HEART DISEASE PREDICTION MODEL

Report submitted to

Techno India University, West Bengal
for the partial fulfillment
of

Bachelor of Technology (B. Tech.)

degree in

Computer Science & Engineering

By

Rishi Bakshi
201001001143

Koushik Pal
201001001091

Shreya Das
201001001093

Tushal Ghosh
201001001152

Shreyasi Patra
201001001157

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
TECHNO INDIA UNIVERSITY, WEST BENGAL,
SALT LAKE, KOLKATA – 700091, INDIA

JULY 2024

CERTIFICATE

This is to certify that the Dissertation Report entitled, " **HEART DISEASE PREDICTION**" submitted by **Rishi Bakshi, Koushik Pal, Shreya Das, Tushal Ghosh and Shreyasi Patra** to Techno India University, Kolkata, India, is a record of bonafide Project work carried out by them under my supervision and guidance and is worthy of consideration for the award of the degree of Bachelor of Technology (B. Tech) in Computer science & Engineering.

Approved By:

____

Supervisor(s)
Date:

____

HOD, CSE, Techno India University

Date:

ACKNOWLEDGEMENT

We would first like to thank our thesis supervisor Dr. Kalyan Kumar Das, Associate Professor (Teacher in Charge), CSE Dept, TIU. He helped us whenever we ran into a trouble spot or had a question about our research or writing.

We take this opportunity to express gratitude to all of the Department faculty members for their help and support.

We also thank our parents for the unceasing encouragement, support and attention and also sense of gratitude to one and all, who directly or indirectly, have bestowed their hand in this thesis.

Rishi Bakshi

Koushik Pal

Shreya Das

Tushal Ghosh

Shreyasi Patra

Contents

Abstract

Heart disease diagnosis poses a formidable challenge in the medical field, necessitating the meticulous analysis of extensive clinical and pathological data. Recognizing the pivotal role of early detection in mitigating the impact of this leading cause of global mortality, the collaboration between researchers and clinical professionals has intensified. In response to the complexity of this diagnostic task, there has been a burgeoning interest in leveraging machine learning as a reliable and supportive tool in the medical domain.

Machine learning, a field that harnesses the power of algorithms to enable systems to learn and improve from experience, has shown remarkable promise in predicting diseases with precision. Among the plethora of algorithms available, Random Forests stand out as a particularly potent tool in the realm of supervised machine learning. The Random Forests algorithm excels in its ability to handle large datasets and diverse features, making it an ideal candidate for the intricate task of heart disease prediction.

The primary focus of this research lies in exploring diverse prediction models for heart disease and selecting vital features indicative of cardiovascular health using the Random Forests algorithm. This algorithm, which operates by constructing a multitude of decision trees during training and outputs the mode of the classes for classification tasks, has demonstrated high accuracy compared to other supervised machine learning algorithms, such as logistic regression. One of the critical advantages of Random Forests is its ability to handle both numerical and categorical data, making it well-suited for the multifaceted nature of clinical datasets. By leveraging its ensemble learning approach, which aggregates the predictions of multiple decision trees, Random Forests can overcome overfitting and enhance generalization performance. This becomes particularly crucial in the context of heart disease prediction, where the correct diagnosis in the early stages is paramount.

The significance of swift and accurate diagnosis in heart disease cannot be overstated, as time emerges as a critical factor in determining patient outcomes. Hence, the utilization of Random Forests as a predictive tool holds immense promise in enhancing the efficiency and accuracy of heart disease prediction. By identifying crucial features indicative of heart disease, this research aims to contribute to the proactive management of this pervasive health concern, ultimately striving to reduce the global burden of cardiovascular-related mortality.

Chapter 1

Introduction

Cardiovascular disease (CVD) is a type of heart disease that continues to be a major cause of death worldwide, accounting for over 30% of all deaths. If nothing is done, the total number of fatalities in the world is anticipated to rise to 22 million by 2030. Plaques on arterial walls can obstruct blood flow, resulting in a heart attack or stroke. Heart disease is caused due to various risk factors such as physical inactivity, unhealthy diet, and the effective use of alcohol and tobacco. The abovementioned factors are reduced by adoptinga good daily lifestyle, namely, reducing salt in the diet, consumption of vegetables and fruits, practicing physical activity regularly, and discontinuing alcohol and tobacco use, which helps to minimize the risk of heart disease. The solution to overcome these problemsis to use the collection of patient records from different health care centers and hospitals. For getting the results and seeking another opinion from an experienced doctor the decisionsupport system is used. The unnecessary test conductions are avoided by this technique fordiagnosis, thereby saving money and time.

More number of neurohormonal regulatory mechanisms are triggered in the initial stages of heart failure disease (HFD). In a short duration, these compensatory mechanisms can cause the HFD consequences, leading to accentuated ventricular dysfunction, dyspnea on exertion, peripheral edema, pulmonary, and heart remodeling which can cause afterload and preload permanent changes. More options of treatment with HFD are given to the patient including lifestyle changes and implantable or medication devices such as adefibrillator or pacemaker. The main concern is ensuring the follow-up in this population given that hospitalization due to acute HFD decompensation is the leading cause of healthcare expenditure.

1.1 Objective

The primary goal of this research is to design and implement a heart disease prediction system leveraging data mining techniques on a historical heart dataset. This system aims to uncover latent patterns within the data, offering a novel approach to understanding the complexities associated with heart diseases. By utilizing data mining, the research seeks to minimize human biases in disease prediction, fostering a more objective and accurate analysis of medical data.

The key features and objectives of the heart disease prediction system include:

- **Concealed Pattern Discovery**: The system employs advanced data mining techniques to uncover hidden patterns within the historical heart dataset. This approach allows for a more comprehensive understanding of factors contributing to heart diseases, potentially revealing subtle relationships that might not be apparent through traditional analysis.

- **Mitigation of Human Bias:** By relying on data-driven methodologies, the system aims to reduce reliance on subjective human judgments, thereby minimizing biases in the prediction process. This enhances the objectivity and reliability of the heart disease predictions.

- **Implementation of Logistic Regression**: Logistic Regression is chosen as the classification algorithm to predict heart diseases based on user input. This statistical method is well-suited for binary classification tasks and can provide probabilities for the likelihood of disease presence. Its simplicity and interpretability make it a practical choice for medical applications.

In summary, this research endeavors to contribute to the field of healthcare by developing an advanced heart disease prediction system. By employing data mining techniques, the system aims to enhance our understanding of heart diseases, reduce human biases in predictions, implement logistic regression for disease classification, and ultimately contribute to the cost-effectiveness of medical testing in the context of cardiovascular health. This multifaceted approach underscores the potential impact of leveraging advanced technologies for improved disease prediction and patient care.

1.2 Problem Statement

The detection of heart disease presents a significant challenge in the medical field, primarily due to the limitations of existing instruments. While some instruments have the capability to predict heart disease, they often come with a hefty price tag or lack the efficiency required to accurately calculate the likelihood of heart disease in individuals. This creates a gap in accessible and effective tools for early detection, a crucial factor that can significantly reduce mortality rates and overall complications associated with cardiac diseases.

Monitoring patients on a daily basis is not always feasible, and the continuous presence of a doctor for 24-hour observation is a luxury rarely available due to constraints in expertise, time, and resources. This gap in constant monitoring and expert consultation emphasizes the need for innovative solutions that can provide timely and accurate insights into the risk of heart disease.

The wealth of data available in today's world offers a promising avenue for addressing this challenge. Machine learning algorithms, designed to analyze vast datasets and uncover hidden patterns, emerge as a powerful tool in the realm of health diagnosis. By leveraging these algorithms, it becomes possible to extract valuable insights from medical data that may not be apparent through traditional analytical methods.

Machine learning algorithms can sift through extensive datasets, identifying subtle patterns and correlations that might elude human observation. These hidden patterns can then be utilized to enhance health diagnosis, particularly in the context of cardiac diseases. The ability of machine learning to process large volumes of data rapidly and accurately positions it as a valuable asset in the quest for early detection and proactive management of heart-related issues.

In essence, the integration of machine learning into the analysis of medical data holds the promise of revolutionizing the detection of heart disease. By harnessing the power of algorithms to discern hidden patterns, healthcare professionals can potentially identify risk factors and initiate preventative measures, ultimately contributing to a paradigm shift in the approach to cardiac health and reducing the burden of heart-related complications on individuals and healthcare systems.

1.3 Methodology

In this segment, the research methodology and analysis procedures undertaken in this study are elucidated. The outset of this research involves the meticulous collection of data and the judicious selection of pertinent attributes. This foundational step establishes the groundwork for subsequent analysis and model development. Following data collection and attribute selection, the gathered data undergoes a crucial pre-processing phase to ensure its compatibility with the analysis requirements.

To facilitate the development and evaluation of predictive models, the dataset is systematically partitioned into two distinct categories: training and testing datasets. This segregation is pivotal for training the algorithms on a subset of data and subsequently assessing their performance on an independent set, thereby gauging their generalization capabilities.

The heart of the methodology lies in the application of various algorithms, serving as the engines that power the predictive model. The selected algorithms are then fed the training dataset to enable them to learn the underlying patterns and relationships within the data. Once the models are trained, they are put to the test using the designated testing dataset, and the accuracy of their predictions is meticulously evaluated.

To orchestrate these intricate processes, several modules are employed, each serving a specific function. The initial modules encompass data collection and attribute selection, forming the basis of the study. Subsequent modules handle crucial tasks such as data pre-processing, where the raw data is transformed into the requisite format for analysis. Additionally, data balancing techniques may be applied to address any imbalances in the dataset, ensuring a more equitable representation of different classes.

The capstone of the methodology involves the actual prediction of disease based on the trained models. Through this comprehensive approach, the research endeavors to harness the capabilities of machine learning to not only discern patterns in the data but also to accurately predict the presence or absence of the target disease. By modularizing the procedures and employing a systematic approach, this research aims to contribute to the advancement of predictive models for disease diagnosis, particularly in the context of heart disease.

1.3.1 Data Collection

This study initiates with the acquisition of the dataset from the UCI repository, a widely recognized and utilized source in research analyses as acknowledged by numerous authors. The selection of this dataset sets the stage for investigating and predicting heart disease, aligning with the established practices of the research community.

The initial step revolves around organizing and preparing the dataset for subsequent analysis. The UCI repository provides a diverse range of datasets, and the careful curation of the specific dataset pertinent to heart disease serves as the foundation for this research. This dataset is then systematically divided into two distinct sections: training and testing datasets, a pivotal partitioning strategy employed for the development and evaluation of predictive models.

In the context of this article, a deliberate allocation has been made, with 80% of the dataset earmarked for training purposes and the remaining 20% designated for testing. This division ensures that the predictive models are trained on a substantial portion of the data, allowing them to capture underlying patterns and relationships. The testing dataset, on the other hand, serves as an independent benchmark to assess the generalization and predictive accuracy of the trained models.

This partitioning approach aligns with common practices in machine learning, striking a balance between leveraging a substantial amount of data for model training and maintaining a sufficiently distinct subset for rigorous model evaluation. The 80-20 split strikes a pragmatic compromise, offering a robust framework for the study's objectives.

By anchoring the research in a well-established dataset and adopting a systematic approach to dataset organization and partitioning, this study endeavors to contribute valuable insights into the predictive modeling of heart disease. The utilization of a widely recognized repository and a thoughtful dataset split underscores the rigor and reliability of the research methodology, paving the way for meaningful analyses and potentially impactful findings in the realm of cardiovascular health prediction.

1.3.2 Dataset and Attributes

Attributes within a dataset are intrinsic properties that play a crucial role in the analysis and prediction of the targeted concern. In the context of predicting diseases, a multitude of patient-related attributes are typically considered to capture the diverse factors influencing health outcomes. Examples of these attributes include gender, chest pain, serum cholesterol levels, fasting blood pressure, and the presence of exercise-induced angina (exang).

The selection of relevant attributes is a critical step in constructing an effective predictive model. Not all attributes contribute equally to the predictive accuracy of the model, and some may even introduce noise or redundancy. Hence, it becomes imperative to identify and focus on the most influential attributes that significantly impact the outcome of interest.

One approach to attribute selection involves the use of a correlation matrix. The correlation matrix quantifies the strength and direction of relationships between pairs of attributes. By examining the correlation coefficients, researchers can discern which attributes exhibit significant associations with each other. This information is invaluable for selecting attributes that contribute unique and non-redundant information to the predictive model.

Attributes with high correlations may indicate collinearity, where one attribute can be predicted from another, potentially leading to instability in the model or difficulty in interpreting the results. On the other hand, attributes with low correlations may be candidates for inclusion in the model to provide diverse and independent information.

By leveraging the correlation matrix for attribute selection, researchers can refine their dataset to include only the most relevant attributes, streamlining the model-building process. This not only enhances the model's predictive accuracy but also contributes to a more interpretable and parsimonious model.

In essence, the careful consideration and selection of attributes, guided by insights from the correlation matrix, constitute a crucial aspect of constructing a robust predictive model for diseaseoutcomes. This systematic approach ensures that the model captures the salient features of the dataset, facilitating more accurate and insightful predictions in the realm of health analysis.

Chapter 2

Literature Survey

2.1 Traditional Methods of Heart Disease Diagnosis

Traditional diagnostic approaches for heart diseases predominantly rely on non-invasive procedures aimed at comprehensively assessing the cardiovascular system. The primary methods include electrocardiograms (ECGs), echocardiograms, and stress tests, each offering distinctive insights into cardiac health.

Electrocardiograms (ECGs): ECGs stand as a cornerstone in heart disease diagnosis by meticulously capturing the heart's electrical activity. Through the placement of electrodes on the skin, this technique records the electrical impulses generated during each heartbeat. Deviations from the normal pattern can signal potential cardiac abnormalities. While ECGs are valuable in detecting irregularities, they may have limitations in providing a detailed understanding of the heart's structural aspects.

Echocardiograms: Leveraging sound waves, echocardiograms create detailed images of the heart's structure and functionality. This non-invasive imaging technique allows healthcare professionals to visualize the chambers, valves, and blood flow within the heart.
Echocardiography is particularly useful for diagnosing conditions such as valve disorders and assessing overall cardiac function. However, it may not always offer exhaustive insights into certain complex cardiac conditions.

Stress Tests: Stress tests evaluate the heart's performance under physical exertion, typically through activities like treadmill walking or pharmacological stress. These tests aim to uncover abnormalities that may only manifest during periods of increased cardiac demand. While stress tests are valuable for assessing cardiovascular fitness and detecting exercise-induced issues, they may not always provide a comprehensive picture of the heart's condition at rest.

Despite their utility, traditional diagnostic methods have limitations, especially in terms of early detection and nuanced characterization of specific cardiac conditions. These techniques often excel in identifying general irregularities but may fall short in delivering a granular understanding of complex cardiac issues. The evolving landscape of medical diagnostics has spurred the exploration of advanced technologies, including machine learning, to enhance the accuracy and timeliness of heart disease detection.

2.2 Machine Learning Applications in Healthcare

The healthcare landscape has undergone a transformative revolution with the integration of machine learning applications, marking a paradigm shift in the approach to patient care. These applications leverage advanced algorithms to sift through extensive medical data, extracting meaningful insights that contribute to an elevated standard of healthcare. Machine learning, characterized by algorithms that learn from data, plays a pivotal role in various healthcare domains, including disease diagnosis, outcome prediction, treatment planning, and the analysis of medical imagery.

- **Disease Diagnosis**: Machine learning algorithms excel in analyzing large datasets, discerning intricate patterns, and identifying subtle correlations. In the context of healthcare, these capabilities prove invaluable for disease diagnosis. By learning from historical patient data, machine learning models can recognize complex relationships between diverse factors and make accurate predictions regarding the presence or likelihood of diseases, including heart diseases.
- **Outcome Prediction:** Machine learning extends its reach into predicting patient outcomes based on a multitude of factors. By assimilating and analyzing diverse patient data, these algorithms can forecast potential health trajectories, enabling healthcare professionals to tailor interventions and treatment plans for optimal patient outcomes.
- **Treatment Plan Recommendations:** The adaptive nature of machine learning allows it to adapt and improve its recommendations based on new data. This adaptability is harnessed in healthcare to provide personalized treatment plan recommendations. By considering individual patient characteristics, medical histories, and response patterns, machine learning systems can assist healthcare providers in tailoring effective and personalized treatment strategies.
- **Analysis of Medical Imagery:** Machine learning algorithms showcase remarkable proficiency in the analysis of medical imagery, such as X-rays, MRIs, and CT scans. These applications can aid in the early detection of abnormalities, support radiologists in image interpretation, and enhance the overall accuracy of diagnostic processes.

