



**Islington college**  
(इस्लिङ्टन कलेज)

**Module Code & Module Title**  
**Level 6 – Artificial Intelligence**

**Assessment Type**  
**Semester**  
**2024/25 Autumn**

**Student Name: Divya Shrestha**

**London Met ID: 22085527**

**College ID: NP01CPS230022**

**Assignment Due Date: Tuesday, January 21, 2025**

**Assignment Submission Date: Wednesday, January 22, 2025**

**Submitted To: Dipeshor Silwal**

**Word Count: 5452**

*I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*

# 22085527\_Divya\_Shrestha\_AI\_SIMILARITY\_CHECK.docx

 Islington College, Nepal

## Document Details

Submission ID

trn:oid::3618:79767677

Submission Date

Jan 21, 2025, 10:14 PM GMT+5:45

Download Date

Jan 21, 2025, 10:16 PM GMT+5:45

File Name

22085527\_Divya\_Shrestha\_AI\_SIMILARITY\_CHECK.docx

File Size

36.2 KB

39 Pages





5,452 Words

31,204 Characters




## 20% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

### Match Groups

-  **76 Not Cited or Quoted 16%**  
Matches with neither in-text citation nor quotation marks
-  **14 Missing Quotations 4%**  
Matches that are still very similar to source material
-  **0 Missing Citation 0%**  
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**  
Matches with in-text citation present, but no quotation marks

### Top Sources

- 12%  Internet sources
- 8%  Publications
- 16%  Submitted works (Student Papers)

## Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## Table of Contents

1	Introduction .....	1
1.1	Explanation of topic/AI concepts used .....	1
1.1.1	Introduction to AI .....	1
1.1.2	Introduction to Machine Learning (ML) .....	2
1.2	Problem Domain .....	4
2	Background .....	6
2.1	Research on topic/problem domain .....	6
2.2	Review and analysis of existing work in the problem domain .....	9
2.2.1	Deep Learning for Cardiovascular Risk Prediction .....	9
2.2.2	Feature Selection and Optimization Techniques .....	10
2.2.3	Supervised Machine Learning for CVD Prediction .....	11
2.3	Comparison of the Studies .....	12
2.3.1	Tiwari et al. ....	12
2.3.2	Subasi et al. ....	12
2.3.3	Ahmed et al. ....	13
2.4	My Dataset .....	14
3	Solution .....	15
3.1	AI Algorithms Used .....	21
3.1.1	Decision Tree .....	21
3.1.2	Logistic Regression .....	25
3.1.3	Gaussian Naive Bayes .....	28
3.2	Development Process .....	30
3.2.1	Tools Used .....	30
3.2.2	Toolkit used .....	31
3.2.3	Explanation of Development Process .....	32
3.3	Pseudocode for each Algorithms .....	15
3.3.1	Decision Tree .....	15
3.3.2	Logistic Regression .....	16
3.3.3	Naive Bayes Classifier .....	17
3.4	Flowcharts .....	18

3.4.1	Decision Tree Flowchart .....	18
3.4.2	Logistic Regression Flowchart .....	19
3.4.3	Native Byers .....	20
4	Conclusion .....	48
5	References.....	50

## Table of Figures

Figure 1: AI hierarchy .....	1
Figure 2: Supervised Machine Learning.....	2
Figure 3: Unsupervised Machine Learning.....	3
Figure 4: Death rate by CVD (Science Direct, 2024).....	4
Figure 5: Tiwari et al. model prediction (Achyut Tiwari, 2024) .....	9
Figure 6: Subasi et al. model prediction (Achyut Tiwari, 2024).....	10
Figure 7: Ahmed et al. model prediction (Arsalan Khan, 2024) .....	11
Figure 8:decision tree flowchart .....	18
Figure 9: Logistic Regression Flowchart .....	19
Figure 10: Native Byers flowchart .....	20
Figure 11: Logistic Regression (datacamp, 2025) .....	25
Figure 12: python figure .....	30
Figure 13: Jupyter Notebook .....	30
Figure 14: importing libraries.....	32
Figure 15: Reading the dataset .....	32
Figure 16: information of the dataset.....	33
Figure 17: pie chart .....	34
Figure 18: Data cleaning part 1 .....	35
Figure 19: Data cleaning part 2 .....	35
Figure 20: train test and splitting data .....	36
Figure 21: Decision Tree Classifier modeling .....	37
Figure 22: Decision Tree Classifier accuracy .....	37
Figure 23: Classification Matrix Decision Tree .....	38
Figure 24: Decision Tree ROC curve.....	39
Figure 25: plotting tree code snippet.....	40

Figure 26: plotting tree graph .....	40
Figure 27: Logistic Regression modeling .....	41
Figure 28: Logistic Regression model .....	41
Figure 29: Classification matrix Logistic Regression .....	42
Figure 30: Logistic Regression ROC curve .....	43
Figure 31: Gaussian Naive Bayes .....	44
Figure 32: Gaussian Naive Bayes model .....	44
Figure 33: Confusion matrix Naive Bayes .....	45
Figure 34: Naive Bayes ROC curve .....	46
Figure 35: Confusion Matrix Comparision .....	47
Figure 36: ROC curve Comparison .....	47

## Table of Equation

Equation 1: Entropy Equation (shiksha online, 2025).....	22
Equation 2: Gini Impurity Equation (shiksha online, 2025).....	23
Equation 3: Information Gain.....	24
Equation 4: Linear Regression (datacamp, 2025) .....	26
Equation 5: Sigmoid Function .....	26
Equation 6: Sigmoid function on linear model .....	27
Equation 7: Bayers Theorem Equation (geekforgeeks, 2025).....	28
Equation 8: Gaussian Naive Bayers.....	29

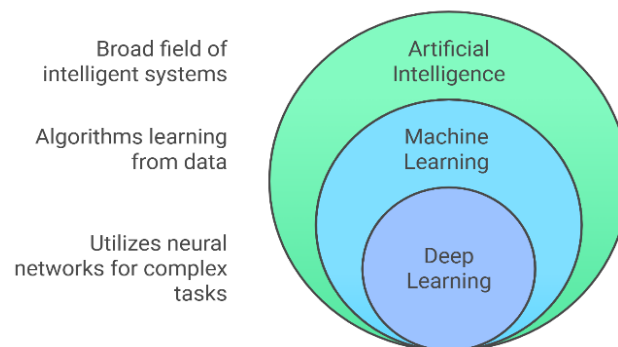
# 1 Introduction

## 1.1 Explanation of topic/AI concepts used

A Cardiovascular Disease Prediction System is a program that predicts individuals at high risk of developing cardiovascular diseases given other attribute relating to health. However, with today's advanced life and heavy schedules the chances of early detection and proper prevention of the health issues is need of the hour and can prove lifesaving. Assessing this risk factors on such diseases requires a lot of time, can easily be done wrong and is very in-efficient. Consequently, the use of AI and machine learning to develop predictive models can go a long way to make the process automated, as well as deliver relative risk probabilities.

### 1.1.1 Introduction to AI

AI is defined as the ability of a computer and other systems to imitate human intelligence capabilities and then some. These include reasoning, learning, problem solving, perception and understanding of language. It covers several technologies and methodologies used for representing cognition. AI can also be broadly defined as the simulation of human intelligence processes by machines. These processes include learning, reasoning, and self-correction (Glover, 2024).



*Figure 1: AI hierarchy*

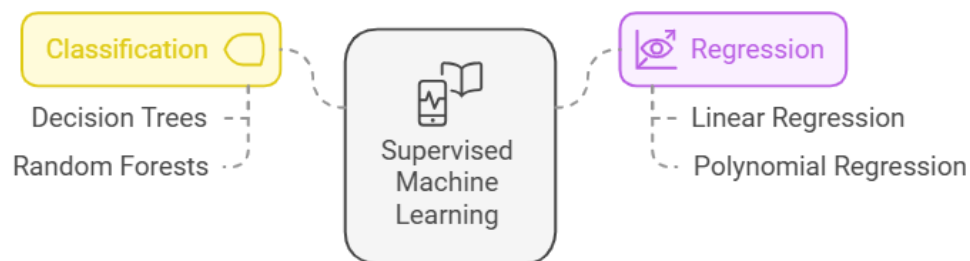
### 1.1.2 Introduction to Machine Learning (ML)

Machine Learning therefore can be understood as the field, which endows algorithms with the capacity to learn from data or information. It is a process of tuning parameters of the model to accomplish specified requirements through computations, such that behavior of the machine aligns with the data or experience. The learning process is iterative, meaning that as new data is introduced, the model continuously updates itself to enhance accuracy and decision-making capabilities (Brown, 2024).

There are two types of Machine Learning:

- Supervised Machine Learning

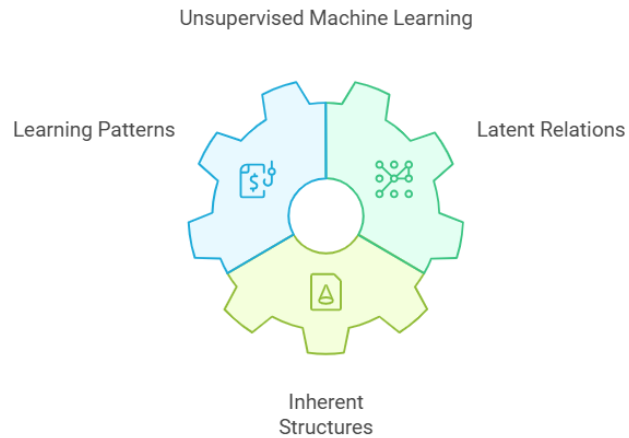
Supervised machine learning is one of the most popular categories of machine learning that use datasets that have been assigned by an instructor. In this context, “labeled” means that each instance is a training example followed by an output label that learner should be able to ‘learn’. It does this while finding out patterns with the input data on which it creates its predictions based on the labels given to it. It is further separated into 2 categories



*Figure 2: Supervised Machine Learning*

- Unsupervised Machine Learning

Unsupervised machine learning aims to learn the existing patterns from demand without existing output patterns. In this case, the goal is to find out latent relations or inherent structures inherent in the input data itself irrespective of what these relationships could be.



*Figure 3: Unsupervised Machine Learning*



## 1.2 Problem Domain

Cardiovascular diseases have found their place among the most life-threatening diseases pressure and increasing the demands for individuals, health-care system, and governments globally in the present era of the world. Cardiovascular diseases (CVDs) are the leading cause of mortality globally, responsible for a significant number of deaths and disabilities. In 2021 alone, CVDs accounted for 20.5 million deaths, comprising approximately one-third of all global deaths (Science Direct, 2024).

