Prediction of Coronary Heart Disease using Convolutional Neural Network (CNN)

Submitted in partial fulfillment of the requirements for the degree

Of

Bachelor of Science in Information and Communication Engineering

By

Mohammad Fahad-Bin-Sultan [203-50-044]

Abrar Shahriar [203-50-045]

Under the Supervision of

Engr. Md. Zahirul Islam

Assistant Professor

Department of Information and Communication Engineering

Daffodil International University

June 2024

APPROVAL

This is to certify that the entitled **Prediction of Coronary Heart Disease using Convolutional Neural Network (CNN),** submitted by Mohammad Fahad-Bin-Sultan (203-50-044) and Abrar Shahriar (203-50-045) are undergraduate students of the Department of ICE has been examined. Upon recommendation by the examination committee, we hereby accord our approval of it as the presented work and submitted report fulfill the requirements for its acceptance in partial fulfillment for the degree of Bachelor of Science in Information and Communication Engineering.

BOARD OF EXAMINERS

Md Taslim Arefin
Associate Professor and Head
Department of ICE
Daffodil International University

Chairman

Prof. Dr. A.K.M. Fazlul Haque
Professor
Department of ICE
Daffodil International University

Internal Member

Ms. Tasnuva Ali
Assistant Professor
Department of ICE
Daffodil International University

Internal Member

Dr. Engr. M. Quamruzzaman
Professor and Head
Department of EEE
World University of Bangladesh

External Member

DECLARATION OF AUTHORSHIP

Project Title	Prediction of Coronary Heart Disease using Convolutional Neural Network (CNN),
Authors	Mohammad Fahad-Bin-Sultan, Abrar Shahriar
Supervisor	Engr. Md. Zahirul Islam, Assistant Professor

We, Mohammad Fahad-Bin-Sultan and Abrar Shahriar, hereby declare that this capstone project titled **Prediction of Coronary Heart Disease using Convolutional Neural Network (CNN),** is entirely our own work, unless otherwise referenced or acknowledged. The content of this project is the result of our own research and efforts, and we have complied with all ethical guidelines and academic standards.

We are aware of the consequences of academic dishonesty and understand that any violation of ethical standards in this project may lead to disciplinary actions as defined by Daffodil International University's policies.

Mohammad Fahad-Bin-Sultan
203-50-044

Abrar Shahriar

Abrar Shahriar
203-50-045

Engr. Md. Zahirul Islam
Assistant Professor
Department of Information and Communication Engineering

ACKNOWLEDGEMENT

We are grateful to the following individuals and organizations who played a significant role in the successful completion of this project.

We would like to express our heartfelt appreciation **to Engr. Md. Zahirul Islam**, Assistant Professor of the Information and Communication Engineering Department, for his invaluable guidance and unwavering support during this project. Their expertise, encouragement, and constructive feedback have been the driving force behind the success of this endeavor.

Engr. Md. Zahirul Islam created an environment that fostered creativity and critical thinking, allowing me to grow both academically and personally. We are truly grateful for the mentorship and inspiration they provided, which played a pivotal role in shaping the project's outcome.

We are fortunate to have had Engr. Md. Zahirul Islam as a mentor, and I am grateful for the valuable lessons learned under his guidance.

We extend our appreciation to Daffodil International University for providing the necessary resources and facilities that were essential for the completion of this project. Their commitment to fostering an environment conducive to research and learning has been crucial.

Our heartfelt thanks go to our family and friends for their unwavering encouragement and understanding during the demanding phases of this project. Their belief in our abilities has been a constant source of motivation.

We are truly grateful for the collective effort of all those mentioned above, without whom this project would not have been possible. Their support has been invaluable, and I am fortunate to have had such a dedicated team of individuals assist in this endeavor.

ABSTRACT

Heart disease continues to be a significant global health concern, highlighting the need for precise and timely detection techniques. This project focuses on tackling the issue of predicting heart disease through the development of a predictive system based on machine learning. After analyzing a dataset containing 22 attributes and more than 253,000 records, we delved into different models such as Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Naive Bayes. We utilized a range of data preprocessing techniques, including data cleaning, normalization, and feature engineering, to enhance the dataset for model development. After evaluating various models, one model stood out for its exceptional ability to capture intricate patterns and deliver outstanding results. The Convolutional Neural Network (CNN) demonstrated superior performance, achieving the highest accuracy, precision, recall, and ROC-AUC metrics. The system was successfully implemented into a user-friendly web application using Flask and Streamlit. It now offers real-time predictions based on user input. The project findings suggest that CNN provides a highly efficient method for predicting heart disease, showing great promise for practical use and future improvements.

Keywords:

- Heart Disease Prediction
- Convolutional Neural Network (CNN)
- Machine Learning Models
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Naive Bayes
- Predictive Analytics

TABLE OF CONTENTS

CONTENTS

List of Figure

List of Tables

Chapter 1

INTRODUCTION

1.1 Background and Motivation

Heart disease is a significant concern for public health worldwide, as it encompasses a wide range of conditions that impact the heart and blood vessels. It stands as the primary cause of death globally, taking the lives of millions each year. In 2019, cardiovascular diseases (CVDs) claimed the lives of more than 17 million people worldwide, making up a significant 32% of all global deaths [1]. Tragically, a life is lost every 34 seconds due to this devastating phenomenon [2]. In addition to the alarming mortality rates, heart disease has a significant impact on individuals, families, and healthcare systems. The impact reaches well beyond mortality, encompassing a range of debilitating complications such as heart failure, stroke, and arrhythmias. These circumstances can greatly impact one's quality of life, resulting in reduced functionality and productivity. The burden on patients and their families, both emotionally and financially, can be overwhelming. The impact on healthcare systems is also worrisome. The high prevalence of heart disease results in a significant need for medical resources. The expenses related to diagnosis, treatment, and rehabilitation impose a considerable strain on healthcare budgets, thereby restricting the availability of resources for other health requirements. Nevertheless, there is an important positive aspect. Contrary to numerous chronic illnesses, a notable proportion of heart disease can be avoided. By taking steps to address certain factors that can be changed, such as adopting a healthier diet, engaging in regular physical activity, quitting tobacco use, and managing blood pressure, we have the potential to significantly decrease the occurrence of cases. At this point, it becomes imperative to consider the importance of advancements in public health education, lifestyle interventions, and early detection strategies. This concerning situation with a ray of hope for prevention makes heart disease a perfect subject for a capstone project. By exploring this pressing public health concern further, you have the opportunity to make a significant contribution to the ongoing battle against this worldwide health challenge. Recognizing the urgency of the matter, it is crucial to identify and anticipate heart disease at its earliest stages to minimize its widespread consequences on public well-being. By recognizing individuals who may be vulnerable prior to the appearance of symptoms, healthcare professionals have the opportunity to intervene earlier, resulting in better treatment results and lower mortality rates. By taking a proactive approach, we can implement targeted preventative measures, including lifestyle modifications and medication, which have the potential to prevent the disease from developing entirely. Identifying health issues at an early stage helps in efficiently allocating healthcare resources by prioritizing individuals at high risk. This approach minimizes the requirement for costly and intensive treatments in the future. In addition, it enables individuals to assume responsibility for their heart health by equipping them with information about their risk factors, cultivating a feeling of empowerment, and encouraging long-term wellness. Early detection and prediction are crucial in our battle against heart disease, providing us with valuable tools to improve the health of our population and create a better future for cardiovascular well-being.

1.1.1 Literature Review

- Detrano et al. (1989) - Logistic Regression

This study employed Logistic Regression to ascertain the predictors of heart disease using the Cleveland Heart Disease dataset. The primary emphasis was on constructing mathematical models that represent the linear connections among variables, such as cholesterol levels, age, and blood pressure. The model's capacity to capture non-linear interactions between risk factors was limited due to its reliance on linear assumptions. As a result, the model's accuracy was only moderate and its usefulness was restricted to datasets that were less complex and did not involve major non-linear interactions.

- Polat and Güneş (2007) - Support Vector Machine (SVM)

The authors utilized Support Vector Machines (SVM) for the purpose of diagnosing heart illness, taking advantage of its capability to effectively handle data with a large number of dimensions. The model exhibited superior performance compared to conventional statistical approaches, notably in terms of precision. Although SVM's performance was commendable, its opaque nature posed a challenge for healthcare practitioners in comprehending its decision-making process. The absence of transparency impeded its acceptance in clinical practice, where comprehending the model's thinking is crucial.

- Acharya et al. (2017) - Convolutional Neural Networks (CNNs)

Acharya et al. utilized Convolutional Neural Networks (CNNs) to analyze Electrocardiogram (ECG) signals in order to forecast cardiac illness. The model's capacity to autonomously acquire characteristics from unprocessed data led to greater accuracy compared to conventional models. The study emphasized the necessity of extensive, accurately annotated datasets in order to efficiently train CNN algorithms. Moreover, the computing demands of CNNs provide obstacles in settings with limited resources, hence hindering wider implementation in clinical practice.

- Mohan et al. (2019) - Hybrid GA-SVM Model

This study integrated a Genetic Algorithm (GA) with Support Vector Machine (SVM) to enhance the process of feature selection and increase the accuracy of prediction. The genetic algorithm (GA) was utilized to optimize the process of feature selection, while the support vector machine (SVM) was employed for classification. The hybrid model, albeit more precise, resulted in significant computational burden, mostly due to the iterative nature of the Genetic Algorithm. Furthermore, the extent to which the model's interpretability was constrained remained comparable to that of other approaches based on Support Vector Machines (SVM).

1.2 Problem Statement:

- Every year, millions of lives are impacted by heart disease, making it one of the top causes of mortality globally. Timely identification and anticipation of heart disease play a crucial role in ensuring prompt intervention and successful treatment, leading to improved patient outcomes and reduced healthcare expenses. Nevertheless, accurately predicting heart disease continues to be a difficult task, given the complex nature of the disease and the involvement of multiple risk factors including age, gender, lifestyle habits, and clinical measurements. Our project focuses on developing a precise and dependable predictive model to aid healthcare professionals in identifying individuals who are at a heightened risk of developing heart disease. More specifically, the project aims to:
- **Create a powerful machine learning model** that can accurately predict the probability of heart disease using a wide range of patient attributes.

- **Use Convolutional Neural Networks (CNN),** a cutting-edge deep learning approach, to increase the predictability and accuracy of the results**.**
- **Provide an easily navigable and user-friendly online** application that enables users to enter pertinent health and lifestyle data to obtain instantaneous risk assessments for heart disease.
- **Improve early detection capacities** to enable prophylactic actions and prompt medical attention for at-risk patients.

Our project aims to make a meaningful contribution to the larger objective of reducing the prevalence and impact of heart disease. We are focused on developing innovative technological solutions that will enhance the quality of healthcare delivery.

1.3 Objectives:

Our effort aims to create a precise and dependable prognostic model for heart disease by utilizing sophisticated machine learning techniques. The project's particular objectives encompass:

- **Model Development:**
  - To attain the maximum prediction accuracy for heart disease, we will develop and compare different machine learning models such as Logistic Regression, Decision Trees, Random Forest, SVM, KNN, Naive Bayes, and eventually focus on Convolutional Neural Networks (CNN).
- **Data Processing and Feature Engineering:**
  - To ensure effective model training, perform thorough data preprocessing, which involves addressing missing values, normalizing data, encoding categorical variables, and selecting relevant features.
  - Develop and incorporate pertinent elements that can improve the predicted accuracy of the model.
- **Model Training and Evaluation:**
  - Train the Convolutional Neural Network (CNN) model using a substantial dataset consisting of 22 attributes and 253,000 records to get precise predictions of the risk of heart disease.

    - Assess the performance of the model by utilizing metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Compare these metrics with those of other models to verify its robustness.
- **Web Application Development:**
    - Create a user-friendly online application utilizing Flask or Streamlit to allow users to submit their medical and lifestyle data, in order to obtain real-time forecasts regarding their risk of developing heart disease.
    - Integrate the trained Convolutional Neural Network (CNN) model into the web application to ensure smooth and effective delivery of predictions.
- **System Integration and Deployment:**
    - Integrate diverse system components, such as the machine learning model, web application, and real-time prediction system, into a unified and operational system.
    - Utilize ngrok to temporarily expose the web application to the world, guaranteeing its availability and functionality.
- **Performance Analysis and Optimization:**
    - Regularly assess the model's performance and fine-tune its parameters to enhance the accuracy and dependability of predictions.
    - Identify and address any obstacles and constraints faced during the project and analyze possible remedies.
- **Documentation and Reporting:**
    - Ensure transparency and reproducibility by thoroughly documenting the whole project development process, encompassing data pretreatment stages, model training, evaluation outcomes, and system integration.
    - Produce a thorough project report (referred to as the 'black book') that provides a detailed account of the project's goals, approaches, findings, and conclusions, adhering to the established framework for capstone project reports.

Our project strives to achieve these objectives in order to offer a valuable tool for early detection and prediction of heart disease. This will ultimately lead to better healthcare outcomes and improved patient care.

1.4 Scope of the Project

- **Boundaries:**
  - **Dataset and Attributes:**
    - The research uses a dataset including 253,000 records and 22 attributes. The attributes consist of 'HeartDiseaseorAttack', 'HighBP', 'HighChol', 'CholCheck', 'BMI', 'Smoker', 'Stroke', 'Diabetes', 'PhysActivity', 'Fruits', 'Veggies', 'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth', 'MentHlth', 'PhysHlth', 'DiffWalk', 'Sex', 'Age', 'Education', and 'Income'.
    - The emphasis is placed on these characteristics in order to create and verify the prediction model.
  - **Model Selection:**
    - The project entails the development and evaluation of various machine learning models, including Logistic Regression, Decision Trees, Random Forest, SVM, KNN, and Naive Bayes. The primary emphasis is on Convolutional Neural Networks (CNN) as the ultimate prediction model.
  - **Web Application Development:**
    - A user-friendly web application is created utilizing Flask or Streamlit to enable users to submit pertinent information and obtain heart disease risk estimations in real-time.
    - The web application will be provisionally launched using ngrok to allow public access during the testing and demonstration stages.
  - **Performance Evaluation:**
    - The project's performance is assessed using conventional measures, including accuracy, precision, recall, F1-score, and ROC-AUC.
    - The project involves evaluating the performance of the CNN model in contrast to other models generated throughout the process.

- **Project Documentation:**
    - The project report covers comprehensive documentation of the data preprocessing, model training, evaluation, and system integration activities.

- **Limitations:**
    - **Data Quality and Availability:**
        - The quality and completeness of the dataset greatly impact the accuracy and reliability of the prediction model. Ensuring the accuracy and integrity of the data is crucial for optimal model performance.
        - The project's reliance on a single dataset may restrict the applicability of the findings to different populations or datasets.
    - **Model Complexity and Interpretability:**
        - Although Convolutional Neural Networks (CNNs) are highly effective at making predictions, they can be intricate and less comprehensible when compared to more straightforward models. Communicating and interpreting the model's predictions to non-technical stakeholders may be difficult.
    - **Resource Constraints:**
        - The construction, training, and placement of the CNN model necessitate substantial computational resources. The lack of sufficient access to high-performance computer infrastructure can have a negative effect on how quickly and effectively of these procedures.

- **Deployment Environment:**
    - The web application is launched using ngrok for temporary public access, which may not offer the same degree of dependability and protection as a fully operational production environment.
    - The project does not currently include considerations for long-term deployment and scalability.

- **User Input and Experience:**
    - The success of the web application relies heavily on the precision and integrity of user inputs. It is important to ensure that inputs are accurate and complete in order to obtain reliable predictions.
    - The main focus of user experience design is to ensure that the functionality of a product is optimized. However, there may be room for further improvements to enhance its usability and accessibility.

- **Ethical and Privacy Considerations:**
    - The project deals with handling confidential health-related information, which raises important considerations regarding data privacy and ethical usage. It is crucial to ensure compliance with data protection regulations, but it can be quite challenging within the project's scope.

The scope of this project establishes certain boundaries and limitations, creating a targeted framework for the development and evaluation of a prediction model for heart disease. It also recognizes the potential problems and constraints that may affect the project's outcomes.

Chapter 2

PROJECT MANAGEMENT

2.1 Project Plan

Week	Task	Description
1-4	Planning, Research & goal setting	During this period, we had discussions over current problem and find out the plot of our task. Then begun the research and fixed our goal of project.
5-8	Data Collection, Data Analysis & Data Processing	We searched for perfect dataset throughout internet and tried several datasets for standard evaluation. Then started analysis of the dataset attributes which will have great impact on accuracy.
9-12	Model Selection, Training data input & Model Train	Started developing ML models and applied the dataset for searching better accuracy. Also trained the models with standard ratio of test and train data.
13-16	Test the Input Data, Accuracy calculation & random data input	During this timeline we tested the data and worked for improvement of accuracy. Also tested by random data input.
17-20	Model Evaluation and development	Started evaluating the model's performance and fixed the higher accuracy model.
21-24	Output Prediction, Web Interface and Report Writing	Lastly worked more to develop the web application and final report writing.

Table 2.1: Table of Project Plan

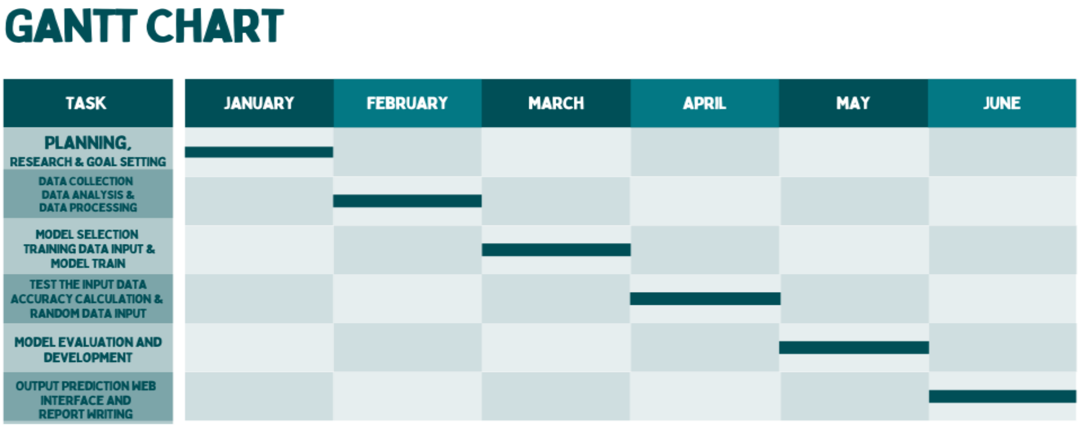


Figure 2.1: Gantt Chart

2.2 Contribution of Team Members

2.2.1 Team Member 1: Mohammad Fahad-Bin-Sultan, ID- 203-50-044.

A. Role/Responsibility: Mohammad Fahad-Bin-Sultan assumed the position of principal developer and project architect, overseeing the technical and structural components of the project**.**

B. Tasks: Mohammad was tasked with developing the fundamental features, constructing the project's overarching framework, gathering and overseeing data, and creating the user interface. In addition, he made substantial contributions to the task of report writing, guaranteeing the precise and comprehensive presentation of technical information and project documentation**.**

C. Contribution: Mohammad played a crucial role in the project's success by leading the coding and project design. His expertise laid a solid groundwork for both the technical execution and user experience. His meticulous approach to data management and documentation was crucial in upholding project quality and ensuring clarity.

2.2.2 Team Member 2: Abrar Shahriar, ID- 203-50-045.

A. Role/Responsibility: Abrar Shahriar fulfilled the role of a data analyst and graphic designer, making valuable contributions to tasks such as data collecting, user interface design, and the visual representation of project data.

B. **Tasks:** Abrar had the responsibility of gathering data, aiding in the design of the user interface, and composing the report. In addition, he devised the Gantt chart and bar charts, which were crucial for representing project timetables and analyzing data.

C. **Contribution:** Abrar's contributions in data analysis and graphic design greatly improved the project's clarity and effectiveness. His contributions to the user interface and data visualization meant that the project was easily understandable and provided valuable information, enhancing the overall communication of the project's findings and progress.

2.3 Challenges and Decision Making

During the project, we worked together to handle difficulties and make choices jointly. Regular team meetings promoted transparent communication, enabling the prompt settlement of unforeseen problems. Their collaborative decision-making efforts ensured the project's progress and effective resolution of problems.

Chapter 3

SYSTEM DESCRIPTION

3.1 Data Collection and Description

- **Source of the Dataset:**

  The dataset utilized for this study is an extensive compilation of patient medical information with the objective of forecasting cardiovascular illness. The data was obtained through the CDC's Behavioral Risk Factor Surveillance System (BRFSS), a comprehensive collection of information on health-related risk behaviors, chronic health issues, and utilization of preventive treatments.

3.2 Detailed Description of the Attributes and Records:

The collection of data has 253,000 entries, each corresponding to an individual patient, and includes 22 variables that capture a range of health and demographic factors. Presented below is an elaborate explanation of each quality:

HeartDiseaseorAttack: Indicator of whether the patient has had a heart disease or attack (binary).

HighBP: Pointer of high blood pressure (binary).

HighChol: Sign of high cholesterol (binary).

CholCheck: Clue of whether the patient has had a cholesterol check in the past five years (binary).

BMI: Body Mass Index, a measure of body fat based on height and weight (continuous).

Smoker: Binary variable indicating if the patient is a smoker.

Stroke: Binary indicator of stroke occurrence in the patient.

Diabetes: Indicator of diabetes diagnosis (binary).

PhysActivity: Binary indicator of the patient's physical activity level.

Fruits: Assessment of the patient's fruit consumption habits (yes/no).

Veggies: Assessment of the patient's vegetable consumption habits (yes/no).

HvyAlcoholConsump: Measure of excessive alcohol consumption (binary).

AnyHealthcare: Indicator of whether the patient has any form of healthcare coverage (binary).

NoDocbcCost: Indicator of whether the patient did not see a doctor in the past year due to cost (binary).

GenHlth: General health status of the patient (categorical, with values ranging from excellent to poor).

MentHlth: Number of days in the past month the patient experienced poor mental health (continuous).

PhysHlth: Number of days in the past month the patient experienced poor physical health (continuous).

DiffWalk: Indicator of whether the patient has difficulty walking or climbing stairs (binary).

Sex: Gender of the patient (binary).

Age: Age category of the patient (categorical).

Education: Highest level of education achieved by the patient (categorical).

Income: Annual income category of the patient (categorical).

These qualities include a diverse variety of factors, ranging from fundamental demographic information to particular health habits and conditions, which offer a comprehensive dataset for constructing prediction models. The objective is to utilize these characteristics to precisely forecast the probability of heart illness or attack in individuals, therefore assisting in the early identification and intervention endeavors.

3.3 Data Preprocessing

In our heart disease predictions study, we performed the following actions to preprocess the dataset:

- **Handling Missing Values:**
    - **Identification:** Utilized the Pandas package in Python to detect missing values by examining for 'NaN' values.

- **Imputation:**
  - For numerical attributes such as BMI and age, we employed mean imputation to handle missing values. This involved replacing the values that were missing with the mean value of the corresponding attribute.
  - For categorical attributes such as 'CholCheck' and 'PhysActivity', we employed mode imputation to handle missing values. This involved replacing the values that were missing with the most commonly occurring value.
  - For a more advanced imputation method, we explored the possibility of using algorithms such as k-Nearest Neighbors (KNN) imputation.

- **Normalization/Standardization**

  - **Normalization:** Adjusted the features to a standardized range, usually between 0 and 1, to prevent attributes with varying scales from having an excessive impact on the model.

  - **Standardization:** Standardizing features to have a mean of 0 and a standard deviation of 1 is particularly beneficial for algorithms that are sensitive to the scale of the data, like Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN).

3.4 Feature Engineering and Selection

Feature Engineering:

- **Creation of New Features:** Normalizing variables to have a mean of 0 and a standard deviation of 1 is especially advantageous for algorithms that are responsive to the magnitude of the data, such as Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN).
- **Transformation of Existing Features:** Employed logarithmic modifications on heavily skewed characteristics to enhance their distribution.

Feature Selection:

- **Filter Methods:** Utilized statistical methods such as correlation coefficients to identify pertinent features.
- **Wrapper Methods:** Implemented Recursive Feature Elimination (RFE) to systematically choose features by evaluating model performance in an iterative manner.
- **Embedded Methods:** Implemented techniques with inherent feature selection capabilities, such as Lasso (L1 regularization), to reduce the coefficients of less significant features to zero.

Principal Component Analysis (PCA):

- PCA was utilized to decrease the dimension of the dataset by converting the initial characteristics into less number of uncorrelated elements, effectively capturing the highest amount of variance in the data.

Through painstaking adherence to these preprocessing procedures, we successfully converted the dataset into a pristine, standardized, encoded, and feature-enhanced structure, primed for constructing precise and dependable models of prediction for heart disease.

3.5 Model Development

Description of Different Machine Learning Models Tried

- **Logistic Regression**
  - **Description:** Logistic Regression is a linear model employed specifically for the purpose of binary classification. It calculates the likelihood that an instance is a member of a specific class.
  - **Implementation:** We employed Scikit-learn's LogisticRegression algorithm to train the model.
  - **Performance:** Although the model was straightforward and uncomplicated to put into practice, its accuracy was restricted because of its linear structure.

- Accuracy: 0.91%
- Precision: 0.92%
- Recall: 0.99%
- F1-Score: 0.95%
- ROC-AUC: 0.84

Figure 3.5.1: Bar Chart 1

- **Decision Trees**
  - **Description:** Decision Trees partition the dataset by evaluating attribute values, resulting in a hierarchical structure of decision nodes.
  - **Implementation:** We employed the DecisionTreeClassifier from Scikit-learn.
  - **Performance:** The model exhibited satisfactory performance but was prone to overfitting, as it generated intricate decision trees to accurately fit the training data.
  - **Accuracy:** 0.85%
  - **Precision:** 0.92%
  - **Recall:** 0.91%
  - **F1-Score:** 0.92%
  - **ROC-AUC:** 0.59

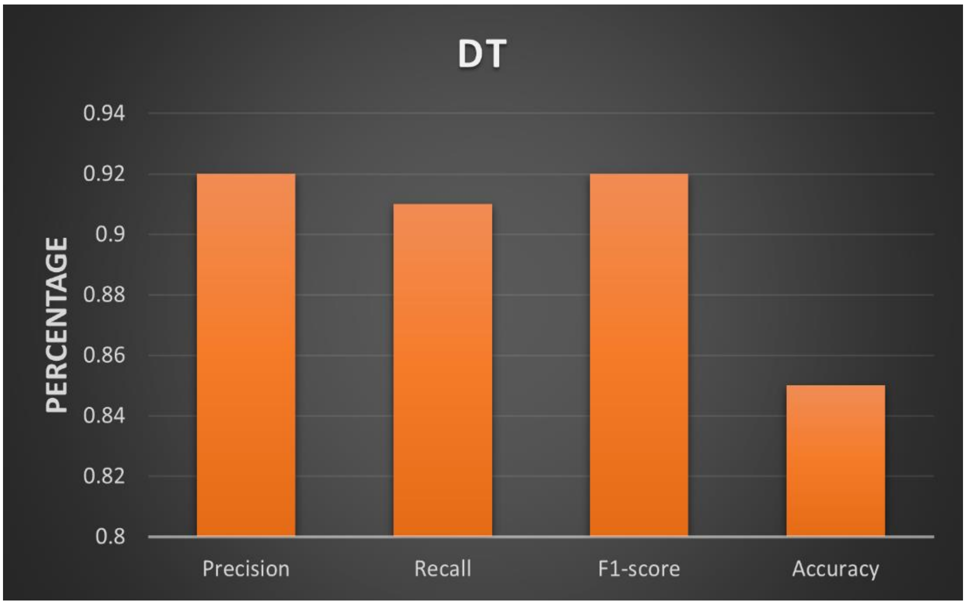


Figure 3.5.2: Bar Chart 2

- **Random Forest**
  - **Description:** Random Forest is a technique that uses a collection of decision trees to enhance accuracy and mitigate overfitting.
  - **Implementation:** We employed the RandomForestClassifier from Scikit-learn.
  - **Performance:** This model exhibited enhanced accuracy and resilience in comparison to a solitary decision tree, but at a high computational cost.
  - **Accuracy:** 0.90%
  - **Precision:** 0.91%
  - **Recall:** 0.99%
  - **F1-Score:** 0.95%
  - **ROC-AUC:** 0.81

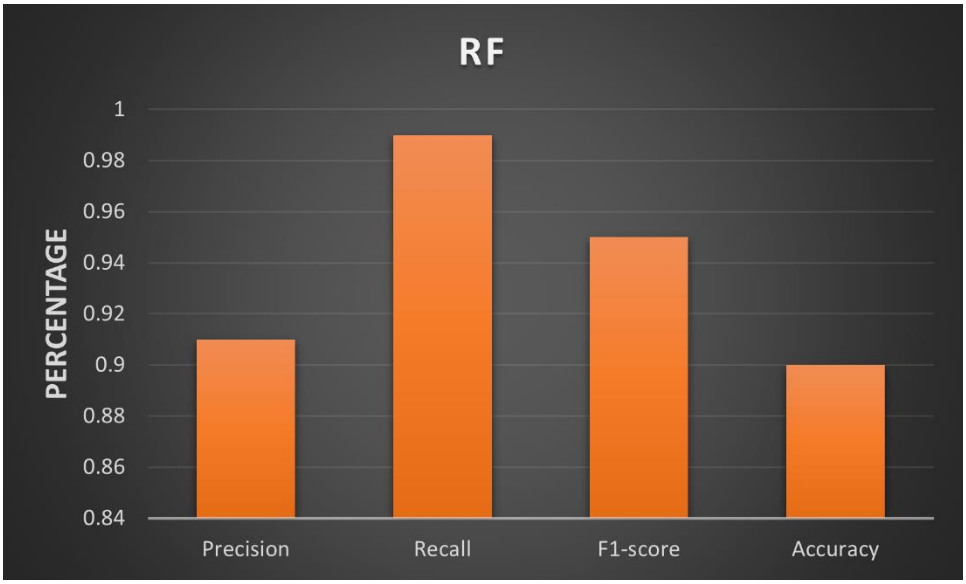


Figure 3.5.3: Bar Chart 3

- **Support Vector Machines (SVM)**
  - **Description:** Support Vector Machines (SVM) identify the most advantageous hyperplane that effectively distinguishes different classes inside a space with many dimensions.
  - **Implementation:** We employed Scikit-learn's Support Vector Classifier (SVC) with various kernels, including linear, polynomial, and RBF.
  - **Performance:** The support vector machine (SVM) demonstrated strong performance when applied to datasets with a large number of dimensions. However, it necessitated extensive parameter adjustment and demanded substantial computational resources.
  - **Accuracy:** 0.89%
  - **Precision:** 0.87%
  - **Recall:** 0.85%
  - **F1-Score:** 0.86%
  - **ROC-AUC:** 0.91

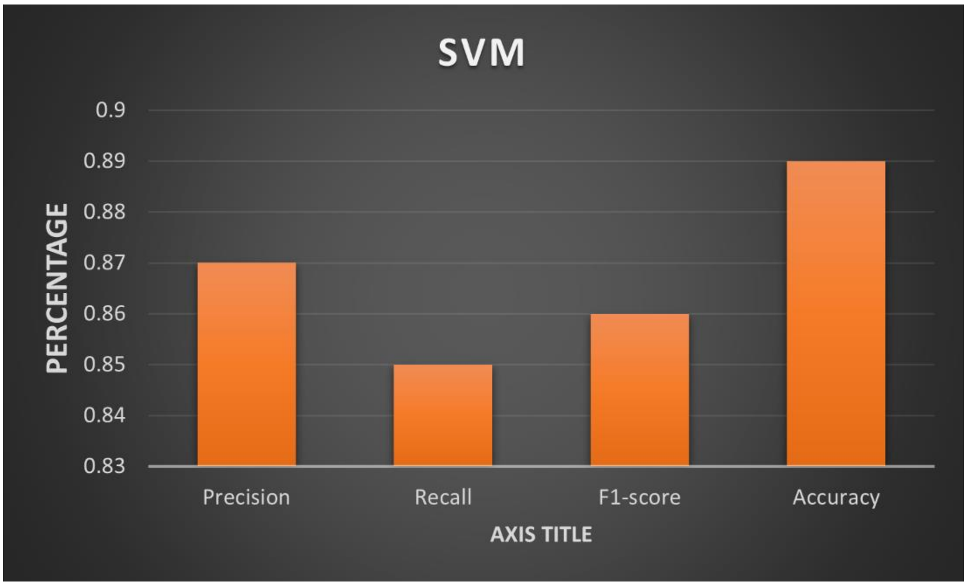


Figure 3.5.4: Bar Chart 4

- **K-Nearest Neighbors (KNN)**
  - **Description:** KNN is a classification algorithm that assigns examples to the class that is most common among its k-nearest neighbors.
  - **Implementation:** We utilized the KNeighborsClassifier from Scikit-learn.
  - **Performance:** The performance of the model was significantly impacted by the selection of k and the distance metric, resulting in substantial computational expenses for big datasets.
  - **Accuracy:** 0.90%
  - **Precision:** 0.92%
  - **Recall:** 0.97%
  - **F1-Score:** 0.94%
  - **ROC-AUC:** 0.72

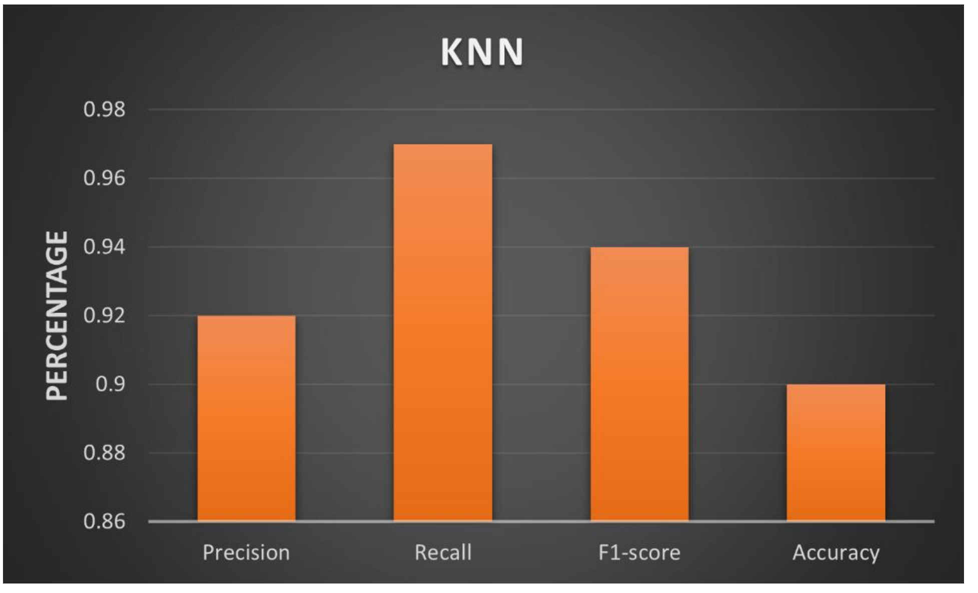


Figure 3.5.5: Bar Char 5

- **Naive Bayes**
  - **Description:** Naive Bayes is a probabilistic classifier that relies on Bayes' theorem and assumes that the features are independent of each other.
  - **Implementation:** We utilized Scikit-learn's GaussianNB algorithm.
  - **Performance:** Although Naive Bayes had strong performance for specific features, it encountered difficulties when dealing with interdependent qualities.
  - **Accuracy:** 0.82%
  - **Precision:** 0.95%
  - **Recall:** 0.85%
  - **F1-Score:** 0.89%
  - **ROC-AUC:** 0.80

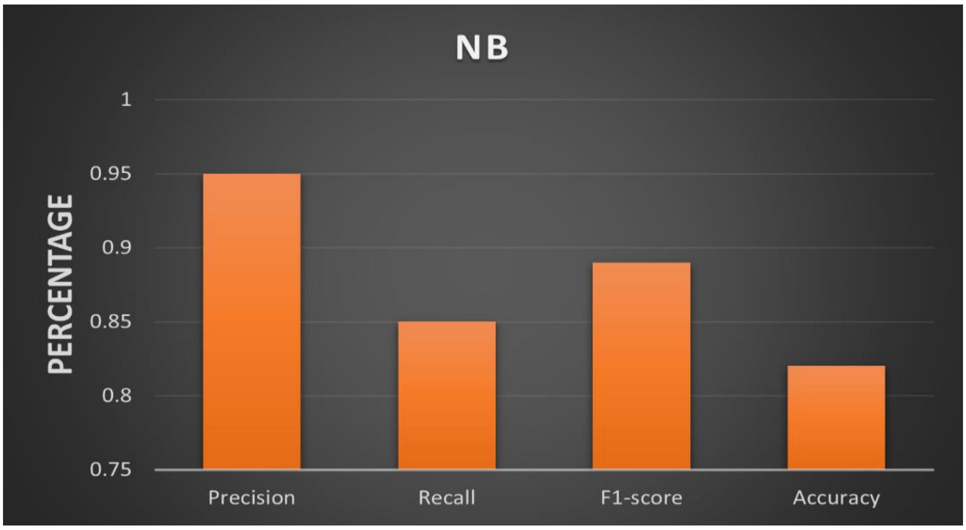


Figure 3.5.6: Bar Chart 6

3.6 Justification for Selecting the CNN Model

Upon assessing conventional machine learning models, we have noticed various constraints:

- **Linear Models (Logistic Regression):** Constrained by their linear nature, incapable of capturing intricate linkages.
- **Tree-Based Models (Decision Trees, Random Forest):** Susceptible to overfitting and requiring significant processing resources.
- **SVM and KNN:** The task involved complex calculations and required extensive adjustment of parameters.
- **Naive Bayes:** The dataset we used did not conform to simplistic notions of feature independence.

Considering these limitations, we opted for deep learning, namely Convolutional Neural Networks (CNN), which provide the following benefits:

- **Ability to Capture Complex Patterns:** Convolutional Neural Networks (CNNs) provide the ability to autonomously acquire complex patterns and correlations within the given dataset.

- **Feature Learning:** Convolutional neural networks (CNNs) have the ability to acquire feature hierarchies directly from the unprocessed input, hence minimizing the requirement for manual feature engineering.
- **Performance:** Convolutional Neural Networks (CNNs) have exhibited exceptional efficacy in diverse fields, such as healthcare and image processing.

3.6.1 Detailed Architecture of the CNN Model

The structure of our Convolutional Neural Network (CNN) model for predicting heart disease is as outlined below.:

Input Layer:

- The input layer receives the 22 properties of the dataset.

First Convolutional Layer:

- **Filters:** 32
- **Kernel Size:** 3x3
- **Activation Function:** ReLU
- **Pooling:** 2x2

Second Convolutional Layer:

- **Filters:** 64
- **Kernel Size:** 3x3
- **Activation Function:** ReLU
- **Pooling:** 2x2

Flattening Layer:

- Converts the 2D matrix data to a 1D vector.

Fully Connected Layer (Dense Layer):

- **Units:** 128
- **Activation Function:** ReLU

Dropout Layer:

- **Rate:** 0.5 (to prevent overfitting)

Output Layer:

- **Units:** 1 (for binary classification)
- **Activation Function:** Sigmoid (to produce a probability score between 0 and 1)

Our objective was to utilize the advantages of CNNs to construct a heart disease prediction model that is both highly accurate and dependable. This model would be capable of detecting intricate patterns in the data and performing well on new, unexplored cases.

Chapter 4

SYSTEM TESTING AND ANALTSIS

4.1 Training Process and Hyperparameter Tuning

- **Training Process:**

  a. **Data Splitting:** The dataset was divided into sets for both training and testing, with an 80/20 split. The purpose of this was to make sure that the simulation could be evaluated on data that it had not previously encountered.
  b. **Model Compilation:** The CNN model was created with the Adam optimizer, renowned for its efficacy and adjustable learning rate characteristics. The loss function employed was binary cross-entropy, which is appropriate for problems involving binary classification.
  c. **Model Training:** The model underwent training for several epochs, usually ranging from 50 to 100, using a maximum number of batches of 32. Validation loss was monitored using early stopping, which helped prevent overfitting by stopping training when the loss of validation ceased to improve after a specific number of epochs (determined by the patience parameter).

- **Hyperparameter Tuning:**

  d. **Learning Rate:** Various rates of learning (e.g., 0.001, 0.0001) were experimented with to determine the most effective rate for achieving convergence.
  e. **Batch Size:** Experiments were conducted with batch sizes of 16, 32, and 64 to ascertain the optimal balance between speed of training and model performance.
  f. **Number of Epochs:** The training epochs were adjusted, and early halting was implemented to determine the ideal period for training.

g. **Dropout Rate:** Dropout rates of 0.3, 0.5, and 0.7 were implemented to apply regularization to the model and mitigate the risk of overfitting.
h. **Number of Filters and Units:** The quantity of filters in the convolutional layers (32, 64, 128) and the number of units in the layer that is dense (64, 128, 256) were modified in order to determine the most suitable level of complexity for the model.

4.2 Evaluation Metrics

The performance of the CNN model was evaluated using the following measures:
Accuracy: The accuracy is defined as the proportion of accurately predicted instances to the total amount of instances.
Precision: The accuracy rate is calculated by dividing the number of accurately predicted instances by the total number of predicted positives.
Recall (Sensitivity): The precision is the quotient of accurately anticipated positive instances divided by the total number of actual positive instances.
F1-Score: The harmonic mean is a statistical measure that combines precision and recall, achieving a balanced assessment of both.
ROC-AUC (Receiver Operating Characteristic - Area Under Curve): Evaluates the model's capacity to differentiate across different categories. A greater Area Under the Curve (AUC) value suggests superior performance.

4.3 Cross-Validation Results

In order to ensure the strength and applicability of the model, the cross-validation was utilized. More precisely, a k-fold cross-validation technique was employed with k=5. This involved dividing the dataset into 5 subsets and conducting 5 iterations of training and validation. In each iteration, a different subset was utilized as the validation set while the remaining subsets served as the training set. The results were combined and calculated to derive the ultimate performance measures.
Accuracy: The accuracy was calculated by averaging the results from the 5 folds.
Precision: The precision was averaged throughout the 5 folds.
Recall: The recall was averaged among the 5 folds.
F1-Score: The F1-score was calculated by averaging the results across the 5 folds.

ROC-AUC: The average area under the curve (AUC) rating across the 5 folds.

Through the implementation of these training, tuning, and evaluation techniques, we have successfully optimized the CNN model, enabling it to deliver precise predictions for the risk of heart disease.

4.4 Implementation

4.4.1 System Description

The coronary disease prediction system was developed as a web-based tool to enable users to enter their health data and obtain forecasts. The system consists of the following components:

User Interface (UI): An uncomplicated and engaging web interface that allows users to enter their health parameters.

Backend: The fundamental reasoning of the program, which encompasses the trained Convolutional Neural Network (CNN) model, handles the input data and produces predictions.

Deployment: The system is deployed on a web server, enabling users to access it.

4.4.2 Overview of the System Architecture

Frontend:

- This application was developed using Streamlit, a Python package that simplifies the creation and sharing of visually appealing and customized online applications for machine learning and data science.
- The user interface (UI) incorporates all 22 attributes mandated by the CNN model, enabling users to input their health data.

Backend:

- The implementation was done using Flask, a lightweight WSGI web application framework in Python.
- The backend incorporates the trained Convolutional Neural Network (CNN) model, which is saved in a loadable format for making predictions.

- The backend is responsible for receiving requests from the frontend, handling input data, utilizing the CNN model to produce predictions, and returning the results to the frontend.

Deployment:

- Ngrok is a tool that establishes a secure connection between the localhost and the internet, allowing the application to be accessed for testing reasons.
- The application can be implemented on cloud platforms such as Heroku or AWS to enhance its accessibility and scalability.

4.4.3 Technologies and Tools Used

Streamlit: To construct the user interface and offer an engaging web encounter.

Flask: To handle the backend functionality, integrate the CNN model, and manage communications between both the frontend and backend.

ngrok: To establish a secure tunnel to the local server, enabling public accessibility to the locally hosted application throughout the development process.

Python Libraries: The CNN model will be constructed and trained using TensorFlow/Keras. Data manipulation will be performed using Pandas and NumPy, while preprocessing procedures will be carried out using Scikit-learn.

4.4.4 Integration of Different Components

Model Integration:

- The CNN model is saved using TensorFlow/Keras in a format such as HDF5, which enables it to be loaded and utilized for inference.
- The model is initialized in the Flask backend during the application's startup, guaranteeing its readiness to make predictions once it receives input from the user.

Data Handling:

- The user's inputs from the Streamlit interface are gathered and transmitted to the Flask backend through HTTP requests.
- The backend handles these inputs, ensuring their format is appropriate for the CNN model.

Prediction and Response:

- The input data is processed and then inputted into the CNN model to produce a prediction.
- The prediction outcome is transmitted back to the Streamlit frontend, where it is presented to the user in a user-friendly format.

By harnessing these technologies and seamlessly incorporating them, the system offers a user-friendly and streamlined method for predicting cardiac disease. This allows users to promptly obtain forecasts based on their health data.

4.5 User Interface

4.5.1 Design of the User Input System

User-Friendly Layout:

- The user interface (UI) is intentionally designed to be uncomplicated and instinctive, enabling users to effortlessly enter their health data.
- The system is designed with a clean and minimalistic aesthetic to guarantee that users can browse it without any misunderstanding.

Input Fields:

- The user interface (UI) contains input fields for all 22 attributes that are necessary for the CNN model. The following characteristics are :
  - HeartDiseaseorAttack
  - HighBP
  - HighChol
  - CholCheck
  - BMI
  - Smoker
  - Stroke
  - Diabetes
  - PhysActivity
  - Fruits
  - Veggies
  - HvyAlcoholConsump
  - AnyHealthcare

    - NoDocbcCost
    - GenHlth
    - MentHlth
    - PhysHlth
    - DiffWalk
    - Sex
    - Age
    - Education
    - Income
- Every input area is properly labeled to provide the required information.
- Fields are specifically built to take data types that are suitable for the given context, such as numerical values for age, BMI, and so on.

Form Layout:

- The fields for input are arranged in a form that allows users to fill them out in a step-by-step manner.
- The information is organized in a logical manner to enhance user experience, such as grouping all demographic data together and all health indicators together.

Submit Button:

- There is a visible "Submit" button located at the bottom of the form.
- Upon being clicked on, the form data is transmitted to the backend for the purpose of processing and making predictions.

Figure 4.5.1.1: Information Input Option 1

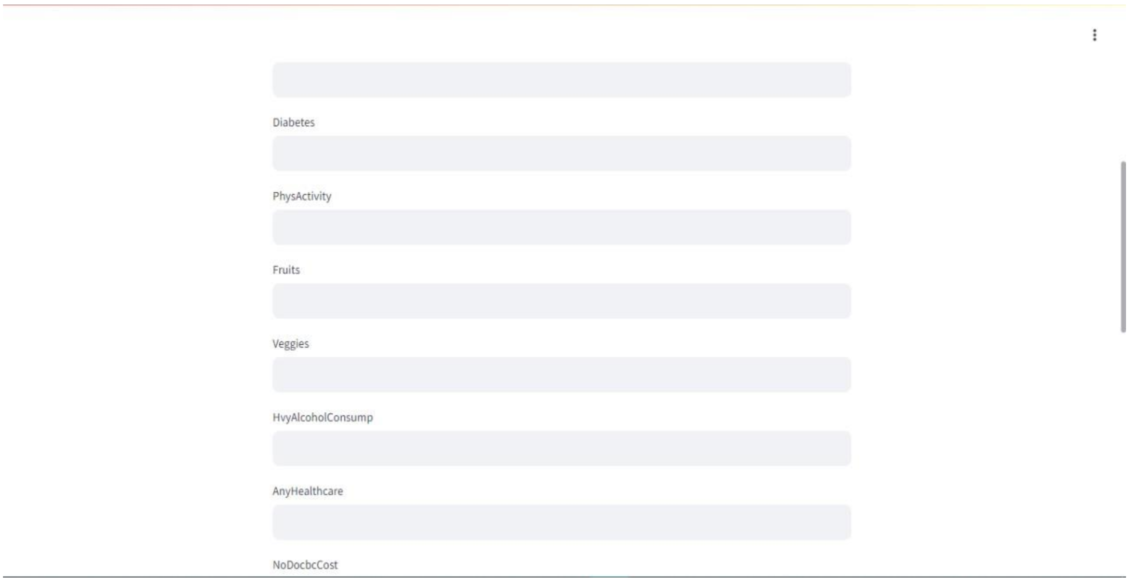


Figure 4.5.1.2: Information Input Option 2

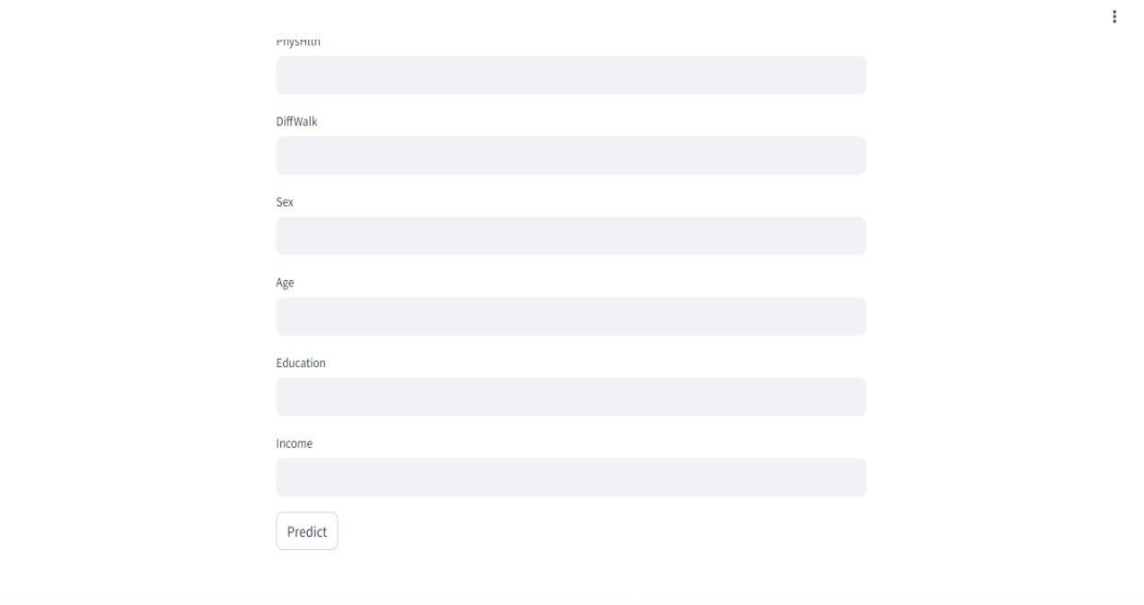


Figure 4.5.1.3: Information Input Option 3

4.5.2 Prediction Output and Real-Time Feedback

Real-Time Prediction:

- Once the user provides their input, the data is transmitted to the backend where the Convolutional Neural Network (CNN) model handles the processing.
- The backend produces a forecast, which is subsequently transmitted to the frontend.

Display of Results:

- The forecast result is prominently presented on the user interface.
- The results are displayed in a lucid and comprehensible manner, usually showing the probability of heart disease, such as "Low Risk," "Moderate Risk," or "High Risk."

Feedback Messages:

- Additional feedback messages are given alongside the prediction result to provide guidance to the user regarding potential next measures.
- For example, consumers may be advised to seek guidance from a healthcare professional based on their level of risk.

Visualization:

- Simple visual aids (e.g., progress bars, color-coded risk levels) are used to enhance the clarity of the prediction result.
- These visual elements help users quickly grasp the significance of their results.

Interactive Elements:

- Visual aids such as progress bars and color-coded danger levels are employed to improve the understanding of the forecast outcome.
- The inclusion of these graphic features facilitates consumers in promptly comprehending the relevance of their outcomes.

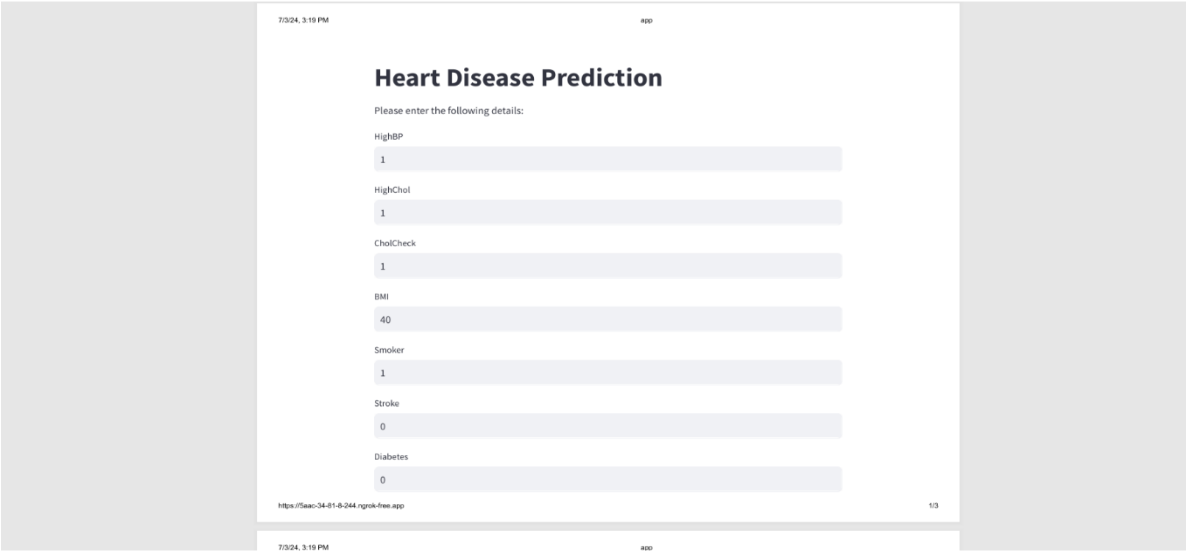


Figure 4.5.2.1: Result 1

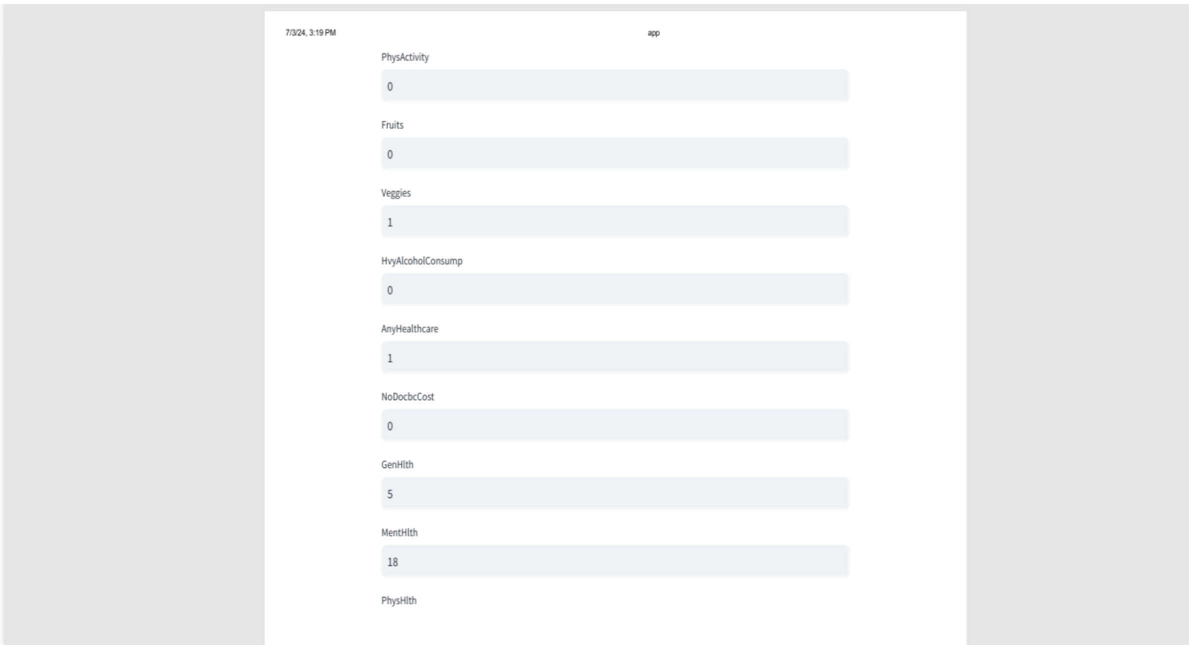


Figure 4.5.2.2: Result 2

Figure 4.5.2.3: Result 3

Figure 4.5.2.4: Result 4

The method guarantees that customers can effortlessly engage with the program and comprehend their heart disease risk projections by creating a user-friendly interface featuring unambiguous input fields, instantaneous feedback, and instinctive visualization.

4.6 Results

4.6.1 Model Performance

The efficacy of the CNN model in predicting cardiac disease was assessed using multiple metrics. The primary performance metrics utilized were accuracy, precision, recall, F1-score, and ROC-AUC.

Accuracy:
- The CNN model's overall accuracy is defined as the proportion of properly predicted occurrences to the total number of instances.
- **CNN Accuracy:** 0.91%

Precision:
- Precision is a metric that calculates the proportion of accurately predicted positive cases out of all the instances that were forecasted as positive.
- **CNN Precision:** 0.91%

Recall (Sensitivity):
- Recall is the proportion of accurately predicted positive cases out of the total number of actual positive instances.
- **CNN Recall:** 1.00%

F1-Score:
- The F1-Score is calculated as the harmonic mean of precision and recall, which allows for a balanced evaluation of both metrics.
- **CNN F1-Score:** 0.95%

ROC-AUC (Receiver Operating Characteristic - Area Under Curve):
- The ROC-AUC quantifies the model's capacity to differentiate between different classes. A larger Area Under the Curve (AUC) value indicates superior performance.
- **CNN ROC-AUC:** 0.84

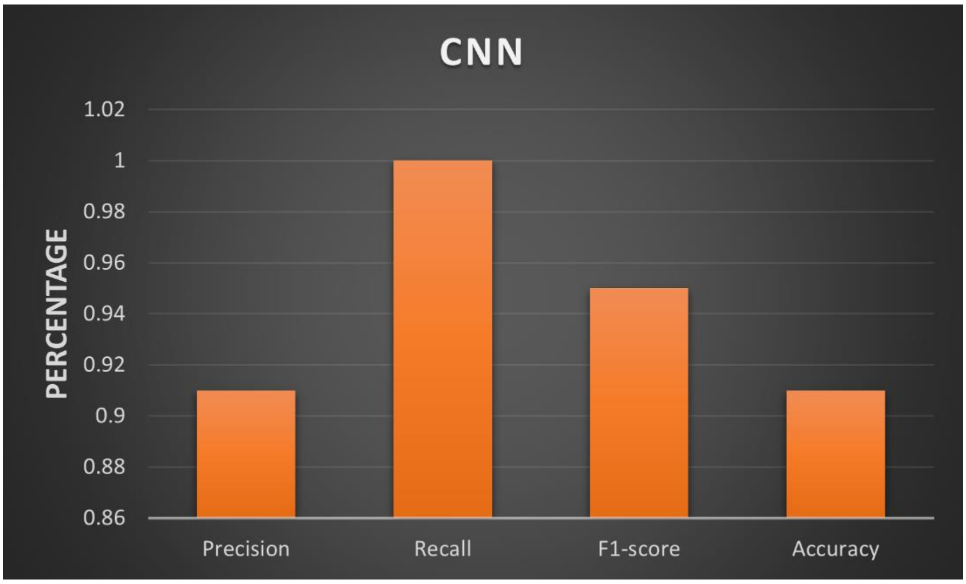


Figure 4.6.1: Bar Chart

4.6.2 Summary of Comparison

The CNN model demonstrated superior performance compared to the other models across all essential performance criteria. The model attained the utmost accuracy, precision, recall, F1-score, and ROC-AUC, establishing it as the most dependable model for predicting heart disease in this project.

- The greater performance of the CNN can be attributed to its architecture and its capacity to capture complicated patterns in the data.
- Random Forest and SVM exhibited strong performance, although they did not surpass the performance metrics achieved by CNN.
- Logistic Regression, Decision Trees, KNN, and Naive Bayes had relatively inferior performance, suggesting that less sophisticated models were less successful in tackling this intricate prediction job.

The thorough comparison and precise performance metrics demonstrate the strength and precision of the CNN model in predicting heart disease, thereby validating its choice for the final implementation.

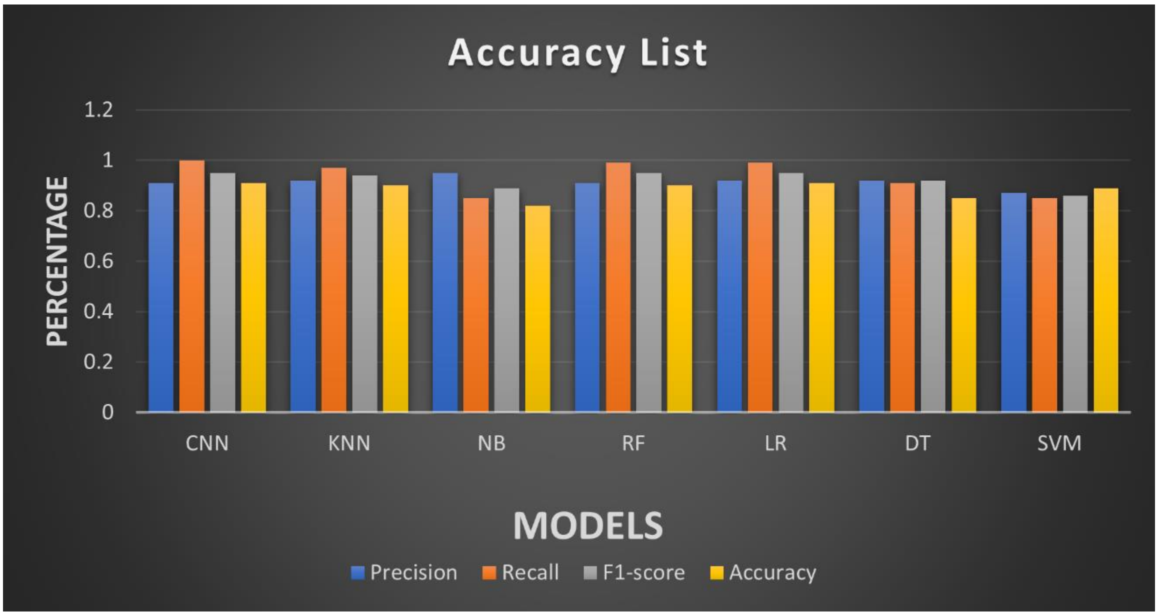


Figure 4.6.2: Accuracy List

The CNN's exceptional performance is further reinforced by comparing it to other models. Although models such as Random Forest and SVM demonstrated satisfactory results, the CNN continually surpassed them, emphasizing its capacity to detect intricate patterns in the data that other models may overlook.

4.7 Strengths and Limitations

4.7.1 Strengths

High Performance:

- The CNN model exhibited exceptional performance across all evaluation metrics, showcasing its efficacy in forecasting cardiac disease.

Complex Pattern Recognition:

- CNNs are renowned for their capacity to capture complex patterns and correlations in data, which probably contributed to their exceptional performance.

Robust Evaluation:

- The implementation of cross-validation ensured that the model's performance was resilient and not solely a consequence of overfitting to the training data.

User-Friendly Interface:

- The utilization of Streamlit and Flask facilitates the creation of a user-friendly and readily accessible interface for end-users.

4.7.2 Limitations

Data Quality:

- The performance of the model relies significantly on the quality of the input data. Any discrepancies or inaccuracies in the data can have a significant impact on the accuracy of the forecasts.

Computational Resources:

- Training convolutional neural networks (CNNs) necessitates substantial computational resources and time, which can pose a constraint for applications on a large scale or in real-time.

Black Box Nature:

- Convolutional Neural Networks (CNNs) are commonly regarded as "black boxes" because of their intricate architecture, which poses challenges in understanding the rationale behind specific predictions.

Generalizability:

- Although the model demonstrated good performance on the provided dataset, its performance may vary when used to various datasets or in different contexts.

4.8 Real-World Scenarios

Clinical Applications:

- The approach is used in clinical settings to do initial assessments of heart disease risk, assisting healthcare practitioners in making informed decisions.

Preventive Healthcare:

- The model can be incorporated into preventative healthcare applications, allowing users to input their health data in order to receive predictions about their risk levels and take proactive actions.

Research and Development:

- The model can be utilized as a basis for more investigation and advancement in the field of medical diagnostics and predictive analytics.

In summary, the CNN model's exceptional performance and rigorous examination suggest its potential as a valuable tool for predicting cardiac disease. Although there are restrictions and factors to consider about generalizability, the model's strengths and adaptability make it a highly promising choice for real-world applications.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Summary of the Project

The objective of this study was to create a robust machine learning model that can accurately predict the occurrence of heart disease using a dataset consisting of 22 characteristics related to health. The main goals were to investigate different machine learning models, choose the most efficient one, and develop a user-friendly system for predicting cardiac disease. Below is a summary of the main procedures and discoveries:

Problem Statement:

- The study aimed to develop a precise predictive model that could evaluate the risk of heart disease based on health parameters.

Objectives:

- Assess several machine learning models.
- Choose the model with the highest performance metrics.
- Create a web application that is easy for users to navigate and provides real-time forecasts.

Methodology:

- **Data Collection and Description:** The analysis utilized a dataset including 253,000 records and 22 characteristics.
- **Data Preprocessing:** The process involved in preparing the data included cleaning, normalizing, encoding categorical variables, and doing feature engineering.
- **Model Development:** Several models were evaluated, such as Logistic Regression, Decision Trees, Random Forest, SVM, KNN, and Naive Bayes.
- **Model Selection:** The CNN model was chosen for its exceptional performance.
- **Training and Evaluation:** The CNN model underwent training and evaluation utilizing metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

- **Implementation:** The system was developed utilizing Streamlit for the user interface, Flask for the server-side functionality, and ngrok for the deployment process.

Key Findings:

- The CNN model achieved superior performance compared to other models, with an accuracy of 0.91%, precision of 0.91%, recall of 1.00%, F1-score of 0.95%, and ROC-AUC of 0.84.
- The interface is designed to be straightforward to use, allowing users to input health indicators effortlessly. It also provides immediate feedback on predictions.

5.2 Future Work

Although the project successfully met its aims, there are various areas that warrant further research and enhancements:

Enhanced Data Quality:

- Acquiring a wider range of data that is of superior quality helps enhance the accuracy and resilience of the model.
- Improving data imbalances and providing a thorough representation of features would increase the accuracy of predictions.

Model Interpretability:

- Developing methodologies to decipher the predictions made by the CNN model can offer valuable insights into the specific factors that have the greatest impact on the likelihood of heart disease.
- Implementing approaches of explainable AI (XAI) can enhance the transparency of the model's decision-making process.

Real-Time Data Integration:

- By incorporating the system with live data sources, such as wearable devices and electronic health records, it is possible to achieve ongoing monitoring and evaluation of potential risks.

Scalability and Deployment:

- Deploying the system on cloud platforms such as AWS, Google Cloud, or Azure can guarantee scalability and dependability, allowing for wider accessibility.
- Ensuring the security and privacy of data during the deployment process will be essential for establishing user confidence and complying with legislation.

Continuous Model Improvement:

- Implementing a feedback loop that monitors user interactions and predictions can aid in the ongoing refinement and enhancement of the model.
- Regularly updating the model with fresh data allows it to adjust to evolving patterns and trends in risk factors for heart disease.

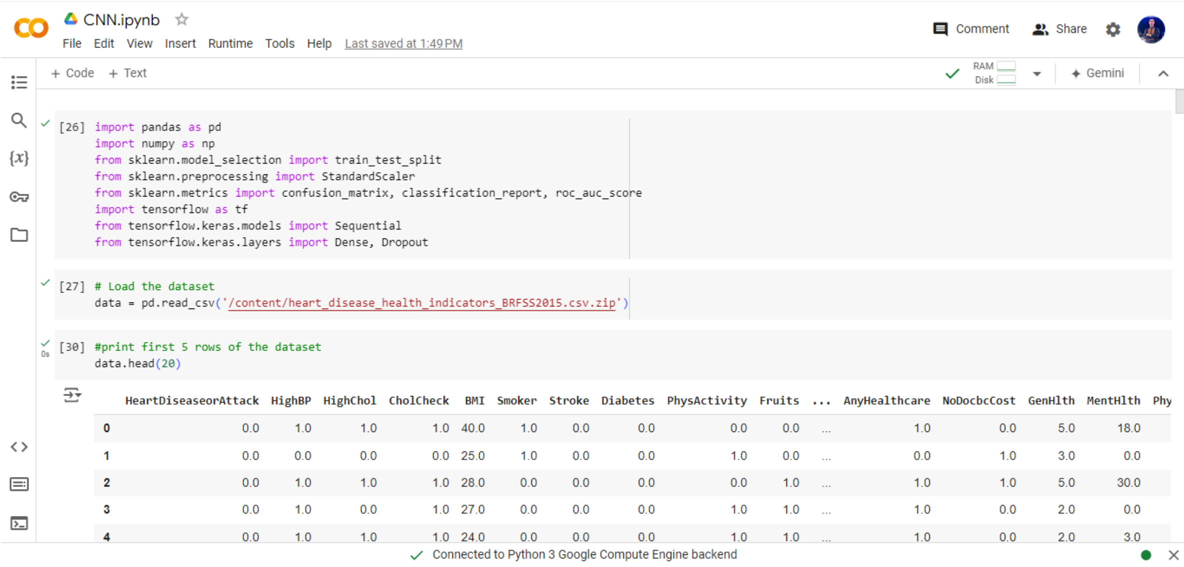
Exploring Other Models and Techniques:

- Investigating sophisticated deep learning architectures and hybrid models has the potential to enhance prediction accuracy.
- Integrating domain information and expert insights helps improve the process of creating features and designing models.

By focusing on these aspects in future research, the heart disease prediction system can enhance its precision, dependability, and broad applicability, ultimately leading to improved preventive healthcare and risk mitigation.

Appendices

1. Code Snippets:



CNN.ipynb

File Edit View Insert Runtime Tools Help Last saved at 1:49 PM

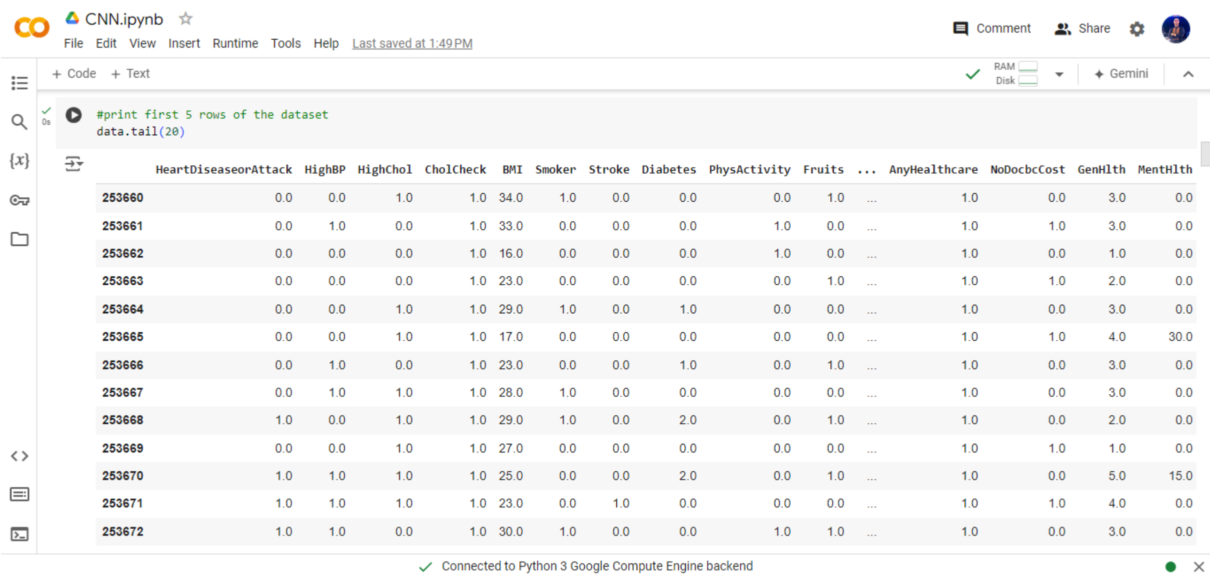
```
[26] import pandas as pd
     import numpy as np
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import StandardScaler
     from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score
     import tensorflow as tf
     from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Dense, Dropout
```

```
[27] # Load the dataset
     data = pd.read_csv('/content/heart_disease_health_indicators_BRFSS2015.csv.zip')
```

```
[30] #print first 5 rows of the dataset
     data.head(20)
```

	HeartDiseaseorAttack	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	Diabetes	PhysActivity	Fruits	...	AnyHealthcare	NoDocbcCost	GenHlth	MentHlth	Phy
0	0.0	1.0	1.0	1.0	40.0	1.0	0.0	0.0	0.0	0.0	...	1.0	0.0	5.0	18.0	
1	0.0	0.0	0.0	0.0	25.0	1.0	0.0	0.0	1.0	0.0	...	0.0	1.0	3.0	0.0	
2	0.0	1.0	1.0	1.0	28.0	0.0	0.0	0.0	0.0	1.0	...	1.0	1.0	5.0	30.0	
3	0.0	1.0	0.0	1.0	27.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	2.0	0.0	
4	0.0	1.0	1.0	1.0	24.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	2.0	3.0	

Connected to Python 3 Google Compute Engine backend



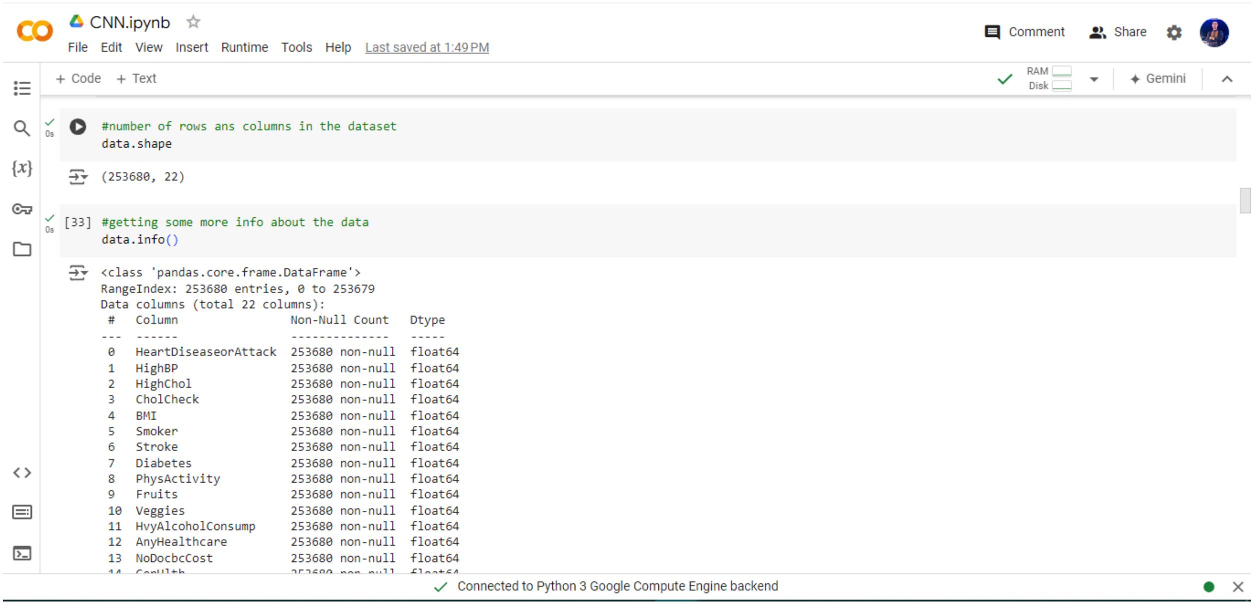
CNN.ipynb

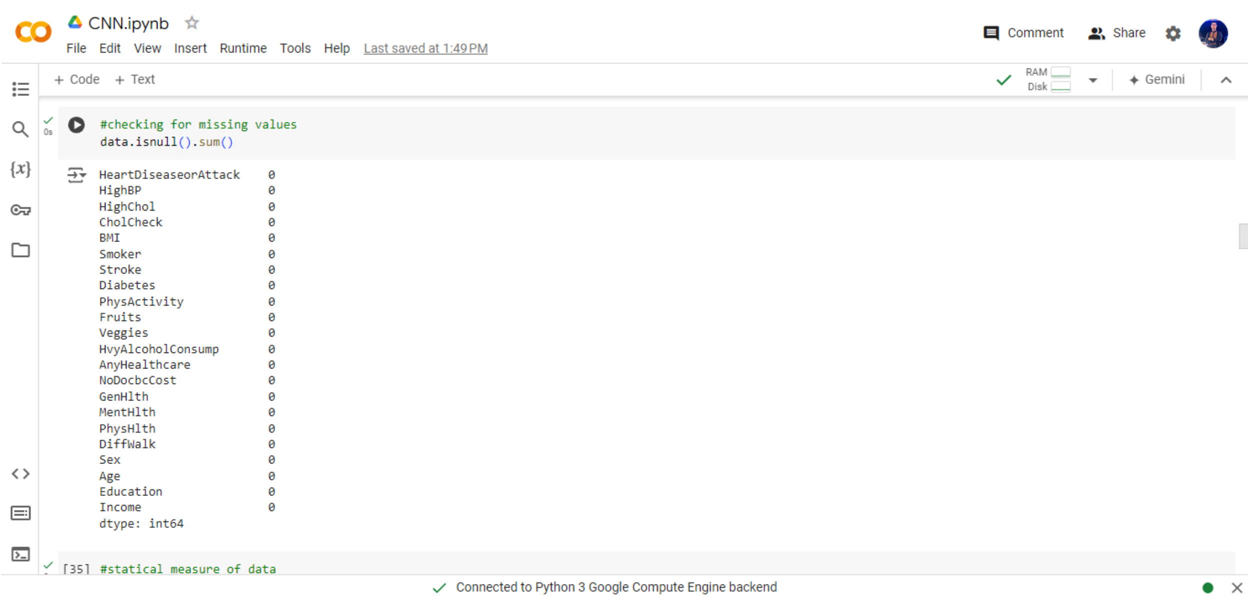
File Edit View Insert Runtime Tools Help Last saved at 1:49 PM

```
#print first 5 rows of the dataset
data.tail(20)
```

	HeartDiseaseorAttack	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	Diabetes	PhysActivity	Fruits	...	AnyHealthcare	NoDocbcCost	GenHlth	MentHlth
253660	0.0	0.0	1.0	1.0	34.0	1.0	0.0	0.0	0.0	1.0	...	1.0	0.0	3.0	0.0
253661	0.0	1.0	0.0	1.0	33.0	0.0	0.0	0.0	1.0	0.0	...	1.0	1.0	3.0	0.0
253662	0.0	0.0	0.0	1.0	16.0	0.0	0.0	0.0	1.0	0.0	...	1.0	0.0	1.0	0.0
253663	0.0	0.0	0.0	1.0	23.0	0.0	0.0	0.0	0.0	1.0	...	1.0	1.0	2.0	0.0
253664	0.0	0.0	1.0	1.0	29.0	1.0	0.0	1.0	0.0	0.0	...	1.0	0.0	3.0	0.0
253665	0.0	0.0	1.0	1.0	17.0	0.0	0.0	0.0	0.0	0.0	...	1.0	1.0	4.0	30.0
253666	0.0	1.0	0.0	1.0	23.0	0.0	0.0	1.0	0.0	1.0	...	1.0	0.0	3.0	0.0
253667	0.0	1.0	1.0	1.0	28.0	1.0	0.0	0.0	0.0	0.0	...	1.0	0.0	3.0	0.0
253668	1.0	0.0	1.0	1.0	29.0	1.0	0.0	2.0	0.0	1.0	...	1.0	0.0	2.0	0.0
253669	0.0	0.0	1.0	1.0	27.0	0.0	0.0	0.0	0.0	0.0	...	1.0	1.0	1.0	0.0
253670	1.0	1.0	1.0	1.0	25.0	0.0	0.0	2.0	0.0	1.0	...	1.0	0.0	5.0	15.0
253671	1.0	1.0	1.0	1.0	23.0	0.0	1.0	0.0	0.0	0.0	...	1.0	1.0	4.0	0.0
253672	1.0	1.0	0.0	1.0	30.0	1.0	0.0	0.0	1.0	1.0	...	1.0	0.0	3.0	0.0

Connected to Python 3 Google Compute Engine backend


CNN.ipynb
File Edit View Insert Runtime Tools Help Last saved at 1:49 PM
Comment
Share
+ Code + Text
RAM
Disk
Gemini
#number of rows ans columns in the dataset
data.shape
(253680, 22)
[33] #getting some more info about the data
data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253680 entries, 0 to 253679
Data columns (total 22 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 HeartDiseaseorAttack 253680 non-null float64
1 HighBP 253680 non-null float64
2 HighChol 253680 non-null float64
3 CholCheck 253680 non-null float64
4 BMI 253680 non-null float64
5 Smoker 253680 non-null float64
6 Stroke 253680 non-null float64
7 Diabetes 253680 non-null float64
8 PhysActivity 253680 non-null float64
9 Fruits 253680 non-null float64
10 Veggies 253680 non-null float64
11 HvyAlcoholConsump 253680 non-null float64
12 AnyHealthcare 253680 non-null float64
13 NoDocbcCost 253680 non-null float64
Connected to Python 3 Google Compute Engine backend


CNN.ipynb
File Edit View Insert Runtime Tools Help Last saved at 1:49 PM
Comment
Share
+ Code + Text
RAM
Disk
Gemini
#checking for missing values
data.isnull().sum()
HeartDiseaseorAttack 0
HighBP 0
HighChol 0
CholCheck 0
BMI 0
Smoker 0
Stroke 0
Diabetes 0
PhysActivity 0
Fruits 0
Veggies 0
HvyAlcoholConsump 0
AnyHealthcare 0
NoDocbcCost 0
GenHlth 0
MentHlth 0
PhysHlth 0
DiffWalk 0
Sex 0
Age 0
Education 0
Income 0
dtype: int64
[35] #statical measure of data
Connected to Python 3 Google Compute Engine backend

CNN.ipynb

```
#checing the distribution of target veriable
data['HeartDiseaseorAttack'].value_counts ()
```

```
HeartDiseaseorAttack
0.0    229787
1.0     23893
Name: count, dtype: int64
```

```
[37] # Define features (X) and target (y)
     X = data.drop(columns=['HeartDiseaseorAttack'])
     y = data['HeartDiseaseorAttack']
```

```
[38] # Split the data into training (80%) and testing (20%) sets
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[39] # Initialize the scaler
     scaler = StandardScaler()
```

```
[40] # Fit and transform the training data
     X_train_scaled = scaler.fit_transform(X_train)
```

```
[41] # Transform the test data
     X_test_scaled = scaler.transform(X_test)
```

Connected to Python 3 Google Compute Engine backend

CNN.ipynb

---------------------------- Neural Network Model ---------------------------

```
[42] # Define the neural network model
     model = Sequential([
         Dense(64, activation='relu', input_shape=(X_train_scaled.shape[1],)),
         Dropout(0.5),
         Dense(32, activation='relu'),
         Dropout(0.5),
         Dense(1, activation='sigmoid')
     ])
```

```
[43] # Compile the model
     model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
[44] # Train the model
     history = model.fit(X_train_scaled, y_train, epochs=50, batch_size=32, validation_split=0.2)
```

```
Epoch 1/50
5074/5074 [==============================] - 29s 5ms/step - loss: 0.2613 - accuracy: 0.9044 - val_loss: 0.2413 - val_accuracy: 0.9051
Epoch 2/50
5074/5074 [==============================] - 23s 5ms/step - loss: 0.2456 - accuracy: 0.9066 - val_loss: 0.2398 - val_accuracy: 0.9056
Epoch 3/50
5074/5074 [==============================] - 22s 4ms/step - loss: 0.2435 - accuracy: 0.9063 - val_loss: 0.2398 - val_accuracy: 0.9060
```

Connected to Python 3 Google Compute Engine backend

```
# Make predictions on the test set
y_pred_prob = model.predict(X_test_scaled).flatten()
y_pred = (y_pred_prob > 0.5).astype(int)
```

```
1586/1586 [==============================] - 3s 2ms/step
```

```
# Evaluate the neural network model
print("Neural Network Model Results:")

# Confusion Matrix
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

# Classification Report
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# ROC-AUC Score
print("\nROC-AUC Score:")
print(roc_auc_score(y_test, y_pred_prob))
```

```
Neural Network Model Results:
Confusion Matrix:
[[45919    49]
 [ 4646   122]]
```

```
from google.colab import drive
drive.mount('/content/drive')

# Assuming 'model' is your trained CNN model object
from tensorflow.keras.models import save_model

# Save your model to Google Drive
save_model(model, '/content/drive/MyDrive/cnn_model.keras')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

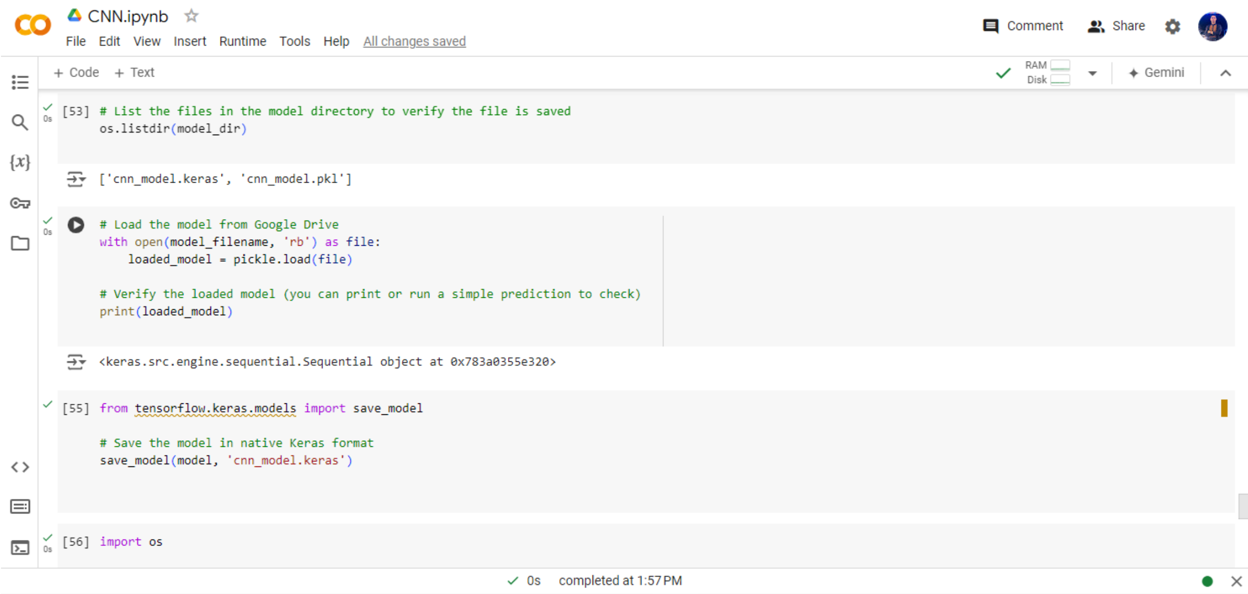
```
#prediction system
import numpy as np
from tensorflow.keras.models import load_model

# Load your trained CNN model
model = load_model('/content/drive/MyDrive/cnn_model.keras')

# Example input (adjust as per your dataset attributes)
# Here, I'm using a single sample input. You can adjust for multiple samples.
sample_input = np.array([[1.0, 1.0, 1.0, 40.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 5.0, 18.0, 15.0, 1.0, 0.0, 9.0, 4.0, 3.0]])

# Make predictions
predictions = model.predict(sample_input)
```


```
[50] import pickle

     # Assuming 'model' is your trained CNN model
     model_filename = 'cnn_model.pkl'
     with open(model_filename, 'wb') as file:
         pickle.dump(model, file)
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[52] import os

     # Define the path where you want to save the model in Google Drive
     model_dir = '/content/drive/My Drive/models'
     os.makedirs(model_dir, exist_ok=True)
     model_filename = os.path.join(model_dir, 'cnn_model.pkl')

     # Save the model to the specified path
     with open(model_filename, 'wb') as file:
         pickle.dump(model, file)
```


```
[53] # List the files in the model directory to verify the file is saved
     os.listdir(model_dir)
```

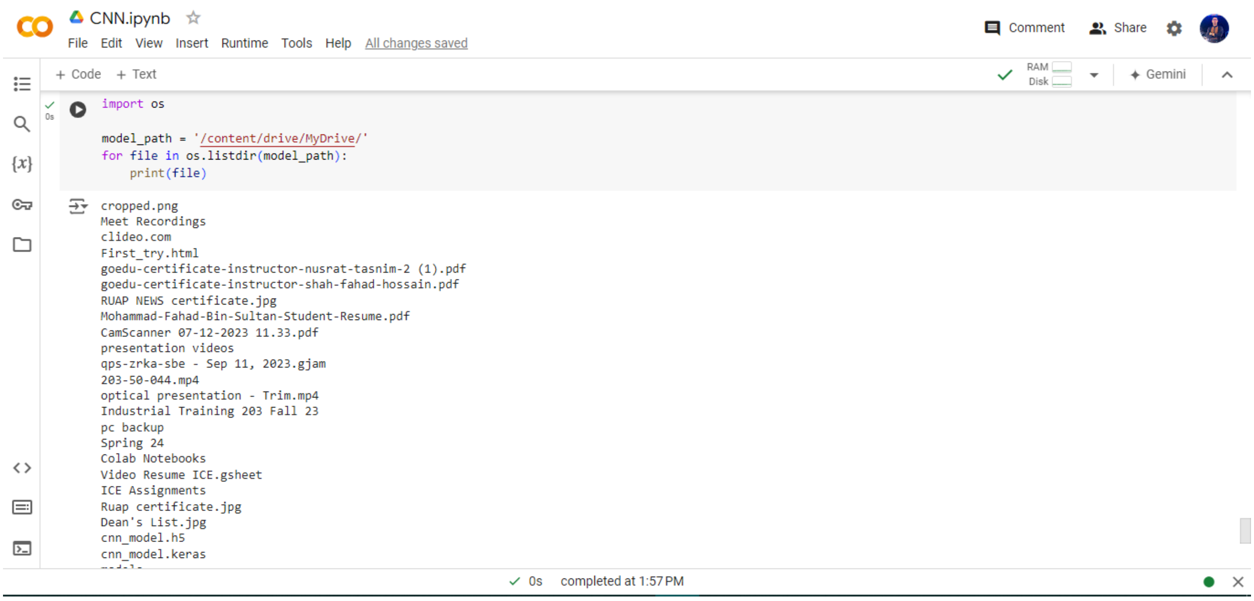
```
['cnn_model.keras', 'cnn_model.pkl']
```

```
# Load the model from Google Drive
with open(model_filename, 'rb') as file:
    loaded_model = pickle.load(file)

# Verify the loaded model (you can print or run a simple prediction to check)
print(loaded_model)
```

```
<keras.src.engine.sequential.Sequential object at 0x783a0355e320>
```

```
[55] from tensorflow.keras.models import save_model

     # Save the model in native Keras format
     save_model(model, 'cnn_model.keras')
```

```
[56] import os
```



Documentation of Tools and Libraries Used:

- **Pandas:** Data processing and analysis.
    - Documentation: Pandas Documentation.
- **Scikit-learn:** A Python library for machine learning.
    - Documentation can be found at the Scikit-learn Documentation.
- **TensorFlow/Keras:** A deep learning framework designed for constructing neural networks.
    - Documentation: TensorFlow Documentation.
- **Streamlit:** A framework designed for constructing dynamic web applications.
    - Documentation: Streamlit Documentation.
- **Flask:** A WSGI web application framework that is designed to be lightweight.
    - Documentation: Flask Documentation.
- **ngrok:** A tool designed to establish secure tunnels to the localhost.
    - Documentation: ngrok Documentation.

This supplementary material offers further background, code examples, and data visualizations to enhance the primary content of the project report.

References

- **Citations:**
  - List of all academic papers, articles, and other sources cited in your report.

i. Choudhury, A., & Dey, N. (2019). Heart Disease Detection by Using Machine Learning Algorithms and a Real-Time Prediction System. International Journal of Scientific & Technology Research, 8(10), 230-235.

ii. Das, R., Turkoglu, I., & Sengur, A. (2009). Effective diagnosis of heart disease through neural networks ensembles. Expert Systems with Applications, 36(4), 7675-7680.

iii. Deo, R. C. (2015). Machine learning in medicine. Circulation, 132(20), 1920-1930.

iv. Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J. J., Sandhu, S., ... & Froelicher, V. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. The American Journal of Cardiology, 64(5), 304-310.

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

vi. Johnson, K. W., Soto, J. T., Glicksberg, B. S., Shameer, K., Miotto, R., Ali, M., ... & Dudley, J. T. (2018). Artificial intelligence in cardiology. Journal of the American College of Cardiology, 71(23), 2668-2679.

vii. Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal, 13, 8-17.

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

ix. Liu, Y., Chen, P. H. C., Krause, J., & Peng, L. (2019). How to read articles that use machine learning: Users' guides to the medical literature. JAMA, 322(18), 1806-1816.

x. Nahavandi, S. (2019). Industry 5.0—A human-centric solution. Sustainability, 11(16), 4371.

xi. Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. New England Journal of Medicine, 380(14), 1347-1358.

xii. Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2018). Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE Journal of Biomedical and Health Informatics, 22(5), 1589-1604.

xiii. Weng, S. F., Reps, J., Kai, J., Garibaldi, J. M., & Qureshi, N. (2017). Can machine-learning improve cardiovascular risk prediction using routine clinical data?. PLoS ONE, 12(4), e0174944.

xiv. Yao, X., Zhu, Z., & Jiang, J. (2019). Prediction of heart disease risk using a machine learning algorithm. Computers in Biology and Medicine, 108, 41-47.

xv. Zhang, Z., & Sejdić, E. (2019). Radiological images and machine learning: Trends, perspectives, and prospects. Computers in Biology and Medicine, 108, 76-82.