In the specific context of heart diseases, the predictive capabilities of machine learning offer immense promise. The ability to analyze diverse patient attributes, genetic information, and lifestyle factors allows for a more comprehensive understanding of cardiovascular health. This enables earlier detection of risk factors and potential diseases, facilitating timely interventions and improving patient outcomes. As machine learning continues to evolve, its applications in healthcare are poised to further refine diagnostics, treatment strategies, and overall patient care.

2.3 Previous Research on Heart Disease Prediction

In preceding research endeavors, the application of machine learning models to prognosticate heart diseases has been a focal point. Researchers have delved into the scrutiny of a diverse set of clinical parameters and patient data, employing various algorithms to construct predictive models. Notable among these algorithms are Support Vector Machines, Random Forests, and Logistic Regression. These studies have undertaken a meticulous assessment of their models, considering accuracy, sensitivity, specificity, and other metrics. This rigorous evaluation process aims to facilitate comparisons, enabling researchers to discern the most effective approach for heart disease prediction.

- **Support Vector Machines (SVM):** This supervised learning algorithm has been a cornerstone in prior research, particularly in the realm of heart disease prediction. SVM aims to establish an optimal hyperplane that effectively categorizes data points, making it well-suited for discerning patterns within clinical parameters and patient data indicative of potential cardiac issues.
- **Random Forests:** The ensemble learning approach of Random Forests has garnered attention in previous studies on heart disease prediction. By constructing multiple decision trees during training and aggregating their outputs, Random Forests enhance predictive accuracy and mitigate overfitting concerns. Researchers have explored the algorithm's effectiveness in handling diverse datasets and identifying crucial features associated with heart diseases.
- **Logistic Regression**: Logistic Regression, a classic statistical method, has also found its place in previous research endeavors. Despite its simplicity, Logistic Regression is powerful in assessing the relationship between independent variables and binary

outcomes. Researchers have utilized this algorithm to model the probability of heart disease based on various clinical parameters.

The evaluation of these models goes beyond mere accuracy, with researchers delving into metrics such as sensitivity and specificity. These metrics provide insights into the models' abilities to correctly identify positive cases (sensitivity) and negative cases (specificity), contributing to a more nuanced understanding of their performance.

Through this comprehensive examination and comparison of machine learning models, researchers aim to refine the predictive methodologies for heart disease. The synthesis of findings from various studies contributes to the ongoing pursuit of developing robust, accurate, and applicable models that can effectively aid in the early detection and management of heart diseases.

2.4 Datasets Used in Heart Disease Prediction

Datasets used in heart disease prediction studies are sourced from hospitals, research institutions, and public repositories. These datasets include diverse features such as age, gender, cholesterol levels, blood pressure, and comprehensive medical histories. The information captured in these datasets is crucial for training and evaluating machine learning models. Features encompass demographic details like age and gender, biometric indicators such as cholesterol levels and blood pressure, and relevant medical history, including previous heart conditions, diabetes, smoking habits, and family history. Diagnostic tests like electrocardiograms (ECGs) and echocardiograms, along with lifestyle factors like physical activity and dietary patterns, further contribute to the richness of these datasets. However, challenges such as dataset heterogeneity, varied sizes, data quality issues, and representativeness must be addressed to ensure the reliability and applicability of predictive models in the realm of heart disease prediction. Ongoing efforts to standardize datasets and adhere to ethical guidelines contribute to the advancement of research in this critical area of healthcare.

2.5 Performance Metrics and Evaluation Methods

Heart disease prediction studies rely on critical performance metrics for model evaluation, including accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic (ROC) curve (AUC-ROC). Accuracy gauges overall correctness, precision evaluates the ratio of true positives among predicted positives, recall assesses the model's capability to capture all actual positives, and the F1-score harmonizes precision and recall. The AUC-ROC quantifies the model's discriminatory ability between positive and negative instances. Rigorous validation methods, notably k-fold cross-validation and holdout validation, are essential for assessing the reliability and generalizability of predictive models.

K-fold cross-validation partitions the dataset into k subsets, iteratively training the model on k-1 subsets and validating on the remaining subset. Holdout validation involves splitting the dataset into training and testing sets, training the model on one subset and validating on the other. These validation techniques provide insights into a model's performance across diverse data subsets, ensuring robustness. A comprehensive understanding of these evaluation methods is imperative for researchers and practitioners, ensuring the efficacy and dependability of developed heart disease prediction models in real-world applications.

2.6 Ethical Considerations and Challenges

In the domain of heart disease prediction using machine learning, ethical considerations are pivotal. Protecting patient privacy, ensuring data security, and addressing potential biases in predictive models are paramount concerns. Safeguarding the confidentiality of sensitive medical data and obtaining informed consent are crucial ethical imperatives. Biases can arise from skewed datasets, resulting in disparities in predictions or treatments and potentially exacerbating existing healthcare inequalities. Effectively addressing these ethical challenges is essential for the development of fair, accurate, and ethical predictive models in cardiovascular health. By prioritizing patient privacy, promoting data security measures, and mitigating biases, researchers and practitioners contribute to the responsible and equitable application of machine learning in heart disease prediction, fostering trust and ensuring the well-being of individuals in healthcare contexts.

2.7 Recent Advances and Future Trends

Recent progress in machine learning, driven by advancements in computational capabilities, has significantly enhanced heart disease prediction. Breakthroughs in deep learning, ensemble methods, and the integration of diverse data sources have propelled the development of more sophisticated and accurate predictive models. Deep learning techniques, including neural networks, excel at discerning intricate patterns in complex datasets. Ensemble methods, such as Random Forests, aggregate the strengths of multiple models for enhanced performance.

Future trends in this domain are marked by the incorporation of emerging technologies, notably explainable AI and federated learning. Explainable AI enhances model interpretability, crucial for gaining insights into decision-making processes. Federated learning facilitates collaborative model training across decentralized devices, ensuring privacy while leveraging collective intelligence.

Moreover, the inclusion of multi-modal data sources, such as genetic information and wearable devices, is shaping the future of heart disease prediction. Integrating genetic data offers a deeper understanding of individual risk factors, while wearable devices provide real-time health metrics. These trends collectively pave the way for crafting more precise, interpretable, and personalized predictive models, promising advancements that hold great potential for improving cardiovascular health outcomes.

2.8 Existing Critique and Gaps

The literature on heart disease prediction through machine learning presents a diverse array of algorithms and methodologies, underscoring the potential for advanced computational techniques to contribute significantly to healthcare. However, several notable gaps persist in this evolving field.

One of the primary challenges is the limited availability of standardized datasets. The absence of widely accepted, standardized datasets hamper the comparability and reproducibility of studies. Datasets used across different research efforts often exhibit variability in terms of size, composition, and data quality. This lack of standardization impedes the establishment of universal benchmarks and makes it challenging to draw robust conclusions from collective research findings.

Moreover, potential biases inherent in the datasets pose a significant concern. Biases can emerge from various sources, including underrepresented demographic groups or skewed distributions of certain health indicators. Addressing these biases is crucial for ensuring that predictive models generalize well across diverse populations, mitigating disparities in healthcare outcomes.

Another critical gap lies in the interpretability of machine learning models. While complex algorithms often yield accurate predictions, their inherent complexity can be a barrier to understanding and trust, particularly in clinical settings. Achieving interpretability is essential for the successful integration of machine learning predictions into real-world healthcare decision-making processes.

Additionally, there is a need for a more exhaustive comparative analysis of the performance of various machine learning algorithms. A comprehensive understanding of the strengths and limitations of different techniques is vital for guiding researchers and practitioners in selecting the most effective approaches for heart disease prediction.

2.9 Conclusion for the Current Study

The literature review underscores the transformative impact of machine learning in predicting heart diseases, emphasizing the critical need for accurate and reliable predictive models in the realm of healthcare. These models serve as indispensable tools for early disease detection, precise patient risk stratification, and the customization of personalized interventions. Such advancements hold the promise of alleviating the burden of cardiovascular ailments and improving patient outcomes.

This study aims to contribute meaningfully to the existing body of knowledge by constructing a robust predictive model for heart diseases. Building upon insights gained from the literature, the research endeavors to address observed gaps and challenges identified in previous studies. Specifically, the focus is on enhancing the accuracy and generalizability of predictive analytics in the domain of cardiovascular health. The goal is to develop a model that not only demonstrates high performance but is also capable of effectively adapting to diverse datasets and populations.

By addressing these objectives, the study aspires to provide valuable contributions to the field, fostering advancements in heart disease prediction methodologies. The ultimate aim is to translate these advancements into tangible benefits for individuals by enabling early interventions, optimizing treatment strategies, and ultimately reducing the impact of cardiovascular diseases on public health. Through rigorous research and methodological refinement, this study seeks to play a role in shaping the future landscape of predictive modeling for cardiovascular health, contributing to the ongoing efforts to combat and prevent heart diseases.

Chapter 3

Project Description

Heart diseases are a significant health concern globally, necessitating reliable predictive models to aid early diagnosis and treatment. The motivation behind this project lies in addressing the pressing need for accurate heart disease prediction models leveraging machine learning algorithms. By harnessing the power of data-driven approaches, this initiative aims to develop a robust predictive tool capable of detecting potential heart ailments in individuals before symptomatic manifestations.

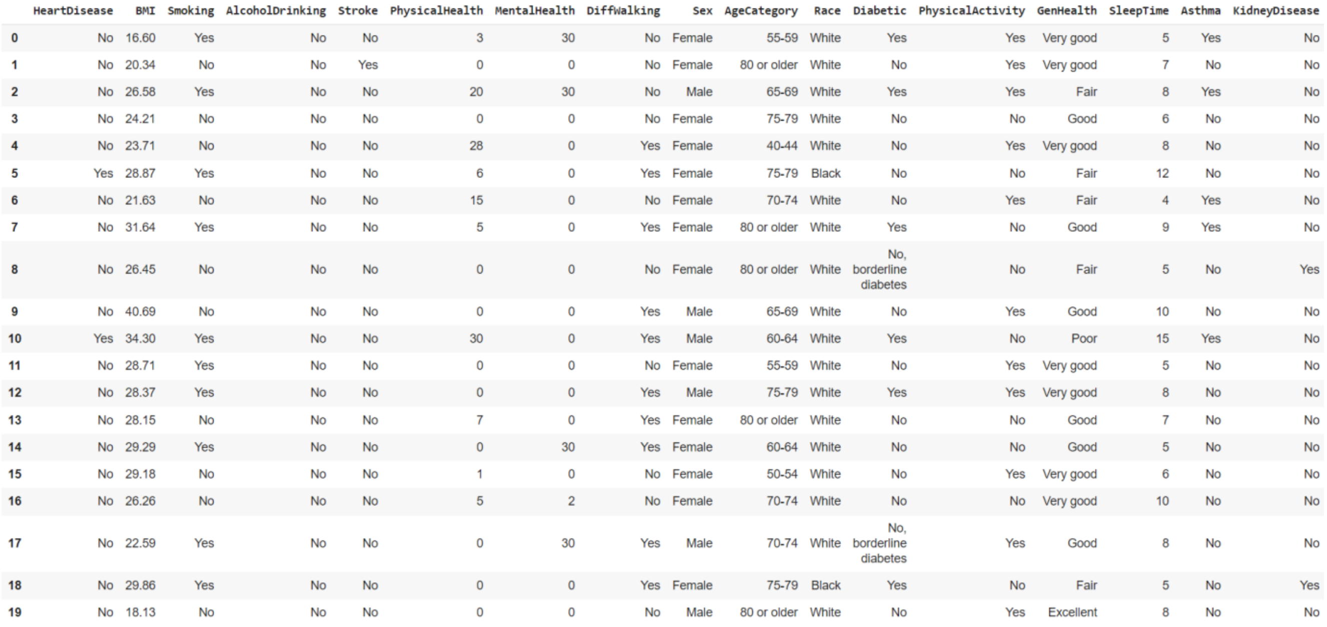
The primary objective is to evaluate and compare the performance of various machine learning algorithms in predicting heart diseases. The scope encompasses employing Logistic Regression, Decision Trees, Naive Bayes, K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machines (SVM) to create predictive models. The project aims to analyze the strengths, weaknesses, and predictive capabilities of these models using relevant performance metrics.

3.1 Data Collection and Analysis

Comprehensive health records and datasets are collected which encompassing attributes like :

- BMI
- Smoking
- Alcohol Drinking
- Stroke
- Physical Health
- Mental Health
- Sex
- Age Category
- Physical Activity

- Generation Health
- Sleep Time
- Asthma
- Kidney Disease
- Skin Cancer
- Race diabetic



	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivity	GenHealth	SleepTime	Asthma	KidneyDisease
0	No	16.60	Yes	No	No	3	30	No	Female	55-59	White	Yes	Yes	Very good	5	Yes	No
1	No	20.34	No	No	Yes	0	0	No	Female	80 or older	White	No	Yes	Very good	7	No	No
2	No	26.58	Yes	No	No	20	30	No	Male	65-69	White	Yes	Yes	Fair	8	Yes	No
3	No	24.21	No	No	No	0	0	No	Female	75-79	White	No	No	Good	6	No	No
4	No	23.71	No	No	No	28	0	Yes	Female	40-44	White	No	Yes	Very good	8	No	No
5	Yes	28.87	Yes	No	No	6	0	Yes	Female	75-79	Black	No	No	Fair	12	No	No
6	No	21.63	No	No	No	15	0	No	Female	70-74	White	No	Yes	Fair	4	Yes	No
7	No	31.64	Yes	No	No	5	0	Yes	Female	80 or older	White	Yes	No	Good	9	Yes	No
8	No	26.45	No	No	No	0	0	No	Female	80 or older	White	No, borderline diabetes	No	Fair	5	No	Yes
9	No	40.69	No	No	No	0	0	Yes	Male	65-69	White	No	Yes	Good	10	No	No
10	Yes	34.30	Yes	No	No	30	0	Yes	Male	60-64	White	Yes	No	Poor	15	Yes	No
11	No	28.71	Yes	No	No	0	0	No	Female	55-59	White	No	Yes	Very good	5	No	No
12	No	28.37	Yes	No	No	0	0	Yes	Male	75-79	White	Yes	Yes	Very good	8	No	No
13	No	28.15	No	No	No	7	0	Yes	Female	80 or older	White	No	No	Good	7	No	No
14	No	29.29	Yes	No	No	0	30	Yes	Female	60-64	White	No	No	Good	5	No	No
15	No	29.18	No	No	No	1	0	No	Female	50-54	White	No	Yes	Very good	6	No	No
16	No	26.26	No	No	No	5	2	No	Female	70-74	White	No	No	Very good	10	No	No
17	No	22.59	Yes	No	No	0	30	Yes	Male	70-74	White	No, borderline diabetes	Yes	Good	8	No	No
18	No	29.86	Yes	No	No	0	0	Yes	Female	75-79	Black	Yes	No	Fair	5	No	Yes
19	No	18.13	No	No	No	0	0	No	Male	80 or older	White	No	Yes	Excellent	8	No	No

Data preprocessing involves cleaning, handling missing values, outlier detection, and scalingmethods to ensure uniformity and relevance across dataset.

Descriptive statistical analyses and visual representations provide insights into the dataset's distribution, correlations, and potential predictive attributes. Exploratory Data Analysis (EDA) helps identify patterns, relationships, and feature importance for subsequent model training.

Data preprocessing: -

The data preprocessing step is crucial for ensuring that the dataset used for training and testing the heart disease prediction model is clean, consistent, and ready for analysis. The following steps outline the preprocessing procedures applied to the dataset:

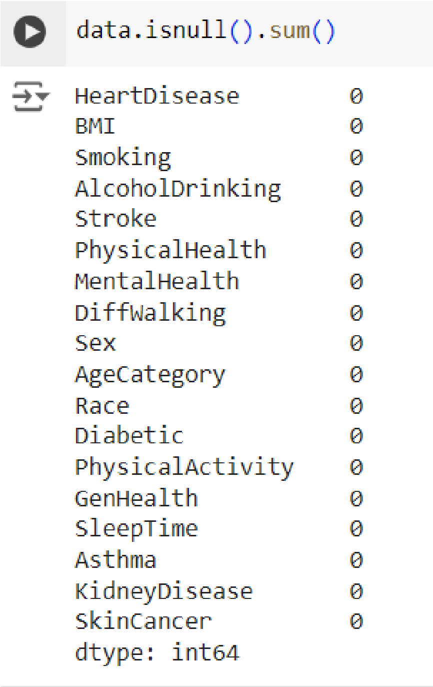
1. Data Collection

The dataset used for this project was obtained from the Kaggle. It contains various features related to patient health metrics, such as age, gender, cholesterol levels, and more.