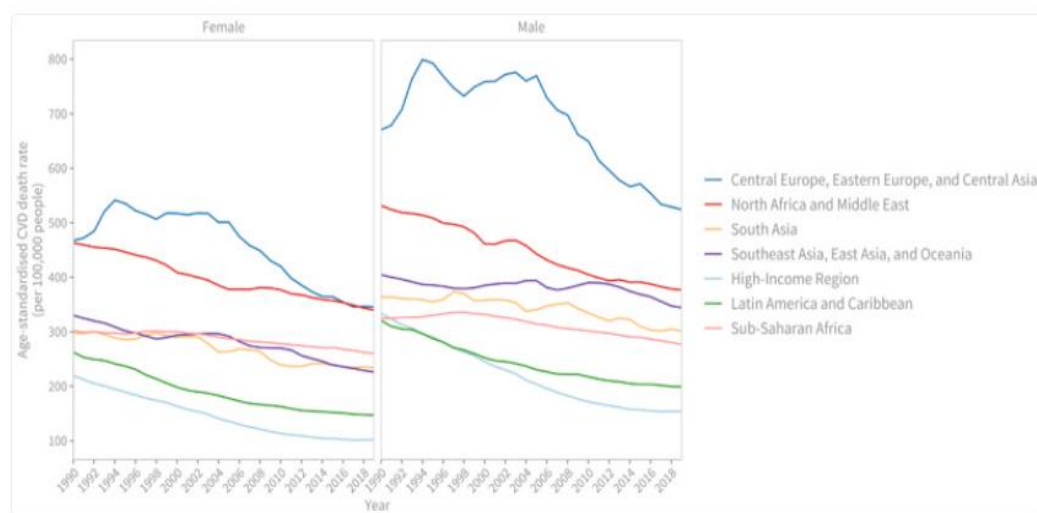


Figure 4: Death rate by CVD (Science Direct, 2024)

For cardiovascular disease risk assessment, a significant amount of health-related data needs to be sorted, which can consume time and might involve errors. Inadequate identification of the signs and the wrong perception of their meaning becomes a direct cause of aggravated consequences and extensive expenses for the treatment. Cardiovascular diseases alone result for over \$ 200 billion in healthcare costs and lost productivity each year globally.

A cardiovascular disease risk assessment through a machine learning algorithm provides a feasible approach to accomplishing this goal. The model can input general health indicators including blood pressure, cholesterol levels, glucose levels to determine the odds of an individual getting cardiovascular diseases hence properly early enough treatment can be taken.

### 1.3 Objectives

- Build an AI system that identifies people who may develop heart diseases based on their health and daily routine.
- Review the performance of Decision Trees, Logistic Regression, and Naive Bayes methods in predicting cardiovascular risk levels through comparative testing.
- Learn predicting models by feeding them with data that includes age, blood pressure, cholesterol levels, glucose levels and physical activity.
- Test and compare the three models through their performance metrics to find which one shows the highest accuracy in predicting outcomes.
- Clean and normalize your input data before dividing it into training and testing groups to support proper model development.
- Visualize model results through confusion matrices, ROC curves, and decision tree plots to assess performance and boundary limits.
- Share what makes each algorithm strong and weak plus point out places where the system can develop further.
- Demonstrate the value of machine learning in spotting diseases early so healthcare works better and gets better results for patients.
- Assess techniques to enhance the capabilities of developed models through increased data size and feature sets to achieve better accuracy and dependability.
- Advise using ensemble learning approaches alongside deep learning frameworks while creating real-time prediction systems.

## **2 Background**

### **2.1 Research on topic/problem domain**

Machine learning is a subfield of Artificial Intelligence that allows systems to train on data, make sense of them and make their choices without much external inputs. In the field of cardiovascular disease machine learning bears new approaches of prevention, early diagnosis and better management of the heart related diseases.

One such study conducted by D. Bhattacharyya et al. (2020) applied Logistic Regression, Random Forest and Support Vector Machine algorithms on patient's health data to predict cardiovascular risks (Bhannu Prakash Doppala, 2024). Their research demonstrated that machine learning models could achieve higher accuracy and sensitivity compared to traditional risk scoring methods, such as the Framingham Risk Score.

Another significant contribution is the study by A. Tiwari et al. (2019), which implemented deep learning models to process complex datasets containing clinical and lifestyle attributes (Sudarshan Singh, 2024). Their results highlighted the potential of neural networks to identify hidden patterns and improve CVD prediction accuracy.

Machine Learning can consider various health indicators including age, blood pressure, cholesterol levels, and most importantly, lifestyle habits, in real data feeds and look for possible correlations and patterns that machine learning algorithms can find. Using supervised learning methods, which include classification models, it becomes possible to build systems for predicting CVD risk from trained data. Also, with unsupervised learning approach one can potentially discover novel risks or patient subgroups within the dataset. Besides enhancing the reliability on risk assessment, this strategy also addresses points of early identification of potential risks and subsequent consultations and interventions may be offered. It helps doctors to target high risk patients and hence, the mortality rate comes down and stress in healthcare is alleviated.

Advantages of using Machine Learning in CVD are:

- **Early Detection:** It is possible to use machine learning for the analysis of health information to discover which patient is most likely to develop cardiovascular diseases at the early stage of the diseases' development.
- **Improved Accuracy:** Based on such data, it is evident that the application of ML algorithms provides a more accurate prediction of CVD risks than conventional diagnostic instruments.
- **Personalized Treatment:** In addition, ML makes it possible to develop individual programs of treatment with the help of such parameters as a patient's age, occupation, and medical record.
- **Efficiency in Data Analysis:** Contrasting with traditional approaches, machine learning is capable to analyze big amounts of health information and provide results in short time.
- **Real-Time Monitoring:** Such integration with wearable devices enables the ML models to give real-time evaluation of patient's health and the potential risk that the patient is likely to face.
- **Reduced Human Error:** Actual Affairs devoid of prejudice and mistakes related to manual assessments.
- **Cost-Effective:** Such early interventions include preventative measures hence cutting down losses that accrue from costly treatments and admissions more so to the healthcare facility.
- **Scalability:** Artificial intelligence can be adapted to process information from millions of patients and makes it applicable for giant medical centers.
- **Discovery of New Patterns:** With ML, earlier overlooked risk factors and assumptions in health data can be discovered more easily.
- **Remote Access:** AI based solutions allow clinicians to have an approach of CVD risk assessments of the patients and remote consultation in regions with limited access.

Disadvantages of using Machine Learning in CVD are:

- **Data Dependency:** This gives much emphasis on the labeled data which are used in training the machine learning models and this might not be easily come by.
- **Overfitting Risk:** The models when not trained properly can work like 'black boxes' and give good results, only when fed the training data; they perform poorly in front of real patients.
- **Interpretability Issues:** Some of the most common machine learning algorithms are not easily interpretable; many modern machine learning algorithms such as the deep learning models remain 'black boxes'.
- **Ethical Concerns:** Making use of patient information also raises questions on issues to do with privacy, security and the proper usage of sensitive data.
- **Bias in Data:** Forecasts made by these models may be unfair or inaccurate particularly towards disadvantaged groups because the models have been shaped by biased data.
- **High Computational Costs:** Teaching and retention of complex machine learning models require a lot of computational power.
- **Dependency on Technology:** It is likely that with the increased use of ML systems, clinical judgment and professional analysis can be somewhat diminished by practitioners.
- **Difficulty in Generalization:** It has been found that models extrapolated from specific data sets may not generalize well on different population or geographical databases.
- **Regulatory Challenges:** Compliance with the most recognized healthcare regulations and standards present a significant challenge for multi-level ML-based systems.
- **Risk of Misclassification:** It also reveals that false positive or negative predictions may create anxiety, over-treatment and undertreatment respectively.

## 2.2 Review and analysis of existing work in the problem domain

### 2.2.1 Deep Learning for Cardiovascular Risk Prediction

Tiwari et al. utilized the deep learning algorithms, particularly neural networks using big healthcare datasets to predict cardiovascular risk. These advanced models showed enhanced ability to learn more complex relationships and were more accurate than conventional models such as Logistic Regression and Random Forest (Achyut Tiwari, 2024). Applications of deep learning algorithms are especially useful in analyzing big and diverse data where numerous sets of intricate dependencies may exist. This advanced feature is more important in the health sector since timely and accurate risk assessment can produce positive impacts on patients and preventions.

However, it has been observed that deep learning models are 'black box' models, and this makes interpretability, which is very important especially in clinics, difficult. Medical professionals require actual understanding of rationale for the predictions to be able to trust and follow them which is why transparent nature is crucial<sup>1</sup>. However, as presented by Tiwari et al. deep learning has potential to transform cardiovascular risk prediction models by providing better analysis of patient data than conventional methods (Achyut Tiwari, 2024).

Comparison of ML models:

Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	MCC
Random Forest	90.21%	87.31%	95.12%	84.82%	91.05%	89.97%	80.65%
MLP	84.25%	82.08%	89.43%	78.57%	85.60%	84.00%	68.60%
KNN	80.85%	78.67%	86.99%	74.10%	82.62%	80.54%	61.80%
Extra Tree Classifier	90.93%	88.54%	94.30%	86.60%	91.33%	90.45%	81.36%
XGB	91.91%	90.62%	94.30%	89.28%	92.43%	91.79%	83.83%
SVC	82.55%	80.14%	88.61%	75.89%	84.16%	82.25%	65.25%
SGD	82.12%	80.00%	87.80%	75.89%	83.72%	81.84%	64.34%
Adaboost	83.40%	81.43%	88.61%	77.67%	84.82%	83.14%	66.88%
CART	84.25%	83.59%	86.99%	81.25%	85.25%	84.12%	68.44%
GBM	84.25%	81.61%	90.24%	77.67%	85.71%	83.96%	68.70%

Figure 5: Tiwari et al. model prediction (Achyut Tiwari, 2024)

### 2.2.2 Feature Selection and Optimization Techniques

Subasi et al. explored feature selection methods like Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) to identify the most relevant health indicators for cardiovascular disease (CVD) prediction (Aditya Ranade, Nitin Pise, 2024). Their approach was quite helpful as it enhanced computational speed and the quality of the models by minimizing the number of features it dealt with. With RFE, they were able to proceed with feature selections by eliminating the features with low significance while maintaining those features with high significance on the performance of the model. While PCA simplified the model via a Singular Value Decomposition that reduced the variables in the model to a small set that were not highly correlated, PCA took the variables of the model and created a set of new, uncorrelated components which made the model simpler and easier to understand.

However, one of the issues that remained even with these improvements included the ability to exclude other important but marginally significant features during feature reduction. This exclusion could minimize generalization capability and its ability to perform well on unseen data. It is these subtle variables that could contain perhaps high levels of information that would aid in producing accurate predictions despite not being obvious. Therefore, despite the enhancement brought by feature selection such as RFE and PCA, there is always a tradeoff between time complexity and the inclusion of other critical features (Aditya Ranade, Nitin Pise, 2024).

Models	Accuracy	Classification error	Precision	F-measure	Sensitivity	Specificity
Naive Bayes	75.8	24.2	90.5	84.5	79.8	60.0
Generalized Linear Model	85.1	14.9	88.8	91.6	94.9	20.0
Logistic Regression	82.9	17.1	89.6	90.2	91.1	25.0
Deep Learning	87.4	12.6	90.7	92.6	95	33.3
Decision Tree	85	15.0	86	91.8	98.8	0.0
Random Forest	86.1	13.9	87.1	92.4	98.8	10.0
Gradient Boosted Trees	78.3	21.7	94.1	86.8	80.7	60.0
Support Vector Machine	86.1	13.9	86.1	92.5	100	0.0
VOTE	87.41	12.59	90.2	84.4	-	-
HRFLM (proposed)	<b>88.4</b>	<b>11.6</b>	<b>90.1</b>	<b>90</b>	<b>92.8</b>	<b>82.6</b>

Figure 6: Subasi et al. model prediction (Achyut Tiwari, 2024)

### 2.2.3 Supervised Machine Learning for CVD Prediction

Research by Ahmed et al. emphasized the superiority of supervised learning algorithms, such as Support Vector Machines (SVM) and Naïve Bayes, for cardiovascular disease (CVD) prediction (Md Mamun Ali, 2024). Compared to unsupervised models, these methods were also more accurate and reliable in terms of the resulted predictions. In supervised learning algorithms the models are trained on labeled data, which means that the models can learn from the outcome of a few training samples and then extend this learning on to other similar samples. This approach is most suitable for healthcare situations which require diagnoses and treatments based on prognostications.

However, these methods require extensive labeled datasets, which are not always readily available in healthcare settings (Md Mamun Ali, 2024). As with most real-world applications, one of the major problems is the lack of quality labeled data; this poses a serious issue for supervised learning methods which depend significantly on high quality data for training as well as validation.

Output	DT	SVM	NB	LR	RF
Accuracy	0.8372	0.8308	0.7474	0.8308	0.8501
95% confidence interval	(0.6608, 0.8043)	(0.654, 0.7986)	(0.567, 0.7221)	(0.654, 0.7986)	(0.6745, 0.8158)
Sensitivity	0.9028	0.8472	0.8889	0.8333	0.8611
Specificity	0.5952	0.631	0.4405	0.6429	0.6548
+Predicted value	0.6566	0.663	0.5766	0.6667	0.6813
−Predicted value	0.8772	0.8281	0.8222	0.8182	0.8862
Prevalence	0.4615	0.4615	0.4615	0.4615	0.4615
Detection rate	0.4167	0.391	0.4103	0.3846	0.3974
Detection prevalence	0.6346	0.5897	0.7115	0.5769	0.5833
Balanced accuracy	0.849	0.8391	0.7647	0.8381	0.8579

Figure 7: Ahmed et al. model prediction (Arsalan Khan, 2024)



## **2.3 Comparison of the Studies**

### **2.3.1 Tiwari et al.**

- Precision: Deep learning models, especially neural networks, excelled in detecting hidden and complex patterns, resulting in improved precision.
- Accuracy: Achieved the highest accuracy among all methodologies, particularly in large and complex datasets.
- Effective for analyzing intricate dependencies in healthcare data.
- The “black box” nature makes it less interpretable, which is a limitation in clinical settings.
- Requires significant computational resources and high-quality data.

### **2.3.2 Subasi et al.**

- Precision: Enhanced with the help of feature selection methods such as Recursive Feature Elimination (RFE), and Feature Principal Component Analysis (PCA) that concern only significant signals..
- Accuracy: Enhanced through computational efficiency and reduced dimensionality, although excluding marginally significant features may slightly impact performance on unseen data.
- Optimizes speed and simplifies models.
- Balances accuracy with computational demands.
- Risks losing subtle but important data variables during feature reduction.

### **2.3.3 Ahmed et al**

- Precision: Delivered reliable precision by training supervised learning models (SVM, Naïve Bayes) on labeled datasets, minimizing prediction errors.
- Accuracy: Consistently high accuracy provided the availability of extensive and high-quality labeled datasets.
- Highly suitable for healthcare applications requiring clear and reliable predictions.
- Limited by the need for labeled data, which is often scarce in real-world healthcare settings.
- Easier to interpret compared to deep learning models.

## 2.4 My Dataset

The dataset used in this study consists of seventy thousand entries and thirteen attributes that are salient factors that have cardinality to cardiovascular fitness. The principal use of this dataset is for model development to predict the likelihood of a cardiovascular disease diagnosis. Below is a detailed breakdown of the features:

- Id: A unique identifier
- Age: Age of an individual
- Gender: Gender of an individual
- Height: Height of an individual, in cm
- Weight: Weight of individual, in kg
- ai\_hi: Systolic blood pressure
- ap\_low: Diastolic blood pressure
- cholesterol: 3 cholesterol level, 1, 2 and 2
- gluc: Category which shows glucose levels
- smoke: A binary feature that shows either 0 or 1 if the individual smokes
- alco: A binary feature that shows either 0 or 1 if the individual drinks
- actice: A binary feature that shows either 0 or 1 if the individual is physically active
- cardio: The targeted value

The material is especially beneficial for training classification models that focus on the probability of development of cardiovascular disease relying on health and lifestyle factors. Moreover, the extensive number of features make the investigations on features possible for data exploration and feature construction for enhancing the models.

Source: [Heart Failure Dataset](#)

### 3 Solution

#### 3.1 Pseudocode for each Algorithms

##### 3.1.1 Decision Tree

**IMPORT** required libraries pandas, numpy, matplotlib, sklearn modules

**READ** "Cardiovascular dataset.csv" into dataframe

**CHECK** for null values

**DROP** null values

**CHECK** for duplicate values

**DROP** duplicate values

**INITIALIZE** X by dropping 'id' and 'cardio' columns

**INITIALIZE** Y with 'cardio' column

**SPLIT** data into train and test sets training set to 80% test set to 20%  
random\_state to 42

**CREATE** object DecisionTreeClassifier with max\_depth=4 as model

**FIT** model with X and Y

**PREDICT** on test set

**DISPLAY** accuracy\_score

**DISPLAY** classification report with precision, recall, f1-score

**PLOT** confusion matrices

**PLOT** ROC curve

**PLOT** decision tree

### 3.1.2 Logistic Regression

**IMPORT** required libraries pandas, numpy, matplotlib, sklearn modules

**READ** "Cardiovascular dataset.csv" into dataframe

**CHECK** for null values

**DROP** null values

**CHECK** for duplicate values

**DROP** duplicate values

**INITIALIZE** X by dropping 'id' and 'cardio' columns

**INITIALIZE** Y with 'cardio' column

**SPLIT** data into train and test sets training set to 80% test set to 20%  
random\_state to 42

**CREATE** LogisticRegression model with max\_iter=1000

**FIT** model with training data (x\_train, y\_train) **PREDICT** on test set

**CALCULATE** accuracy score

**DISPLAY** accuracy percentage

**DISPLAY** classification report with precision, recall, f1-score

**PLOT** confusion matrices

**PLOT** ROC curve

### 3.1.3 Naive Bayes Classifier

**IMPORT** required libraries pandas, numpy, matplotlib, sklearn modules

**READ** "Cardiovascular dataset.csv" into dataframe

**CHECK** for null values

**DROP** null values

**CHECK** for duplicate values

**DROP** duplicate values

**INITIALIZE** X by dropping 'id' and 'cardio' columns

**INITIALIZE** Y with 'cardio' column

**SPLIT** data into train and test sets training set to 80% test set to 20%  
random\_state to 42

**CREATE** object GaussianNB model

**FIT** model with training data

**PREDICT** on test set

**DISPLAY** accuracy score

**DISPLAY** classification report, with precision, recall, f1-score

**PLOT** confusion matrices

**PLOT** ROC curve

## 3.2 Flowcharts

### 3.2.1 Decision Tree Flowchart

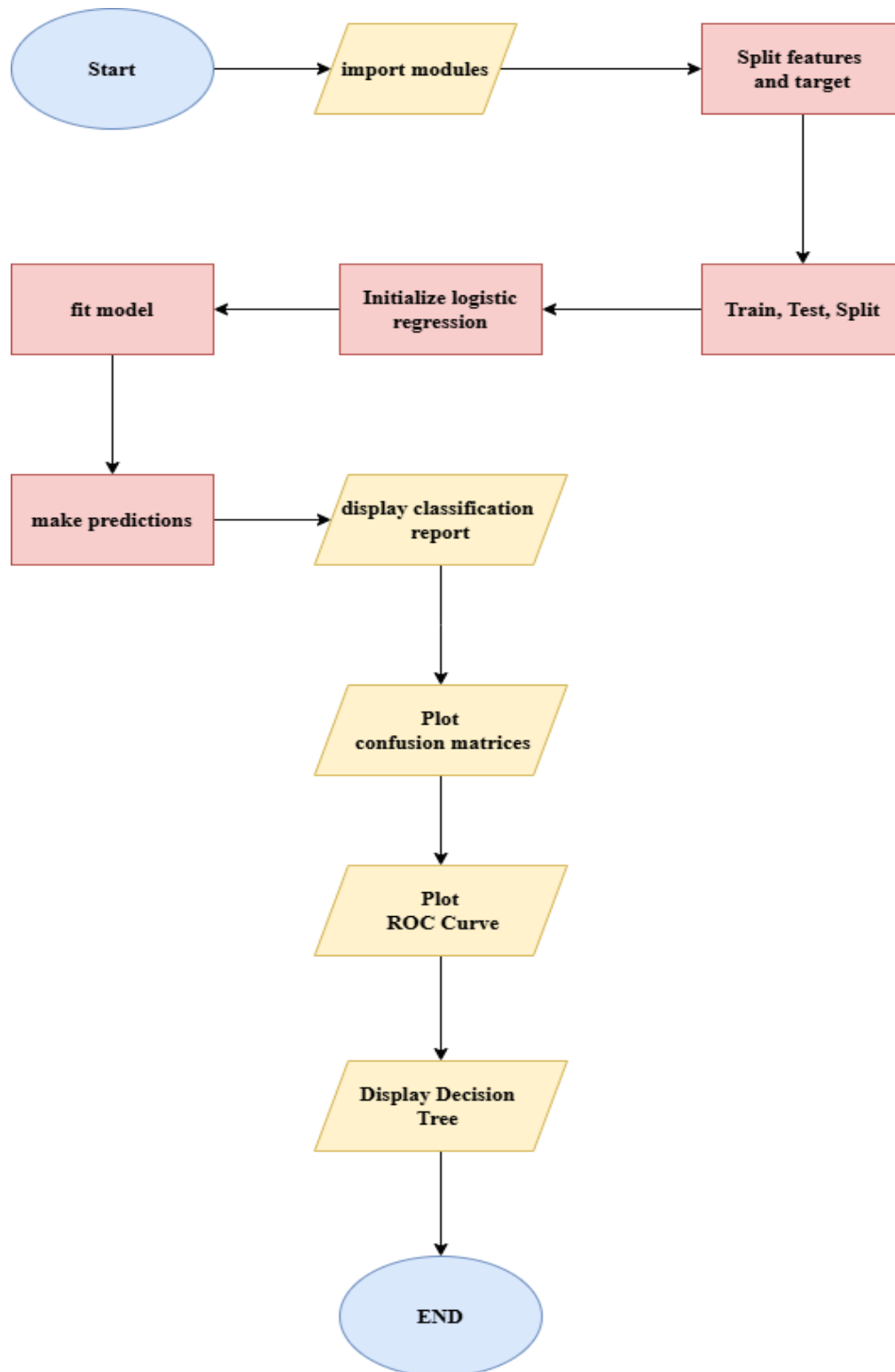


Figure 8:decision tree flowchart

### 3.2.2 Logistic Regression Flowchart

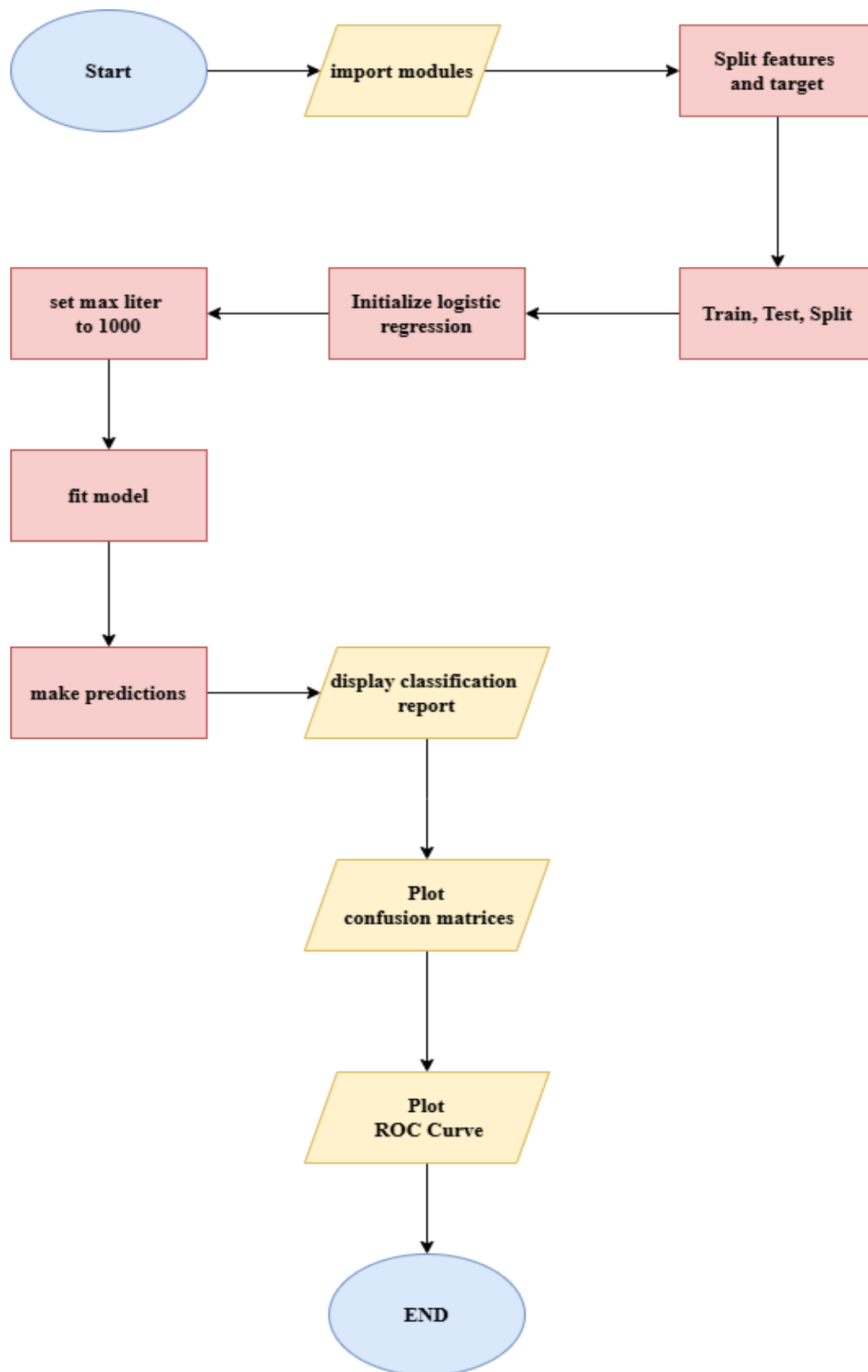


Figure 9: Logistic Regression Flowchart



### 3.2.3 Native Byers

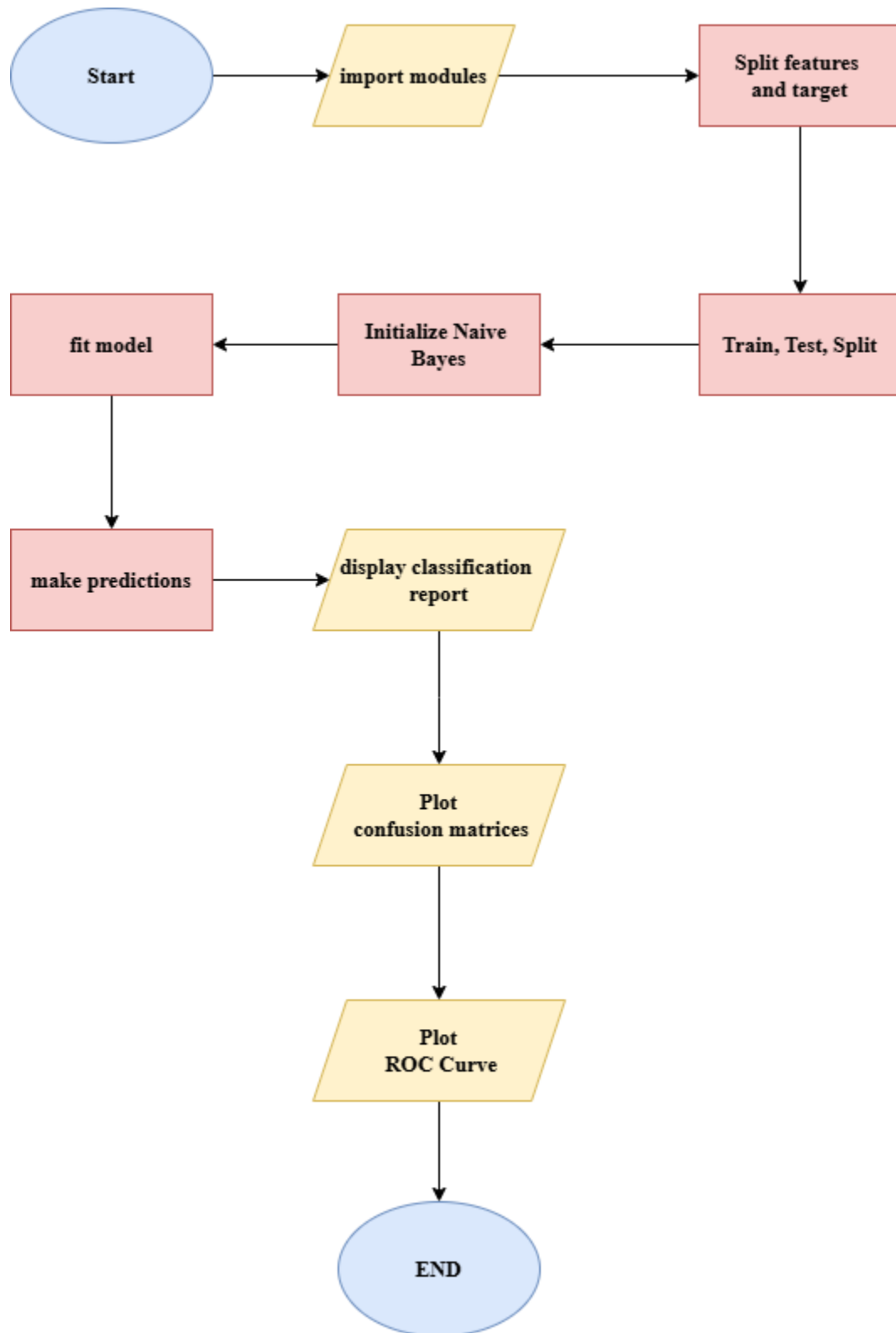


Figure 10: Native Byers flowchart

### 3.3 AI Algorithms Used

#### 3.3.1 Decision Tree

A Decision Tree is a supervised machine learning algorithm employed for classification and regression tasks (Trevor Hastie, 2024). It operates in a manner of repeatedly dividing data sets according to feature value and leading to a tree structure. The inner nodes give a decision based on an attribute, the branches are the outcome of the decision possessed by a node, and each node at the outermost level represents a prediction.

##### Key Components

- **Root Node:** The initial node that contains the entire dataset and makes the first decision.
- **Internal Nodes:** Nodes that split the data further based on a condition.
- **Leaf Nodes:** Terminal nodes that contain the output label or predicted value.

##### Working Mechanism

- **Splitting:** Dataset is partitioned into sub-sets by one feature with a condition, such as a threshold or feature category. In this context, the focus is to achieve high purity of the subsets.
- **Stopping Criteria:** Splitting stops when all data points in a node belong to the same class (pure node), a predefined maximum depth of the tree is reached, or the minimum number of samples required to split is not met.
- **Prediction:** For classification tasks, the majority class in a leaf node is the predicted class. For regression tasks, the average value in a leaf node is the predicted value.

Splitting Criteria for Decision Tree:

➤ **Entropy**

Entropy is a measure of purity or the degree of uncertainty, impurity, or disorder of a random variable in dataset. It is, in essence, the assessment of impurity or unpredictability in data points (shiksha online, 2025). It's value ranges from 0 to 1 where sometimes, it can go higher than 1 depending upon the number of features in the dataset.

Mathematical formula for Entropy is:

$$Entropy = \sum_{i=1}^c - p_i * \log_2(p_i)$$

*Equation 1: Entropy Equation (shiksha online, 2025)*

Where,

P<sub>i</sub>: Proportion of instances belonging to class

c: Total number of classes.

➤ **Gini Impurity**

Gini Impurity is a binary measure of how a split reduces Gini Index, Gini Index representing entropy. Describing Gini impurity on a scale of 0-1 for all, 0 for all elements belonging to the same class and 1 for only existence of one class. A Gini impurity of 1 suggests that all items are scattered randomly across various classes, whereas a value of 0.5 shows that the elements are distributed uniformly across some classes (shiksha online, 2025).

$$Gini = 1 - \sum_{i=1}^n (p_i)^2$$

*Equation 2: Gini Impurity Equation (shiksha online, 2025)*

Where,

Pi: Proportion of instances belonging to class

c: Total number of classes.

➤ **Information Gain**

The Information Gain is related to the determination of which attributes have a most relative amount of information about a class. The entropy principle is applied about targeting at minimizing entropy from the root node down to the leaf nodes. Information gain is the difference in entropy before and after splitting, which describes the impurity of in-class items (shiksha online, 2025).

$$\text{Information Gain} = 1 - \text{Entropy}$$

*Equation 3: Information Gain*

**Advantages**

- Easy to interpret and visualize.
- Handles both categorical and numerical data.
- Nonlinear relationships between features are captured.

**Disadvantages**

- Overfit without proper regularization
- Small variations in data can lead to different trees.

### 3.3.2 Logistic Regression

Logistic Regression is a supervised learning algorithm used for binary and multi-class classification tasks (David W. Hosmer, 2024). However, it is found regression algorithm used for the classification problems which gives a probability of a given class using log (or sigmoid) function. The logistic function ensures that what is being outputted are probabilities that fall between 0 and 1 making the model ideal for classification.

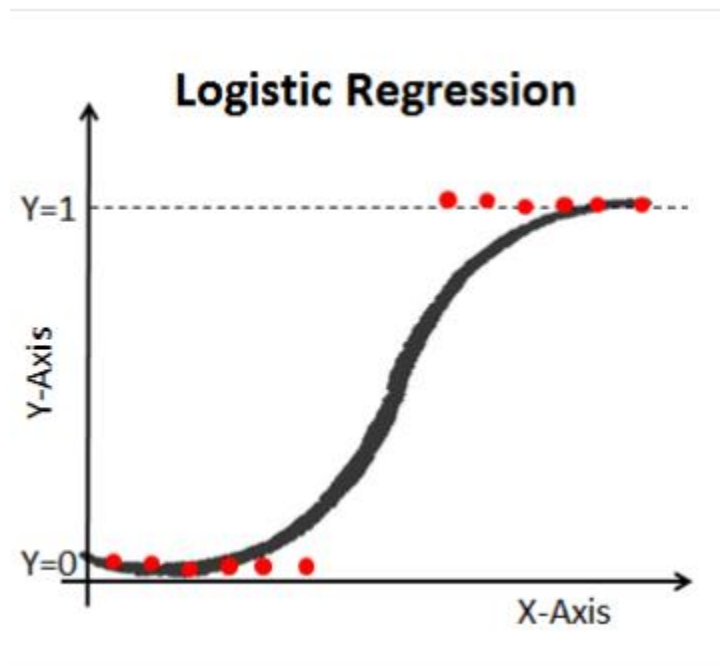


Figure 11: Logistic Regression (datacamp, 2025)

## Key Components

- **Logistic Function (Sigmoid Function):** Converts the output of a linear equation into a probability.
- **Probability Threshold:** The predicted probability  $h(x)$  is compared to a threshold (commonly 0.5).
- **Cost Function:** Logistic Regression uses the **log-loss** or binary cross-entropy as its cost function.
- **Gradient Descent:** Optimizes the model coefficients ( $\beta$ ) by minimizing the cost function.
- **Multinomial Logistic Regression:** Extends binary logistic regression to handle multiple classes using the SoftMax function

## How it works

- Fit linear model to the data

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

*Equation 4: Linear Regression (datacamp, 2025)*

Where,

Y: Linear combination of features and coefficients.

$\beta_0$ : Intercept term.

$X_1$  = Coefficient of the feature

- Apply the sigmoid function:

$$p = \frac{1}{1 + e^{-y}}$$

*Equation 5: Sigmoid Function*

Applying sigmoid function on linear model:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}}$$

*Equation 6: Sigmoid function on linear model*

**Advantages:**

- Effective for linear separable data
- Provides probabilistic interpretation
- Simple and efficient for small dataset

**Disadvantages**

- Sensitive to outliers
- Not suitable for complex, nonlinear problems



### 3.3.3 Gaussian Naive Bayes

Naive Bayes is a supervised machine learning algorithm based on Bayes' Theorem, primarily used for classification tasks. It assumes that features are independent from each other if the class label is given, thus making the computations easier and the algorithm fast. Even with this “naive” assumption it proves to be relatively efficient in several practical applications.

Key Concepts:

- **Bayes Theorem:** The algorithm relies on Bayes' Theorem, which calculates the probability of a class given the input features
- **Classification:** The algorithm predicts the class with the highest posterior probability

How it works:

Bayes Theorem calculates the likelihood of the event occurring before it occurs, based on prior knowledge of conditions related to the event. It is based on the following formula:

$$p(A/B) = \frac{p(B/A) \cdot p(A)}{p(B)}$$

*Equation 7: Bayes Theorem Equation (geekforgeeks, 2025)*

Where,

P(A|B): probability of A is calculated when B is already provided.

P(B|A): probability of B given the occurrence of A

P(A): previous probability of A

P(B): previous probability of B

Gaussian Naive Bayes is the application of Naive Bayes on a normally distributed data. Gaussian Naive Bayes assumes that the likelihood ( $P(x_i / y)$ ) follows the Gaussian Distribution for each  $x_i$  within  $y_i$  (geekforgeeks, 2025) . Therefore,

$$P(x_i / y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

*Equation 8: Gaussian Naive Bayes*

## 3.4 Development Process

### 3.4.1 Tools Used

- **Python**

Python is a versatile first level programming language used mostly in web application and artificial intelligence.



*Figure 12: python figure*

- **Jupyter Notebook**

Jupyter Notebook is an open-source IDE used to create online documents with live code.



*Figure 13: Jupyter Notebook*

### 3.4.2 Toolkit used

- **Pandas**

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive (pypi , 2025). In this project, pandas is used for manipulating data structures and data processing, especially for importing csv into a Data Frame

- **Matplotlib**

Matplotlib is a Python library that allows you to create static, animated, and interactive visualizations. In this project, it is used to plot charts and confusion matrix

- **Sklearn**

Sklearn is most used and powerful Machine Learning library in python which is used for machine learning and statistical modeling (tutorialpoint , 2025). In this project we have used Sklearn for evaluating models, Classification Report and Confusion Matrix.

### 3.4.3 Explanation of Development Process

- **Importing Libraries**

All the necessary libraries for development are imported

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, ConfusionMatrixDisplay
```

*Figure 14: importing libraries*

- **Reading the dataset**

Then the raw dataset that is in CSV format is imported and read in a Data Frame.

```
df=pd.read_csv("Cardiovascular dataset.csv")
```

*Figure 15: Reading the dataset*

- **Data Exploration and Visualization**

First, we take a look at the data extracted from the dataset, including the data types and number of not null values using `.info()` method

```
df.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70000 entries, 0 to 69999
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   id              70000 non-null  int64
1   age             70000 non-null  int64
2   gender          70000 non-null  int64
3   height          70000 non-null  int64
4   weight          70000 non-null  int64
5   ap_hi           70000 non-null  int64
6   ap_lo           70000 non-null  int64
7   cholesterol     70000 non-null  int64
8   gluc            70000 non-null  int64
9   smoke           70000 non-null  int64
10  alco            70000 non-null  int64
11  active          70000 non-null  int64
12  cardio          70000 non-null  int64
dtypes: int64(13)
memory usage: 6.9 MB
```

*Figure 16: information of the dataset*

Now, as we are about to predict the cardio column of the dataset, we can check how many of the rows have cardiovascular disease or not.

```
plt.figure(figsize=(8,8))
df['cardio'].value_counts().plot(kind='pie',autopct='%1.1f%%')
plt.title("Cardiovascular chart")
```

✓ 0.1s

Text(0.5, 1.0, 'Cardiovascular chart')

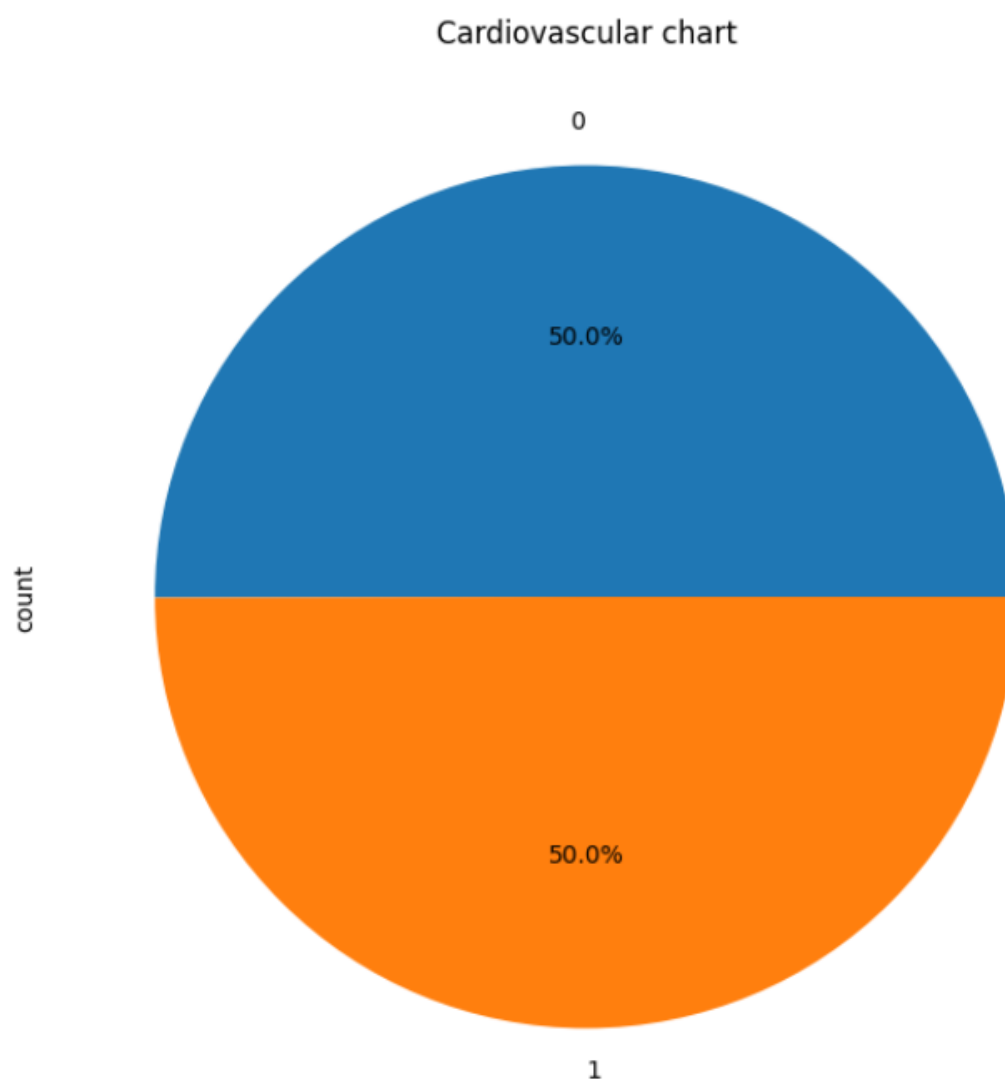


Figure 17: pie chart

- **Data Cleaning**

After gaining a good understanding of the data, the data cleaning phase is initiated. In the data cleaning phase, the primary goal is to ensure the dataset's integrity by addressing missing values and duplicate values.

The initial step involved identifying the presence of null values in the dataset.

```
df.isna().sum()

✓ 0.0s
```

id	0
age	0
gender	0
height	0
weight	0
ap_hi	0
ap_lo	0
cholesterol	0
gluc	0
smoke	0
alco	0
active	0
cardio	0
dtype:	int64

*Figure 18: Data cleaning part 1*

Next step is to check if there are any duplicate values present in the dataset

```
# Check if there is a
df.duplicated().sum()

✓ 0.0s
```

np.int64(0)

*Figure 19: Data cleaning part 2*

Since, there are no duplicate values in the dataset we can do not need to perform any kind of operation for clearing it.



- **Separate train and test and data**

Here, the feature and category columns are separated into x and y axis then it is split into two groups for training and testing. The data is split into an 80/20 ratio where the former is train data, and the latter is test data. The random state parameter is set which provides the same split each time to check if the accuracy is consistent across runs.

```
x=df.drop(columns=['id','cardio'])  
y=df['cardio']
```

✓ 0.0s

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

✓ 0.0s

*Figure 20: train test and splitting data*

After these functions are applied, the data is now ready to be used in the models

- **Modeling**

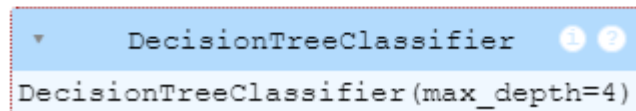
Three models have been selected for the dataset, Decision Tree Classifier, Naïve Bayes, and Logistic Regression

- **Decision Tree Classifier**

First of all, the Decision Tree Classifier model is initialized using `DecisionTreeClassifier()` class from scikit learn. Then the model is fit with training data that was modified in the previous phases.

```
model=DecisionTreeClassifier(max_depth=4)
```

```
model.fit(x,y)
```



DecisionTreeClassifier

DecisionTreeClassifier(max\_depth=4)

*Figure 21: Decision Tree Classifier modeling*

Then the test data is used to predict them by the model and a classification report of the test data is then computed. The classification report includes the accuracy for each class by presenting the precision, recall, f1-score and support.

```
y_prediction=model.predict(x_test)
```

```
print(f"accuracy: {accuracy_score(y_test,y_prediction)*100}")  
print(classification_report(y_test,y_prediction))
```

```
accuracy: 73.32142857142857  
      precision    recall  f1-score   support  
  
     0       0.73      0.74      0.73       6988  
     1       0.74      0.73      0.73       7012  
  
   accuracy          0.73          0.73          0.73      14000  
  macro avg          0.73          0.73          0.73      14000  
weighted avg          0.73          0.73          0.73      14000
```

*Figure 22: Decision Tree Classifier accuracy*

The confusion matrix for testing data is constructed and represented to understand the true positive, true negative, false positive and false negative values of the dataset

```
cm = confusion_matrix(y_test, y_prediction)
cmd = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=[0, 1])
cmd.plot(cmap="BuPu_r")
plt.title("Decision Tree - Confusion Matrix")
plt.show()
```

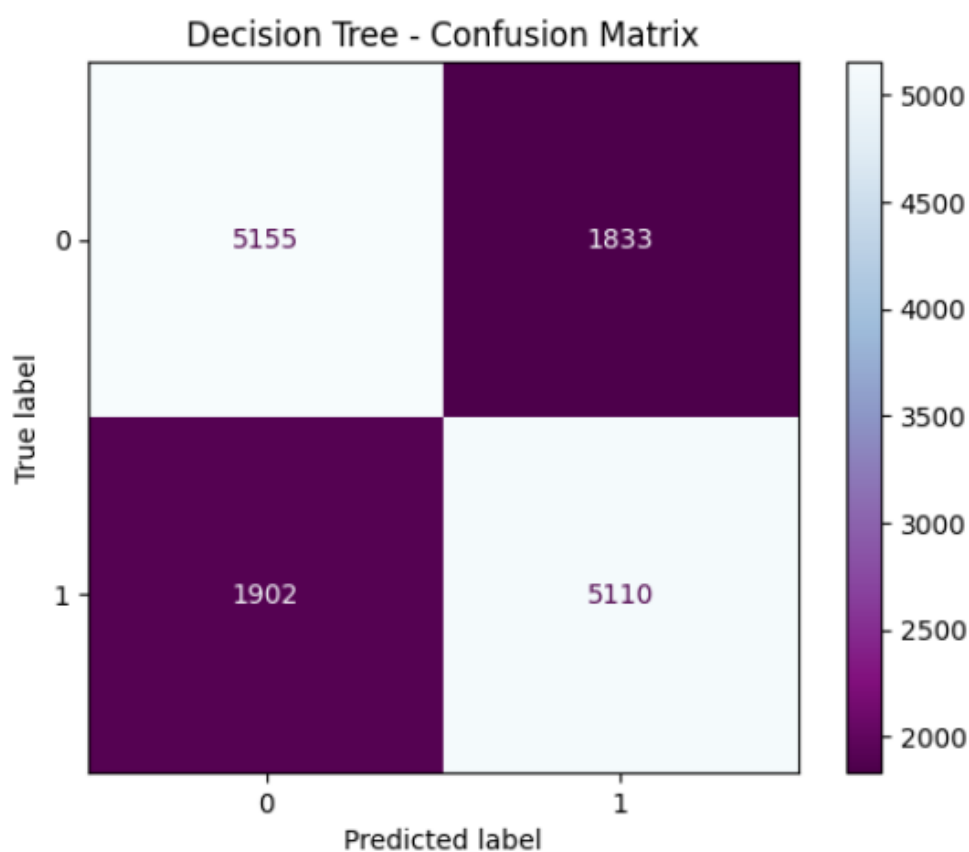


Figure 23: Classification Matrix Decision Tree

The ROC curve below shows how well the model can balance true positives and false positives, which gives the clear difference between classes at different thresholds.

```
problities = model.predict_proba(x_test)[:,-1]
fpr, tpr, _ = roc_curve(y_test_binarized, problities)
```

[133] ✓ 0.0s

```
plt.figure(figsize=(4, 4))
plt.plot(fpr, tpr)
plt.plot([0, 1], [0, 1], 'k--', lw=2)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Decision Tree - ROC Curve')
```

[134] ✓ 0.2s

```
... Text(0.5, 1.0, 'Decision Tree - ROC Curve')
```

```
...
```

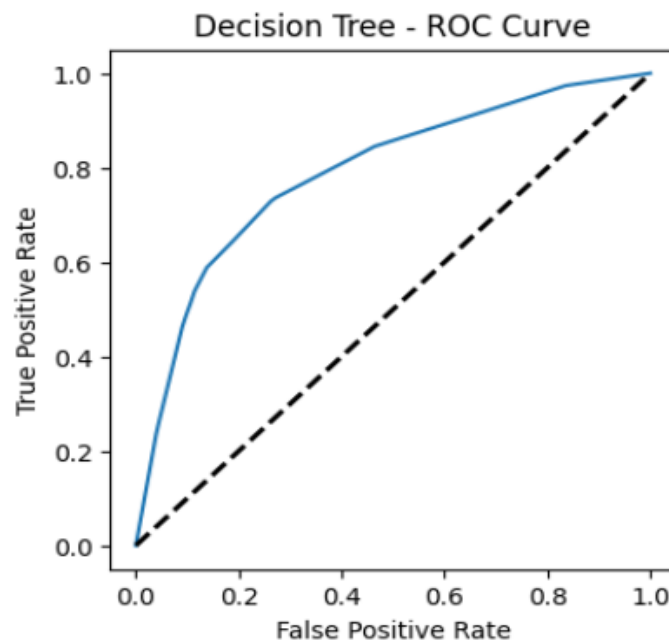


Figure 24: Decision Tree ROC curve

For we further plot classification tree for this model. Max depth for this tree in 4

Figure 25: plotting tree code snippet

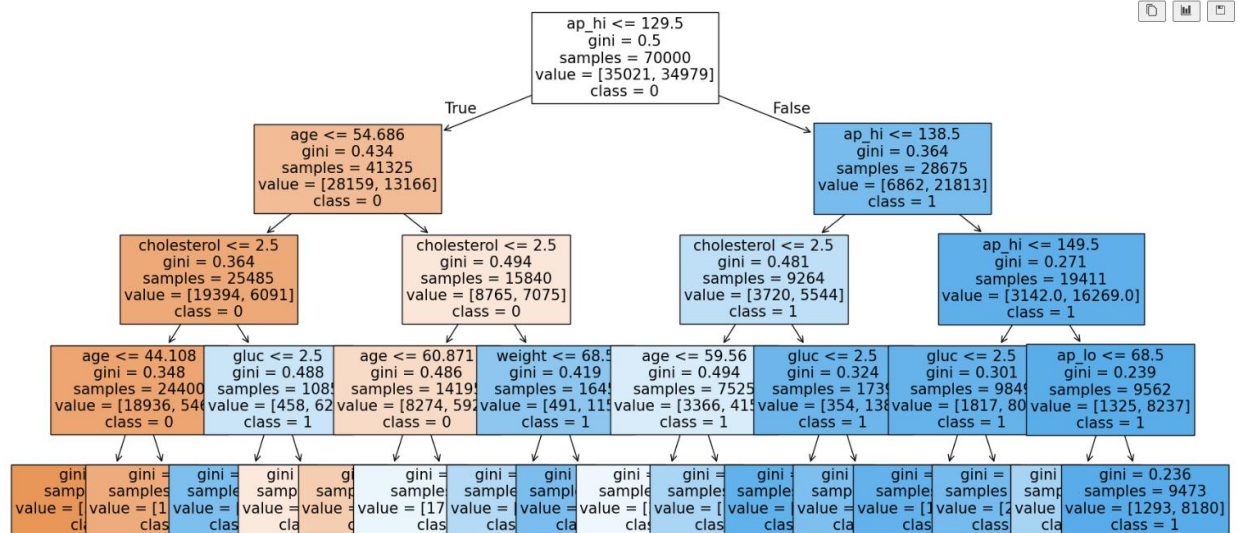


Figure 26: plotting tree graph

- **Logistic Regression**

First of all, the Logistic Regression model is initialized using `LogisticRegression()` class from scikit learn. Then the model is fit with training data that was modified in the previous phases.

```
model=LogisticRegression(max_iter=1000)
```

✓ 0.0s

---

```
model.fit(x_train,y_train)
```

✓ 2.4s

*Figure 27: Logistic Regression modeling*

Then the test data is used to predict them by the model and a classification report of the test data is then computed. The classification report includes the accuracy for each class by presenting the precision, recall, f1-score and support.

```
y_prediction=model.predict(x_test)
```

---

```
accuracy = accuracy_score(y_test, y_prediction)
print(f"Accuracy: {accuracy * 100:.2f}%\n")
print("Classification Report:")
print(classification_report(y_test, y_prediction))
```

Accuracy: 72.24%

Classification Report:

	precision	recall	f1-score	support
0	0.70	0.77	0.73	6988
1	0.74	0.68	0.71	7012
accuracy			0.72	14000
macro avg	0.72	0.72	0.72	14000
weighted avg	0.72	0.72	0.72	14000

*Figure 28: Logistic Regression model*

The confusion matrix for testing data is constructed and represented to understand the true positive, true negative, false positive and false negative values of the dataset

```
cm = confusion_matrix(y_test, y_prediction)
cmd = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=[0, 1])
cmd.plot(cmap="BuPu_r")
plt.title("LogisticRegression - Confusion Matrix")
plt.show()
```

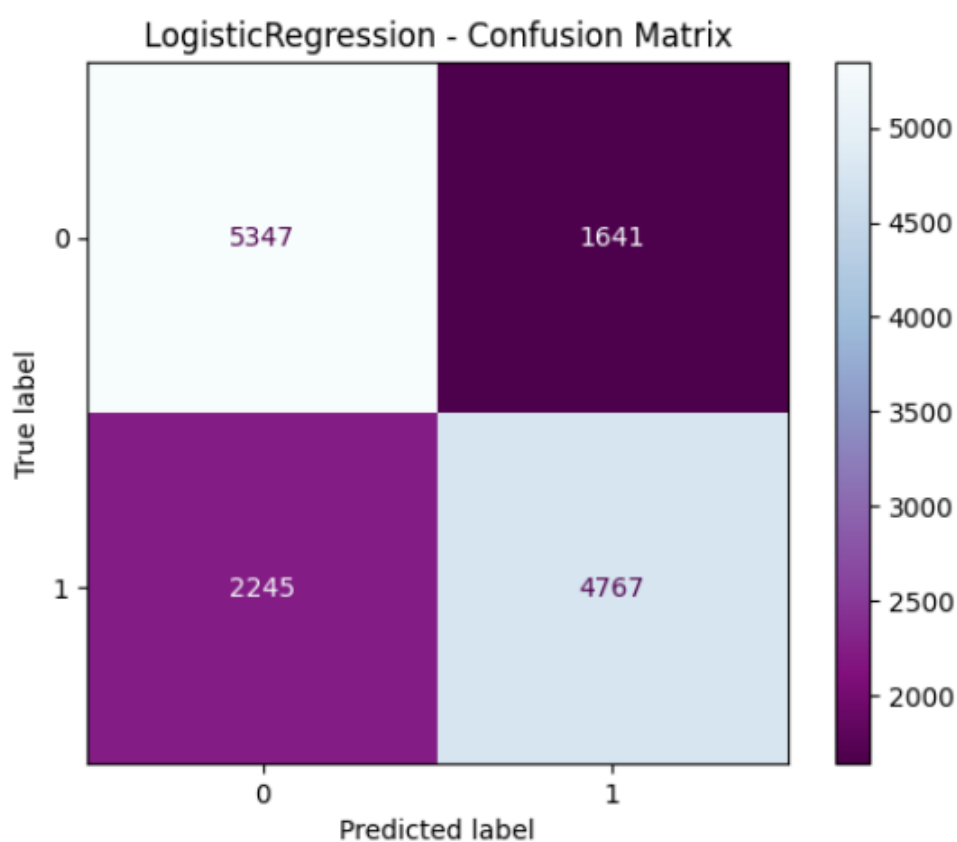


Figure 29: Classification matrix Logistic Regression

The ROC curve below shows how well the model can balance true positives and false positives, which gives the clear difference between classes at different thresholds.

```
problities = model.predict_proba(x_test)[: ,1]
fpr, tpr, _ = roc_curve(y_test_binarized, problities)
```

2] ✓ 0.0s

```
plt.figure(figsize=(4,4))
plt.plot(fpr, tpr)
plt.plot([0, 1], [0, 1], 'k--', lw=2)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Logistic Regression - ROC Curve')
```

3] ✓ 0.2s

Text(0.5, 1.0, 'Logistic Regression - ROC Curve')

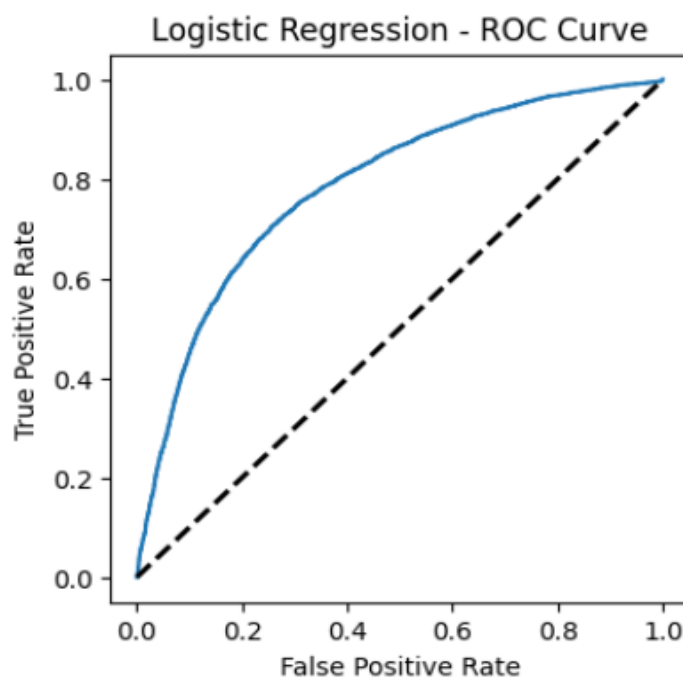


Figure 30: Logistic Regression ROC curve

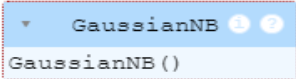


- **Gaussian Naïve Bayes**

Gaussian Naïve Bayes model is initialized using Gaussian Naïve Bayes class from scikit learn. Then the model is fit with training data that was modified in the previous phases.

```
model=GaussianNB()

model.fit(x_train, y_train)
```



*Figure 31: Gaussian Naive Bayes*

Then the test data is used to predict them by the model and a classification report of the test data is then computed. The classification report includes the accuracy for each class by presenting the precision, recall, f1-score and support.

```
y_prediction=model.predict(x_test)

print(accuracy_score(y_test, y_prediction))
accuracy=accuracy_score(y_test, y_prediction)*100

0.5932857142857143

print(classification_report(y_test, prediction))
```

	precision	recall	f1-score	support
0	0.70	0.77	0.73	6988
1	0.74	0.68	0.71	7012
accuracy			0.72	14000
macro avg	0.72	0.72	0.72	14000
weighted avg	0.72	0.72	0.72	14000

*Figure 32: Gaussian Naive Bayes model*

The confusion matrix for testing data is constructed and represented to understand the true positive, true negative, false positive and false negative values of the dataset

```
cm = confusion_matrix(y_test, y_prediction)
cmd = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=[0, 1])
cmd.plot(cmap="BuPu_r")
plt.title("Naive Bayes - Confusion Matrix")
plt.show()
```

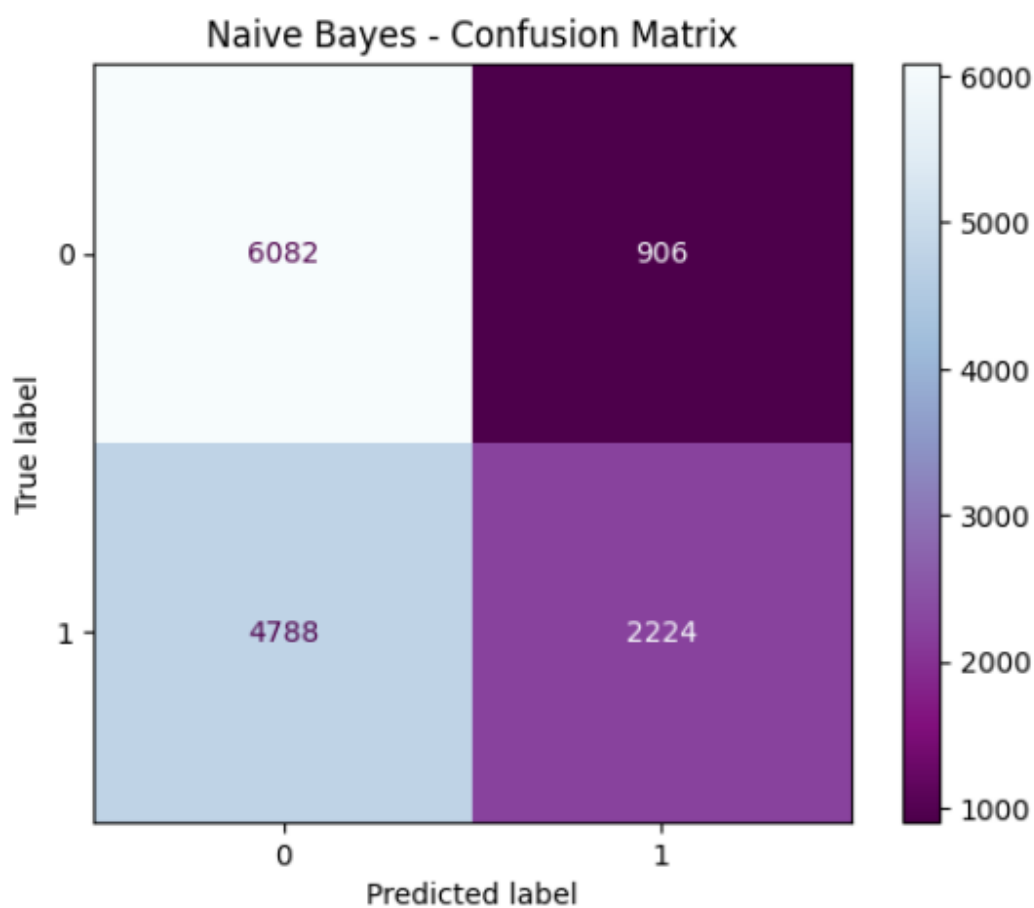


Figure 33: Confusion matrix Naive Bayes

The ROC curve below shows how well the model can balance true positives and false positives, which gives the clear difference between classes at different thresholds.

▷ ▾

```
problities = model.predict_proba(x_test)[: ,1]
fpr, tpr, _ = roc_curve(y_test_binarized, problities)
```

[150]

✓ 0.0s

▷ ▾

```
plt.figure(figsize=(4, 4))
plt.plot(fpr, tpr)
plt.plot([0, 1], [0, 1], 'k--', lw=2)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Naive Bayes - ROC Curve')
```

[152]

✓ 0.3s

... Text(0.5, 1.0, 'Naive Bayes - ROC Curve')

...

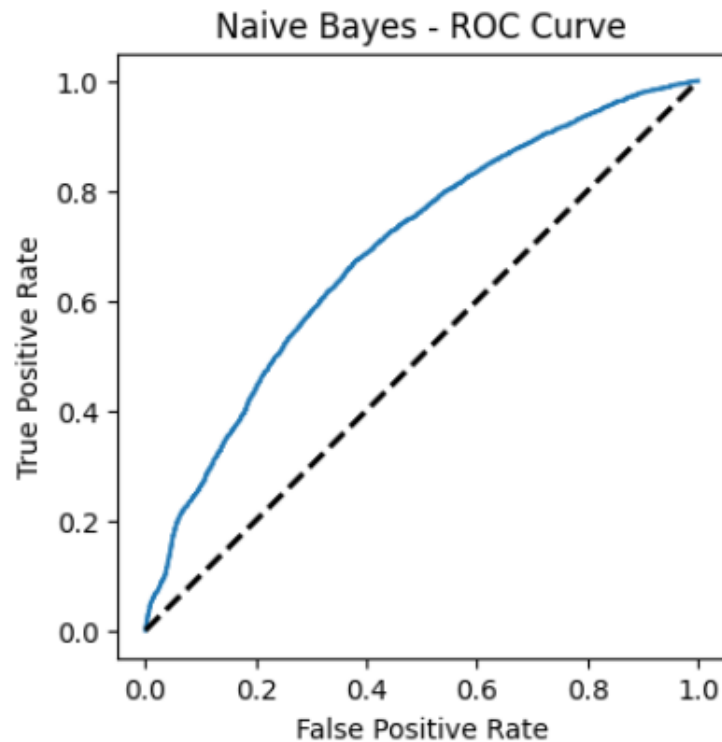


Figure 34: Naive Bayes ROC curve

- **Comparison Between All 3 metrices**

The visualization of confusion matrices and ROC curves demonstrates the effectiveness of the three models. The Naive Bayes classifier had limited success predicting class 1 as it recorded 2,224 accurate predictions but misidentified 4,788 cases as class 0. The ROC curve shows moderate distinction between positive and negative classes suggesting better methods should be found to separate them. Logistic Regression outperformed other models by correctly identifying 5,347 cases for class 0 and 4,767 cases for class 1 while incorrectly classifying 1,641 class 0 instances into class 1. The ROC curve should display strong class differentiation due to its balanced performance. The Decision Tree classifies 5,155 items as class 0 and 5,110 items as class 1 with the highest accuracy rate yet fails to correctly classify 1,902 instances of class 1 as class 0. The ROC curve reveals excellent class distinction confirming the model's reliable performance on this data set. The Decision Tree leads in performance, but ROC curves help us better understand how accurately each model can classify data.

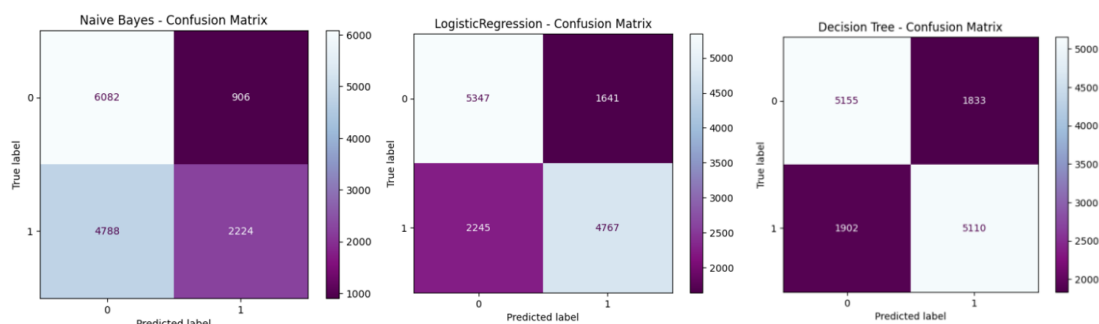


Figure 35: Confusion Matrix Comparison

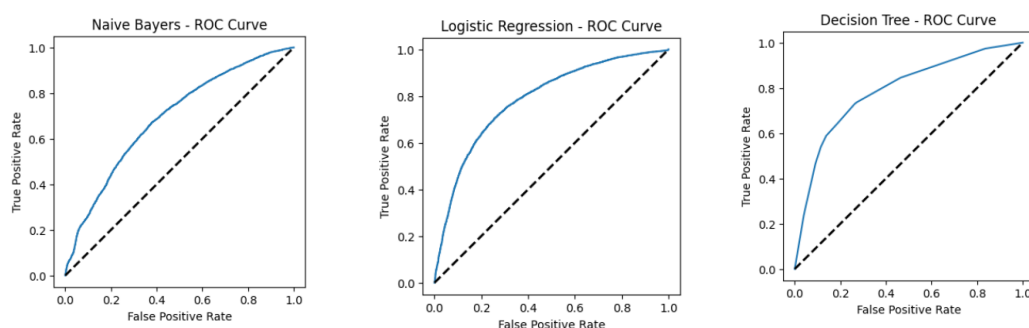


Figure 36: ROC curve Comparison

## 4 Conclusion

This project embarked seeks to employ Decision Tree, Logistic Regression, and Naive Bayes as algorithm frameworks to risk prediction of cardiovascular disease (CVD). Each algorithm was carried out on a data set consisting of health and lifestyle parameters that included, blood pressures, cholesterol levels, glucose levels, smoking habits and physical activity among others. Decision Tree gave the user understandable decision rules, Logistic Regression offered the probability of the inputs belonging to each of the groups and Naive Bayes proved to be the fastest. Each of these models was judged based on precision, accuracy, and interpretability; Logistic Regression was found to be precise, while Decision Trees were most interpretable. In the experiment conducted, naive Bayes was shown to be compact and ideal for use in large datasets.

To achieve this work, the project concentrated on features such as the Decision Tree, Logistic Regression, and Naive Bayes, which are the machine learning algorithms to predict CVD risk. Each algorithm was carried out on a data set consisting of health and lifestyle parameters that included, blood pressures, cholesterol levels, glucose levels, smoking habits and physical activity among others. Decision Tree gave the user understandable decision rules, Logistic Regression offered the probability of the inputs belonging to each of the groups and Naive Bayes proved to be the fastest. Each of these models was judged based on precision, accuracy, and interpretability; Logistic Regression was found to be precise, while Decision Trees were most interpretable. In the experiment conducted, naive Bayes was shown to be compact and ideal for use in large datasets.

**Limitations:**

- Naive Bayes assumes feature independence, which is often unrealistic in medical datasets where variables like cholesterol and glucose levels are interdependent.
- Decision Trees, particularly deeper ones, are prone to overfitting, reducing their generalizability to unseen data.
- In cases, where data structures are less straightforward, Logistic Regression has a limitation of assuming a linear relationship between predictor variables and the logarithm of odds of the response variable.
- While the algorithms were effective on the current dataset, handling significantly larger datasets with diverse attributes might require additional optimization.

**Future Directions**

- Combining models like Random Forest or Gradient Boosting with the existing approaches can improve overall accuracy and robustness.
- Exploring additional features such as genetic data or incorporating time-series health records could enhance model performance.
- For Logistic Regression, incorporating advanced regularization methods (e.g., Elastic Net) could help mitigate overfitting while preserving model performance.
- Creating hybrid models by integrating deep learning techniques (e.g., neural networks) with traditional algorithms may capture non-linear relationships and improve predictive power.
- Implementing real-time prediction capabilities using wearable health monitoring devices could make the system more practical and impactful.
- Expanding the solution to other domains, such as diabetes prediction or cancer risk assessment, could demonstrate its adaptability and utility.

## 5 References

- Achyut Tiwari. (2024, December 21). *deepai.org*. Retrieved from deepai web site:  
<https://deepai.org/publication/ensemble-framework-for-cardiovascular-disease-prediction>
- Achyut Tiwari, A. C. (2024, December 23). *arxiv*. Retrieved from arxiv web site:  
<https://arxiv.org/pdf/2306.09989v1>
- Aditya Ranade, Nitin Pise. (2024, Decmber 21). *Springer Nature Link*. Retrieved from Springer Nature Web site: [https://link.springer.com/chapter/10.1007/978-981-99-8479-4\\_34](https://link.springer.com/chapter/10.1007/978-981-99-8479-4_34)
- Arsalan Khan, M. Q. (2024, December 23). *onlinelibrary*. Retrieved from onlinelibrary website: <https://onlinelibrary.wiley.com/doi/full/10.1155/2023/1406060>
- Bhannu Prakash Doppala. (2024, December 21). *researchgate*. Retrieved from researchgate website:  
[https://www.researchgate.net/publication/348647049\\_A\\_Novel\\_Approach\\_to\\_Predict\\_Cardiovascular\\_Diseases\\_Using\\_Machine\\_Learning](https://www.researchgate.net/publication/348647049_A_Novel_Approach_to_Predict_Cardiovascular_Diseases_Using_Machine_Learning)
- Brown, S. (2024, December 21). *mitsloan*. Retrieved from mitsloan Web site:  
<https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained>
- datacamp. (2025, January 05). *datacamp* . Retrieved from datacamp website:  
[https://www.datacamp.com/tutorial/understanding-logistic-regression-python?utm\\_source=google&utm\\_medium=paid\\_search&utm\\_campaignid=19589720824&utm\\_adgroupid=157156376311&utm\\_device=c&utm\\_keyword=&utm\\_matchtype=&utm\\_network=g&utm\\_adpostion=&utm\\_creative=72](https://www.datacamp.com/tutorial/understanding-logistic-regression-python?utm_source=google&utm_medium=paid_search&utm_campaignid=19589720824&utm_adgroupid=157156376311&utm_device=c&utm_keyword=&utm_matchtype=&utm_network=g&utm_adpostion=&utm_creative=72)
- David W. Hosmer, J. S. (2024, December 24). *books.google*. Retrieved from books.google website:  
[https://books.google.com.np/books/about/Applied\\_Logistic\\_Regression.html?id=64JYAwAAQBAJ&redir\\_esc=y](https://books.google.com.np/books/about/Applied_Logistic_Regression.html?id=64JYAwAAQBAJ&redir_esc=y)
- Dr. R. Deepa, V. B. (2024, December 21). *pubs.aip*. Retrieved from pubs.aip web site:  
<https://pubs.aip.org/aip/adv/article/14/3/035049/3279524/Early-prediction-of-cardiovascular-disease-using>
- geekforgeeks. (2025, January 8). *geekforgeeks*. Retrieved from geekforgeeks website:  
<https://www.geeksforgeeks.org/gaussian-naive-bayes/>
- Glover, E. (2024, December 20). *builtin* . Retrieved from builtin website:  
[https://builtin.com/artificial-intelligence?trk=article-ssr-frontend-pulse\\_little-text-block](https://builtin.com/artificial-intelligence?trk=article-ssr-frontend-pulse_little-text-block)

Md Mamun Ali, B. K. (2024, December 21). *researchoutput*. Retrieved from researchoutput Web site:  
<https://researchoutput.csu.edu.au/en/publications/heart-disease-prediction-using-supervised-machine-learning-algori>

pypi . (2025, January 7). *pypi* . Retrieved from pypi website:  
<https://pypi.org/project/pandas/>

Science Direct. (2024, December 21). *Science Direct*. Retrieved from Science Direct website:  
<https://www.sciencedirect.com/science/article/pii/S0735109722072497?via%3Dihub>

shiksha online. (2025, January 08). *shiksha*. Retrieved from shiksha website:  
<https://www.shiksha.com/online-courses/articles/splitting-in-decision-tree/>

Sudarshan Singh, S. T. (2024, December 20). *thepharmajournal* . Retrieved from thepharmajournal website: <https://www.thepharmajournal.com/special-issue?year=2019&vol=8&issue=1S&ArticleId=25234>

Trevor Hastie, R. T. (2024, December 24). *springer*. Retrieved from springer web site:  
<https://link.springer.com/book/10.1007/978-0-387-84858-7>

tutorialpoint . (2025, January 7). *tutorialpoint* . Retrieved from tutorialpoint website:  
[https://www.tutorialspoint.com/scikit\\_learn/scikit\\_learn\\_introduction.htm](https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm)