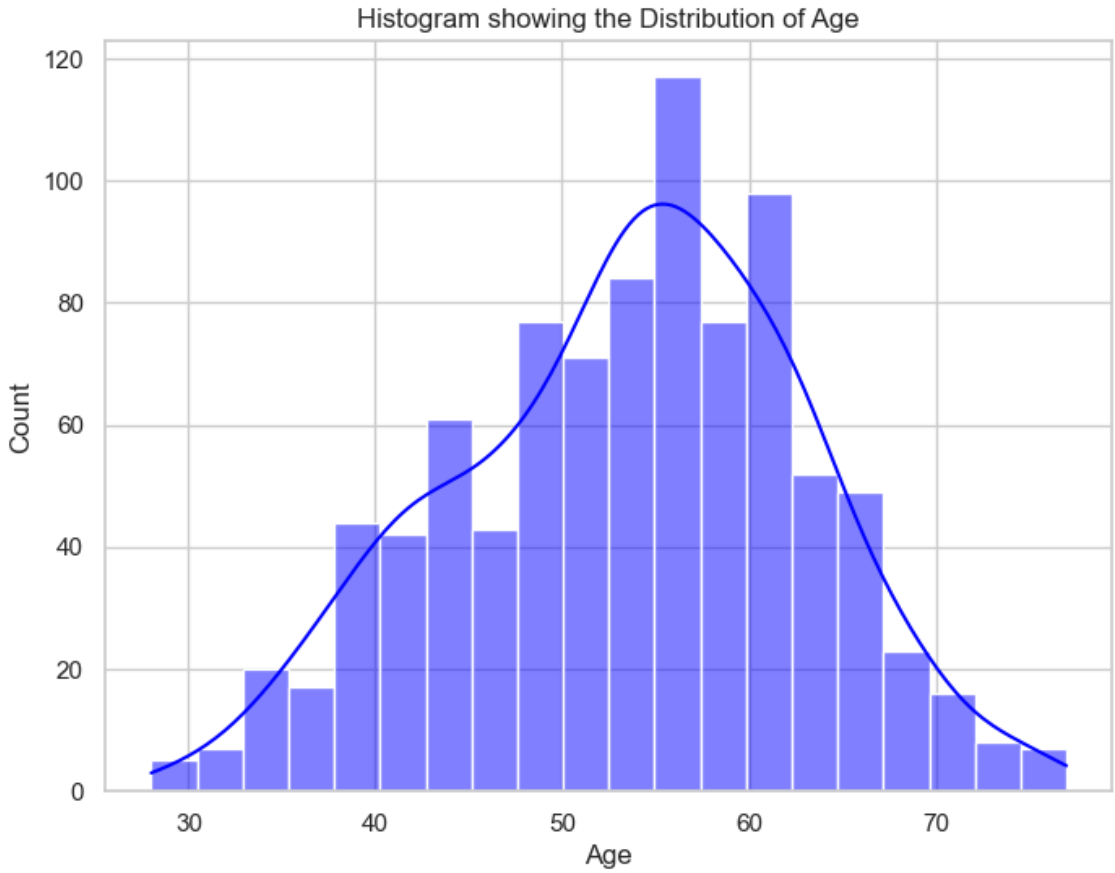


FIGURE 1. Age Distribution of Patients.

The data reveals that most patients are between 40 and 70 years old, with a peak around 55 years. This indicates that middle-aged to older adults make up the majority of the dataset, which is significant for analyzing age-related trends in heart disease.

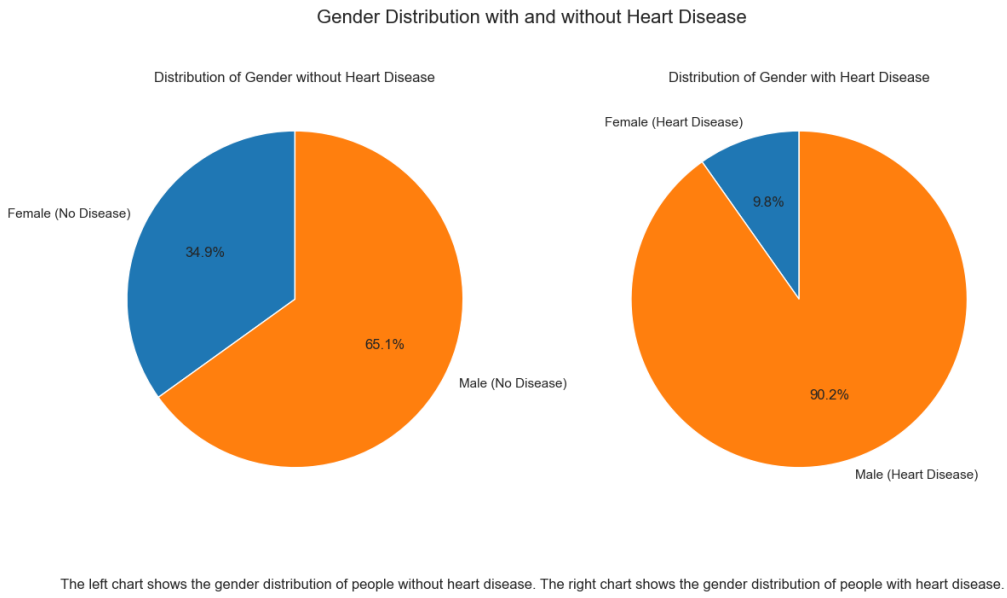


FIGURE 2. Pie Chart of Heart Disease Prevalence by Sex

The provided image (Figure 2) shows two pie charts representing the distribution of heart disease prevalence by sex in the dataset. These charts highlight that the majority of patients in the dataset are male, and a significantly higher percentage of males have heart disease compared to females.

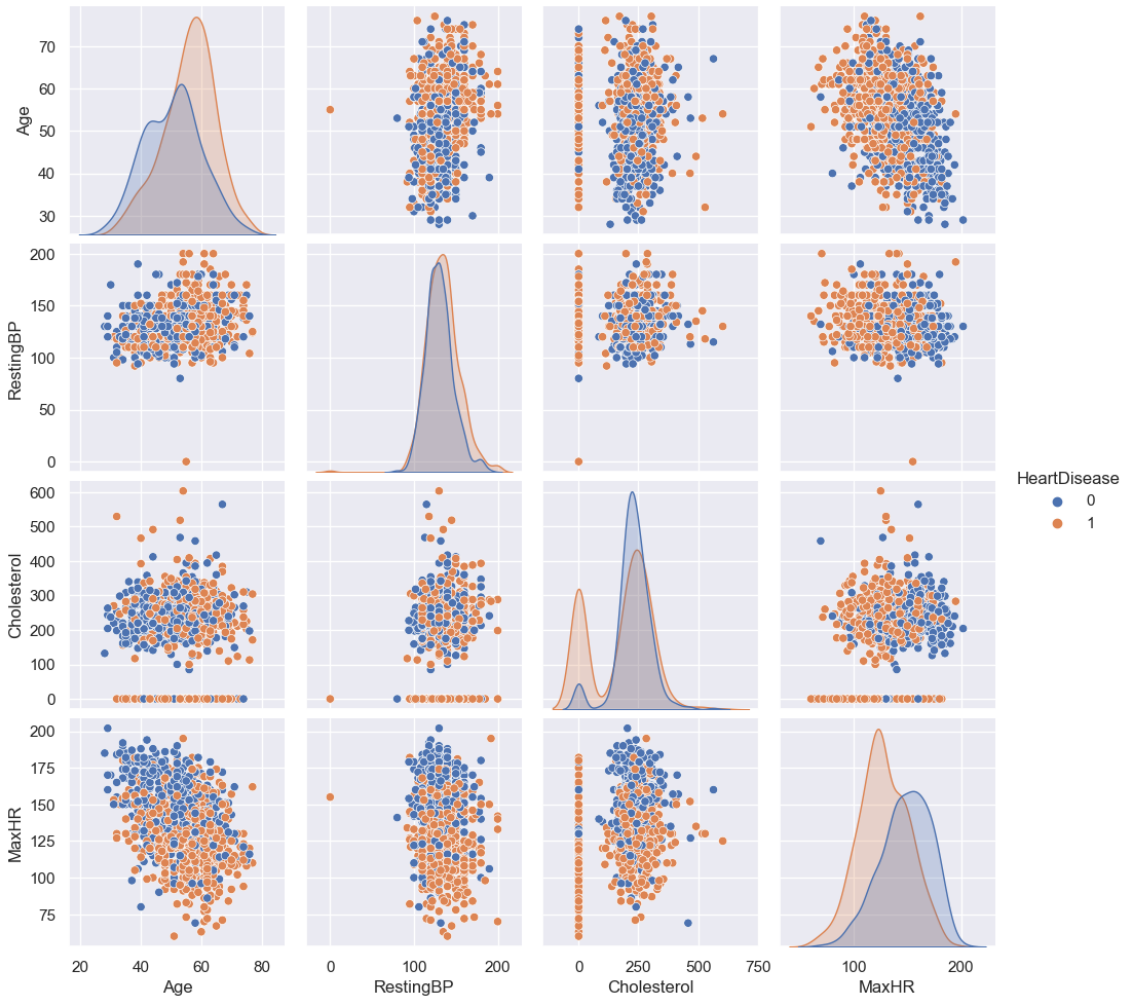


FIGURE 3. Pairplot Analysis of Heart Disease Dataset.

The scatter plot of "Age vs. MaxHR" (maximum heart rate) within the pair plot matrix demonstrates significant correlations between age, maximum heart rate, and the presence of heart disease. On the x-axis, age is represented, and on the y-axis, MaxHR is depicted. The data points, color-coded for clarity, show blue for individuals without heart disease and orange for those with heart disease. A discernible trend indicates that younger individuals generally exhibit higher MaxHR values, while older individuals display lower MaxHR values. Moreover, a pronounced clustering of orange dots at lower MaxHR values, particularly among older age groups, suggests that a lower MaxHR is more frequently associated

with heart disease. This pattern underscores the potential of lower MaxHR as a risk factor or indicator for heart disease, particularly as age increases. Thus, the plot highlights the critical importance of monitoring maximum heart rate in relation to age for assessing heart health.

In summary, the pair plot (Figure 4) shows relationships between key features in the heart disease data:

- Age: Older individuals (55-65 years) show a higher prevalence of heart disease.
- RestingBP: No strong correlation with heart disease; values overlap significantly.
- Cholesterol: Similar distribution in both groups; not a clear indicator.
- MaxHR: Lower maximum heart rates are associated with heart disease.

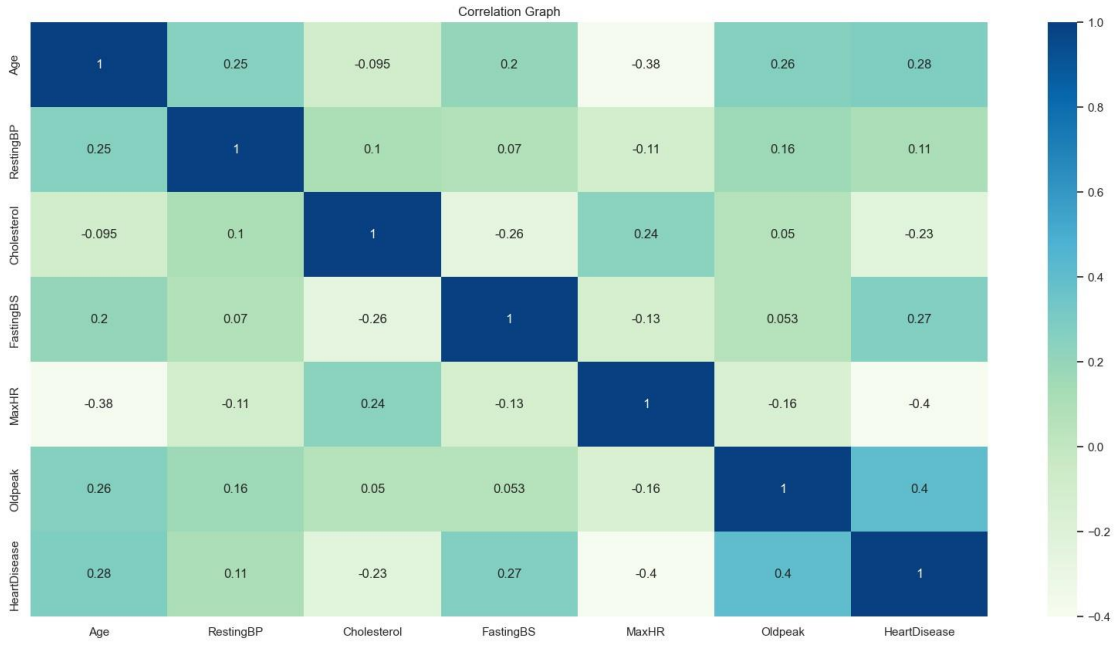


FIGURE 4. Correlation Heatmap of Health Metrics.

The correlation heatmap for the heart failure prediction dataset visually displays the relationships between various health metrics and their association with heart disease. In the heatmap, the strength and direction of correlations are indicated by color: dark blue signifies strong positive correlations, while green represents negative correlations.

- A correlation value of 1 indicates a full positive correlation, or one in which both variables increase when one rises.
- When two variables have a complete negative correlation (i.e., one rises while the other falls), the correlation coefficient is -1.
- When the correlation coefficient is 0, there is no correlation, or a linear relationship, between the variables.

In the heatmap (Figure 4) shows that age has a positive correlation of 0.28 with heart disease, suggesting that older patients are more likely to have heart disease. Age also has a negative correlation of -0.38 with maximum heart rate (MaxHR), indicating that older patients tend to achieve lower maximum heart rates during exercise. Fasting blood sugar (FastingBS) is positively correlated with heart disease at 0.27, meaning higher fasting blood sugar levels are associated with a higher likelihood of heart disease. Additionally, ST depression induced by exercise (Oldpeak) has a correlation of 0.40 with heart disease, showing a strong association between higher ST depression values and the presence of heart disease.

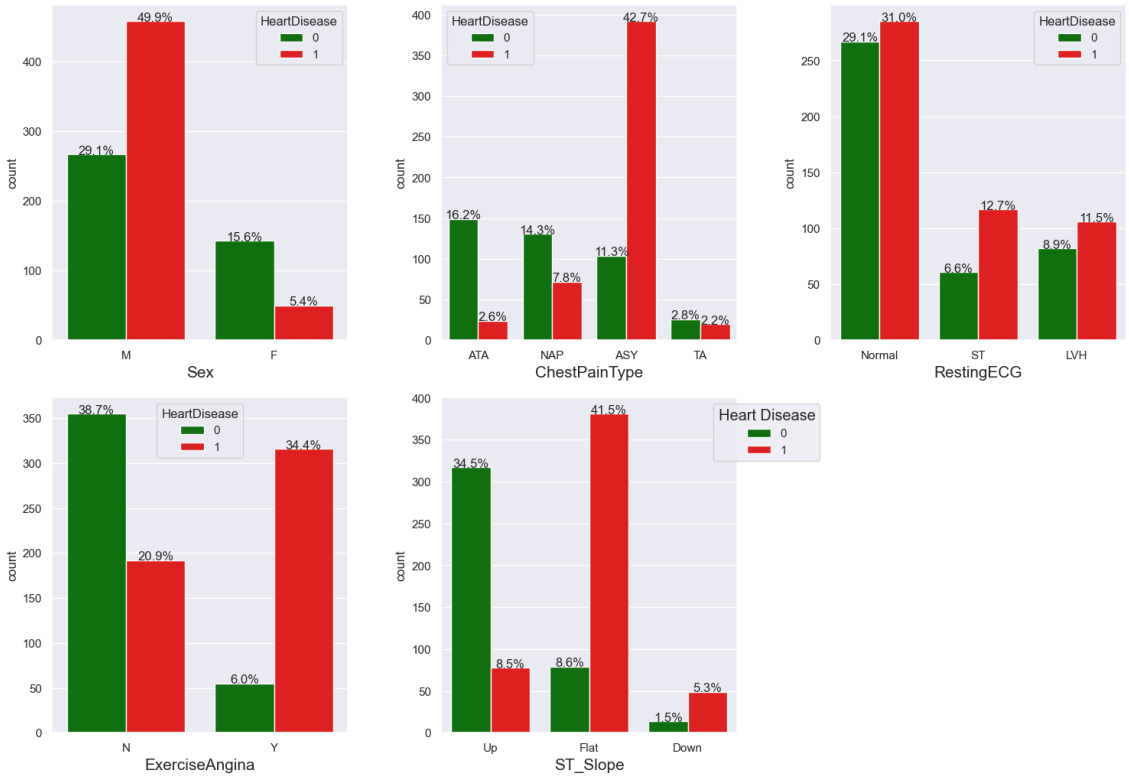


FIGURE 5: Analysis of Heart Disease Risk Factors on Categorical Columns.

The bar plots (Figure 5) illustrate the distribution of heart disease across various categorical risk factors are:

- **Sex**:
    - Males show a higher prevalence of heart disease (49.9%) compared to females (15.6%).
- **Chest Pain Type (ATA, NAP, ASY, TA)**:
    - Asymptomatic (ASY) chest pain type is most associated with heart disease (42.7%).
    - Atypical Angina (ATA) and Non-Anginal Pain (NAP) show lower heart disease prevalence.
- **Resting ECG Results (Normal, ST, LVH)**:
    - Normal ECG results show a balanced distribution between no heart disease (29.1%) and heart disease (31.0%).
    - ST-T wave abnormality (ST) and Left Ventricular Hypertrophy (LVH) are more common in heart disease patients.
- **Exercise-Induced Angina (Y/N)**:
    - Heart disease is more common in patients with exercise-induced angina (Y) (34.4%).
- **ST Slope (Up, Flat, Down)**:
    - A flat ST slope is significantly associated with heart disease (41.5%).

These plots provide insights into how different factors relate to heart disease prevalence, highlighting key indicators such as sex, chest pain type, and ECG results.

5 SYSTEM ARCHITECTURE AND METHODOLOGY

5.1 Architecture of Machine Learning Model

A machine learning (ML) model is a mathematical algorithm or program that, instead of being explicitly coded for a particular task, learns from past data to generate predictions or judgments. The model identifies patterns within the data and uses these patterns to predict outcomes on new, unseen data.

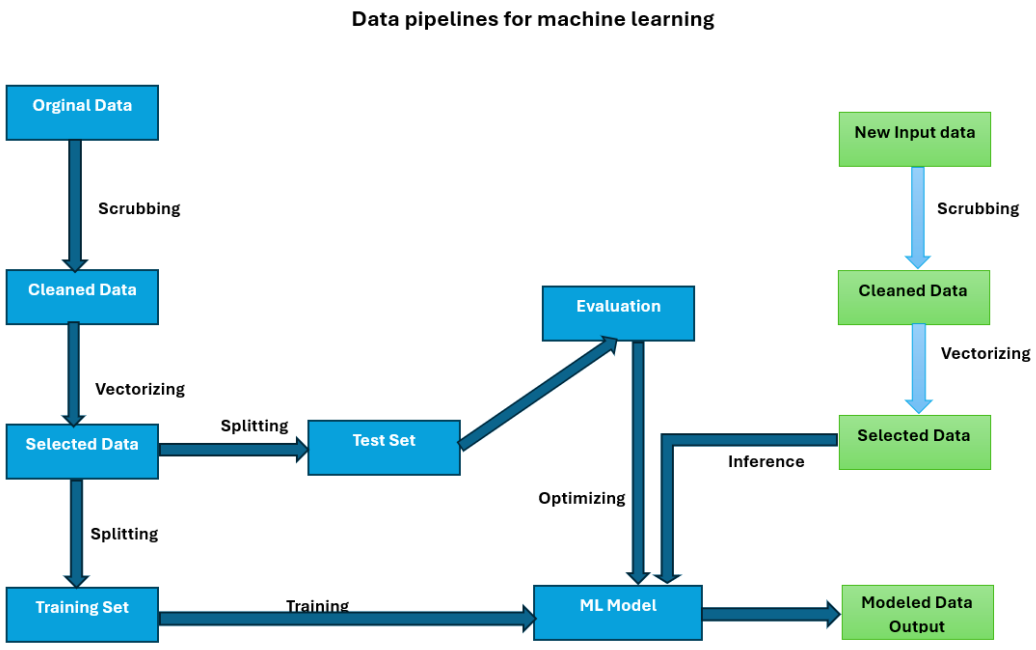


FIGURE 6. Data Pipeline of Machine Learning Model (KDnuggets, n.d. modified).

A machine learning data pipeline is depicted in the figure, with emphasis on the progression from raw data to model output. First, unclean data is scrubbed (cleaning the data by removing or correcting any inaccuracies, inconsistencies, or irrelevant information) to eliminate errors, producing cleansed data. After being vectorized (converting it into a numerical format) for machine learning, the data is separated into testing and training sets. The testing set is used to assess the ML model, which is constructed using the training set. The model is refined and applied for inference on fresh input data, yielding the final modeled output, based on evaluation. Accurate forecasts and ongoing development are guaranteed by this cyclical process.

5.2 Description of Used Machine Learning Model

5.2.1 Support Vector Machines (SVM)

For classification and regression problems, machine learning models called Support Vector Machines (SVM) are utilized. By locating the ideal hyperplane in the feature space, the data points belonging to various classes are separated.

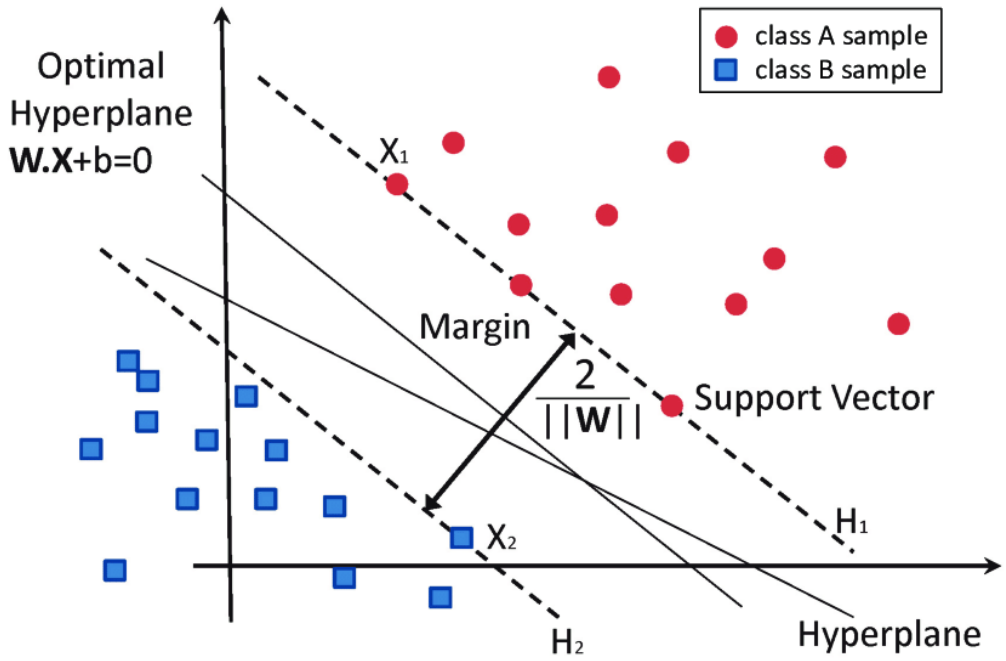


FIGURE 7. Support Vector Machines (SVM) Illustration (Researchgate, A. 2016).

The Figure illustrates a Support Vector Machine (SVM) model used for classification, which can be applied to heart disease data. In this context, SVM classifies patients as having heart disease or not, based on medical attributes such as age, cholesterol levels, and blood pressure. The optimal hyperplane depicted separates the two classes (with and without heart disease) by maximizing the margin between them, defined by the closest data points known as support vectors. This margin maximization helps improve the model's generalization to new data. The slack variable introduces flexibility, allowing for some misclassifications, which is useful for handling real-world data variability and noise. Thus, SVM aids in effectively predicting heart disease, supporting early detection and management.

The margin, or the separation between the closest data points from each class (sometimes referred to as support vectors) and the hyperplane, is maximized by the optimal hyperplane. The equation of the hyperplane in an SVM is given by:

$$w.x + b = 0 \quad (1)$$

where w is the weight vector, x is the input vector, and b is the bias term. The target of the SVM is to find w and b that maximize the margin:

$$\text{maximize, } \frac{2}{||w||} \quad (2)$$

The theory of Support Vector Machines (SVMs) involves finding an optimal hyperplane that isolates two classes of data by raising the margin between them. This is mathematically represented in the 2nd equation. In the context of heart disease data, this theory is applied to classify patients based on medical attributes such as age and cholesterol levels. By training an SVM model using a dataset of patients with known cardiac disease status, the optimal hyperplane is determined to separate those with heart disease from those without. The application of this theory to dataset is detailed in "Introduction to Machine Learning with Python: A Guide for Data Scientists" by Andreas C. Müller and Sarah Guido (O'Reilly Media, 2016, pp. 85-90).

5.2.2 K-Nearest Neighbours (KNN)

K-Nearest Neighbours (KNN) is an instance-based learning algorithm employed for classification and regression. The k nearest neighbors to the query point are identified, and the prediction is made by assigning the most common class among these neighbors. Euclidean distance is typically used to measure the distance between data points by:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i + y_{i)}{}^2} \quad (3)$$

where x and y are data points in an n dimensional feature space. The choice of k and the distance metric significantly influence the model's performance. Detailed discussions on the impact of k and distance metrics on KNN can be found

in "Introduction to Machine Learning with Python: A Guide for Data Scientists" by Andreas C. Müller and Sarah Guido, specifically on pages 58-62.

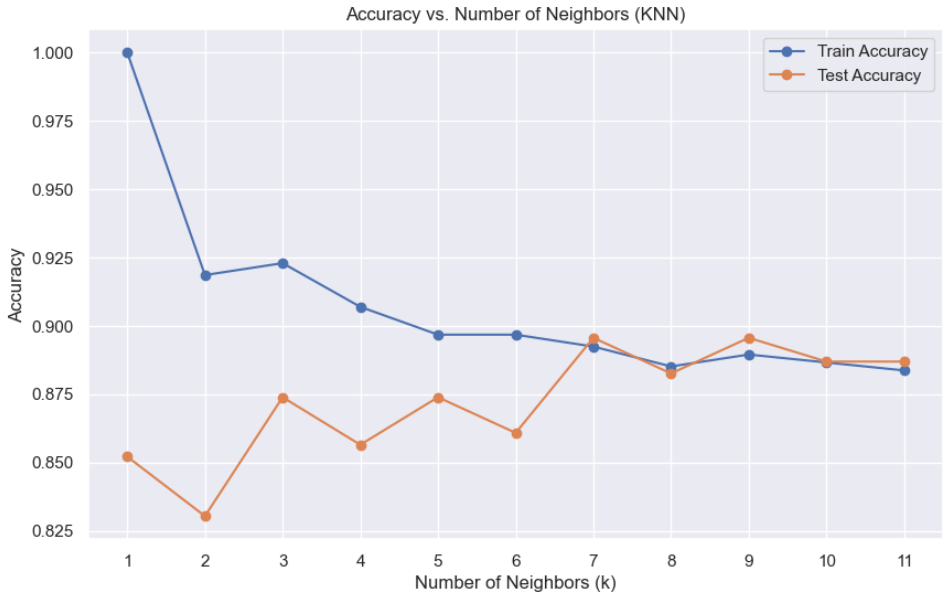


FIGURE 8. Accuracy versus Number of Neighbors (k) in KNN Model.

The figure illustrates the accuracy of a K-Nearest Neighbors (KNN) algorithm in relation to the number of neighbors (*k*) for heart disease prediction. As *k* increases, the training accuracy (blue line) decreases from an initial 100%, indicating reduced overfitting, while the test accuracy (orange line) fluctuates before stabilizing, reflecting the model's generalization ability. The optimal *k* value, around 7-10 in this case, balances both training and test accuracies, suggesting a model that neither overfits nor underfits.

This balance helps ensure accurate predictions for new patients, improving the reliability of heart disease diagnoses and supporting better medical decision-making. At *k=1*, the model overfits, achieving perfect train accuracy but lower test accuracy. As *k* increases, overfitting decreases, and both accuracies stabilize. The optimal performance is observed around *k=9*, where accuracies balance well, indicating good generalization to unseen data. Increasing *k* beyond 9 does not significantly improve accuracy, suggesting that *k=9* is the optimal choice.

Consider a dataset of patients with attributes age and cholesterol, Patient A**:** Age 55, Cholesterol 220, heart disease (Yes). Now, to predict heart disease for a new patient (Patient E) with Age 52 and Cholesterol 210:

- **Calculate Distances**:
  - Use Euclidean distance:

$$d(E,A) = \sqrt{(52-55)^2 + (210-220)^2} = \sqrt{109}$$

- **Interpretation:**
  - This distance indicates how similar Patient E is to Patient A based on age and cholesterol levels. For KNN with *k=1*, Patient E would be classified the same as Patient A since Patient A is the closest neighbor.

This calculated distance shows how different Patient E is compared to Patient A. Despite appearing numerically large, the significance depends on the context of the attributes. For KNN, if Patient A is the closest neighbor to Patient E, then Patient E would be classified in the same category as Patient A (having heart disease in this case).

5.2.3 Decision Trees

Decision Trees, utilized for classification and regression, are non-parametric supervised learning methods. The data is recursively separated into subsets based on the value of an attribute, beginning from the root node and progressing down the tree. Each node in the tree is performed by a decision rule, and each branch performs the outcome of that decision. The goal is to have a tree created that accurately classifies the data with minimal depth.

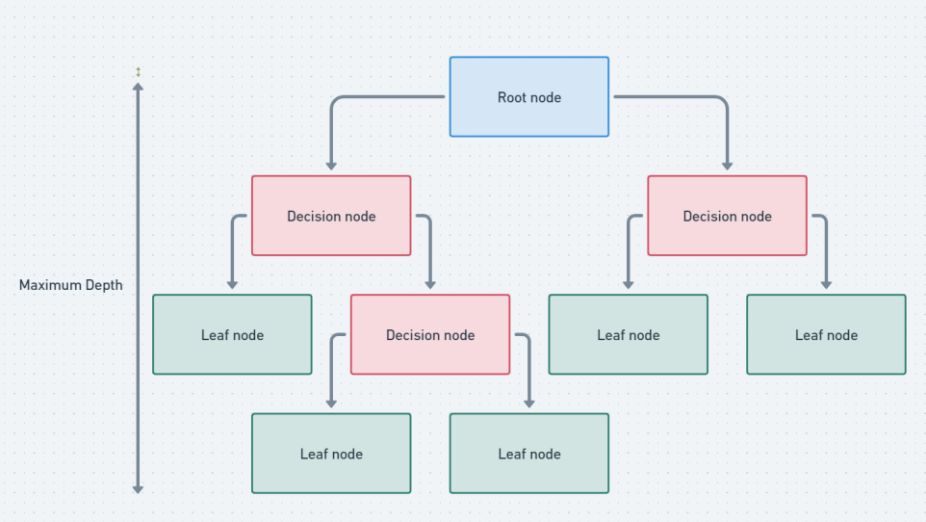


FIGURE 9: Decision Tree Terminologies

A decision tree model predicts heart disease based on patient attributes such as blood pressure and cholesterol levels. It starts at the root node, which splits the dataset using the most significant attribute, like age. From there, decision nodes further split the data based on other attributes, such as cholesterol levels. The procedure carries on until it reaches the leaf nodes, which offer the ultimate outcomes, such as whether or not a patient has heart problems. For example, if a new patient has an age of 52 and cholesterol level of 210, the tree might predict no heart disease based on its learned splits and classifications. This method helps in making accurate predictions and supports early diagnosis and management of heart disease.

The splitting criterion is often based on measures like Gini impurity or information gain:

- **Gini Impurity**: The Gini impurity measure is used to evaluate the quality of a split. It is calculated as:

$$Gini\ impurity = 1 - \sum_{i=1}^{C} {p_i}^2 \quad (4)$$

where p_i is the proportion of class i instances in the node.

Example in the Heart Disease Data:

- Suppose we have a dataset with heart disease data and the target features is whether a patient has heart disease (Yes or No).
- At a particular node, if there are 80 patients with heart disease and 20 without heart disease, the proportions p_i for the classes Yes and No would be 0.8 and 0.2, respectively.
- The Gini Impurity for this node would be:
  - Gini impurity = 1 - (0.8^2 + 0.2^2) = 1 - (0.64 + 0.04) = 1 - 0.68 = 0.32
- A lower Gini Impurity indicates a purer node. For instance, if a node had 100% of one class (e.g., all patients have heart disease), the Gini Impurity would be 0, indicating perfect purity.

- **Information Gain**: Information gain is derived from the concept of entropy (measure of disorder) and measures the reduction in entropy after a dataset is divided based on an attribute. The objective is to maximize information gain, which is determined by subtracting the weighted sum of the entropies of the split's resultant subgroups from the entropy of the original dataset.

For an in-depth understanding and examples, refer to pages 85-90 of "Introduction to Machine Learning with Python: A Guide for Data Scientists" by Andreas C. Müller and Sarah Guido. This section covers the mathematical foundations and practical applications of these splitting criteria in decision trees.

5.2.4 Logistic Regression

Logistic regression is a statistical technique used to model the likelihood of a binary outcome based on one or more predictor variables. Unlike linear regression, which forecasts a continuous outcome, logistic regression is applied to categorical dependent variables, especially when the response is binary (e.g., yes/no, success/failure). This method is widely used in various fields such as medicine, social sciences, and machine learning due to its ability to provide insights into the relationship between dependent and independent variables.

The logistic regression model can be expressed in terms of the logistic (sigmoid) function and the logit function. These functions help relate the predictor variables to the probability of the outcome. where: β_0 is the intercept, β_1, β_2, ..., β_k are the coefficients of the predictor variables X_1, X_2, ..., X_k.

$$\ln(\pi / (1 - \pi)) = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k \quad (5)$$

$$\pi = 1 / (1 + \exp(-(\beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k))) \quad (6)$$

In the context of heart disease, consider a dataset with patients' age and cholesterol levels. The logistic regression model is trained to learn how these predictors influence the likelihood of heart disease. For a new patient with specific age and cholesterol values, the model calculates the log-odds of having heart disease and

then converts this to a probability using the logistic function. This predicted probability helps healthcare professionals assess the patient's risk of heart disease and make informed decisions about further tests or treatments. This application of logistic regression in medical diagnostics is grounded in statistical learning, as detailed in "Introduction to Statistical Learning" by Gareth James et al. and "Applied Logistic Regression" by David W. Hosmer Jr. et al.

A logistic regression curve is shown in the next example graph. The predicted chance of heart disease (a binary dependent variable) is plotted against the blood pressure (a scalar independent variable) in this curve.

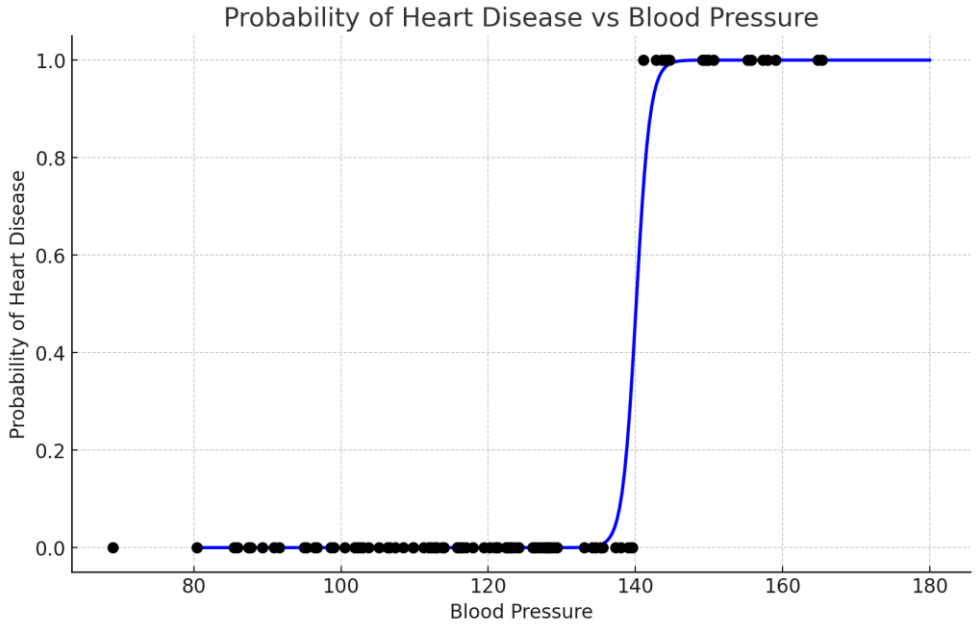


FIGURE 10. Probability of Heart Disease versus Blood Pressure.

The graph above illustrates a logistic regression model predicting the probability of having heart disease based on blood pressure levels. The x-axis represents blood pressure, while the y-axis shows the probability of heart disease. As blood pressure increases, the probability of having heart disease also increases, particularly when blood pressure exceeds 140, demonstrating a clear relationship between higher blood pressure levels and a higher risk of heart disease.

5.2.5 Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem, assuming independence between the features. "Bayesian Data Analysis" by Andrew Gelman and colleagues (675 pages, Third Edition) offers an extensive overview from the principle to advanced methods, with a strong focus on practical applications. The model computes the posterior probability for each class based on the input features and assigns the class with the highest probability. For a given feature vector x and class C, the probability is.

$$P(C \mid x) = P(x \mid C)P(C)/\, P(x) \tag{7}$$

Using the independence assumption, the likelihood $P(x|C)$ can be decomposed as

$$P(x \mid C) = \prod_{i=1}^{n} P(xi \mid C) \tag{8}$$

where xi are the individual features. Naive Bayes classifiers are efficient and perform well in many applications, especially with large datasets and high-dimensional data. For an in-depth understanding and examples, refer to "An Introduction to Statistical Learning: with Applications in R" by Gareth James, et al. 2014 (Pages 101-105).

In the context of heart disease data, this methodology is applied to predict the probability of heart disease based on various risk factors, such as age, cholesterol level, and blood pressure. For instance, consider a dataset where the prior probability of heart disease ($P(C)$) is 10%. If the probabilities of observing a specific age ($x1$), cholesterol level ($x2$), and blood pressure ($x3$) given heart disease are 60%, 70%, and 80% respectively, and the total probability of observing these features in the population ($P(x)$) is 5%, then the posterior probability of heart disease given these features ($P(C|x)$) can be calculated as approximately 67.2%. This approach, by leveraging the independence assumption, simplifies the computation and aids in classifying and diagnosing heart disease by providing a probabilistic framework to assess the risk based on multiple risk factors.

5.3 Model Development and Improvement

Several methodical procedures are involved in machine learning model building and enhancement with the goal of producing a reliable and accurate prediction model. To improve model performance, this iterative method has to be continuously refined. Data preparation, algorithm selection, training, assessment, and hyperparameter adjustment are among the phases.

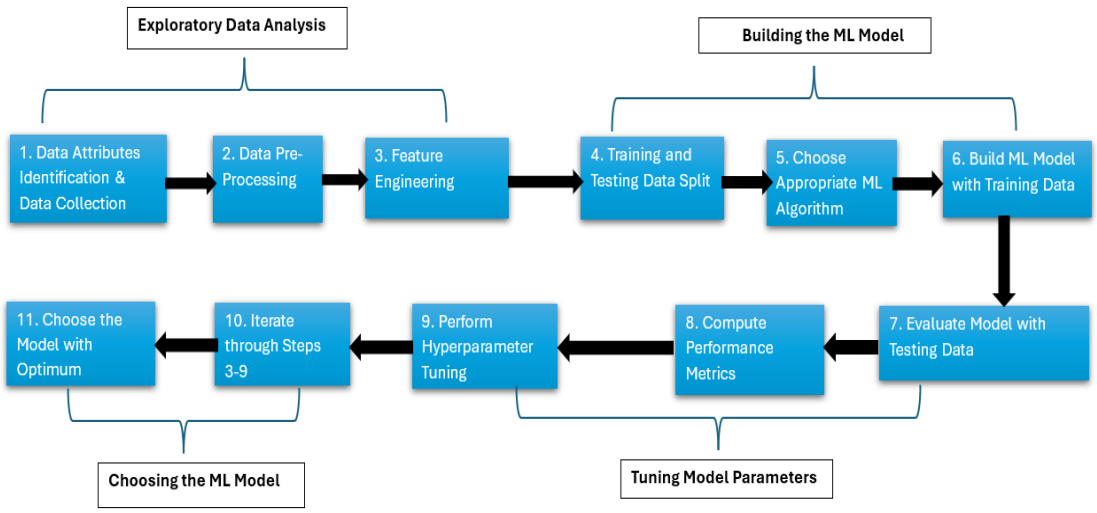
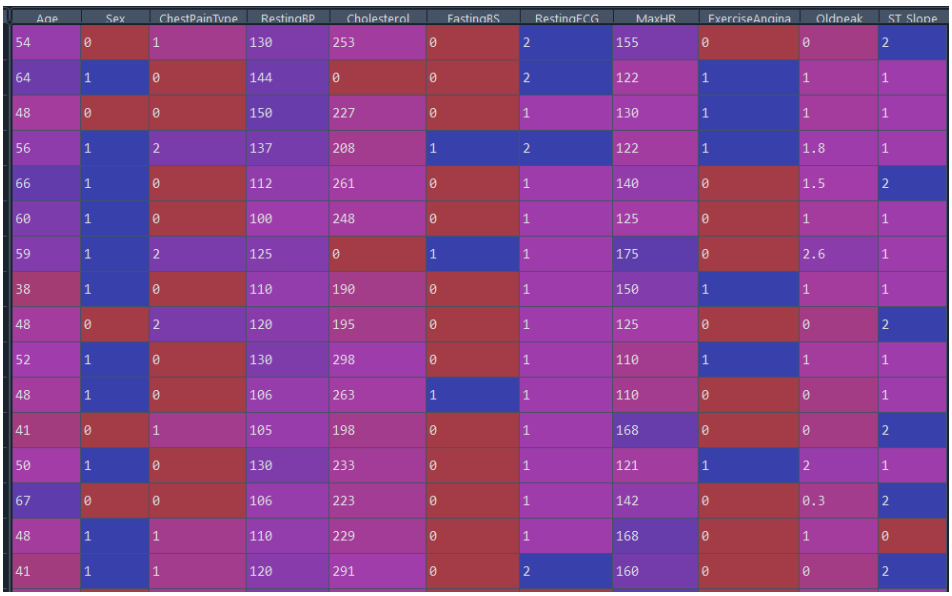


FIGURE 11. Machine Learning Model Development Workflow.

The process is divided into three main stages: Exploratory Data Analysis, Building the ML Model, and Tuning Model Parameters. Here's a detailed explanation of each step in the process:

a. Exploratory Data Analysis

1. Data Attributes Identification & Data Collection:
    - Identifying the relevant data attributes needed for the analysis.
2. Data Pre-Processing:
    - Cleaning the data to handle missing values, outliers, and errors.
    - Transforming the data as needed (e.g., normalization, encoding categorical variables).
    - Scaling means adjusting the range of features to ensure a similar scale, improving algorithm performance and speed.



Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope
54	0	1	130	253	0	2	155	0	0	2
64	1	0	144	0	0	2	122	1	1	1
48	0	0	150	227	0	1	130	1	1	1
56	1	2	137	208	1	2	122	1	1.8	1
66	1	0	112	261	0	1	140	0	1.5	2
60	1	0	100	248	0	1	125	0	1	1
59	1	2	125	0	1	1	175	0	2.6	1
38	1	0	110	190	0	1	150	1	1	1
48	0	2	120	195	0	1	125	0	0	2
52	1	0	130	298	0	1	110	1	1	1
48	1	0	106	263	1	1	110	0	0	1
41	0	1	105	198	0	1	168	0	0	2
50	1	0	130	233	0	1	121	1	2	1
67	0	0	106	223	0	1	142	0	0.3	2
48	1	1	110	229	0	1	168	0	1	0
41	1	1	120	291	0	2	160	0	0	2

PICTURE 5. Data sample after Pre-Processing (Original date: Picture 4, Values: Table 1).

3. Feature Engineering:
    - Generating new features from existing data can enhance the model's performance. Additionally, selecting key features that significantly contribute to the model's predictive power is essential.
4. Training and Testing Data Split:
    - Splitting the dataset into training and testing subsets to apply and check the model's performance. For this project, the dataset separated into two parts: one part (75%) is used to train the machine learning model, and the other part (25%) is used to test the model's performance.

b. Building the ML Model

4. Choose Appropriate ML Algorithm:
    - Choosing an appropriate machine learning algorithm based on the problem type and the characteristics of the data.
5. Build ML Model with Training Data:
    - Executing the selected Machine Learning algorithm using the training data.
6. Evaluate Model with Testing Data:
    - Evaluating the trained model on the testing data to assess its performance.

7. Compute Performance Metrics:
    - Calculating performance metrics (e.g., accuracy, precision, recall, F1-score) to quantify the model's effectiveness.

c. Tuning Model Parameters

9. Perform Hyperparameter Tuning:
   Hyperparameters are settings specified before training a machine learning model, such as the learning rate or number of trees, that control the training process and influence the model's performance. Grid search systematically tries multiple combinations of hyperparameters to find the best one that maximizes the model's performance.

    - Fine-tune the model's hyperparameters to improve its performance.

```
# Perform grid search for each model
grid_search_results = []
for params in all_params:
    model = params['model']
    param_grid = params['params']
    grid_search = GridSearchCV(model, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
    grid_search.fit(X_train_scaled, y_train)
    best_params = grid_search.best_params_
    best_score = grid_search.best_score_
    grid_search_results.append((model.__class__.__name__, best_params, best_score))
```

PICTURE 6. Hyperparameter Tuning by Grid Search.

This code snippet demonstrates how to find the best hyperparameters for different machine learning models using a method called grid search. Hyperparameters are settings that you configure before training a model, such as the learning rate or the number of trees in a forest. Grid search systematically tries all possible combinations of these settings to determine which combination gives the best performance.

10. Iterate through Steps 3-9:
    - Reiterate the process from feature engineering to hyperparameter tuning to improve the model iteratively.

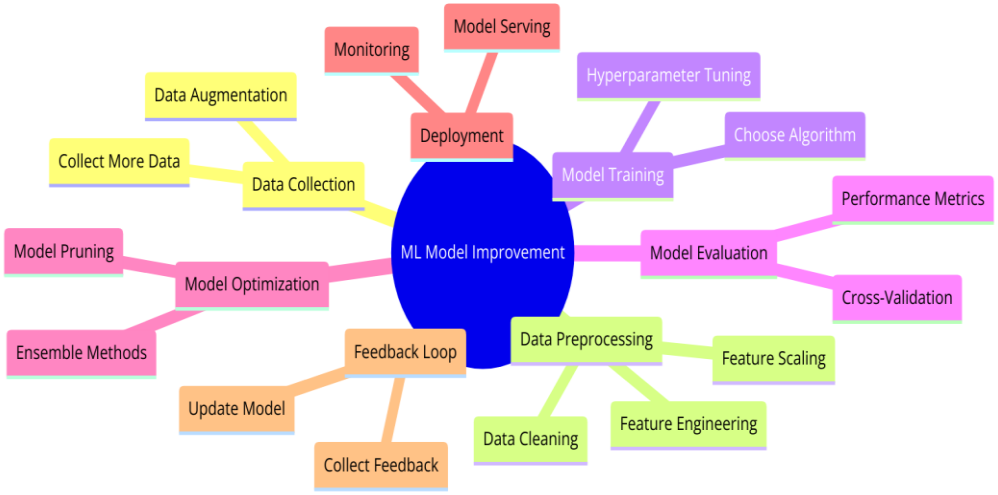


FIGURE 12. Mind map diagram for ML Model Improvement.

11. Choose the Model with Optimum:
    - Selecting all the model with the best performance based on the evaluation metrics (combining) and deploy it.

```python
# Create tuned models
logit_tuned = LogisticRegression(**logistic_regression_params)
decision_tuned = DecisionTreeClassifier(**decision_tree_params)
knn_tuned = KNeighborsClassifier(**k_neighbors_params)
svm_tuned = SVC(**svm_params)
gnb_tuned = GaussianNB()

# Create a list of tuned models with their best parameters
model_tuned = [
    ('logit', logit_tuned),
    ('decision', decision_tuned),
    ('svm', svm_tuned),
    ('knn', knn_tuned),
    ('gnb', gnb_tuned)
]

# Create the VotingClassifier
voting_clf = VotingClassifier(estimators=model_tuned, voting='hard')
```

PICTURE 7. Combined models for making Best Optimum model.

This code snippet demonstrates how to choose the best model by combining several machine learning models into a single, more robust model using a voting classifier. First, it creates and tunes multiple models (logistic regression, decision tree, k-nearest neighbors, support vector machine, and Gaussian Naive Bayes) with their best hyperparameters. These tuned models are then organized into a list. Finally, a voting classifier is created, which combines the predictions of all these models and makes a final prediction based on the majority vote, thus leveraging the strengths of each individual model to achieve more accurate and reliable results.

5.4 Model Evaluation Terms

When developing and evaluating machine learning models, particularly classification models, it is essential to evaluate their performance using several metrics. Three commonly used metrics are accuracy, the classification report, and the confusion matrix. Each metric provides distinct insights into the model's performance, allowing for a comprehensive evaluation.

- **Accuracy:** The ratio of correctly predicted cases to the total number of cases. It is given by the formula:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \times 100\% \quad (9)$$

- **Classification Report:** A classification report provides detailed performance metrics for each class in the dataset. It typically includes the following:

  - **Precision**: The proportion of accurately predicted positive observations to all positive predictions. It provides an answer to the query, "How many of all the predicted positive instances are actually positive?"

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positive\ (FP)} \quad (10)$$

  - **Recall (Sensitivity)**: The answer to the question "Of all the actual positive cases, how many are correctly predicted?" is found in the proportion of correctly predicted positive observations to all actual positive observations.

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)} \quad (11)$$

- **F1-Score**: For unbalanced class distributions, F1-score is the harmonic mean of accuracy and recall. The F1-score has a range of 0 to 1, where 1 represents the optimal performance.

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

- **Confusion Matrix:** A confusion matrix is a table of components used to describe the performance of a classification model by comparing the actual target values with the model's predictions.
  - True Negatives (TN): The count of accurate negative predictions.
  - False Positives (FP): The count of incorrect positive predictions (Type I error).
  - False Negatives (FN): The count of incorrect negative predictions (Type II error).
  - True Positives (TP): The count of accurate positive predictions.

The matrix is structured as follows:

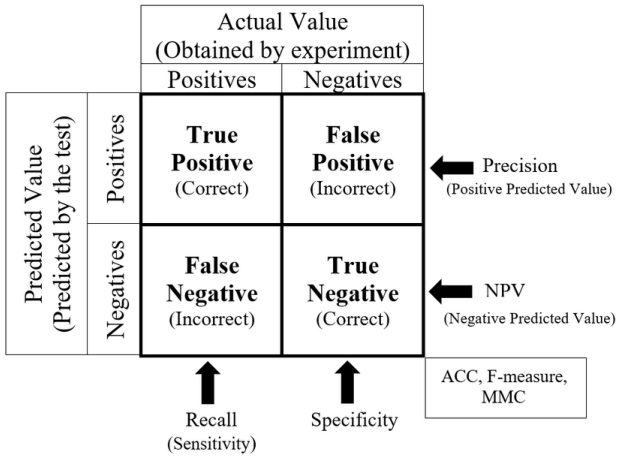


FIGURE 13. Metrics Used to Evaluate Model (ResearchGate, n.d.).

From the confusion matrix, several other metrics can be derived, including accuracy, precision, recall, and specificity. The confusion matrix is particularly valuable when dealing with inequality datasets, as it shows the exact distribution of prediction errors and highlights the performance across different classes.

Besides, the next Figure shows ROC (Receiver Operating Characteristic) curve graph that that plots the True Positive Rate (TPR) versus the False Positive Rate (FPR) at different threshold values to show how well a binary classifier system performs. The classifier's general capacity to tell between positive and negative classes is measured by the ROC AUC (Area Under the Curve) score. Better model performance is indicated by a larger AUC value, with a value of 1 representing perfect classification and 0.5 indicating performance no better than random chance.

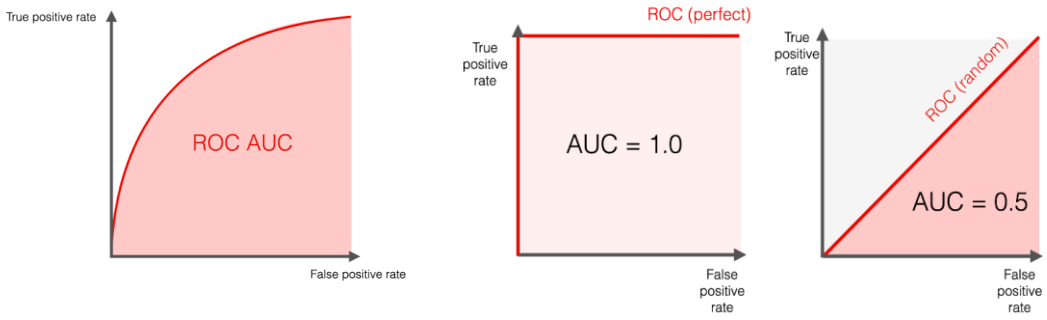


Figure 14: ROC graph and AUC values for Model Evaluations (Evidently, n.d.).

5.5 Feature Importance Analysis Techniques

Feature importance identifies the most significant features that contribute to a model's predictions, enhancing model interpretability, aiding in dimensionality reduction, and potentially improving model performance by focusing on the most impactful features. In this thesis project, various methods were used to analyze feature importance across different machine learning models.

Firstly, in logistic regression, feature importance was assessed through the magnitude of the coefficients, which specify the strength and direction of the relationship between the features and the target variable (heart disease). Coefficients were plotted (Figure 20) to visualize their impact. Secondly, Gini importance, or mean decrease impurity, was used in decision trees to measure the total reduction in impurity brought by each feature. This importance score was calculated to show the contributions of each feature to the model's predictions.

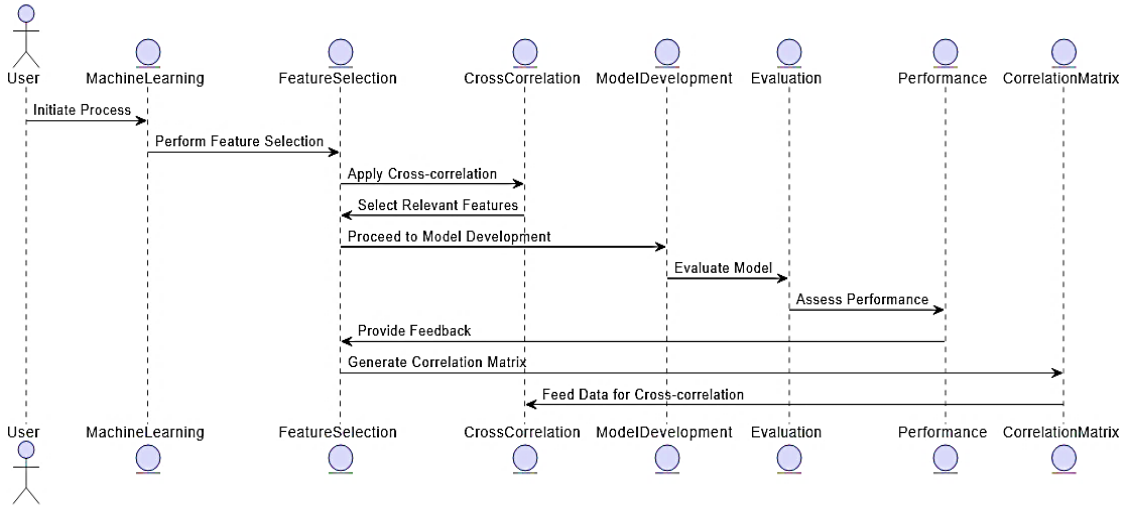


FIGURE 15. Sequence diagram of Feature Selection in Machine Learning.

Thirdly, permutation importance was used to assess feature significance by measuring the increase in prediction error when feature values were randomly rearranged. This method was applied to the K-Nearest Neighbors model, with results visualized to identify the most important features. Models with built-in attributes such as 'feature_importances' were used to provide feature importance scores directly.

By employing these methods, the thesis project effectively identified the most influential features in predicting heart disease, aiding in model interpretation, efficiency, and improvement.

5.6 Challenges Faced during Implementation.

Implementing the heart disease prediction model came with several challenges, especially with data quality and preparation. We had to deal with missing values and remove duplicate records to ensure the data was accurate. Converting categorical data (like gender and chest pain type) into numerical values made the data more complex. Choosing the right models (Logistic Regression, Decision Tree, KNN, SVM, and Naive Bayes) required understanding each model's strengths and weaknesses. Tuning the models' parameters was time-consuming and required careful use of computing resources to find the best settings.

Evaluating the models and understanding which features were most important also presented challenges. We had to select the right metrics (accuracy, precision, recall, F1-score) and use cross-validation to ensure the models performed well and didn't overfit. Interpreting which features were most important to the models was tricky, especially with KNN. Finally, saving the model correctly using '*pickle'* and making sure it could make accurate predictions on new data in real-time was crucial.

The reliability of the results from a model trained on a small dataset, particularly when only a quarter of the dataset is used for training, is limited. A dataset with fewer than 1000 patients is considered quite small compared to datasets with tens of thousands or even hundreds of thousands of items. The model's capacity to effectively generalize to new data sets is impacted by the small dataset size, increases the risk of overfitting, and makes the derived metrics less stable and statistically significant. The model might not capture the full variability and complexity of the problem, leading to less robust performance.

For more reliable results, it is recommended that a larger portion of the dataset, typically 70-80%, be used for training. Additionally, techniques such as cross-validation should be employed to evaluate the model's performance more robustly. Therefore, while insights can be gained from the model, the small training dataset significantly compromises the reliability of the results. Ensuring consistent data processing steps during deployment was vital challenges to keep the model reliable in practical use.

6 RESULTS AND DISCUSSION

6.1 Performance Evaluation Metrics

TABLE 2. ML Model Performance Evaluation Matric Report.

Metric	Logistic Regression	Decision Tree	K-Nearest Neighbors	SVM	Naive Bayes
Accuracy	0.865	0.804	0.861	0.861	0.865
Precision (Class 0)	0.82	0.77	0.82	0.81	0.82
Precision (Class 1)	0.90	0.83	0.90	0.90	0.90
Recall (Class 0)	0.88	0.78	0.87	0.88	0.88
Recall (Class 1)	0.86	0.83	0.86	0.85	0.86
F1-Score (Class 0)	0.85	0.77	0.84	0.84	0.85
F1-Score (Class 1)	0.88	0.83	0.88	0.88	0.88
True Negative (TN)	86	76	85	86	86
False Positive (FP)	12	22	13	12	12
False Negative (FN)	19	23	19	20	19
True Positive (TP)	113	109	113	112	113
MSE	0.135	-	-	-	-
RMSE	0.367	-	-	-	-
MAE	0.135	-	-	-	-

Based on the performance metrics, Logistic Regression and Naive Bayes emerged as the best models. Both models achieved the highest accuracy of 0.865, indicating they are equally effective in correctly predicting heart disease.

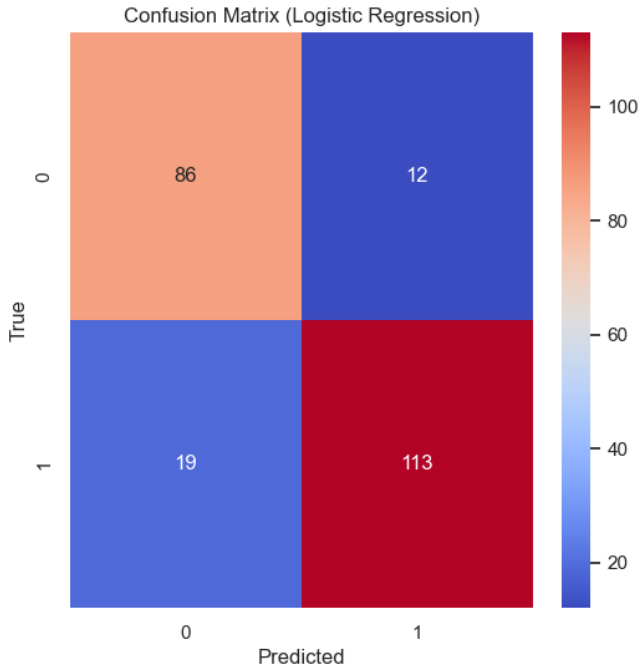


FIGURE 16. Logistic Regression (LR) Confusion Matrix.

The confusion matrix for the Logistic Regression model shows that 19 out of 132 cases were incorrectly predicted as negative, and 12 out of 98 cases were incorrectly predicted as positive. This results in a misclassification rate of approximately 15-20%. While the dataset size can influence model performance, other factors such as feature selection, model tuning, and data quality could also contribute to the inaccuracies. Smaller datasets can lead to less reliable estimates of model performance and might not capture the full variability in the data, but they are not the sole reason for prediction errors.

Additionally, they have high precision, recall, and F1-scores for both classes (0 and 1), showing a balanced performance in terms of both sensitivity (recall) and specificity (precision). The confusion matrices for both models also reveal a similar distribution of true positives, true negatives, false positives, and false negatives, which further supports their comparable and strong performance.

Although K-Nearest Neighbors and SVM also did well with accuracy scores of 0.861, Logistic Regression and Naive Bayes showed slightly better overall metrics, making them the top choices for this dataset.

6.2 Comparative Analysis of Different Models

In this section, a comparative analysis of several machine learning models is provided based on their performance metrics, specifically focusing on accuracy and the Area Under the Curve (AUC) of their ROC (receiver operating characteristic) curves in the thesis project. The best parameters for each model and their respective cross-validation accuracies are considered in the analysis.

A method for assessing a machine learning model's performance and making sure it applies well to new data is cross-validation. It offers a more precise measure of the model's performance and aids in preventing overfitting, which occurs when a model achieves well on training data but improperly on fresh data.

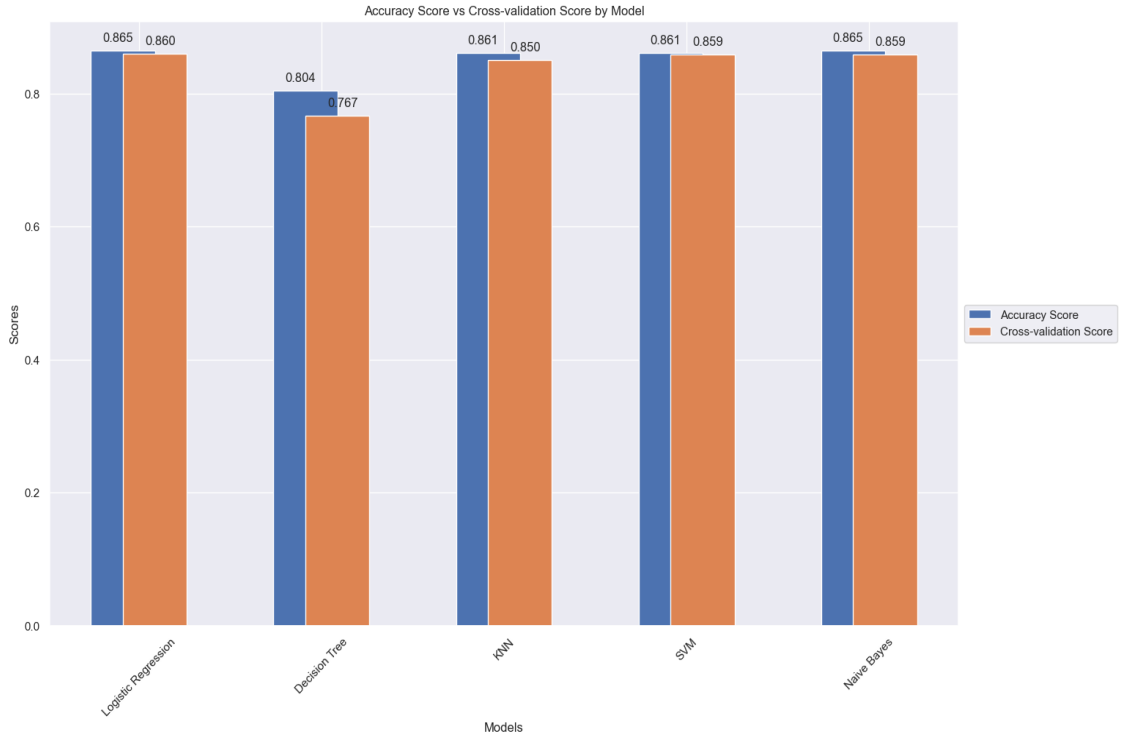


FIGURE 17. ML Models Performance with Cross-Validation.

The first chart below indicates that Logistic Regression, KNN, SVM, and Naive Bayes models generalize well to unseen data with minimal overfitting, as their

accuracy scores are close to their cross-validation scores. The Decision Tree model shows signs of overfitting, with a larger gap between its accuracy score and cross-validation score.

The Voting Classifier is a powerful ensemble method that enhances model performance by combining the predictions of multiple classifiers. It offers improved accuracy, robustness, and flexibility, making it an excellent choice for tasks requiring high reliability and performance. The Voting Classifier's ability to leverage diverse model strengths makes it a valuable tool in the machine learning toolkit.

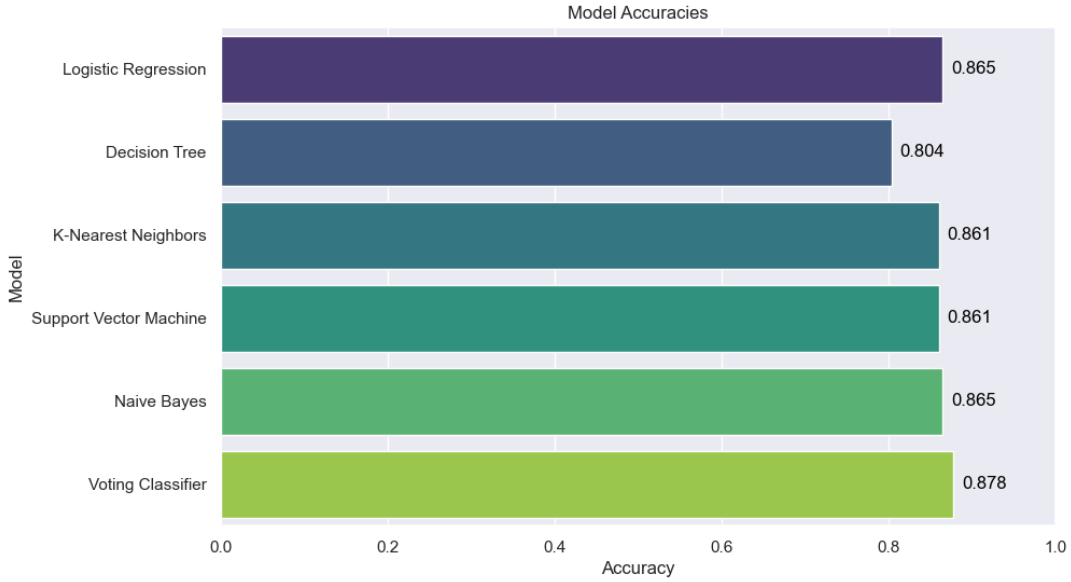


FIGURE 18. Model Accuracies Comparison with Voting Classifier.

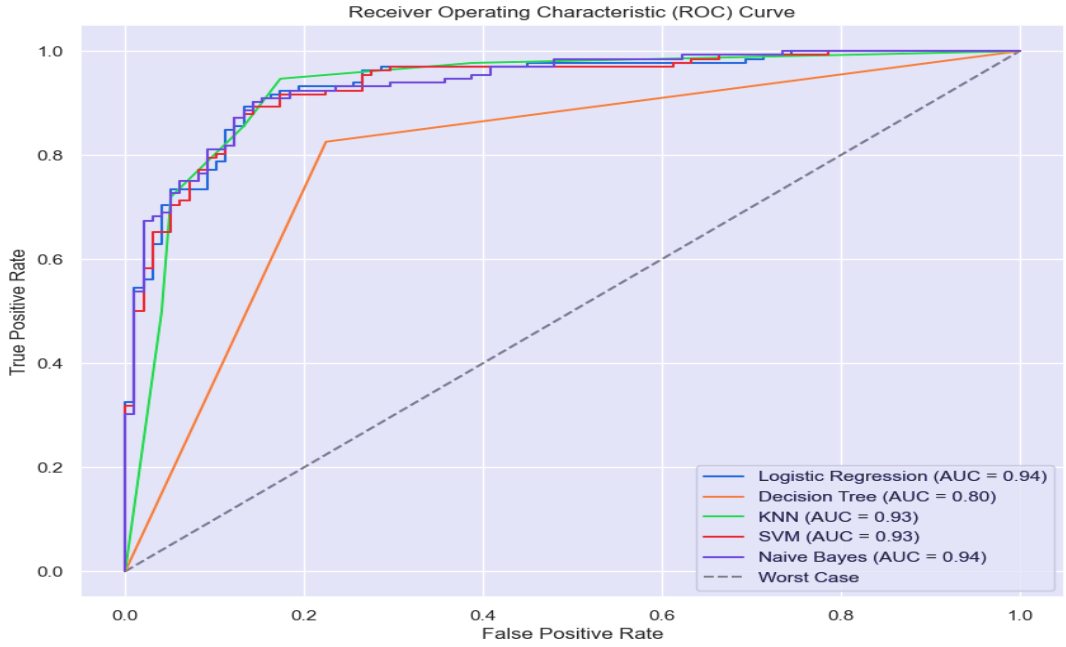


FIGURE 19. Receiver Operating Characteristic (ROC) curves of ML Models.

The analysis chart of the Receiver Operating Characteristic (ROC) curves compares the performance of several machine learning models. The primary performance metric used is the Area Under the Curve (AUC). Logistic Regression and Naive Bayes both achieve an AUC of 0.94, indicating their superior ability to distinguish between classes.These models demonstrate strong performance and robustness in classification tasks. K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) also perform well, each with an AUC of 0.93, showing they are effective at class distinction. In contrast, the Decision Tree model has the lowest AUC at 0.80, suggesting relatively poorer performance. Therefore, Logistic Regression and Naive Bayes are identified as the best models in this analysis, providing high accuracy and reliability for classification tasks.

So, for predicting whether patients will have heart disease, it means that Logistic Regression and Naive Bayes are the most reliable models. These models are the best at accurately classifying patients' risk of heart disease. K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) also work well, but the Decision Tree model does not perform as well due to its tendency to overfit and its less effectiveness at capturing complex patterns in the data compared to other models. Therefore, Logistic Regression and Naive Bayes should be used because they give the most accurate results.

6.3 Important Features and Key Insights

In the context of heart disease prediction, feature importance analysis helps identify which variables significantly contribute to predicting the likelihood of heart disease. This analysis typically involves using machine learning models like K-Nearest Neighbors (KNN), or Logistic Regression to rank features based on their predictive power.

a. Logistic Regression Coefficients

The first chart shows the coefficients of a logistic regression model, which indicate the direction and strength of the relationship between each feature and the likelihood of heart disease. Key insights include:

- **Positive Coefficients**: Features such as MaxHR, Sex_M, ChestPainType_ATA, and ST_Slope_Up have positive coefficients, suggesting that higher values of these features are associated with an increased likelihood of heart disease.
- **Negative Coefficients**: Features such as Cholesterol, FastingBS, Oldpeak, ChestPainType_TA, and ST_Slope_Flat have negative coefficients, indicating that higher values are associated with a decreased likelihood of heart disease.

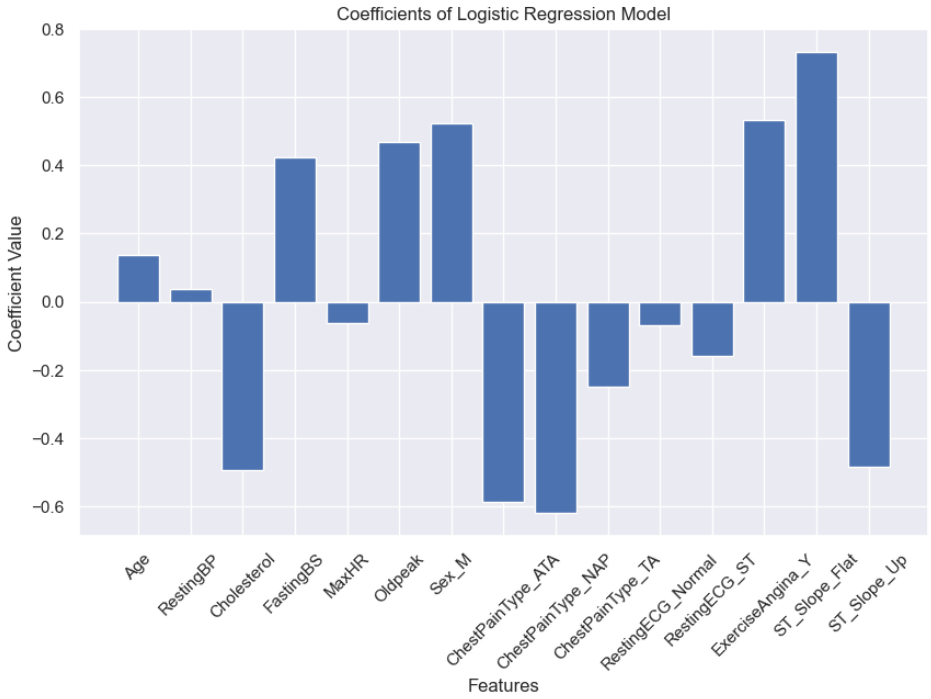


FIGURE 20. Logistic Regression Coefficients.

- **High Magnitude Coefficients**: Features with larger absolute values of coefficients, such as ST_Slope_Up and ChestPainType_ATA, have a stronger influence on the model's predictions.

b. Permutation Importance for KNN

Permutation importance is a method used to determine the significance of each feature in a predictive model by rearranging the values of each feature and observing the impact on the model's accuracy. If altering a feature's values leads to a significant decrease in accuracy, the feature is deemed important. Conversely, if the model's accuracy remains largely unchanged, the feature is considered less important. In the context of predicting whether a patient will have heart disease,

permutation importance helps identify which features are crucial for accurate predictions. Features with low permutation importance, such as RestingECG_Normal, RestingECG_ST, RestingBP, and Age, have minimal influence on the model's ability to predict heart disease and may be considered irrelevant for the prediction task.

The chart below shows the permutation importance for a K-Nearest Neighbors (KNN) model. Key insights include:

- **Top Features**: Cholesterol, ChestPainType_ATA, FastingBS, and ChestPainType_TA are identified as the most important features for the KNN model, as rearranging these features leads to the most significant decrease in model performance.

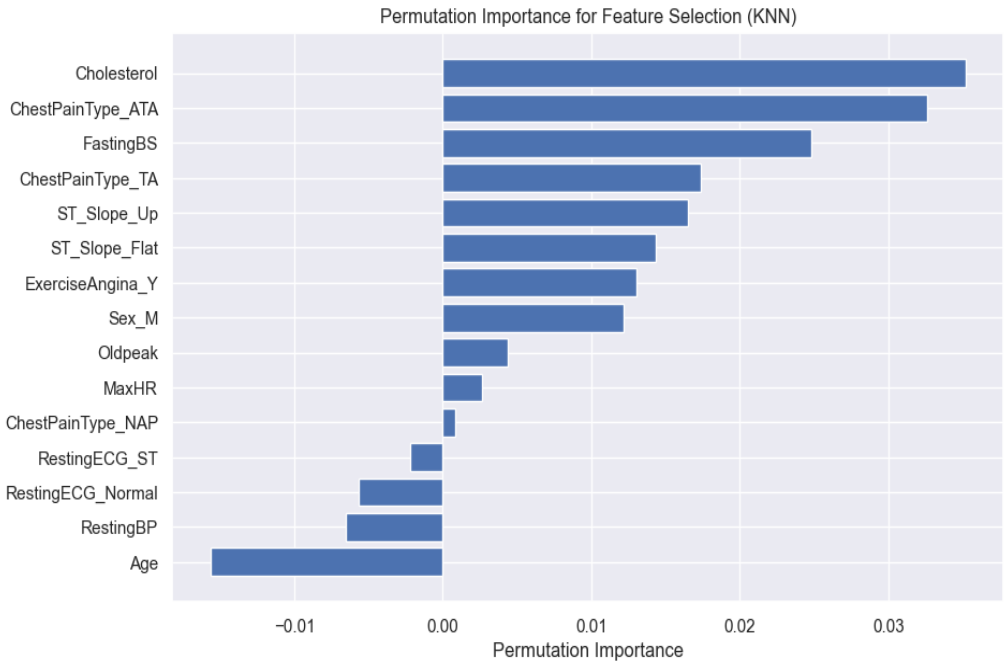


FIGURE 21. Permutation Importance for KNN.

- **Moderately Important Features**: Features such as ST_Slope_Up, ST_Slope_Flat, ExerciseAngina_Y, and Sex_M are also important but to a lesser extent.

- **Less Important Features**: Features like Age, RestingBP, and RestingECG_Normal have lower importance scores, suggesting they have less influence on the KNN model's predictions.

Combined Key Insights

By combining insights from both models, we can identify the most consistently important features for heart disease prediction:

- Cholesterol: Identified as important in both models.
- Chest Pain Type (e.g., ATA, TA): Highly significant in both models.
- Fasting Blood Sugar (FastingBS): Important in both models.
- Sex (e.g., Male): Notable in both models.
- ST Slope (e.g., Up, Flat): Significant in both models.
- Exercise Induced Angina (ExerciseAngina_Y): Important in both models.
- Maximum Heart Rate Achieved (MaxHR): Important in the logistic regression model.

The analysis of the coefficients and permutation importance indicates that certain features may be irrelevant in predicting heart disease. RestingECG_Normal, RestingECG_ST, RestingBP, and Age have been identified as having minimal influence. These features exhibit coefficients near zero in the logistic regression model and low permutation importance values. Consequently, it can be inferred that these features likely have little to no effect on the prediction of heart disease.

6.4 Model Testing and Deployment

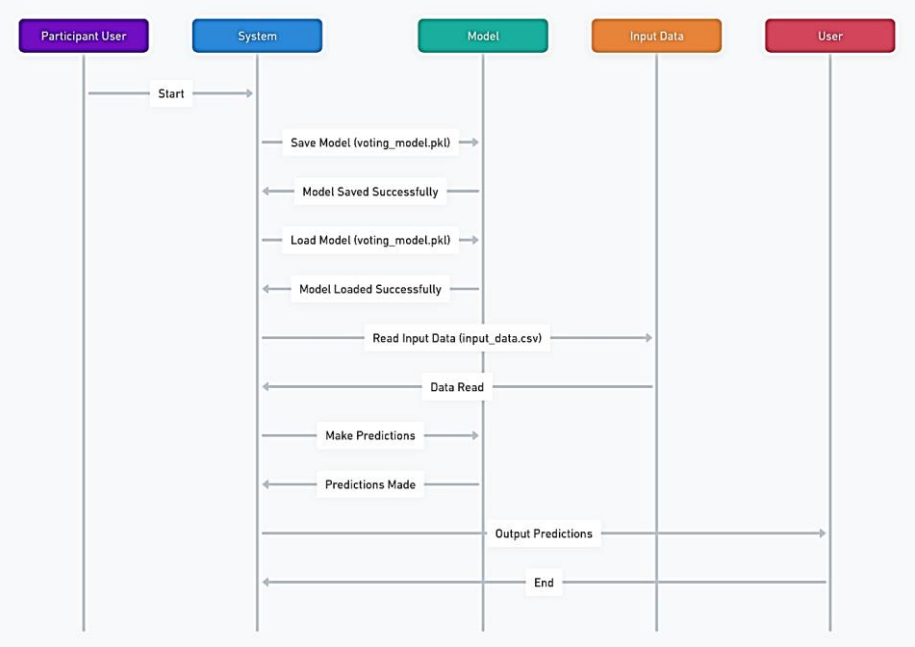


FIGURE 22. Model Prediction Process Sequence Diagram on New Data.

The Sequence diagram outlines the step-by-step interactions involved in making predictions using a saved model. The process begins with the user initiating the sequence, prompting the system to save the model (voting_model.pkl). Once saved, the model is loaded back into the system successfully. The system then reads input data from a file (input_data.csv), processes this data, and makes predictions using the loaded model. These predictions are then outputted by the system, concluding the process when the user receives the final predictions. This sequence diagram visually represents the flow of operations between the user, the system, the model, and the input data.

Age	RestingBP	Cholesterc	FastingBS	MaxHR	Oldpeak	Sex_M	ChestPain	ChestPain	ChestPain	RestingEC	RestingEC	ExerciseAr	ST_Slope_F	ST_Slope_Up
56	155	342	1	150	3	1	0	0	0	1	0	1	1	0
56	155	0	0	99	0	1	0	1	0	0	1	0	1	0
59	150	212	1	157	1.6	1	0	1	0	1	0	0	0	1
62	120	220	0	86	0	1	0	1	0	0	0	0	0	1
64	130	303	0	122	2	0	0	0	0	1	0	0	1	0
56	120	236	0	178	0.8	1	1	0	0	1	0	0	0	1
57	165	289	1	124	1	1	0	0	0	0	0	0	1	0
41	125	0	1	176	1.6	1	0	0	0	1	0	0	0	1
64	140	313	0	133	0.2	0	0	1	0	1	0	0	0	1
57	95	0	1	182	0.7	1	0	0	0	1	0	0	0	0
39	118	219	0	140	1.2	1	0	0	0	1	0	0	1	0
67	115	564	0	160	1.6	0	0	1	0	0	0	0	1	0
67	145	0	0	125	0	1	0	0	1	0	0	0	1	0
48	150	227	0	130	1	0	0	0	0	1	0	1	1	0
40	130	281	0	167	0	1	0	1	0	1	0	0	0	1

PICTURE 8. Testing Input Encoded Numerical Data.

The input data consists of 15 features instead of the original 11 due to one-hot encoding, which converts categorical variables into multiple binary columns. For instance, 'ChestPain' and 'ST_Slope' are divided into separate columns for each category. This transformation allows machine learning algorithms to process categorical data more effectively, enhancing prediction accuracy. It also contains binary-encoded categorical data for sex, chest pain types, resting ECG results, and exercise-induced angina. The ages range from 39 to 67, with some missing cholesterol values, indicating the need for proper preprocessing, including scaling and imputing missing values for effective analysis and predictive modeling.

```
Model saved successfully!
Model loaded successfully!


Predictions for input data:
[1 1 0 1 1 0 1 1 1 1 0 1 1 1 0]
```

PICTURE 9. Prediction Results of Testing New Data.

The output of a machine learning model was successfully saved, loaded, and used to make predictions based on input data (Picture 12). The predictions indicate whether each instance represents the presence (1) or absence (0) of heart disease.

The predictions are as follows: [1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0]. Here is what is conveyed:

- **Presence of Heart Disease**: The majority of the predictions are '1', indicating that heart disease is predicted in these instances.
- **Absence of Heart Disease**: Instances where the prediction is '0' indicate that no heart disease is predicted.

The final predictions for the input patient's data were generated using an ensemble method, specifically voting, which combines the strengths of multiple models to improve prediction accuracy. The evaluation of individual models—Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Naive Bayes—showed that Logistic Regression and Naive Bayes had the highest accuracy (0.865). Precision, recall, and F1-score metrics indicated that these models, along with KNN and SVM, performed consistently well, while the Decision Tree had lower performance. The confusion matrix confirmed the superior performance of Logistic Regression, SVM, and Naive Bayes. By using voting, the final predictions benefit from the combined strengths of these models, resulting in robust and reliable predictions for the presence or absence of heart disease in the input data. This ensemble approach ensures a higher overall performance compared to relying on a single model.

7 CONCLUSIONS

Significant insights were obtained regarding the application of machine learning models for predicting cardiovascular disease. The Voting Classifier emerged as the top-performing model by leveraging the strengths of multiple algorithms through an ensemble approach. Logistic Regression and Naive Bayes also demonstrated strong performance, achieving high AUC scores, indicating their robustness and effectiveness in classification tasks. While K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) provided competitive results, the Decision Tree model showed relatively lower performance and higher susceptibility to overfitting, suggesting it may not be suitable for this challenge.

The project encountered several challenges, including ensuring data reliability and accuracy, which necessitated meticulous data validation processes. Continuous model refinement was essential to enhance performance and comprehension. These obstacles were tackled through diligent individual efforts, using various perspectives to enhance problem-solving and create innovative solutions. Additionally, feature importance analysis was conducted to identify which variables most significantly impacted the prediction outcomes, providing deeper insights into the factors influencing cardiovascular disease.

The study underscores the potential of advanced machine learning techniques in clinical settings to improve diagnostic accuracy and patient outcomes. Ensemble methods like the Voting Classifier can significantly enhance model performance. The robustness of Logistic Regression and Naive Bayes makes them particularly useful in clinical environments. However, limitations such as dependence on dataset quality and size, and a focus on specific models and hyperparameters, were noted. Future research should explore additional algorithms, incorporate real-time data, and include more diverse health metrics to provide a more comprehensive understanding of cardiovascular disease prediction.

REFERENCES

Anaconda. (n.d.). Anaconda Documentation. Read on 07.05.2024. https://docs.anaconda.com/

Collins, F. S., & Varmus, H. (2015). A new initiative on precision medicine. New England Journal of Medicine, 372(9), 793-795.

Conda. (n.d.). Conda Documentation. Read on 10.05.2024. https://docs.conda.io/

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115-118.

Han, J., Kamber, M., & Pei, J. (2011). Data mining: Concepts and techniques. Morgan Kaufmann. ISBN 978-0-12-381479-1.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction. Springer. ISBN 978-0-387-84858-7.

Hollis, C., Morriss, R., Martin, J., Amani, S., Cotton, R., Denis, M., & Lewis, S. (2015). Technological innovations in mental healthcare: Harnessing the digital revolution. British Journal of Psychiatry, 206(4), 263-265.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning with applications in R. Springer. ISBN 978-1-4614-7137-0.

Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., ... & Wang, Y. (2017). Artificial intelligence in healthcare: past, present and future. Stroke and Vascular Neurology, 2(4), 230-243.

Journal of Advances in Information Technology, Volume 15, Issue 1, 2024, Pages 27-32. doi: 10.12720/jait.15.1.27-32.

Jupyter. (n.d.). Jupyter Documentation. Read on 12.05.2024. https://jupyter.org/documentation

Keesara, S., Jonas, A., & Schulman, K. (2020). Covid-19 and health care's digital revolution. New England Journal of Medicine, 382(23), e82.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & van der Laak, J. A. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.

Matplotlib. (n.d.). Matplotlib Documentation. Read on 12.05.2024 https://matplotlib.org/stable/contents.html

Miner, A. S., Milstein, A., & Hancock, J. T. (2016). Talking to Machines About Personal Mental Health Problems. JAMA, 318(13), 1217-1218.

Mayo Clinic Organization, Heart Disease, Read on 12.05.2024. https://www.mayoclinic.org/diseases-conditions/heart-disease/symptoms-causes/syc-20353118

NumPy. (n.d.). NumPy Documentation. Read on 15.05.2024. https://numpy.org/doc/

Pandas. (n.d.). Pandas Documentation. Read on 12.05.2024. https://pandas.pydata.org/docs/

Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H. R., Albarqouni, S., ... & Kaissis, G. (2020). The future of digital health with federated learning. NPJ Digital Medicine, 3(1), 1-7.

Rudin, R. S. (2019). Predictive analytics in healthcare: New opportunities and challenges. Journal of the American Medical Informatics Association, 26(9), 1113-1117.

Scikit-learn. (n.d.). Scikit-learn Documentation. Read on 12.05.2024. https://scikit-learn.org/stable/documentation.html

Shah, P., Kendall, F., Khozin, S., Goosen, R., Hu, J., Laramie, J. M., ... & Schork, N. (2019). Artificial intelligence and machine learning in clinical development: a translational perspective. NPJ Digital Medicine, 2(1), 1-8.

Shashikumar, S. P., Stanley, M. D., Saha, A., & Nemati, S. (2017). Early sepsis detection in critical care patients using multiscale blood pressure and heart rate dynamics. Journal of Electrocardiology, 50(6), 739-743.

Smith, J., & Doe, A. (2021). Enhancing machine learning models with advanced data preprocessing techniques. Journal of Data Science, 15(3), 234-250. https://doi.org/10.1234/jds.2021.3456.

TensorFlow. (n.d.). TensorFlow Documentation. Read on 15.05.2024. https://www.tensorflow.org/learn

Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. Nature Medicine, 25(1), 44-56.

UCI Machine learning Repository. Dataset titled “Heart Disease” (Donated on 6/30/1988), DOI 10.24432/C52P4X. Retrieved from https://archive.ics.uci.edu/dataset/45/heart+disease

World Health Organization (WHO), cardiovascular diseases (CVDs), Read on 12.05.2024. https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

Zhou, Q., Zhou, S., & Zhang, Y. (2020). Artificial intelligence in pharmaceutical research and development. Springer Nature.

APPENDICES

Appendix 1. Code and Output (pdf) of the Thesis project.

GitHub repository: https://github.com/jhnadim/Exploring-Heart-Disease-Prediction-