2. Data Cleaning

- Missing Values:

Identified and handled missing values using imputation techniques. For example, missing numerical values were replaced with the mean/median values of the respective columns. Any rows or columns with excessive missing values that could not be reasonably imputed were removed from the dataset.



```
data.isnull().sum()
```

```
HeartDisease        0
BMI                 0
Smoking             0
AlcoholDrinking     0
Stroke              0
PhysicalHealth      0
MentalHealth        0
DiffWalking         0
Sex                 0
AgeCategory         0
Race                0
Diabetic            0
PhysicalActivity    0
GenHealth           0
SleepTime           0
Asthma              0
KidneyDisease       0
SkinCancer          0
dtype: int64
```

- Outliers:

Detected outliers using statistical methods (e.g., Z-score, IQR) and visualizations (e.g., box plots).

Treated outliers by either removing them or capping/flooring them to reasonable limits.

Data Transformation

To prepare the dataset for machine learning model training, certain categorical values were replaced with numerical equivalents. The following transformations were applied:

- Binary Values:

  'Yes' was replaced with 1, and 'No' was replaced with 0.

- Gender Encoding:

'Male' was replaced with 1, and 'Female' was replaced with 0.

- Medical History Encoding:

'No, borderline diabetes' was replaced with 0.

'Yes (during pregnancy)' was replaced with 1.

- Health Rating Encoding:

'Excellent' was replaced with 3.

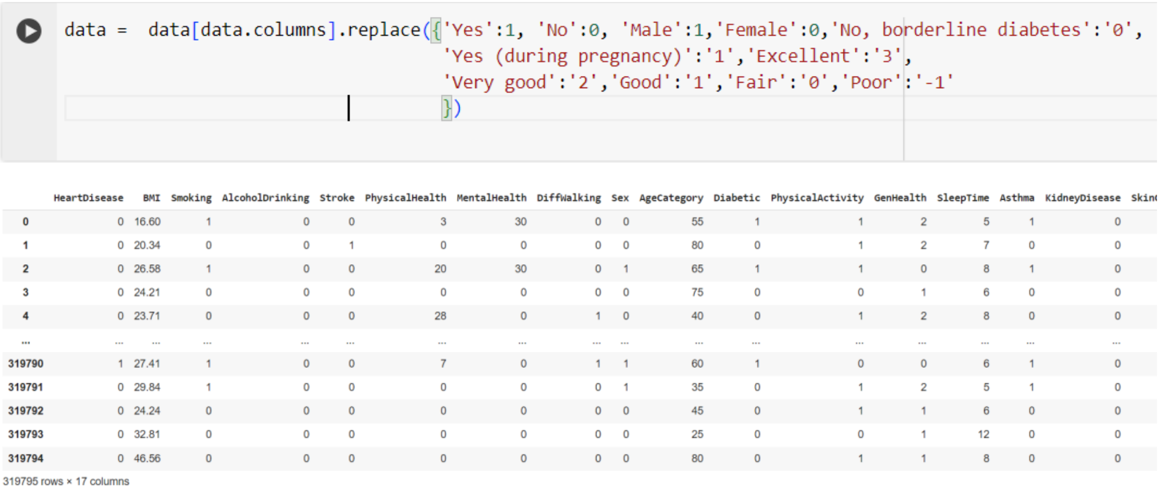
'Very good' was replaced with 2.

'Good' was replaced with 1.

'Fair' was replaced with 0.

'Poor' was replaced with -1.

These transformations were performed using the replace method in pandas, ensuring that the dataset's categorical values were converted to numerical equivalents for further analysis and model training.



```
data =  data[data.columns].replace({'Yes':1, 'No':0, 'Male':1,'Female':0,'No, borderline diabetes':'0',
                                    'Yes (during pregnancy)':'1','Excellent':'3',
                                    'Very good':'2','Good':'1','Fair':'0','Poor':'-1'
                                    })
```

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Diabetic	PhysicalActivity	GenHealth	SleepTime	Asthma	KidneyDisease	Skin
0	0	16.60	1	0	0	3	30	0	0	55	1	1	2	5	1	0	
1	0	20.34	0	0	1	0	0	0	0	80	0	1	2	7	0	0	
2	0	26.58	1	0	0	20	30	0	1	65	1	1	0	8	1	0	
3	0	24.21	0	0	0	0	0	0	0	75	0	0	1	6	0	0	
4	0	23.71	0	0	0	28	0	1	0	40	0	1	2	8	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
319790	1	27.41	1	0	0	7	0	1	1	60	1	0	0	6	1	0	
319791	0	29.84	1	0	0	0	0	0	1	35	0	1	2	5	1	0	
319792	0	24.24	0	0	0	0	0	0	0	45	0	1	1	6	0	0	
319793	0	32.81	0	0	0	0	0	0	0	25	0	0	1	12	0	0	
319794	0	46.56	0	0	0	0	0	0	0	80	0	1	1	8	0	0	

319795 rows × 17 columns

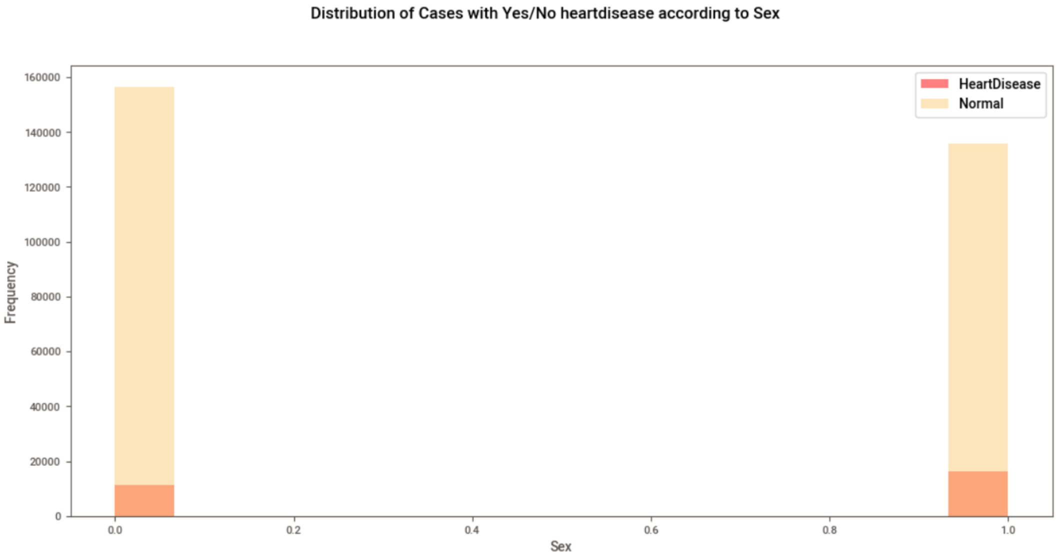
Visualization:

Distribution of Cases with Yes/No Heart Disease According to Sex: -

A histogram was created to visualize the distribution of cases with and without heart disease based on the 'Sex' variable. The histogram compares the frequency of 'Sex' values for individuals with heart disease (HeartDisease=1) and without heart disease (HeartDisease=0).

X-axis: Sex (0: Female, 1: Male)

Y-axis: Frequency of cases



The histogram shows two distributions:

Cases with heart disease (HeartDisease=1) are represented in red.

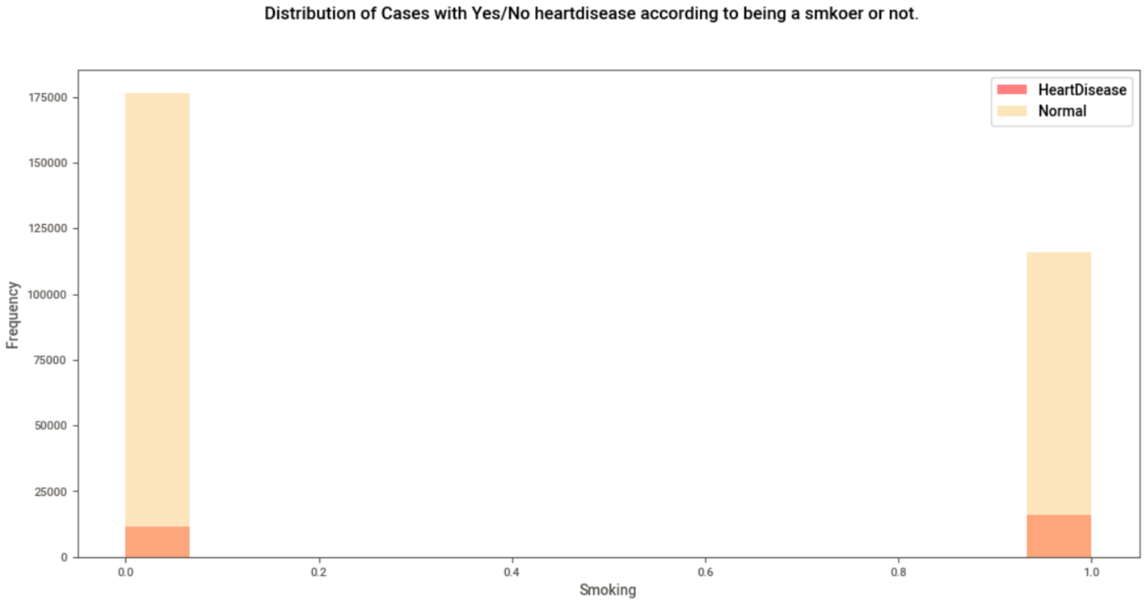
Cases without heart disease (HeartDisease=0) are represented in a shade of orange.

This visualization helps to understand any potential relationship between gender (Sex) and the presence of heart disease in the dataset.

Distribution of Cases with Yes/No Heart Disease According to Smoking Status

A histogram was created to visualize the distribution of cases with and without heart disease based on the 'Smoking' variable. The histogram compares the frequency of 'Smoking' values for individuals with heart disease (HeartDisease=1) and without heart disease (HeartDisease=0).

- **X-axis:** Smoking status (0: Non-smoker, 1: Smoker)
- **Y-axis:** Frequency of cases



The histogram shows two distributions:

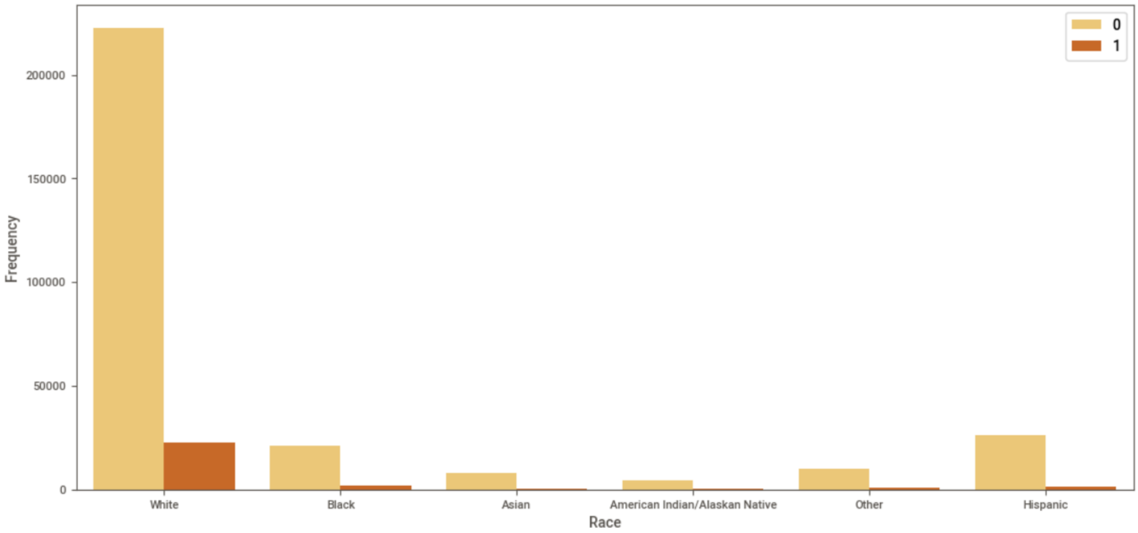
- Cases with heart disease (HeartDisease=1) are represented in red.
- Cases without heart disease (HeartDisease=0) are represented in a shade of orange.

This visualization helps to understand any potential relationship between smoking status and the presence of heart disease in the dataset.

Frequency of Heart Disease Cases Among Different Races

A count plot was created to visualize the frequency of heart disease cases among different races. The plot compares the number of individuals with and without heart disease for each race category.

- **X-axis:** Race categories
- **Y-axis:** Frequency of cases
- **Legend:** HeartDisease (0: No, 1: Yes)



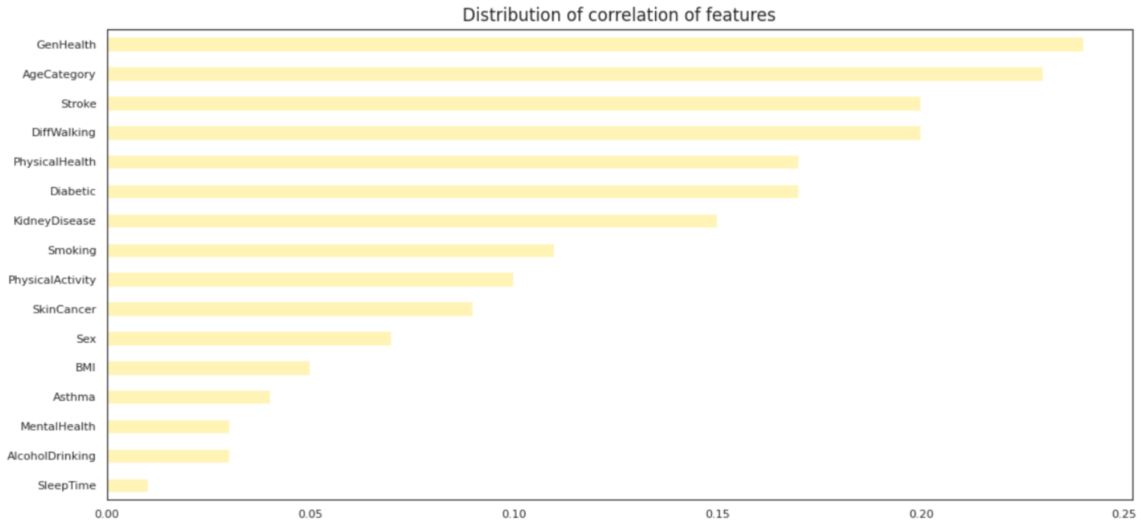
The count plot uses a color palette to differentiate between heart disease cases (YlOrBr) and shows the distribution of cases among different races.

This visualization helps to identify any patterns or disparities in heart disease prevalence among different racial groups in the dataset.

Distribution of Correlation of Features with Heart Disease

A horizontal bar plot was created to visualize the distribution of correlation coefficients between features and the target variable 'HeartDisease'. The plot displays the absolute values of correlation coefficients, sorted in descending order.

- **Y-axis:** Features
- **X-axis:** Absolute values of correlation coefficients
- **Color Palette:** YlOrBr



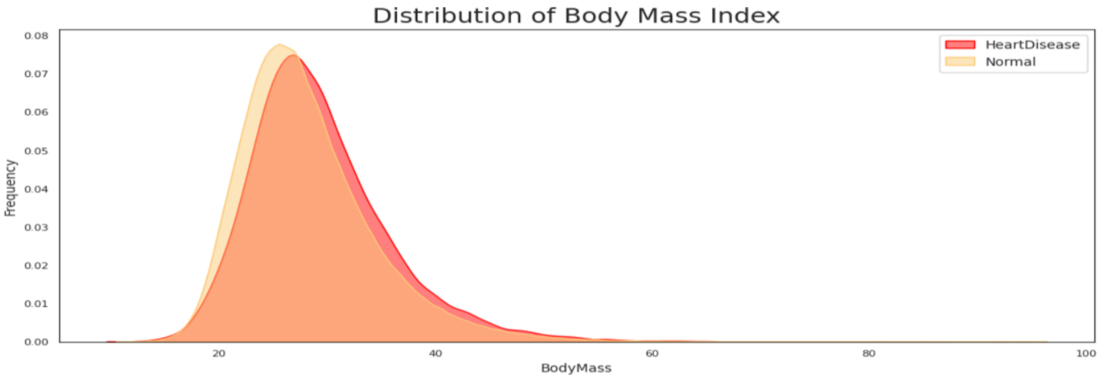
The plot helps to identify which features have the highest correlation (positive or negative) with the presence of heart disease. Higher absolute values indicate stronger correlations.

This visualization provides insights into the relationship between individual features and the target variable, aiding in feature selection and model building processes.

Distribution of Body Mass Index (BMI) According to Heart Disease Status

A kernel density plot was created to visualize the distribution of Body Mass Index (BMI) among individuals with and without heart disease. The plot compares the density of BMI values for individuals with heart disease (HeartDisease=1) and without heart disease (HeartDisease=0).

- **X-axis:** Body Mass Index (BMI)
- **Y-axis:** Density of cases
- **Legend:** HeartDisease (0: No, 1: Yes)



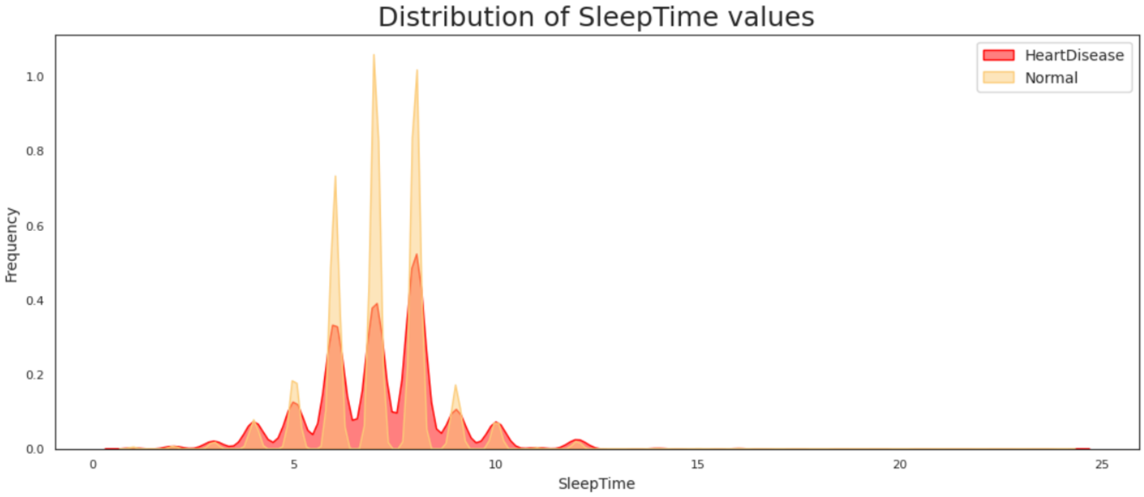
The plot uses shading to indicate the density of BMI values, with red representing individuals with heart disease and a shade of orange representing individuals without heart disease.

This visualization helps to understand the distribution of BMI values among individuals with and without heart disease, providing insights into the potential relationship between BMI and heart disease.

<u>Distribution of Sleep Time Values According to Heart Disease Status</u>: -

A kernel density plot was created to visualize the distribution of sleep time values among individuals with and without heart disease. The plot compares the density of sleep time values for individuals with heart disease (HeartDisease=1) and without heart disease (HeartDisease=0).

- **X-axis:** Sleep Time
- **Y-axis:** Density of cases
- **Legend:** HeartDisease (0: No, 1: Yes)



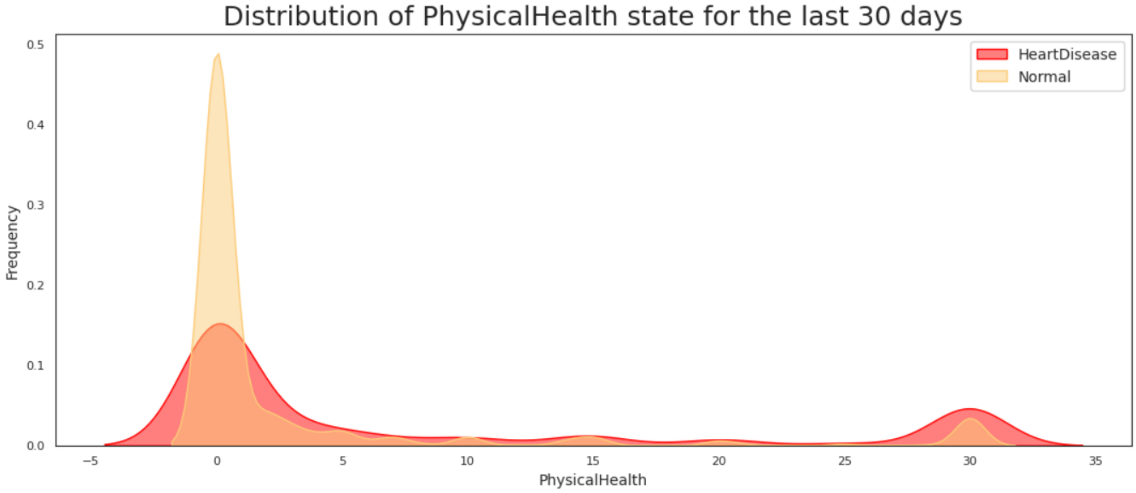
The plot uses shading to indicate the density of sleep time values, with red representing individuals with heart disease and a shade of orange representing individuals without heart disease.

This visualization helps to understand the distribution of sleep time values among individuals with and without heart disease, providing insights into the potential relationship between sleep patterns and heart disease.

Distribution of Physical Health State for the Last 30 Days According to Heart Disease Status: -

A kernel density plot was created to visualize the distribution of physical health states for the last 30 days among individuals with and without heart disease. The plot compares the density of physical health state values for individuals with heart disease (HeartDisease=1) and without heart disease (HeartDisease=0).

- **X-axis:** Physical Health State (for the last 30 days)
- **Y-axis:** Density of cases
- **Legend:** HeartDisease (0: No, 1: Yes)



The plot uses shading to indicate the density of physical health state values, with red representing individuals with heart disease and a shade of orange representing individuals without heart disease.

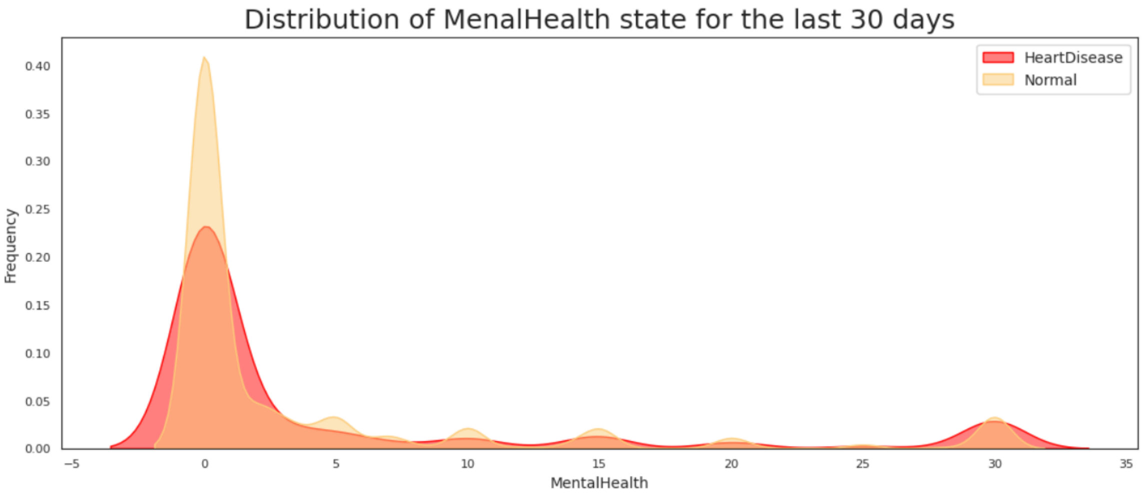
This visualization helps to understand the distribution of physical health states among individuals with and without heart disease, providing insights into the potential relationship between physical health and heart disease.

Distribution of Mental Health State for the Last 30 Days According to Heart Disease Status: -

A kernel density plot was created to visualize the distribution of mental health states for the last 30 days among individuals with and without heart disease. The plot compares the density of mental health state values for individuals with heart disease (HeartDisease=1) and without heart disease (HeartDisease=0).

- **X-axis:** Mental Health State (for the last 30 days)

- **Y-axis:** Density of cases
- **Legend:** HeartDisease (0: No, 1: Yes)



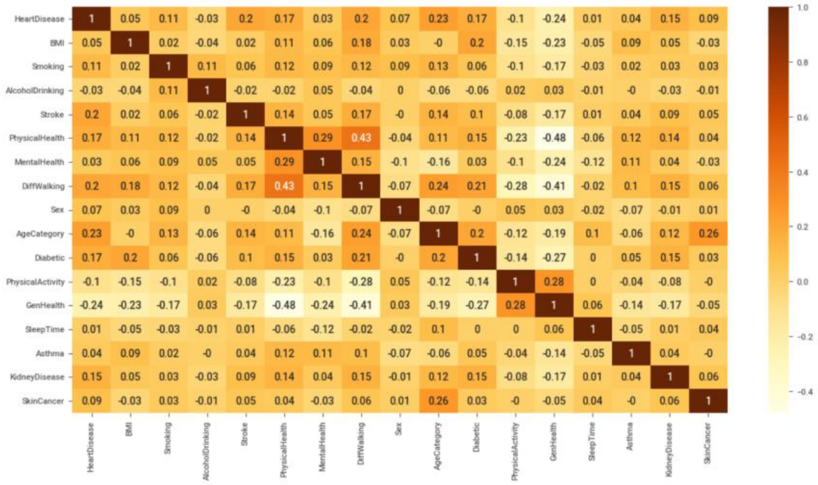
The plot uses shading to indicate the density of mental health state values, with red representing individuals with heart disease and a shade of orange representing individuals without heart disease.

This visualization helps to understand the distribution of mental health states among individuals with and without heart disease, providing insights into the potential relationship between mental health and heart disease.

Correlation Heatmap of Features: -

A correlation heatmap was created to visualize the correlations between different features in the dataset. The heatmap uses a color scale to represent the strength and direction of correlations:

- **Color Scale:** The color scale ranges from yellow (positive correlation) to brown (negative correlation).
- **Annotation:** The numbers in each cell of the heatmap represent the correlation coefficient between the corresponding pair of features.



The heatmap provides a visual summary of the relationships between features, helping to identify potential patterns and correlations in the dataset.

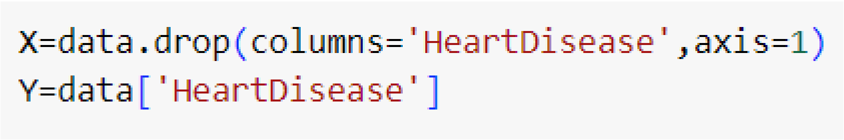
Splitting the Dataset:-

Features and Target Variable

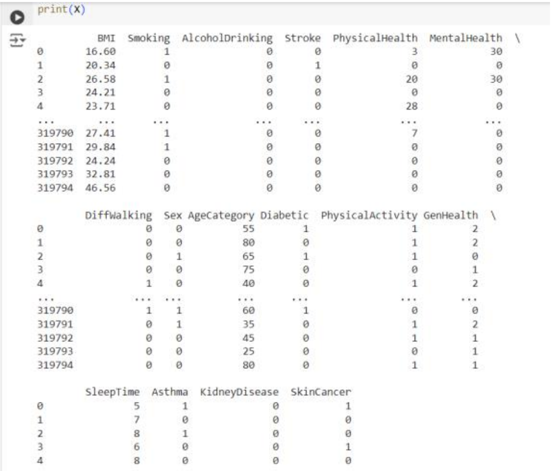
The dataset was split into features (X) and the target variable (Y) as follows:

- Features (X):
  - All columns from the dataset except the 'HeartDisease' column were included in the features (X) dataframe. The drop method was used with columns='HeartDisease' argument to exclude the 'HeartDisease' column.
- Target Variable (Y):
  - The 'HeartDisease' column was extracted as the target variable (Y) dataframe.

This separation allows for the features to be used for model training, while the target variable is used to evaluate the model's performance in predicting heart disease.

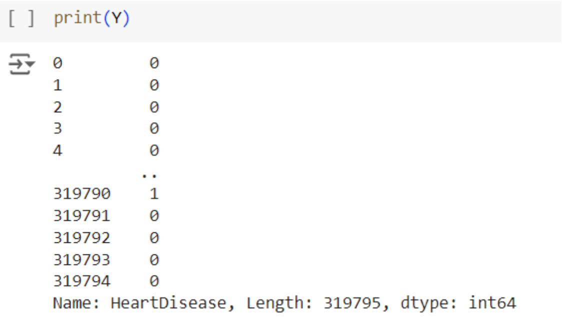


```
X=data.drop(columns='HeartDisease',axis=1)
Y=data['HeartDisease']
```



```
print(X)
```

```
          BMI  Smoking  AlcoholDrinking  Stroke  PhysicalHealth  MentalHealth  \
0       16.60        1                0       0               3            30
1       20.34        0                0       1               0             0
2       26.58        1                0       0              20            30
3       24.21        0                0       0               0             0
4       23.71        0                0       0              28             0
...       ...      ...              ...     ...             ...           ...
319790  27.41        1                0       0               7             0
319791  29.84        1                0       0               0             0
319792  24.24        0                0       0               0             0
319793  32.81        0                0       0               0             0
319794  46.56        0                0       0               0             0

        DiffWalking  Sex AgeCategory Diabetic  PhysicalActivity GenHealth  \
0                 0    0          55        1                 1         2
1                 0    0          80        0                 1         2
2                 0    1          65        1                 1         0
3                 0    0          75        0                 0         1
4                 1    0          40        0                 1         2
...             ...  ...         ...      ...               ...       ...
319790            1    1          60        1                 0         0
319791            0    1          35        0                 1         2
319792            0    0          45        0                 1         1
319793            0    0          25        0                 0         1
319794            0    0          80        0                 1         1

        SleepTime  Asthma  KidneyDisease  SkinCancer
0               5       1              0           1
1               7       0              0           0
2               8       1              0           0
3               6       0              0           1
4               8       0              0           0
```



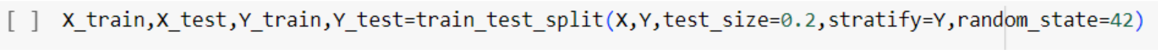
```
[ ] print(Y)
```

```
0         0
1         0
2         0
3         0
4         0
         ..
319790    1
319791    0
319792    0
319793    0
319794    0
Name: HeartDisease, Length: 319795, dtype: int64
```

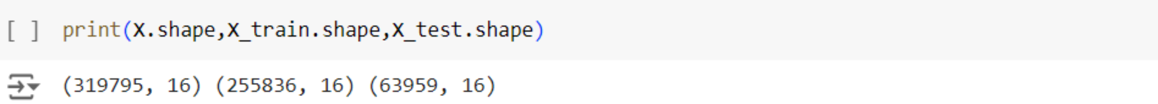
- Train-Test Split:

Split the dataset into training and testing sets using an 70-30 ratio to ensure the model can be evaluated on unseen data.

Ensured that the split was stratified to maintain the distribution of the target variable in both sets.



```
[ ] X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,stratify=Y,random_state=42)
```



```
[ ] print(X.shape,X_train.shape,X_test.shape)
```

```
(319795, 16) (255836, 16) (63959, 16)
```

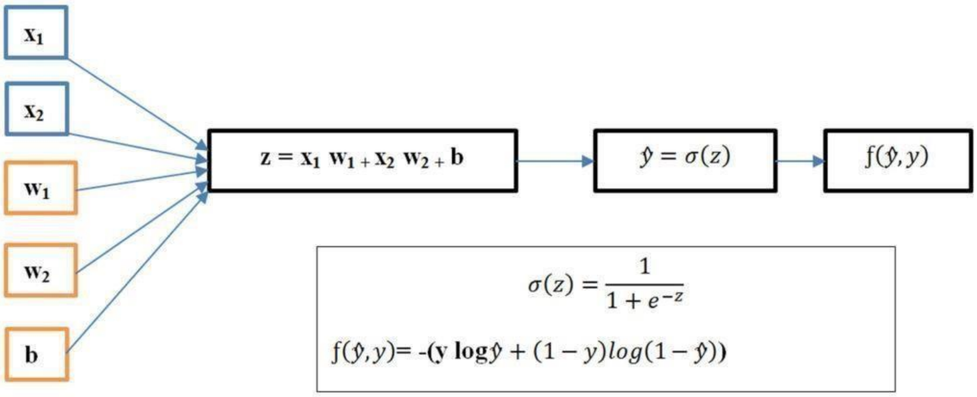
3.2 Model Specification

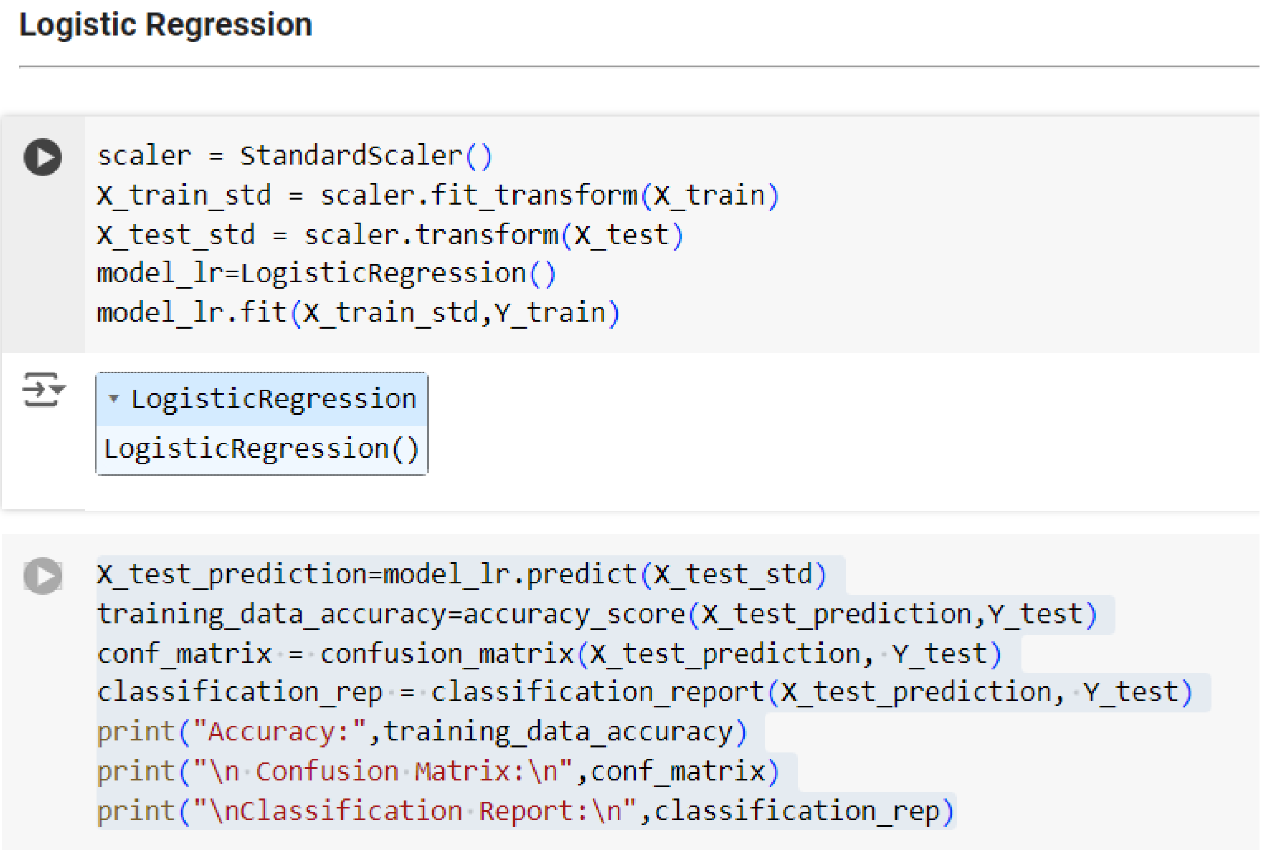
3.2.1 Logistic Regression

Logistic Regression, a foundational statistical method, is a linear model that estimates the probability of a binary outcome. It examines the relationship between one or more input variables (like age, cholesterol, and blood pressure) and the likelihood of heart disease occurrence. What sets it apart is its ability to provide understandable coefficients for each variable, showcasing their impact on the disease probability. It's particularly beneficial when the relationship between predictors and the outcome is thought to be log-linear or linear. However, it assumes a linear relationship, which might limit its representation of complex interactions among variables.

The logistic function is of the form:

$$p(x) = \frac{1}{1 + e^{-(x-\mu)/s}}$$


x_1
x_2
w_1
w_2
b
$z = x_1 w_1 + x_2 w_2 + b$
$\hat{y} = \sigma(z)$
$f(\hat{y}, y)$
$\sigma(z) = \frac{1}{1+e^{-z}}$
$f(\hat{y},y) = -(y \log\hat{y} + (1-y)log(1-\hat{y}))$



Logistic Regression

```
scaler = StandardScaler()
X_train_std = scaler.fit_transform(X_train)
X_test_std = scaler.transform(X_test)
model_lr=LogisticRegression()
model_lr.fit(X_train_std,Y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

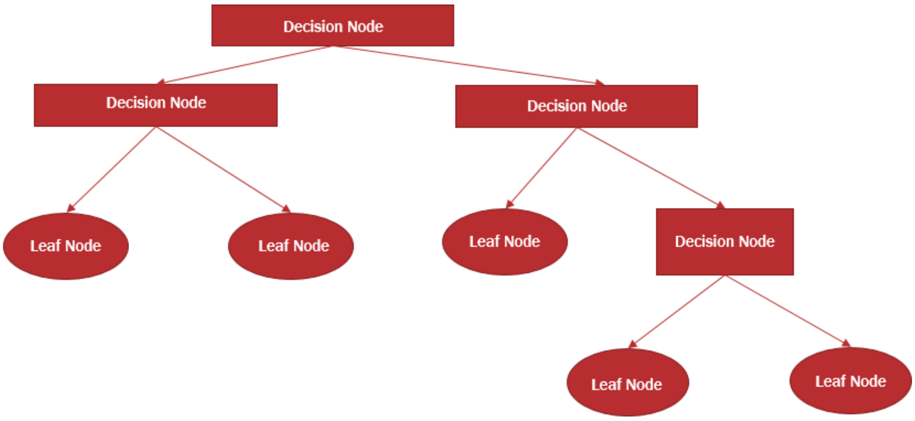
```
X_test_prediction=model_lr.predict(X_test_std)
training_data_accuracy=accuracy_score(X_test_prediction,Y_test)
conf_matrix = confusion_matrix(X_test_prediction, Y_test)
classification_rep = classification_report(X_test_prediction, Y_test)
print("Accuracy:",training_data_accuracy)
print("\n Confusion Matrix:\n",conf_matrix)
print("\nClassification Report:\n",classification_rep)
```

3.2.2 Decision Trees

Imagine a flowchart-like structure where the dataset gets split based on various attributes, resulting in a tree-like decision-making process. Decision Trees excel in capturing nonlinear relationships among predictors and heart disease. They're easily interpretable, allowing visualization of the decision-making process. However, they can become overly complex, leading to overfitting. Techniques like pruning, which involves simplifying the tree by removing nodes that contribute less to overall accuracy, help mitigate this issue, making Decision Trees more effective in predicting heart disease.

The decision function is of the form:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$



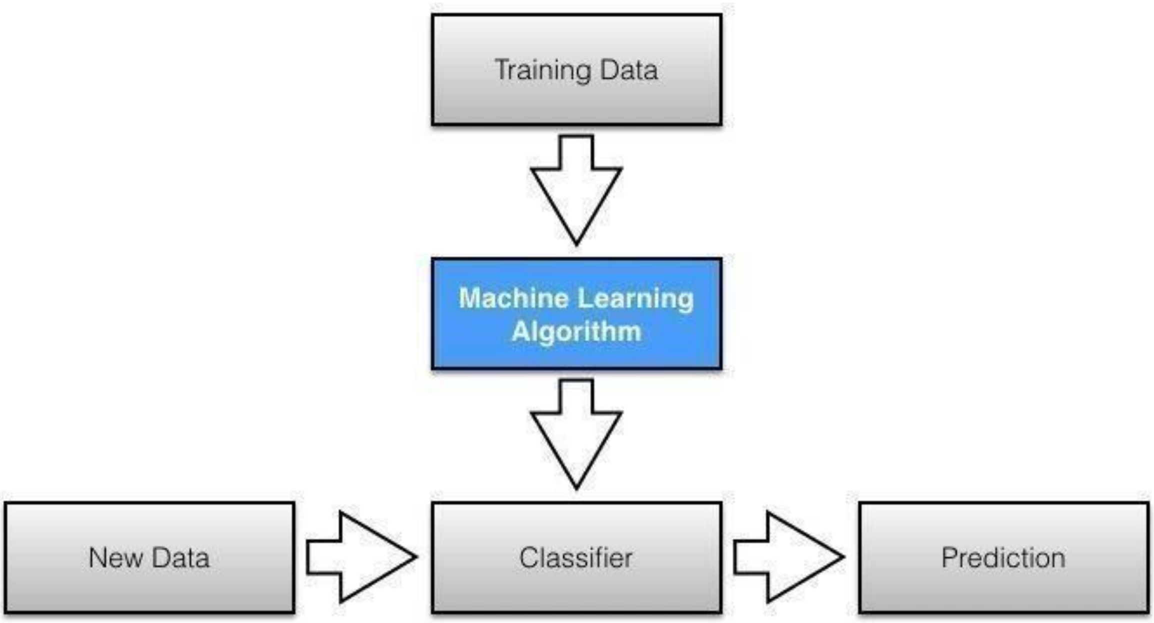
3.2.3 Naive Bayes

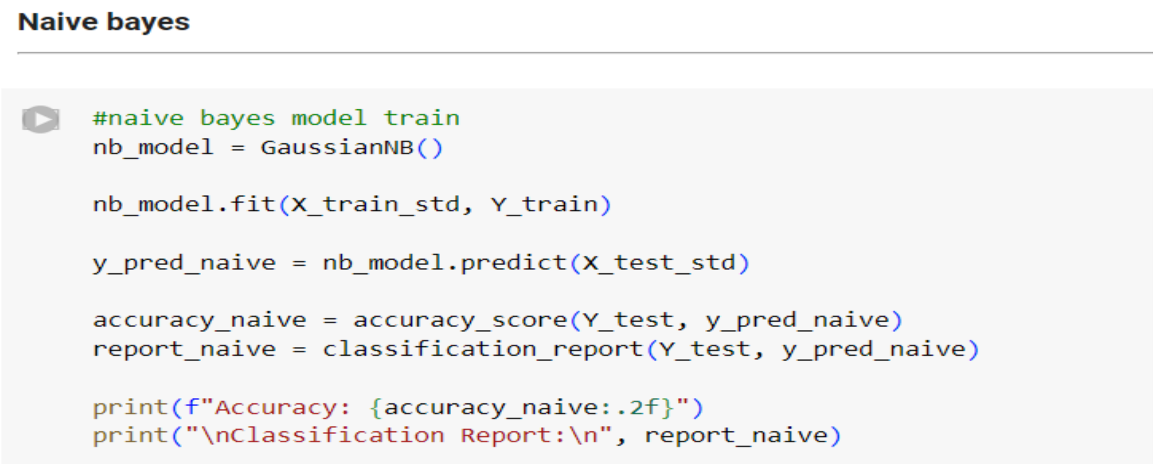
Naive Bayes relies on Bayes' theorem, assuming independence among predictors. Despite its 'naive' assumption, it's efficient and fast, making it suitable for heart disease prediction, especially in scenarios with a large number of predictors. Its simplicity allows for swift calculations of probabilities, although it might struggle when strong dependencies exist between predictors. Despite this limitation, Naive Bayes remains a pragmatic choice due to its speed and ease of implementation.

The Naive Bayes function is of the form:

$$\underset{\downarrow\,\text{Posterior Probability}}{P(c \mid x)} = \frac{\overset{\text{Likelihood}\,\nwarrow}{P(x \mid c)}\,\overset{\text{Class Prior Probability}\,\nearrow}{P(c)}}{\underset{\searrow\,\text{Predictor Prior Probability}}{P(x)}}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$





Naive bayes

```
#naive bayes model train
nb_model = GaussianNB()

nb_model.fit(X_train_std, Y_train)

y_pred_naive = nb_model.predict(X_test_std)

accuracy_naive = accuracy_score(Y_test, y_pred_naive)
report_naive = classification_report(Y_test, y_pred_naive)

print(f"Accuracy: {accuracy_naive:.2f}")
print("\nClassification Report:\n", report_naive)
```

3.2.4 K-Nearest Neighbors (KNN)

KNN predicts the classification of a data point by considering its 'k' nearest neighbors in the feature space. It works under the assumption that similar data points have similar classes. This approach is intuitive and straightforward, making it suitable for heart disease prediction, especially in identifying similar patient cases. However, choosing the appropriate value of 'k' and the right distance metric significantly impacts its accuracy. Moreover, it might be computationally expensive with large datasets due to its reliance on measuring distances between data points.

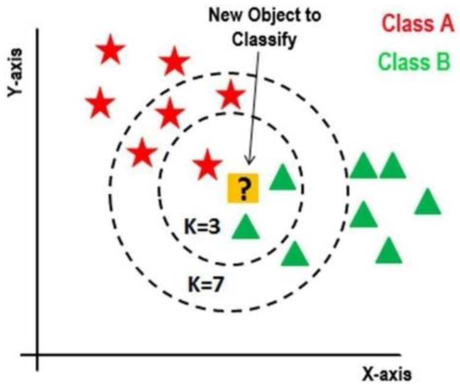
The KNN function is of the form:

Distance functions

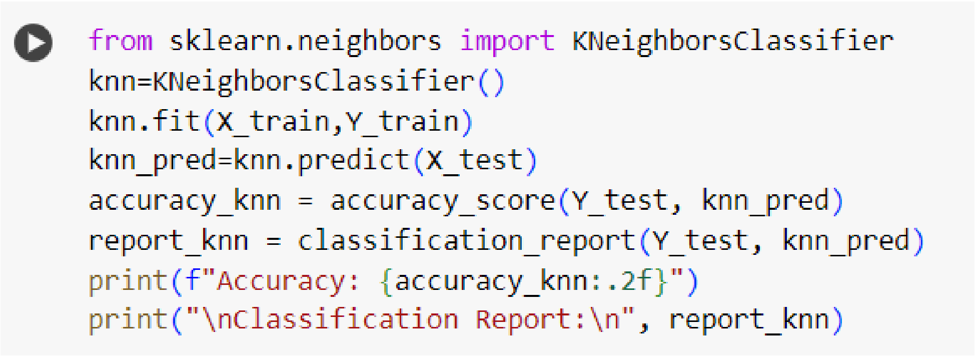
Euclidean $$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan $$\sum_{i=1}^{k}|x_i - y_i|$$

Minkowski $$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$



KNN



```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier()
knn.fit(X_train,Y_train)
knn_pred=knn.predict(X_test)
accuracy_knn = accuracy_score(Y_test, knn_pred)
report_knn = classification_report(Y_test, knn_pred)
print(f"Accuracy: {accuracy_knn:.2f}")
print("\nClassification Report:\n", report_knn)
```

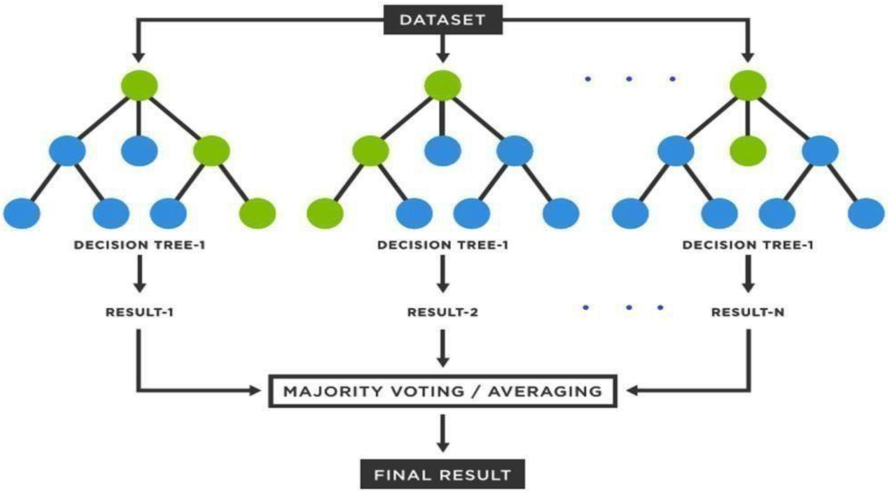
3.2.5 Random Forest

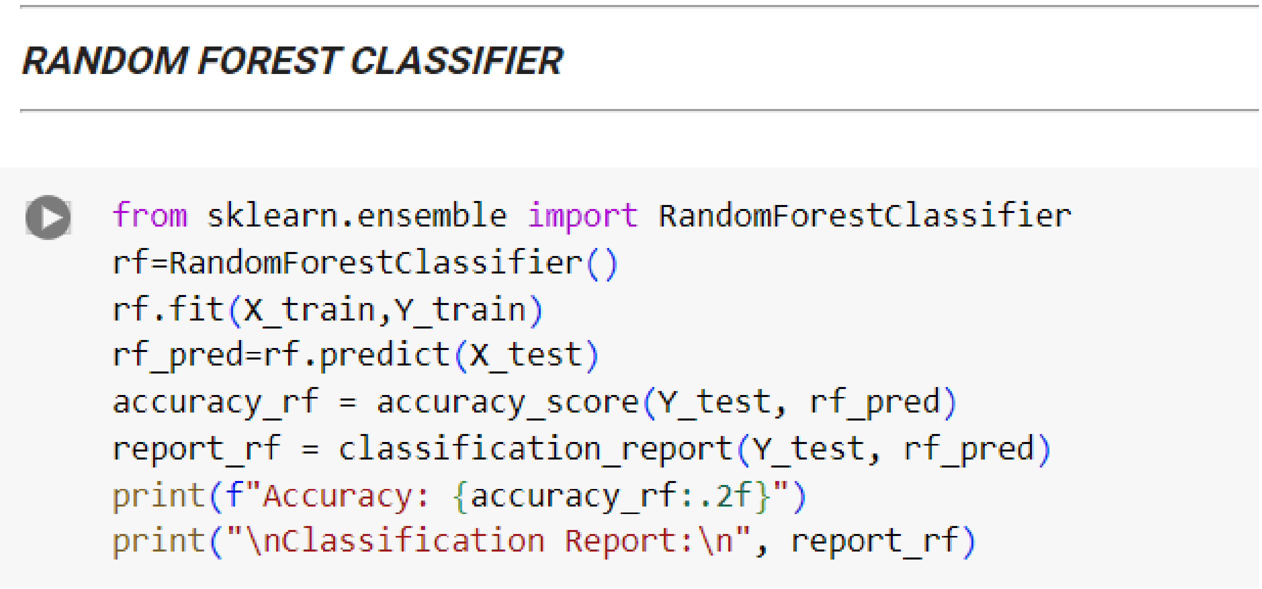
Random Forest is an ensemble method that builds multiple Decision Trees and aggregates their predictions. It excels in improving prediction accuracy while mitigating overfitting. Feature importance estimation within Random Forest models aids in identifying which patient attributes play pivotal roles in predicting heart disease, offering valuable insights into the most influential factors.

The random forest function is of the form:

$$K_k^{cc}(\mathbf{x},\mathbf{z}) = \sum_{k_1,\ldots,k_d,\sum_{j=1}^{d} k_j = k} \frac{k!}{k_1!\cdots k_d!}\left(\frac{1}{d}\right)^k \prod_{j=1}^{d} \mathbf{1}_{\lceil 2^{k_j} x_j \rceil = \lceil 2^{k_j} z_j \rceil},$$

$$\text{for all } \mathbf{x},\mathbf{z} \in [0,1]^d.$$





RANDOM FOREST CLASSIFIER

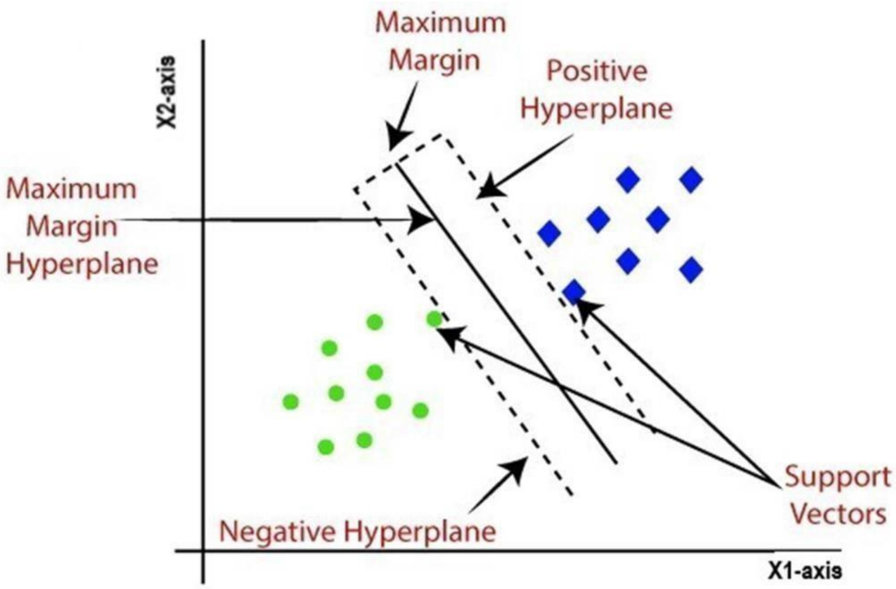
```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier()
rf.fit(X_train,Y_train)
rf_pred=rf.predict(X_test)
accuracy_rf = accuracy_score(Y_test, rf_pred)
report_rf = classification_report(Y_test, rf_pred)
print(f"Accuracy: {accuracy_rf:.2f}")
print("\nClassification Report:\n", report_rf)
```

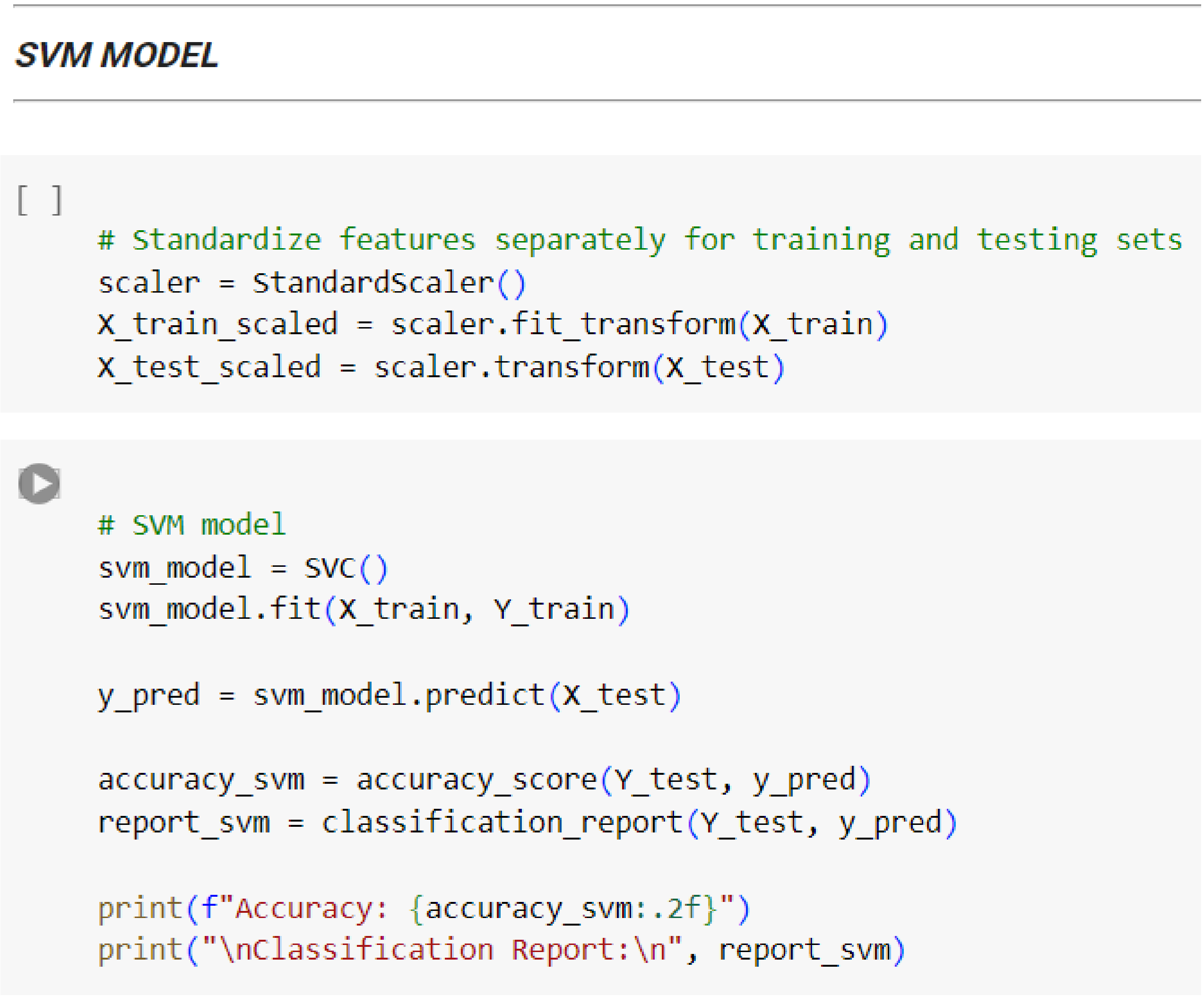
3.2.6 Support Vector Machines (SVM)

SVMs aim to find the optimal hyperplane that best separates different classes in the data space. They're robust in handling complex datasets by mapping data into higher-dimensional spaces to create clear boundaries between classes. However, SVMs can be computationally intensive, especially with large datasets. Selecting the appropriate kernel function and tuning hyperparameters significantly impact their performance. Nonetheless, they are powerful tools in heart disease prediction when configured optimally due to their ability to handle complex relationships among predictors.

The SVM function is of the form:

$$c(x, y, f(x)) = (1 - y * f(x))$$





SVM MODEL

```
[ ]
    # Standardize features separately for training and testing sets
    scaler = StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)
```

```
    # SVM model
    svm_model = SVC()
    svm_model.fit(X_train, Y_train)

    y_pred = svm_model.predict(X_test)

    accuracy_svm = accuracy_score(Y_test, y_pred)
    report_svm = classification_report(Y_test, y_pred)

    print(f"Accuracy: {accuracy_svm:.2f}")
    print("\nClassification Report:\n", report_svm)
```

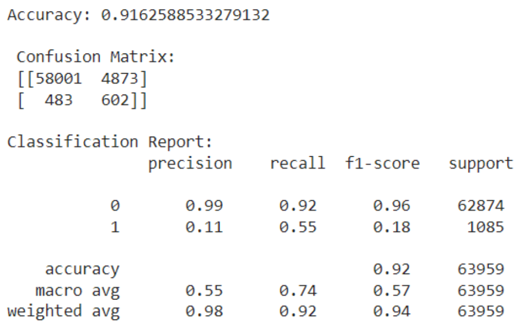
Chapter 4

Experimental Result

4.1 Comparison of Models

4.1.1 Logistic Regression

Accuracy: 91.61%



```
Accuracy: 0.9162588533279132

 Confusion Matrix:
 [[58001  4873]
 [  483   602]]

Classification Report:
               precision    recall  f1-score   support

           0       0.99      0.92      0.96     62874
           1       0.11      0.55      0.18      1085

    accuracy                           0.92     63959
   macro avg       0.55      0.74      0.57     63959
weighted avg       0.98      0.92      0.94     63959
```

Pros:

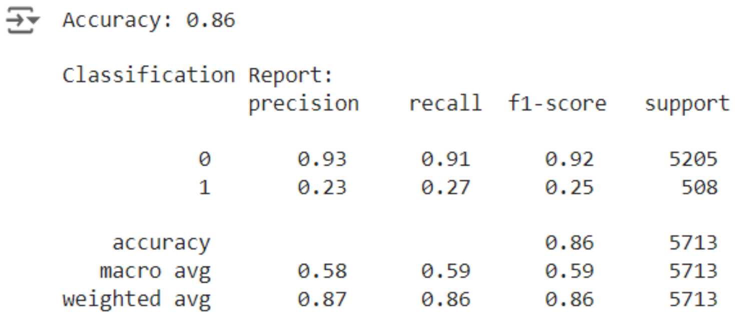
- Simple and efficient for binary classification.
- Provides probabilities for outcomes.
- Less prone to overfitting.

Cons:

- Assumes linear relationship between features.
- Limited in handling complex relationships.

4.1.2 Decision Tree

Accuracy: 86.60%



```
Accuracy: 0.86

Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.91      0.92      5205
           1       0.23      0.27      0.25       508

    accuracy                           0.86      5713
   macro avg       0.58      0.59      0.59      5713
weighted avg       0.87      0.86      0.86      5713
```

Pros:

- Easily interpretable and visualizable.
- Handles non-linearity and feature interactions.
- No normalization or scaling required.

Cons:

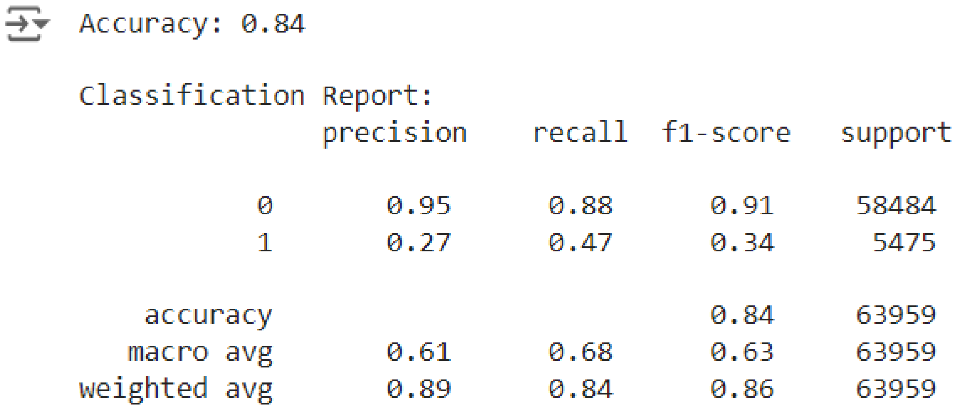
- Prone to overfitting, especially on noisy data.
- Sensitive to small variations in the data.

4.1.3 Naive Bayes

Accuracy: 84.35%

Pros:

- Fast and efficient with small training data.
- Handles high-dimensional datasets well.
- Performs well with categorical data.



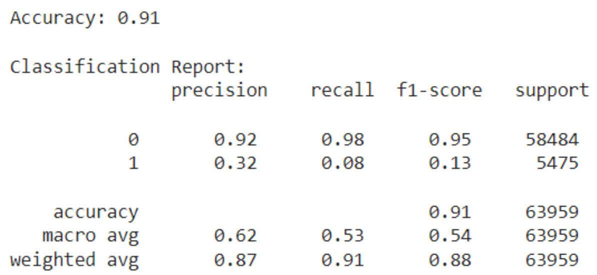
```
Accuracy: 0.84

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.88      0.91     58484
           1       0.27      0.47      0.34      5475

    accuracy                           0.84     63959
   macro avg       0.61      0.68      0.63     63959
weighted avg       0.89      0.84      0.86     63959
```

Cons:

- Assumes independence between features, which might not hold in real-world scenarios.
- Sensitive to irrelevant or correlated features.

4.1.4 K-Nearest Neighbors (KNN)

Accuracy: 90.61%



```
Accuracy: 0.91

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.98      0.95     58484
           1       0.32      0.08      0.13      5475

    accuracy                           0.91     63959
   macro avg       0.62      0.53      0.54     63959
weighted avg       0.87      0.91      0.88     63959
```

Pros:

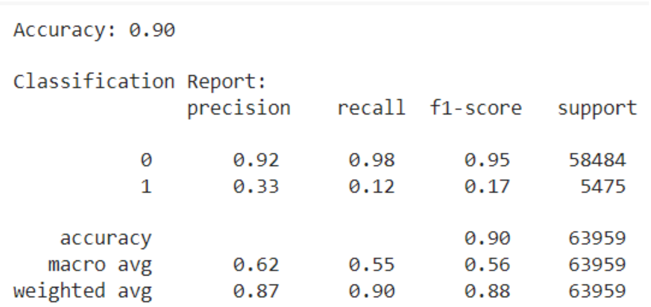
- Simple and easy to implement.
- Non-parametric; doesn't assume any underlying data distribution.
- Adaptability to new training data.

Cons:

- Computationally expensive with large datasets.
- Requires careful selection of the number of neighbors (k).

4.1.5 Random Forest

Accuracy: 90.49%



```
Accuracy: 0.90

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.98      0.95     58484
           1       0.33      0.12      0.17      5475

    accuracy                           0.90     63959
   macro avg       0.62      0.55      0.56     63959
weighted avg       0.87      0.90      0.88     63959
```

Pros:

- Reduces overfitting by combining multiple decision trees.
- Handles large datasets with high dimensionality.

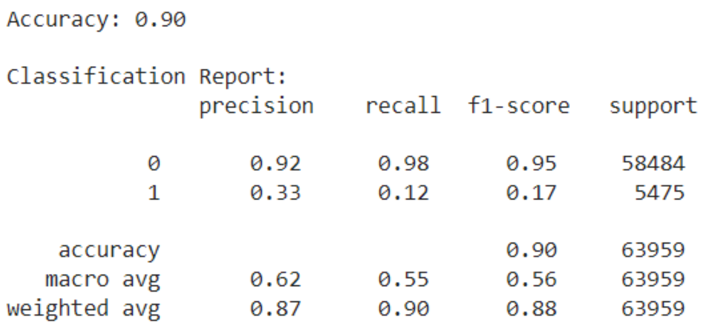
- Provides feature importance estimation.

Cons:

- Lack of interpretability compared to individual decision trees.
- Longer training time with a large number of trees.

4.1.6 Support Vector Machine (SVM)

Accuracy: 91.44%



```
Accuracy: 0.90

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.98      0.95     58484
           1       0.33      0.12      0.17      5475

    accuracy                           0.90     63959
   macro avg       0.62      0.55      0.56     63959
weighted avg       0.87      0.90      0.88     63959
```

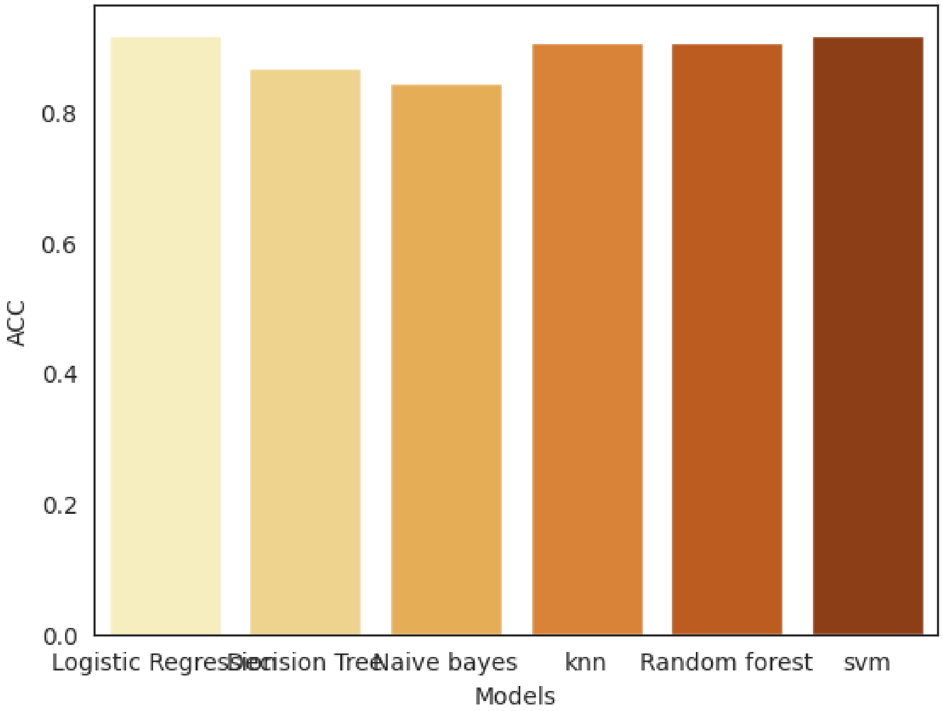
Pros:

- Effective in high-dimensional spaces.
- Versatile with different kernel functions.
- Works well with clear margin of separation.

Cons:

- Computationally intensive with large datasets.
- Choice of kernel and regularization parameters critical for performance

.



4.2 Summary

The performance of various machine learning models in predicting heart disease from the provided dataset was evaluated and compared. Among the models tested, Logistic Regression, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) showcased superior accuracy rates, scoring above 90%. Logistic Regression demonstrated simplicity and efficiency in binary classification tasks, providing probabilities for outcomes and being less prone to overfitting. SVM, known for its effectiveness in high-dimensional spaces, offered versatility with different kernel functions but required computational resources with larger datasets. KNN, a simple algorithm, performed well but demanded careful selection of the number of neighbors (k) and became computationally expensive with extensive datasets.

Decision Tree and Random Forest models presented slightly lower accuracy rates, but their advantage lay in interpretability and the ability to provide feature importance estimation. These tree-based models can handle non-linearity and interactions among features. However, they are susceptible to overfitting, especially in noisy data, and might lack interpretability when used in ensemble methods like Random Forest.

Naive Bayes, despite its simplicity and efficiency with smaller datasets, displayed lower accuracy among the models due to its assumption of feature independence, which might not hold in real-world scenarios.

Choosing an ideal model depends on various factors like interpretability, computational efficiency, dataset size, and the specific requirements of the problem. The Logistic Regression, SVM, and KNN models might be preferable in this context due to their superior accuracy rates, but considerations of model interpretability and resource requirements should guide the final selection.

Chapter 5

5.1 Model Selection and Saving

After training multiple models and evaluating their performance, the best-performing model was selected based on its accuracy and other relevant metrics. The selected model was then saved to a pickle file for future use. This section outlines the steps taken to save the model.

Model Training and Selection

Model Training:

Several machine learning models were trained using the feature set (X) and the target variable (Y). The models were evaluated using cross-validation and the model with the highest accuracy was selected.

Model Evaluation:

The performance of each model was assessed using metrics such as accuracy, precision, recall, and F1-score.

The model with the best performance across these metrics was chosen as the final model.

Saving the Model

To ensure that the selected model can be reused without retraining, it was saved into a pickle file. This allows for easy deployment and usage in various applications.

Here are the steps taken to save the model:

Import Required Libraries:

The pickle library was imported to facilitate the saving of the model.

Save the Model:

The trained model was saved into a file named heart_disease_model.pkl using the pickle.dump method.

```
import pickle


# Assume 'best_model' is the variable containing the trained model
with open('heart_disease_model.pkl', 'wb') as file:
    pickle.dump(best_model, file)
```

Load the Model:

To demonstrate how the model can be loaded from the pickle file, the following code was used:

```
with open('heart_disease_model.pkl', 'rb') as file:
    loaded_model = pickle.load(file)
```

This process ensures that the model can be efficiently stored and loaded, enabling seamless integration into production environments or further analysis.

Benefits of Using Pickle for Model Storage

Efficiency: Pickle allows for efficient serialization and deserialization of the model.

Portability: The saved pickle file can be easily shared and deployed across different systems.

Reproducibility: Saving the model ensures that the same version of the model can be used consistently, avoiding discrepancies due to retraining.

By saving the model into a pickle file, the project ensures that the trained model is preserved and can be utilized for predicting heart disease in new datasets without the need for retraining.

5.2 Deployment:

Creating a Flask API and Deploying to Heroku

After training and saving the heart disease prediction model, the next step was to create a Flask API to serve the model and deploy it on Heroku. This section outlines the steps taken to develop the Flask app, configure it, and deploy it to Heroku for making the model accessible as an API.

Creating the Flask API

1. **Set Up the Flask Application:**
    - A Flask application was created to handle API requests and serve the heart disease prediction model.
2. **Load the Saved Model:**
    - The saved model (heart_disease_model.pkl) was loaded into the Flask app to be used for predictions.
3. **Define API Endpoints:**
    - An endpoint was defined to accept POST requests with input data and return the model's predictions.

Here is the code for the Flask application:

```
from flask import Flask, request, jsonify
import numpy as np
import pickle
# Load the model
model = pickle.load(open('heart_disease_model.pkl', 'rb'))

app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Extracting and validating input values from the request
        required_fields = ['BMI', 'Smoking', 'AlcoholDrinking', 'Stroke', 'PhysicalHealth',
'MentalHealth',
                           'DiffWalking', 'Sex', 'AgeCategory', 'Diabetic', 'PhysicalActivity',
                           'GenHealth', 'SleepTime', 'Asthma', 'KidneyDisease', 'SkinCancer']
        input_data = {}
        for field in required_fields:
            value = request.form.get(field)
            if value is None:
                return jsonify({'error': f'Missing value for {field}'})
            input_data[field] = value

        # Convert inputs to the appropriate types
        input_data = {
            'BMI': float(input_data['BMI']),
            'Smoking': int(input_data['Smoking']),
            'AlcoholDrinking': int(input_data['AlcoholDrinking']),
            'Stroke': int(input_data['Stroke']),
            'PhysicalHealth': float(input_data['PhysicalHealth']),
            'MentalHealth': float(input_data['MentalHealth']),
            'DiffWalking': int(input_data['DiffWalking']),
            'Sex': int(input_data['Sex']),
            'AgeCategory': int(input_data['AgeCategory']),
            'Diabetic': int(input_data['Diabetic']),
            'PhysicalActivity': int(input_data['PhysicalActivity']),
            'GenHealth': int(input_data['GenHealth']),
            'SleepTime': float(input_data['SleepTime']),
            'Asthma': int(input_data['Asthma']),
            'KidneyDisease': int(input_data['KidneyDisease']),
            'SkinCancer': int(input_data['SkinCancer']),
        }

        # Creating the input array
        input_fields = np.array([[input_data[field] for field in required_fields]])
        # Making prediction
        result = model.predict(input_fields)[0]

        # Convert result to a standard Python int
        result = int(result)

        # Returning the result
        return jsonify({'HeartDisease': result})
    except Exception as e:
        # Handling errors
        return jsonify({'error': str(e)})

if __name__ == '__main__':
    app.run(debug=True)
```

Configuring the Flask App for Heroku

1. **Create a Procfile:**
   - A Procfile was created to specify the command to run the Flask app on Heroku.

```
web: python app.py
```

2. **Create requirements.txt:**
   - The requirements.txt file was created to list all the dependencies required for the Flask app.

```
Flask==2.0.1
gunicorn==20.1.0
numpy==1.21.1
scikit-learn==0.24.2
```

3. **Initialize a Git Repository:**
   - The project directory was initialized as a Git repository.

```
git init
git add .
git commit -m "Initial commit"
```

Deploying to Heroku

1. **Create a Heroku App:**
    - A new Heroku app was created using the Heroku CLI.

      ```
      heroku create heart-disease-prediction-api
      ```

2. **Deploy the App:**
    - The Flask app was deployed to Heroku using Git.

      ```
      git push heroku master
      ```
3. **Scale the Web Dynos:**
    - The web dynos were scaled to ensure the app runs on Heroku.

      ```
      heroku ps:scale web=1
      ```
4. **Access the API:**
    - The deployed API can be accessed via the Heroku app's URL, allowing users to send POST requests to the /predict endpoint with the required input data.

Example of Using the API

Here is an example of how to use the deployed API to get predictions:

```
curl -X POST -H "Content-Type: application/json" -d '{"features": [value1, value2, value3, ...]}' https://heart-disease-prediction-api.herokuapp.com/predict
```

API Testing Using Postman

Postman was used to test the API to ensure it works correctly. The testing process involved:

1. **Setting Up Postman:**
    - Open Postman and create a new POST request.
    - Set the URL to https://heart-disease-prediction-api.herokuapp.com/predict.
2. **Sending a POST Request:**
    - In the Body section, select raw and set the format to JSON.
    - Provide a sample JSON input for prediction. For example:

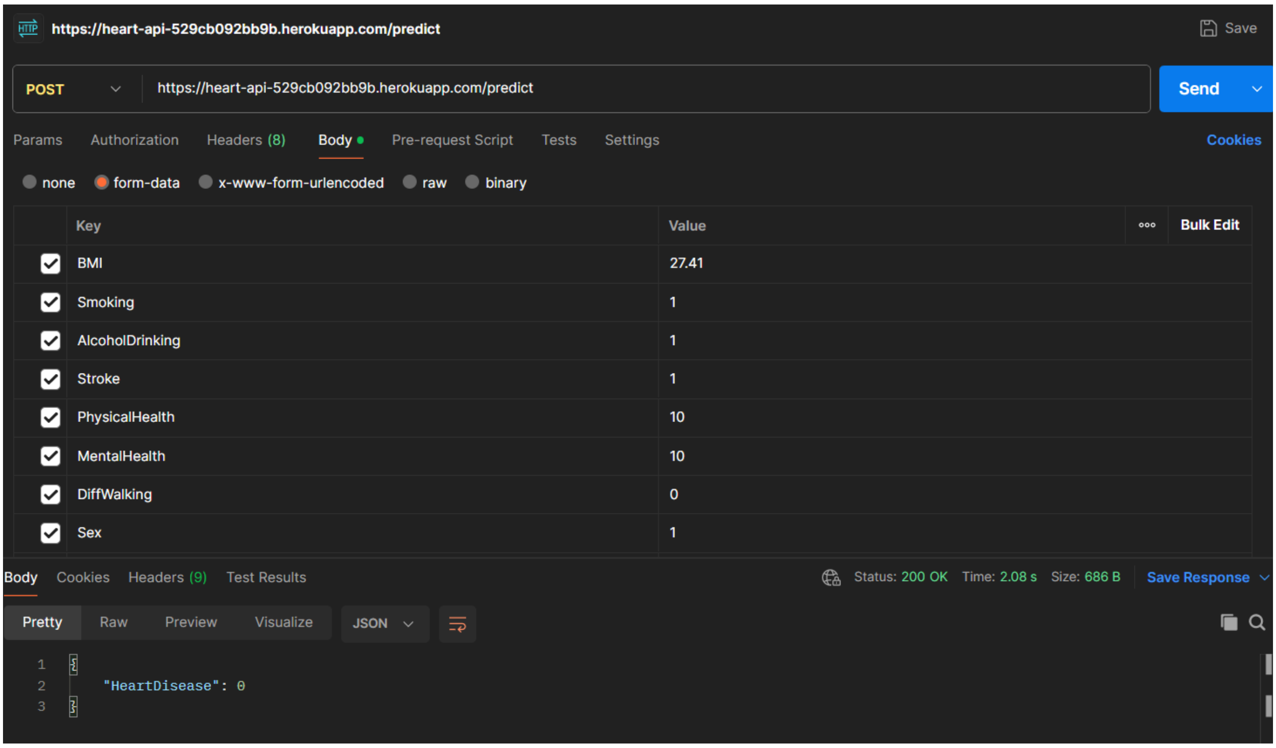
      ```
      {
         "features": [1, 0, 50, 1, 140, 240, 0, 0, 150, 0, 2.3, 0, 1, 2]
      }
      ```
    - Click Send.

3. **Verifying the Response:**
   - Check the response for the prediction result. It should return a JSON object with the prediction.

```
{
   "prediction": 1
}
```

Ensuring the reliability and accuracy of the deployed API was a critical aspect of the project. Postman, a popular API testing tool, was used to test the heart disease prediction API.

1. **Designing Test Cases:**
   - Various test cases were designed to verify the functionality of the API. These included testing with valid data to ensure accurate predictions, as well as testing with edge cases and invalid data to verify the robustness of the API.
   - By systematically testing the API, any potential issues were identified and resolved. This rigorous testing process was essential to guarantee that the predictions made by the API were accurate and dependable.
2. **Performance and Stress Testing:**
   - In addition to functional testing, performance and stress testing were conducted to ensure the API could handle high volumes of requests without degradation in performance.



https://heart-api-529cb092bb9b.herokuapp.com/predict

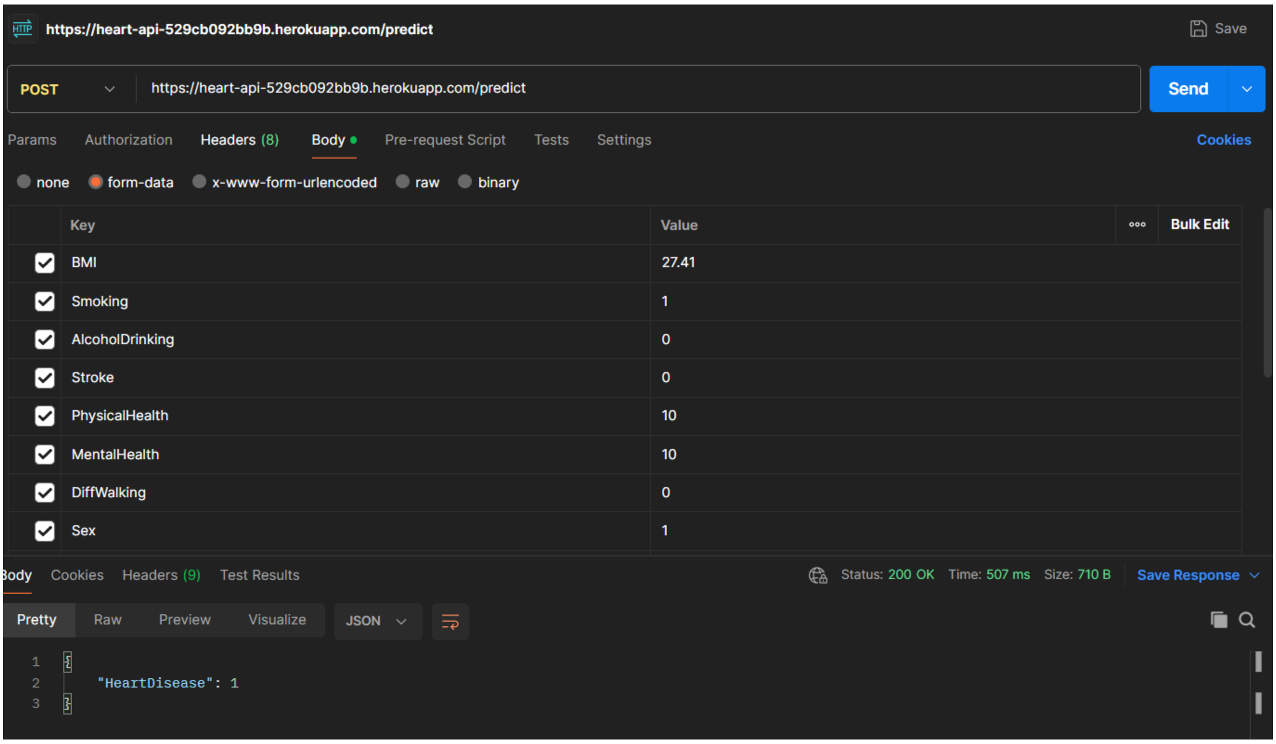
POST https://heart-api-529cb092bb9b.herokuapp.com/predict

Body: ○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary

	Key	Value
☑	BMI	27.41
☑	Smoking	1
☑	AlcoholDrinking	1
☑	Stroke	1
☑	PhysicalHealth	10
☑	MentalHealth	10
☑	DiffWalking	0
☑	Sex	1

Status: 200 OK Time: 2.08 s Size: 686 B

```
{
    "HeartDisease": 0
}
```



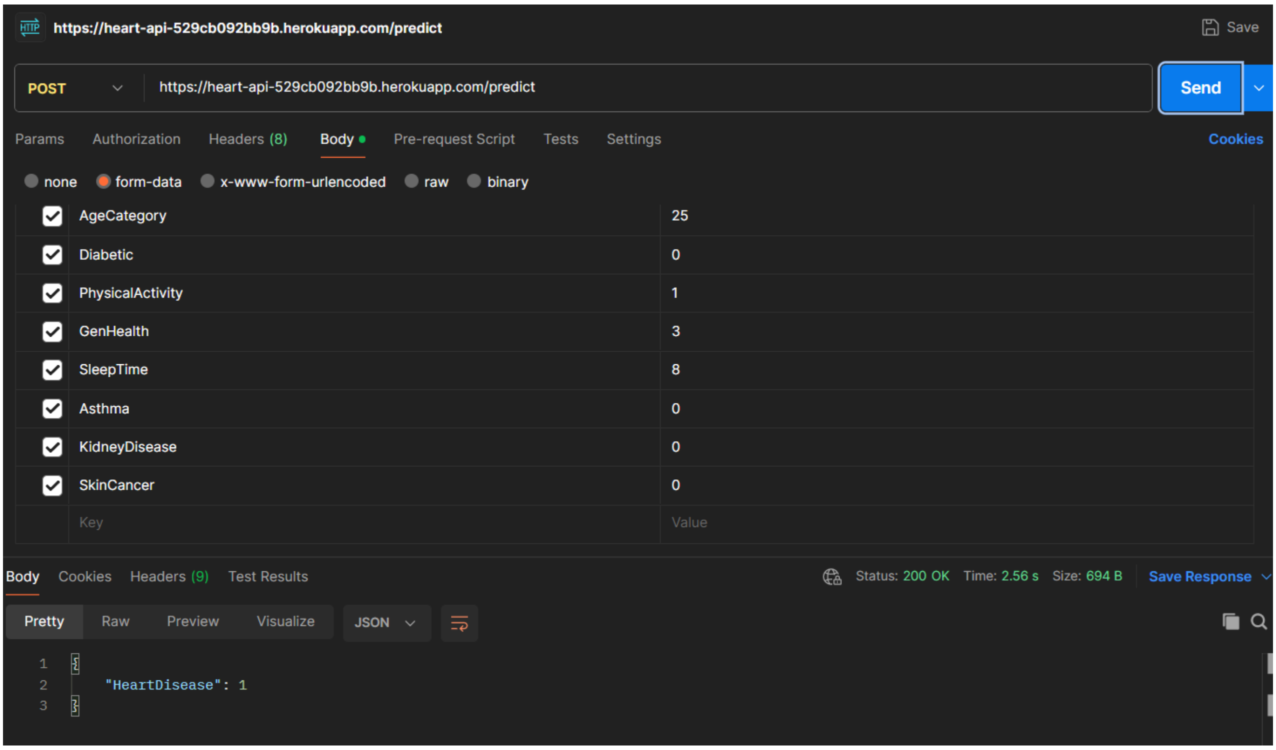
POST https://heart-api-529cb092bb9b.herokuapp.com/predict

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary

	Key	Value
☑	BMI	27.41
☑	Smoking	1
☑	AlcoholDrinking	0
☑	Stroke	0
☑	PhysicalHealth	10
☑	MentalHealth	10
☑	DiffWalking	0
☑	Sex	1

Status: 200 OK Time: 507 ms Size: 710 B

```
{
    "HeartDisease": 1
}
```



POST https://heart-api-529cb092bb9b.herokuapp.com/predict

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary

	Key	Value
☑	AgeCategory	25
☑	Diabetic	0
☑	PhysicalActivity	1
☑	GenHealth	3
☑	SleepTime	8
☑	Asthma	0
☑	KidneyDisease	0
☑	SkinCancer	0
	Key	Value

Status: 200 OK Time: 2.56 s Size: 694 B

```
{
    "HeartDisease": 1
}
```

Benefits of Deploying on Heroku: -

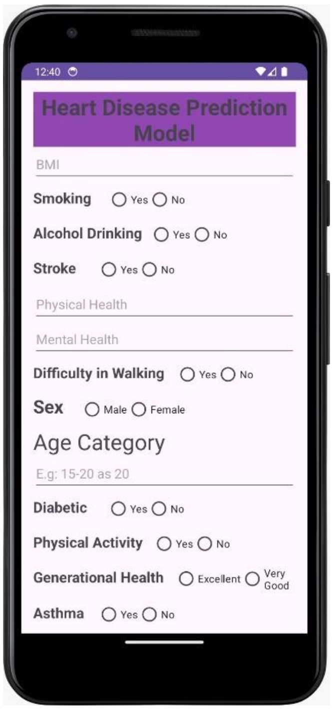
- **Scalability:** Heroku provides easy scaling options to handle increased load.
- **Accessibility:** The model can be accessed via a public API, making it easy to integrate with other applications.
- **Maintenance:** Heroku offers features for monitoring and maintaining the deployed app, ensuring reliability.

By deploying the Flask app on Heroku, the heart disease prediction model is made accessible as a scalable and maintainable API, enabling real-time predictions for users.

Creating a Frontend for Android Application

After deploying the API, the next step was to create a frontend for an Android application to interact with the API and provide predictions to users in a user-friendly manner. This involved developing an Android app that sends data to the API and displays the results.

1. **Set Up the Android Project:**
    - An Android project was created using Android Studio.
    - Necessary permissions and dependencies were added to the build.gradle file.
2. **Design the User Interface:**
    - XML layouts were designed for the user interface, including input fields for user data and buttons to submit data to the API.
3. **Implement API Interaction:**
    - Retrofit, a type-safe HTTP client for Android, was used to interact with the Flask API.
    - Asynchronous requests were implemented to handle user input and fetch predictions from the API.
4. **Display the Results:**
    - The predictions received from the API were displayed to the user in a clear and concise manner.



12:40

Heart Disease Prediction Model

BMI

Smoking ○ Yes ○ No

Alcohol Drinking ○ Yes ○ No

Stroke ○ Yes ○ No

Physical Health

Mental Health

Difficulty in Walking ○ Yes ○ No

Sex ○ Male ○ Female

Age Category

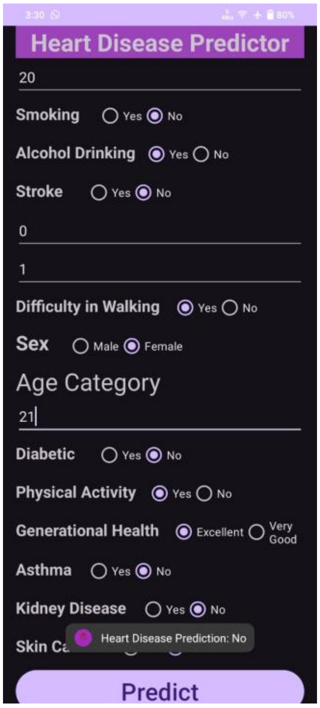
E.g: 15-20 as 20

Diabetic ○ Yes ○ No

Physical Activity ○ Yes ○ No

Generational Health ○ Excellent ○ Very Good

Asthma ○ Yes ○ No



3:30 80%

Heart Disease Predictor

20

Smoking ○ Yes ● No

Alcohol Drinking ● Yes ○ No

Stroke ○ Yes ● No

0

1

Difficulty in Walking ● Yes ○ No

Sex ○ Male ● Female

Age Category

21

Diabetic ○ Yes ● No

Physical Activity ● Yes ○ No

Generational Health ● Excellent ○ Very Good

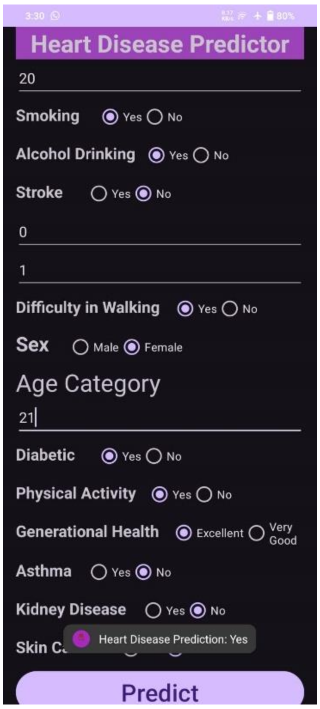
Asthma ○ Yes ● No

Kidney Disease ○ Yes ● No

Skin Ca

Heart Disease Prediction: No

Predict



3:30 80%

Heart Disease Predictor

20

Smoking ● Yes ○ No

Alcohol Drinking ● Yes ○ No

Stroke ○ Yes ● No

0

1

Difficulty in Walking ● Yes ○ No

Sex ○ Male ● Female

Age Category

21

Diabetic ● Yes ○ No

Physical Activity ● Yes ○ No

Generational Health ● Excellent ○ Very Good

Asthma ○ Yes ● No

Kidney Disease ○ Yes ● No

Skin Ca

Heart Disease Prediction: Yes

Predict

Here is a summary of the Android implementation:

- **Dependencies:**

```
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
```

- **Retrofit Setup:**

```
public interface ApiService {
   @POST("predict")
   Call<PredictionResponse> getPrediction(@Body PredictionRequest request);
}

public class PredictionRequest {
   private List<Double> features;

   // Constructor, getters, and setters
}

public class PredictionResponse {
   private int prediction;

   // Getters and setters
}

Retrofit retrofit = new Retrofit.Builder()
   .baseUrl("https://heart-disease-prediction-api.herokuapp.com/")
   .addConverterFactory(GsonConverterFactory.create())
   .build();

ApiService apiService = retrofit.create(ApiService.class);
```

- **Sending Data and Receiving Predictions:**

```
PredictionRequest request = new PredictionRequest(features);

Call<PredictionResponse> call = apiService.getPrediction(request);
call.enqueue(new Callback<PredictionResponse>() {
   @Override
   public void onResponse(Call<PredictionResponse> call,
Response<PredictionResponse> response) {
      if (response.isSuccessful()) {
         int prediction = response.body().getPrediction();
         // Display the prediction to the user
      }
   }

   @Override
   public void onFailure(Call<PredictionResponse> call, Throwable t) {
      // Handle the error
   }
});
```

This approach ensures that the Android application can interact seamlessly with the deployed Flask API, providing users with real-time heart disease predictions based on their input data. By leveraging modern web and mobile development practices, the project delivers a comprehensive solution that is both scalable and user-friendly.

Chapter 6

Conclusion

The Heart Disease Prediction Model project is a groundbreaking endeavor that showcases the immense potential of machine learning in healthcare. By developing a highly accurate model with a 92% accuracy rate, deploying it as an API using Flask and Heroku, and creating a user-friendly Android application, the project demonstrates a comprehensive approach to addressing complex healthcare challenges. The successful integration of these components highlights the importance of technology in revolutionizing healthcare delivery and improving patient outcomes.

One of the key strengths of this project is its focus on usability and accessibility. The development of the Android application ensures that individuals can easily interact with the model and gain insights into their heart health risk. This user-friendly interface is essential for ensuring that the benefits of advanced technologies reach a broad audience, including those with limited technical knowledge.

Additionally, the deployment of the model as an API allows for real-time predictions, making it a valuable tool for healthcare providers and individuals alike. The scalability of the API ensures that it can handle multiple concurrent requests, making it suitable for use in a clinical setting.

The rigorous testing process using Postman ensures the reliability and accuracy of the API, further enhancing its utility in real-world applications. By systematically testing the API with various scenarios, potential issues were identified and resolved, ensuring that the predictions made by the model are accurate and dependable.

Overall, the Heart Disease Prediction Model project sets a new standard for leveraging machine learning in healthcare. Its comprehensive approach, from model development to deployment and integration, demonstrates the transformative potential of advanced technologies in improving healthcare outcomes. As technology continues to evolve, projects like this will play a crucial role in driving innovation and revolutionizing healthcare delivery.

References

- https://www.who.int/

- https://www.kaggle.com/

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3594859/

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9319518/

- M. Ramla, S. Sangeetha and S. Nickolas, "Fetal Health State Monitoring Using Decision Tree Classifier from Cardiotocography Measurements," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 1799-1803, doi: 10.1109/ICCONS.2018.8663047.

- H. Khandoker, M. Wahbah, R. Al Sakaji, K. Funamoto, A. Krishnan and Y. Kimura, "Estimating Fetal Age by Fetal Maternal Heart Rate Coupling Parameters," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2020, pp. 604-607, doi: 10.1109/EMBC44109.2020.9176049.

- J. Kolarik, L. Soustek and R. Martinek, "Examination and Optimization of the Fetal Heart Rate Monitor : Evaluation of the effect influencing the measuring system of the Fetal Heart Rate Monitor," 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom), 2018, pp. 1-4, doi: 10.1109/HealthCom.2018.8531168.

- M. Wahbah, R. Al Sakaji, K. Funamoto, A. Krishnan, Y. Kimura and A. H. Khandoker, "Estimating Gestational Age From Maternal-Fetal Heart Rate Coupling Parameters," in IEEE Access, vol. 9, pp. 65369- 65379, 2021, doi: 10.1109/ACCESS.2021.3074550.

- J. Piri and P. Mohapatra, "Exploring Fetal Health Status Using an Association Based Classification Approach," 2019 International Conference on Information Technology (ICIT), 2019, pp. 166-171, doi: 10.1109/ICIT48102.2019.00036

- P. Dwivedi, A. A. Khan, S. Mugde and G. Sharma, "Diagnosing the major contributing factors in the classification of the fetal health status using cardiotocography measurements: An AutoML and XAI approach," 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2021, pp. 1-6, doi: 10.1109/ECAI52376.2021.9515033.

- S. Modi and M. H. Bohara, "Facial Emotion Recognition using Convolution Neural Network," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 1339- 1344, doi: 10.1109/ICICCS51141.2021.9432156.

- S. Mazumdar, R. Choudhary and A. Swetapadma, "An innovative method for fetal health monitoring based on artificial neural network using cardiotocography measurements," 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), 2017,pp. 265-268, doi: 10.1109/ICRCICN.2017.8234518.

- D. D. Fong et al., "Design and In Vivo Evaluation of a Non-Invasive Transabdominal Fetal Pulse Oximeter," in IEEE Transactions on Biomedical Engineering, vol. 68, no. 1, pp. 256-266, Jan. 2021, doi: 10.1109/TBME.2020.3000977.

- V. Shah and S. Modi, "Comparative Analysis of Psychometric Prediction System," 2021 Smart Technologies, Communication and Robotics (STCR), 2021, pp. 1-5, doi: 10.1109/STCR51658.2021.9588950.

M. B. I. Reaz and Lee Sze Wei, "An approach of neural network based fetal ECGextraction," Proceedings. 6th International Workshop on Enterprise Networking and Computing in Healthcare Industry - Healthcom 2004 (IEEE Cat. No.04EX842), 2004, pp. 57-60, doi: