



**Essay / Assignment Title:**

**Cardiovascular Disease Risk Detection using Machine Learning**

**Algorithms**

**Programme title:**

**Predictive Analytics and Machine Learning using Python**

**Name:**

**Mohammed Huraib Qureshi**

**Year:**

**2024**

# CONTENTS

|  |
| --- |
|  |

1. ABSTRACT …………………………………………………………………… 4
2. INTRODUCTION …………………………………………………………………… 5
3. METHODOLOGY …………………………………………………………………… 7
4. CHAPTER ONE: DATA EXPLORATION …………………………………………. 9 a. Data inspection
   1. Distribution of Numerical Variables
   2. Comparison with Target Variable
   3. Correlation Analysis
5. CHAPTER TWO: DATA CLEANING ANDPREPROCESSING ……………… 18 a. Outlier Removal
   1. Data Preprocessing and Splitting
6. CHAPTER THREE: MODEL SELECTION …………………………………………. 22 a. Logistic Regression
   1. Random Forest Classifier
   2. K- Nearest Neighbors
   3. Decision Tree Classifier
7. CHAPTER FOUR: MODEL BUILDING AND EVALUATION ……………………… 26 a. Model Building
   1. Model Performance
8. CHAPTER FIVE: HYPERPARAMETER TUNING ……………………. 33
9. CHAPTER SIX: MODEL EVALUATION …………………………………………. 39
10. CONCLUDING REMARKS …………………………………………. 45
11. REFERENCES ……………………………………….. 48

# ABSTRACT

|  |
| --- |
|  |

For a long time Cardiovascular diseases (CVDs) have been one of the leading causes of death around the world, contributing to the global health burden. With the advancements made in technology, Machine Learning algorithms provide a solution to this problem by early detection and prevention of these lifethreatening diseases. This research is solely based on leveraging these Machine learning models to create a model which successfully detects the risks of cardiovascular disease based on the personal health and lifestyle factors. This study utilizes a dataset consisting of 438,693 records records from the

2021 Behavioral Risk Factor Surveillance System (BRFSS), sourced from the World Health Organization (WHO). The dataset contains various health-related responses, such as smoking habits, physical activities, general health status, and chronic conditions like diabetes, which are very crucial in predicting CVD risk.

This study compares the performance of four machine learning models, namely Logistic Regression, KNearest Neighbours, Decision Tree, and Random Forest to determine which among these models offers best predictions for CVD risks. Each model was trained and evaluated using an 80:20 train-test split, which was then followed by hyperparameter tuning to optimize the performance of these models. The model’s performance was evaluated based on accuracy, precision, recall, and F1 score. The results show that all the models had their strengths and weaknesses.

By identifying key personal attributes and behaviours related with increased CVD risk, such as diabetes status, general health, and smoking history, this study contributes to endless applications of Machine Learning in healthcare industry. The findings hold high importance for developing data driven decision making processes aimed at reducing global impact of cardiovascular diseases.

# INTRODUCTION

Cardiovascular diseases (CVDs) are the leading cause of death around the world, contributing significantly to mortality rates. Early detection of these diseases in an individual at a risk is very important for prevention and management, potentially reducing the burden on the healthcare systems. In recent times, Machine learning algorithms have shown a great potential in identifying these individuals at high risks of CVDs. The relationships between various health-related variables and these CVDs are learned by these machine learning models to guide individuals to early intervention and prevention.

This study compares the performances on four widely used machine learning models namely, Logistic Regression, K-Nearest Neighbours, Decision Tree, and Random Forest for predicting CVD risk based on personal health related data from the Behavioral Risk Factor Surveillance System (BRFSS) in 2021. The dataset consists of over 438,000 records of individual responses related to health behaviour, conditions, and risk factors such as age, smoking history, Body Mass Index, general health status and exercise habits. Previous studies (Lupague et al., 2023) have shown that Logistic Regression is a reliable model for predicting CVD, given its simplicity and interpretability. However, tree-based models like Decision Tree and Random Forest are gaining popularity due to their ability to capture non-linear relationships and complex interactions between features.

In this project, we have followed a stepwise process to ensure accurate model comparison and performance optimization. After acquiring the dataset, we first checked for any kind of missing values and then addressed the outliers in our dataset using the IQR method for our model to be able to perform in the best way possible and detect at-risk individuals.

Next, each model was further enhanced using Hyperparameter tuning using a grid search to optimize key parameters. For Logistic Regression, the regularization parameter C was tuned to balance model complexity and overfitting issues. The KNN model we experimented with different values of K (the number of nearest neighbours) to identify the optimal number of neighbours. In Decision Tree, we tuned the maximum depth and minimum samples per leaf to prevent overfitting. For Random Forest, we tuned the number of trees and maximum features to improve the model’s predictive accuracy.

These models were then further evaluated using cross-validation to ensure that the results were robust. We compared the models using key performance evaluation metrics, including accuracy, precision, recall, and F1 score, we specially targeted the recall and F1 score due to the importance of minimizing false negatives in this case.

The research questions guiding this study are:

1. **Which machine learning model performs best for CVD prediction?** We aimed to identify the model that balances precision and recall most effectively for predicting CVD risk.
2. **How does hyperparameter tuning affect the performance of each model?** By optimizing key parameters for each algorithm, we sought to enhance their predictive accuracy.
3. **What insights can these models provide about CVD risk factors?** By analyzing the feature importance across models, we aimed to identify which personal health factors (e.g., diabetes, smoking, BMI) were most indicative of CVD risk.

By preprocessing the dataset, addressing class imbalance, and tuning hyperparameters we were able to achieve a comparison of these four widely used machine learning models, each offering unique insights. The result of this study tries to contribute to the ongoing research in the use of machine learning in healthcare domain, providing valuable insights into the strengths and the limitation of each of these models while predicting life-threatening conditions live CVD.

# METHODOLOGY

This section is to understand the methodology used to develop and implement the machine learning model for enhancing the telemarketing campaigns at HashSysTech Insurance. The approach encompasses data collection, preprocessing, feature selection, model development, evaluation, and lastly deployment.

1. **Data Collection**

The success of the predictive model is highly dependent on the quality of the data. For this project, personal health related data from the Behavioral Risk Factor Surveillance System (BRFSS) in 2021 was used.

This study utilizes a dataset consisting of 438,693 records records from the 2021 Behavioral Risk Factor Surveillance System (BRFSS), sourced from the World Health Organization (WHO). The dataset contains various health-related responses, such as smoking habits, physical activities, general health status, and chronic conditions like diabetes, which are very crucial in predicting CVD risk.

1. **Data Preprocessing**

Data preprocessing is critical to ensure the quality and usability of the dataset. The following steps were undertaken:

* + **Data Cleaning**: Missing values, duplicates, and inconsistencies if found were addressed.
  + **Outlier Detection and Removal**: Outliers were identified and removed using the Interquartile Range (IQR) method to avoid skewing the model.
  + **Feature Scaling**: Features were standardized or normalized to ensure that all variables contribute equally to the model, particularly for algorithms sensitive to feature scales, such as KNN.

1. **Model Development**

Several machine learning models were trained to predict the likelihood of heart disease, including logistic regression, random forest classifiers, and decision trees.

* + **Logistic Regression**
  + **K-Nearest Neighbors (KNN)**
  + **Random Forest Classifier**
  + **Decision Tree Classifier**

1. **Hyperparameter Tuning:**

To optimize model performance, hyperparameter tuning was conducted using GridSearchCV and RandomizedSearchCV. These techniques tested different combinations of parameters such as:

* + The number of estimators (for Random Forests)
  + The maximum depth of trees
  + The learning rate (for models like gradient boosting)

This step was critical in identifying the best-performing model configurations, ensuring improved accuracy and robustness across various metrics.

1. **Model Evaluation**

Model performance was assessed using the following metrics:

* + **Accuracy**: The proportion of correctly classified instances among the total instances.
  + **Confusion Matrix**: To evaluate the number of true positives, true negatives, false positives, and false negatives.
  + **Precision, Recall, and F1 Score**: For a more nuanced understanding of model performance, particularly for imbalanced datasets.
  + **ROC Curve and AUC**: To evaluate the models’ ability to distinguish between classes across different thresholds.
  + **Feature Importance:** The importance of different features was assessed to determine which factors had the strongest predictive power for cardiovascular disease. This was achieved using feature importance plots, primarily for tree-based models like the random forest.
  + **Validation:** The models were validated using cross-validation techniques to ensure robustness and avoid overfitting. Performance metrics from cross-validation were averaged to give a comprehensive view of model performance.

1. **Summary**

The methodology employed in this research involves a comprehensive approach to data collection, preprocessing, model development, and evaluation. By leveraging historical data and advanced machine learning techniques, the project aims to enhance the efficiency of CVD risk detection and drive significant improvements in healthcare domain.

# CHAPTER ONE: DATA EXPLORATION

**Data exploration** refers to the initial phase of data analysis where analysts examine and visualize a dataset to uncover its main characteristics and underlying patterns. This process helps in understanding the structure, distribution, and relationships within the data, which guides subsequent steps in data preparation and modeling. Data exploration typically includes summary statistics, visualizations (such as histograms, scatter plots, and box plots), and correlation analysis to identify trends, outliers, and anomalies in the data (Tukey, 1977). It plays a crucial role in identifying potential problems with the data, such as skewness or multicollinearity, and provides a foundation for selecting appropriate modeling techniques.

The dataset used in this research comprises of 308,850 records, each representing individual’s demographics, health characteristics and lifestyle choices. The dataset contains 19 variables in total, 12 of them are categorical and 7 of them are numerical. Table 1 provides us with the variable names and the datatype of variables.

This section provides an overview of the data exploration process.

**1.1. Data Inspection**

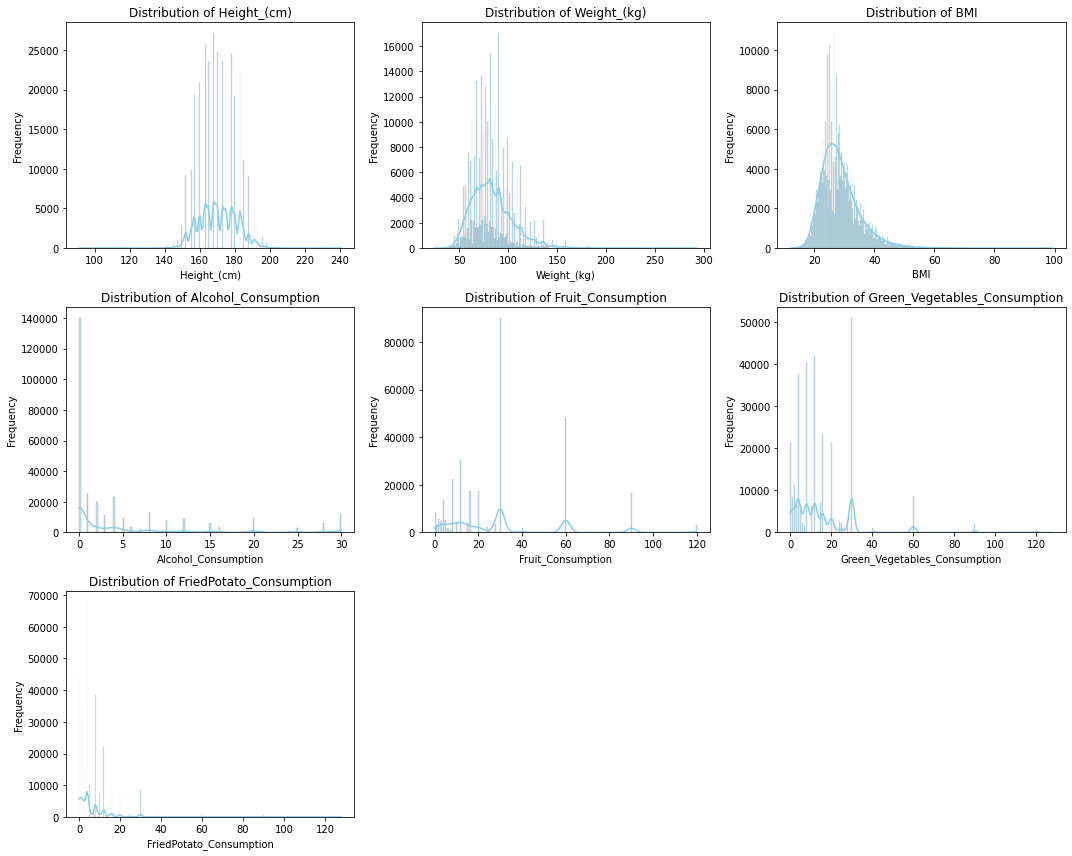
The dataset contains no missing values, and the variables encompass a range of health-related information. Key variables include:

* **General\_Health** (self-reported health status),
* **Heart\_Disease** (target variable indicating the presence or absence of heart disease),
* **BMI**, **Height**, and **Weight** (physical measurements),
* **Smoking\_History** and **Alcohol\_Consumption** (lifestyle factors), and
* **Fruit\_Consumption** and **Green\_Vegetables\_Consumption** (dietary habits).

The features are both categorical and numerical and help us understand the relationships between lifestyle and cardiovascular health.

**1.2 Distribution of Numerical Variables**

The numerical features show a variety of distributions which can be seen in Figure 1.



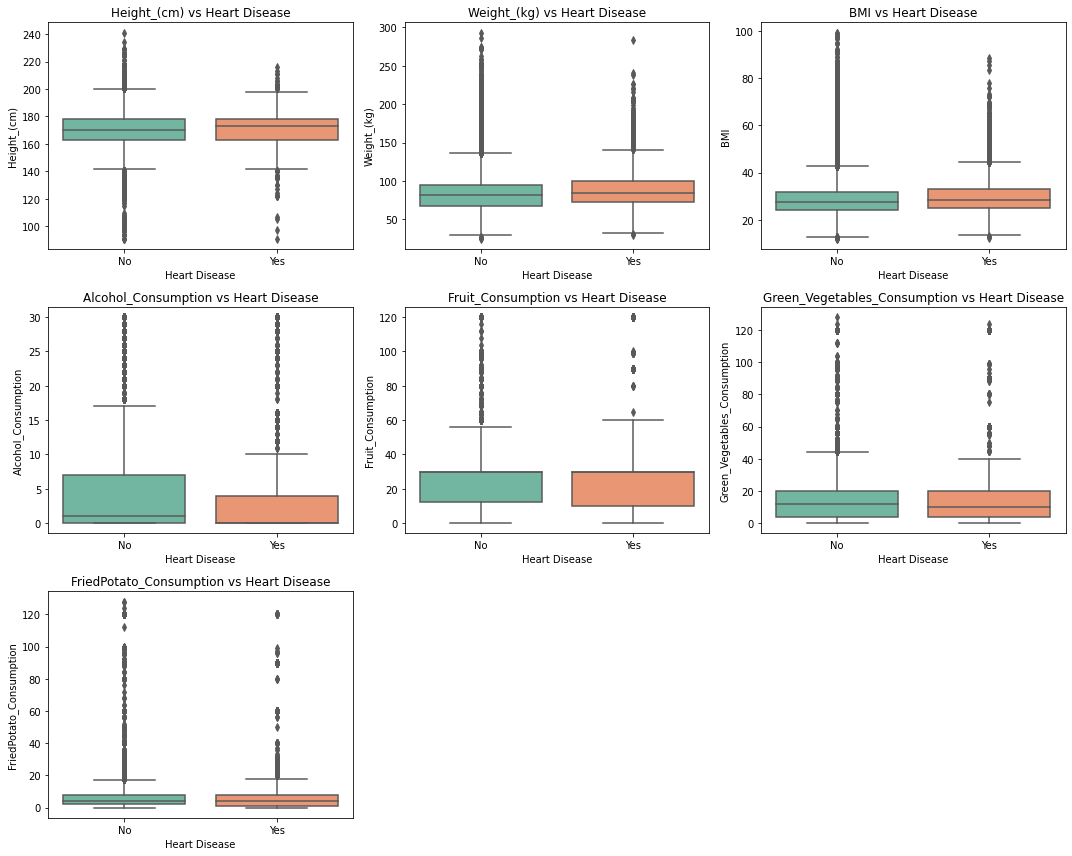
## Figure 1. Distribution of Numerical Variables

* **Distribution of Height (cm)** 
  + The distribution of the height is bell shaped which resembles normal distribution.
  + Most of the individuals have a height clustered around the mean value, with very less people being very short or tall. This is common for humans, therefore follows a normal distribution.

* **Distribution of Weight (kg)** 
  + The distribution of weight is right-skewed.
  + Most people have weight values clustered on the lower side, while fewer individuals have extremely high weights. Weight distributions often show this kind of skew due to extreme values.
* **Distribution of BMI (Body Mass Index)** 
  + The distribution for BMI is right-skewed.
  + High BMI corresponds to overweight or obesity being very rare in the population. Therefore the concentration is more on the lower range.
* **Distribution of Alcohol Consumption** 
  + The distribution of Alcohol consumption is highly right skewed.
  + There are very few people who consume high quantity of alcohol while the number of people consuming very low or no alcohol is quite high therefore there is a large spike at the low end.
* **Distribution of Fruit Consumption** 
  + The distribution of Fruit Consumption is irregular and right-skewed.
  + Most individuals have relatively low fruit consumption, with some spikes indicating certain common consumption levels. Fewer individuals consume large amounts of fruit.
* **Distribution of Green Vegetables Consumption**
* Like fruit consumption, this distribution is irregular and right-skewed.
* Most people consume low amounts of green vegetables, with some consuming more, though these are less frequent.
* **Distribution of Fried Potato Consumption** 
  + - This distribution is also right-skewed but flatter.
    - Many people have low fried potato consumption, but there’s a flatter distribution compared to fruit and vegetables, indicating a more varied consumption pattern, though high consumption is still rare.

**1.3 Comparison with Target Variable.**

Figure 2 shows the comparison between Numerical variables and the Target variables using Boxplot for better understanding the data we are dealing with.



**Figure 2. Boxplot between Target variable and Numerical Variables**

* **Height (cm) vs Heart Disease**

▪ The median of height for individuals with and without heart diseases does not have any significant difference. The interquartile ranges overlap a large amount suggesting height might not be a factor to predict heart diseases.

* **Weight (kg) vs Heart Disease**

▪ Individuals with heart diseases have a higher median of weight compared to the ones without heart problems. The boxplot shows wider range of weights in the “Yes” group, indicating more distribution in the weight of people with heart diseases.

* **BMI vs Heart Disease**

▪ The BMI of the individuals with heart diseases is generally higher as seen in the boxplot. There is also a greater distribution of BMI values for people with heart disease, suggesting BMI may be more closely associated with heart disease risk.

* **Alcohol Consumption vs Heart Disease**

▪ The median of the group of alcohol consumption of individuals without heart disease is slightly higher than those with heart disease, the difference however is very small. Both the groups have many outliers as can be seen in the boxplot.

* **Fruit Consumption vs Heart Disease**

▪ There is a little difference in the fruit consumption between those with and without heart disease. Both groups have a similar median, indicating that fruit consumption may not be a strong indicator of heart disease risks in this dataset.

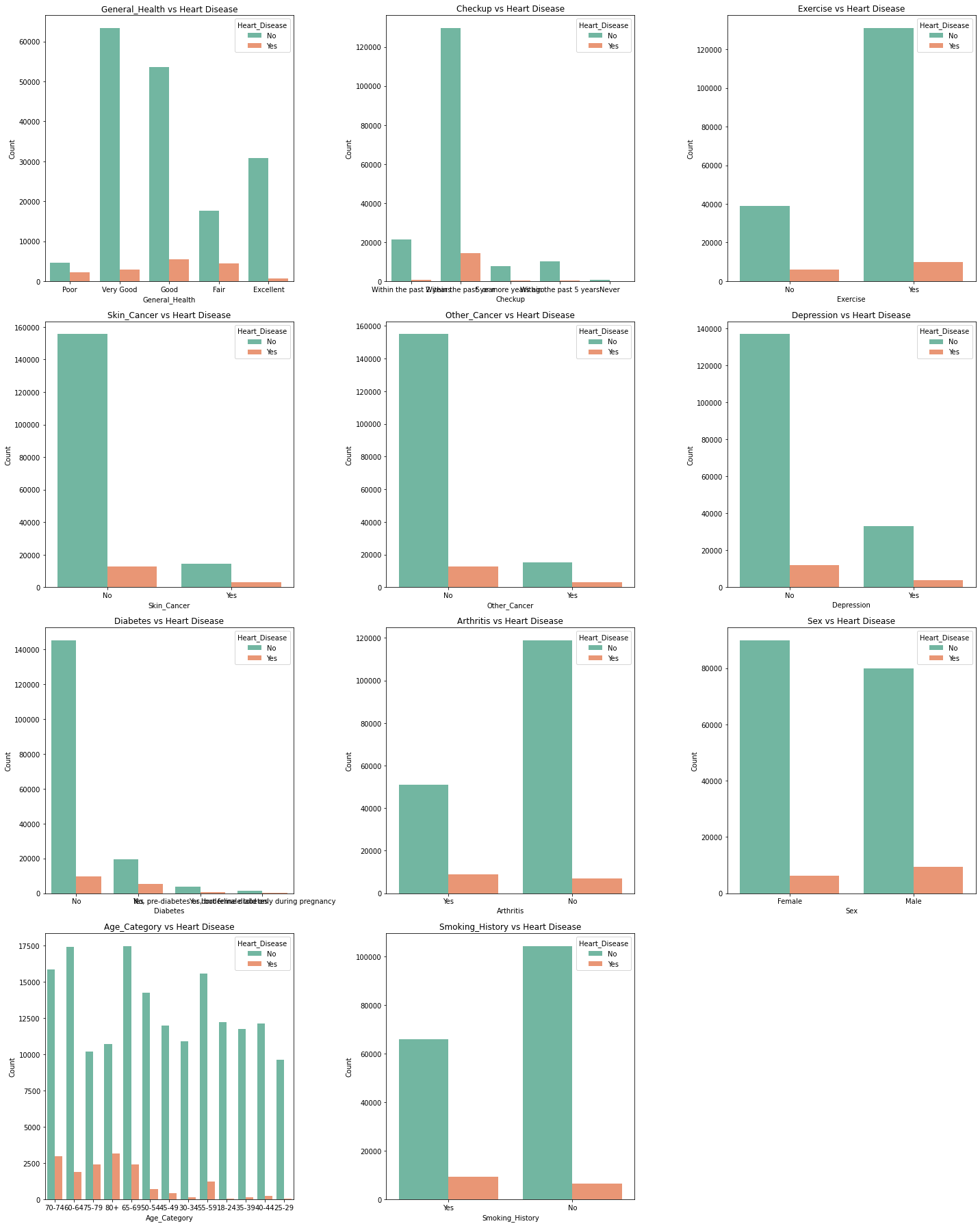
* **Green Vegetables Consumption vs Heart Disease**

▪ The median of green vegetable consumption is surprisingly higher for individuals with heart diseases. Although the overall difference between these groups is not very large, suggesting it is not a strong indicator for predicting heart disease risks.

* **Fried Potato Consumption vs Heart Disease**

▪ Both groups show similar median consumption of fried potatoes, with no significant difference between individuals with or without heart disease. The outliers in both groups indicate that some people consume significantly more fried potatoes, but overall, this factor doesn’t seem to have a strong relationship with heart disease.

Let’s further analyse and visualize the relationships between the categorical variables and the target variable using bar plot which can be seen in figure 3.



**Figure 3. Comparison between Categorical variables and the Target Variable.**

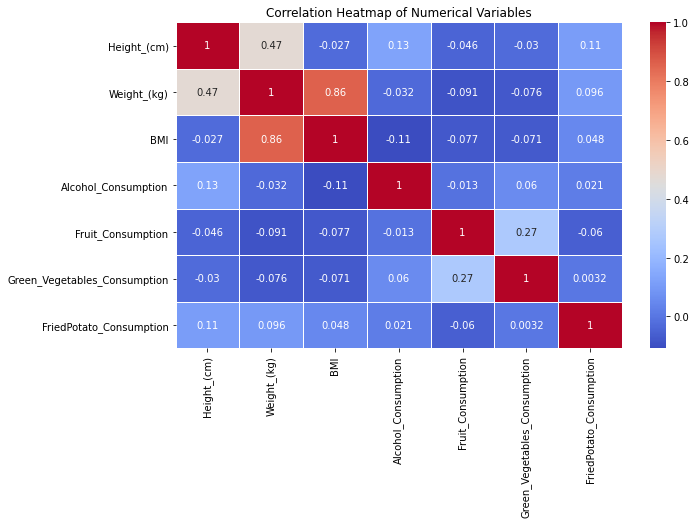
These graphs in Figure 3 illustrate the relationships between various categorical variables and the occurrence of heart disease.

* + 1. **General Health vs. Heart Disease**: The graph shows that the people who have reported their general health as “Poor” or “Fair” tend to have higher rate of heart diseases. People rating “Very Fair” and “Excellent” general health have very few cases of heart diseases.
    2. **Checkup vs. Heart Disease**: Individuals who go for less frequent checkups or no checkups at all tend to have heart diseases more than individual who have checkups within the past year. Regular checkups might correlate with better heart health monitoring.
    3. **Exercise vs. Heart Disease**: The chart highlights that people who do not engage in physical exercise are more likely to suffer from heart disease, compared to those who exercise regularly.
    4. **Skin Cancer vs. Heart Disease**: Most of the individuals in this dataset do not have skin cancer, however the proportion of heart diseases among those having skin cancer is slightly elevated.
    5. **Other Cancer vs. Heart Disease**: Just like skin cancer, the occurrence of heart disease is higher among individuals diagnosed with other forms of cancer, though the majority do not have heart disease.
    6. **Depression vs. Heart Disease**: This graph shows that heart diseases are more dominant in individuals suffering from depression suggesting potential relationship between mental health and cardiovascular health.
    7. **Diabetes vs. Heart Disease**: This graph shows that people without diabetes have a lower occurrence of heart disease. However, those with different types of diabetes show a higher proportion of heart disease cases. This suggests that diabetes is a notable risk factor for heart disease.
    8. **Arthritis vs. Heart Disease:** Individuals with arthritis are more likely to have heart disease compared to those without arthritis. However, the overall count of heart disease in people without arthritis is still higher due to the larger population size, but this suggests the association between arthritis and heart disease.
    9. **Sex vs. Heart Disease:** Males have a higher rate of heart disease compared to females, though the overall distribution shows a higher count of females in the dataset. This finding is consistent with other studies indicating that men may be at a higher risk for cardiovascular disease.
    10. **Age Category vs. Heart Disease:** The graph shows a clear trend where the proportion of heart disease increases with age. The highest rate of heart disease occurs in the oldest age group (80+), while younger age groups (29-44) exhibit very few cases of heart disease. This confirms that age is a major risk factor for heart disease.
    11. **Smoking History vs. Heart Disease:** People with a smoking history have a notably higher incidence of heart disease compared to non-smokers. This is in line with well-established research that smoking is a significant risk factor for cardiovascular problems.

**1.4 Correlation Analysis**

A correlation heatmap is a graphical representation used to display the correlation coefficients between multiple variables in a dataset. It provides a visual summary of the strength and direction of relationships between variables, typically ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation). Each cell in the heatmap represents a pairwise correlation, with color gradients indicating the magnitude of the correlation (McKinney, 2017). Correlation heatmaps are commonly used in exploratory data analysis to quickly identify relationships and potential multicollinearity among features in a dataset.

A correlation heatmap was generated to assess relationships between the numerical variables. Key findings include:



## Figure 4. Correlation Heatmap of Numerical Variables

* **Height and Weight:** 
  + The correlation between height and weight is at 0.47 which is moderately strong. Which can be satisfied as taller individuals tend to weight more, but the relationship is not perfect.
* **Weight and BMI:** 
  + The strongest correlation is between weight and BMI at 0.86, suggesting that weight significantly influences BMI (Body Mass Index), which is expected because BMI is calculated using weight.

* **Alcohol Consumption and Smoking:**
* A weak positive relationship can be seen between Alcohol consumption and smoking having a correlation of 0.27. People who consume alcohol might also likely smoke, but this is a weak relationship.
* **Other Health-Related Behaviours:** 
  + Variables like fruit consumption, vegetable consumption, and nutritional awareness have weak correlations with most other variables. This indicates that these factors might have a more complex or weaker direct relationship with anthropometric variables (height, weight, BMI) and other lifestyle factors like alcohol and smoking.

These exploratory insights laid the groundwork for further statistical analysis and modelling in the next phases of the research. The dataset’s diverse variables allow for comprehensive analysis of factors affecting cardiovascular health, offering a clear path for assessing the impact of lifestyle choices, demographic factors, and pre-existing health conditions on heart disease outcomes.

# CHAPTER TWO: DATA CLEANING AND PREPROCESSING

|  |
| --- |
|  |

Data cleaning and preprocessing are critical steps in the data science pipeline, ensuring that raw data is transformed into a suitable format for analysis. **Data cleaning** involves identifying and rectifying errors, inconsistencies, or missing values in the dataset to enhance data quality. This process typically includes handling outliers, correcting incorrect or incomplete entries, and ensuring consistency across data points (Kotsiantis, Kanellopoulos & Pintelas, 2006). **Data preprocessing**, on the other hand, is a broader concept that involves preparing the cleaned data for modeling. This includes tasks such as normalization, feature scaling, encoding categorical variables, and partitioning the data into training and test sets (Han, Kamber & Pei, 2011). Together, these steps help improve the performance and accuracy of machine learning models by ensuring that the data is structured and standardized.

**2.1 Outlier Removal:**

The following is a snippet of code used to remove the outliers from our dataset for our models to perform better.



The provided code removes outliers from multiple numerical columns using the **Interquartile Range (IQR)** method. The IQR method is commonly used to detect and eliminate outliers by calculating the range between the 25th percentile (**Q1**) and the 75th percentile (**Q3**) of the data. The outliers are defined as data points that fall below **Q1 - 1.5\*IQR** or above **Q3 + 1.5\*IQR** (Tukey, 1977).

* **IQR Function**:

▪ The function remove\_outliers(df, column) calculates Q1 and Q3 for the specified column. It computes the IQR and filters out rows where the column's values fall outside the lower and upper bounds (1.5 times the IQR below Q1 or above Q3). This ensures that outliers are removed, resulting in a cleaner dataset.

* **Outlier Removal**:

▪ The function is applied iteratively to a list of columns (['Height\_(cm)', 'Weight\_(kg)', 'BMI',

'Alcohol\_Consumption', 'Fruit\_Consumption', 'Green\_Vegetables\_Consumption', 'FriedPotato\_Consumption']), which contain the numerical data. For each column, the function recalculates the IQR and filters the dataset based on that specific column’s outliers.

* **Shape Checking**:

▪ After removing outliers, the final shape of the dataset is printed to check how many rows were removed.

The approach of using IQR to remove outliers is widely recognized for its robustness in handling skewed or non-normally distributed data (Rousseeuw & Leroy, 1987). However, it assumes that the data follows a distribution where outliers are located beyond the 1.5\*IQR threshold, which may not always be the case in every scenario.

**2.2 Data Splitting and Preprocessing:**

Data preprocessing is a crucial step in the machine learning pipeline, involving the preparation and transformation of raw data into a clean, structured format suitable for analysis. This process typically includes tasks such as handling missing values, normalizing, or scaling numerical data, encoding categorical variables, and removing outliers or irrelevant features. Effective preprocessing ensures that machine learning algorithms can process the data efficiently and improve the quality and accuracy of the models (Izenman, 2008). Preprocessing is important because raw data is often noisy, inconsistent, and may contain irrelevant features that can degrade the model's performance (Kotsiantis et al., 2006).



Above is the snippet of code that performs **data preprocessing** steps to prepare the dataset for machine learning models. Below is a breakdown of the key components of the code:

1. **Label Encoding:** 
   * The first part of the code converts the categorical variables into numeric formats using the LabelEncoder from scikit learn. This is a very important process as most of the machine learning algorithms generally work with only numerical data.
   * The LabelEncoder() transforms each category into a unique integer. This method is appropriate for ordinal or nominal categorical data (Pedregosa et al., 2011).
2. **Feature (X) and Target (y) Definition:** 
   * The dataset is split into **features (X)** and **target (y)**.
   * **X** contains all the input features (independent variables), while **y** is the column 'Heart\_Disease', which is the **target variable** or the dependent variable.
3. **Train-Test Split:**

Splitting data into training and testing sets is a method used in machine learning to evaluate the model's performance and generalization ability. Typically, the dataset is divided into two parts: the training set, which is used to train the model, and the testing set, which is kept separate to evaluate the model on unseen data. This approach helps to prevent overfitting, where the model performs well on the training data but poorly on new, unseen data. A common split ratio is 80% for training and 20% for testing, though the exact split can vary depending on the size of the dataset (Kuhn & Johnson, 2013). The use of train-test splitting ensures that the model's performance metrics, such as accuracy or F1 score, reflect its ability to generalize to new data (James et al., 2013).

* + The dataset is divided into training and testing sets using the train\_test\_split function from the scikit learn library.
  + **80% of the data** is used for training the model (X\_train, y\_train), while **20%** is reserved for testing (X\_test, y\_test). The split is crucial for evaluating the model's performance on unseen data, preventing overfitting (Kuhn & Johnson, 2013).

1. **Standard Scaling:** 
   * StandardScaler is used for numerical features scaling, which standardizes the data to have a mean of 0 and a standard deviation of 1. This is important for certain machine learning algorithms (e.g., Logistic Regression, K-Nearest Neighbors) that are sensitive to the scale of features (Izenman, 2008).
   * The scaler is fitted on the training data, then applied to both the training and testing sets to ensure consistency (Pedregosa et al., 2011).

Now the data is all prepared for model building which will be the next phase of the research.

# CHAPTER THREE: MODEL SELECTION

|  |
| --- |
|  |

**3.1. LOGISTIC REGRESSION:**

**Logistic regression** is a statistical and machine learning algorithm used for **binary classification** problems, where the target variable has two possible outcomes (e.g., 0 or 1). It models the relationship between one or more independent variables (features) and a binary dependent variable (target) using the logistic function (sigmoid function), which maps predicted values to probabilities between 0 and 1 (Hosmer et al., 2013). Unlike linear regression, which predicts continuous outcomes, logistic regression predicts the probability of a specific class label, typically using a threshold (e.g., 0.5) to assign the final class (Wright, 1995). This makes it suitable for classification tasks where the output is categorical.

**Why Use Logistic Regression for Cardiovascular Disease Prediction:**

**Logistic regression** is highly suitable for **Cardiovascular disease prediction** for several reasons:

1. **Binary Outcome**: Heart disease prediction is often framed as a binary classification problem (e.g., presence or absence of cardiovascular disease). Logistic regression is designed for binary outcomes, making it an ideal algorithm (James et al., 2013).
2. **Interpretable Model**: Logistic regression provides coefficients that represent the impact of each feature on the likelihood of heart disease. This interpretability helps medical practitioners understand which risk factors (e.g., age, cholesterol levels) contribute to heart disease (Hosmer et al., 2013).
3. **Efficient and Effective**: Logistic regression is computationally efficient and performs well when there is a linear relationship between the features and the target variable, which is often the case with medical data such as blood pressure, cholesterol, and body mass index (BMI) (Wright, 1995).
4. **Probabilistic Output**: The model's output is a probability score, which can be useful in medical decision-making. For instance, patients with a high probability of heart disease can be flagged for further medical investigation, while those with a low probability may require less urgent attention (Kuhn & Johnson, 2013).

Thus, in our research paper we start by using logistic regression model as it offers a balance between performance, simplicity, and interpretability, making is very useful for cardiovascular disease prediction.

**3.2. RANDOM FOREST CLASSIFIER:**

**Random Forest** is an ensemble learning algorithm used for both **classification** and **regression** tasks. It works by constructing multiple decision trees during training and outputting the class (for classification) or mean prediction (for regression) from the individual trees. Random forests operate by combining the predictions of numerous weak learners (decision trees) to create a strong learner, thus reducing overfitting, and improving model generalization (Breiman, 2001).

Each decision tree is trained on a random subset of the data, both in terms of samples and features. The final prediction is made through **majority voting** in classification problems or **averaging** in regression problems, which leads to more accurate and stable predictions. The algorithm leverages **bagging** (Bootstrap Aggregation) to improve model performance and decrease variance (Ho, 1995).

**Why Use Random Forest Classifier for Cardiovascular Disease Prediction:**

**Random Forest Classifier** is highly suitable for heart disease prediction due to several key reasons:

1. **Handling Complex Relationships**: Random forest can capture complex, non-linear relationships between features and the target variable, which is beneficial in medical datasets where multiple risk factors interact in predicting heart disease. For example, a combination of blood pressure, cholesterol levels, age, and BMI may be essential to predict heart disease (James et al., 2013).
2. **Robustness to Overfitting**: Unlike individual decision trees, which are prone to overfitting, random forests reduce the risk of overfitting by averaging the predictions from multiple trees, leading to a more **generalized model** (Breiman, 2001). This is important in healthcare, where making reliable predictions on unseen data is critical.
3. **Feature Importance**: Random forest provides insights into the **importance of features**. In the case of heart disease prediction, this allows medical professionals to identify the most influential factors (e.g., cholesterol, blood pressure, and smoking status) that contribute to heart disease risk. This can help in better diagnosis and understanding of risk factors (Kuhn & Johnson, 2013).
4. **Works Well with Imbalanced Data**: Medical datasets are often imbalanced, with fewer instances of patients having heart disease than those who do not. Random forests handle imbalanced datasets effectively by assigning weights to classes or by focusing on the more challenging cases (Chen et al., 2004).
5. **Versatility and Accuracy**: Random forests generally provide high accuracy and can handle both **numerical** and **categorical data** without requiring much preprocessing, making them versatile for medical datasets that often contain a mix of data types (Ho, 1995).

For our research the random forest classifiers make an excellent choice due to the advantages mentioned above. It provides a balance of accuracy, interpretability, and resistance to overfitting.

**3.3.K-NEAREST NEIGHBOUR CLASSIFIER:**

**K-Nearest Neighbours (KNN)** is a simple, instance-based machine learning algorithm used for both **classification** and **regression** tasks. In KNN, the algorithm predicts the class of a new data point by considering the classes of its **K nearest neighbours** (points closest in distance). The predicted class is usually determined by **majority voting** from these neighbours. For regression, the prediction is the average of the neighbours’ values (Cover & Hart, 1967).

The **distance metric** commonly used in KNN is the **Euclidean distance**, but other metrics such as Manhattan or Minkowski distance can also be applied depending on the problem. Since KNN is a **nonparametric** algorithm, it does not assume any underlying distribution for the data, making it highly flexible (Altman, 1992).

**Why Use K-Nearest Neighbours for Heart Disease Prediction:**

**K-Nearest Neighbours (KNN)** can be a valuable method for **heart disease prediction** due to several key reasons:

1. **Simple and Intuitive**: KNN is straightforward and easy to implement. It makes predictions based on the similarity of a patient's features to other patients in the dataset. This can be easily understood and explained in the medical domain, where understanding relationships between patient symptoms and outcomes is important (James et al., 2013).
2. **Effective for Non-Linear Data**: Heart disease prediction often involves non-linear relationships between features. For example, the combined effect of smoking, age, and food habits may not be linearly related to heart disease. KNN, which is a **non-parametric model**, does not assume a linear relationship and can model these complex interactions effectively (Hastie et al., 2009).
3. **No Training Phase**: Unlike other models that require a training phase, KNN works by storing the entire dataset and only makes decisions when a query point is provided (lazy learning). This can be beneficial in real-time medical applications where decisions need to be made instantly (Cover & Hart, 1967).
4. **Sensitive to Distance**: KNN relies on the **closeness** of points to make predictions. For heart disease, this means that patients with similar health profiles (e.g., similar blood pressure, cholesterol, BMI) are likely to have similar outcomes, allowing KNN to be an intuitive approach for predicting heart disease risk based on patient history (James et al., 2013).

However, **KNN can be computationally expensive** when working with large datasets and may perform poorly if there is **high dimensionality** or irrelevant features. Thus, it may require careful data preprocessing such as scaling or feature selection (Kuhn & Johnson, 2013).

**3.4. DECISION TREE CLASSIFIER:**

A **Decision Tree Classifier** is a popular, tree-based machine learning algorithm used for both **classification** and **regression** tasks. In a decision tree, the data is split recursively into subsets based on feature values that provide the best separation according to some criteria (e.g., **Gini impurity** or **information gain** for classification). The algorithm works by creating a tree-like model of decisions, where internal nodes represent feature tests, branches represent outcomes of these tests, and leaf nodes represent class labels or continuous values for regression (Breiman et al., 1984).

The main advantage of decision trees is their **interpretability**, as the structure resembles human decision-making processes. Additionally, decision trees are **non-parametric**, meaning they do not assume any specific distribution of the data (Quinlan, 1986).

**Why Use Decision Tree Classifier for Heart Disease Prediction:**

**Decision Tree Classifiers** are particularly suitable for **heart disease prediction** for several reasons:

1. **Interpretability and Transparency**: Decision trees are highly interpretable, which is important in healthcare applications. For heart disease prediction, a decision tree can provide clear, step-bystep explanations of how everyone’s features influence the model's decision.
2. **Feature Importance**: Decision trees can rank features by their importance, showing which factors most strongly influence heart disease predictions. This is useful in identifying key risk factors such as age, cholesterol levels, and blood pressure, allowing clinicians to focus on the most critical predictors in patient diagnosis (Hastie et al., 2009).
3. **Handling Non-linear Relationships**: Heart disease risk is influenced by a complex set of factors, and the relationships between these factors are often non-linear. Decision trees can model such non-linear interactions without requiring any transformations of the input data, which makes them highly adaptable to real-world medical datasets (Breiman et al., 1984).
4. **Handling Missing Data**: Decision trees can manage missing data without requiring extensive imputation, a useful feature in medical datasets where patient data may be incomplete. This can be beneficial in heart disease prediction when certain patient health indicators are unavailable (Quinlan, 1986).
5. **Versatility and Flexibility**: Decision trees can handle both **categorical** and **numerical** data, making them versatile for medical datasets that often include a mix of data types (Breiman et al., 1984).

Despite these advantages, decision trees can suffer from **overfitting**, especially if the tree is allowed to grow too deep. However, techniques like **pruning** or using ensemble methods (e.g., Random Forest) can mitigate this risk and improve model performance.

# CHAPTER FOUR: MODEL BUILDING AND EVALUATION

|  |
| --- |
|  |

**4.1 Model Building:**

**Model building** in machine learning refers to the process of selecting and developing an algorithm that captures patterns in data and uses these patterns to make predictions or decisions. It involves several stages, including **data preprocessing**, **feature selection**, **training** the model on a training dataset, and **evaluating** its performance on unseen test data. The goal is to create a model that generalizes well, meaning it can make accurate predictions on new, unseen data, not just the data it was trained on (Bishop, 2006).

Model building often requires iterating through different algorithms, tuning hyperparameters, and evaluating multiple performance metrics, such as accuracy, precision, recall, and F1 score, to select the best-performing model for the task at hand (Hastie et al., 2009). In many applications, particularly in fields like healthcare, finance, and engineering, building accurate predictive models can assist in critical decision-making processes.

The below code snippet shows us how we have built every model.



This code snippet implements four different machine learning classification models (Logistic Regression, Random Forest, K-Nearest Neighbors, and Decision Tree) to predict a target variable, likely for a classification problem such as heart disease prediction. Below is a detailed explanation of each step:

1. **Logistic Regression:** 
   * + **LogisticRegression**: This line creates an instance of the **Logistic Regression** model. We use random\_state=42 so that the model produces the same result each time the code is run.
     + **fit(X\_train\_scaled, y\_train)**: The model is trained using the scaled training data (X\_train\_scaled) and the target variable (y\_train). Logistic Regression requires features to be scaled, so the training data is scaled.
     + **predict(X\_test\_scaled)**: After training, the model is used to make predictions on the scaled test data (X\_test\_scaled). The predictions are stored in y\_pred\_logreg.
2. **Random Forest Classifier:** 
   * + **RandomForestClassifier**: An instance of the **Random Forest Classifier** model is created. The n\_estimators=100 specifies that the model will use 100 decision trees.
     + **fit(X\_train, y\_train)**: Random Forest does not require scaling. Therefore, model is trained using the training data (X\_train) and target variable (y\_train).
     + **predict(X\_test)**: The trained model is used to predict the target variable for the test data (X\_test), and predictions are stored in y\_pred\_rf.
3. **K-Nearest Neighbors (KNN):** 
   * + **KNeighborsClassifier**: An instance of the **K-Nearest Neighbors (KNN)** model is created.

n\_neighbors=5 is used to specify that the model will consider 5 nearest neighbours for prediction.

* + - **fit(X\_train\_scaled, y\_train)**: KNN is sensitive to scale of the data. Therefore, KNN model is trained using the scaled training data (X\_train\_scaled) and target variable (y\_train).
    - **predict(X\_test\_scaled)**: After training, the model predicts the target variable for the scaled test data (X\_test\_scaled). The predictions are stored in y\_pred\_knn.

1. **Decision Tree Classifier:** 
   * + **DecisionTreeClassifier**: An instance of the **Decision Tree Classifier** model is created. The random\_state=42 ensures reproducibility.
     + **fit(X\_train\_scaled, y\_train)**: The model is trained on the scaled training data (X\_train\_scaled) and the target variable (y\_train). In this case, scaling isn't typically necessary for Decision Trees, but it seems scaling was applied consistently for simplicity.
     + **predict(X\_test)**: The trained Decision Tree model is used to predict the target variable for the test data (X\_test). Predictions are stored in y\_pred\_dt.
   1. **Model Performance:**

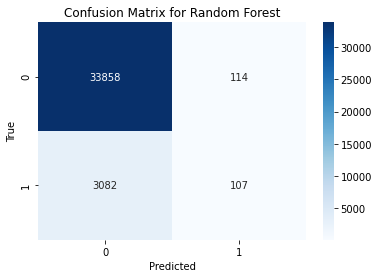
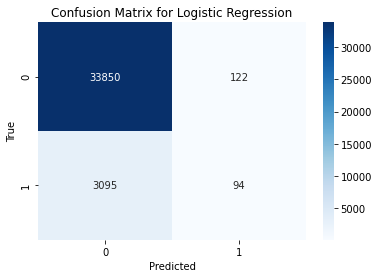
**4.2.1. Confusion Matrix:**

A **confusion matrix** is a performance measurement tool for classification models, providing a summary of prediction results on a classification problem. It is a **table** that compares the actual target values to the model’s predicted values. The matrix contains four key components: **True Positives (TP)**, **True Negatives (TN)**, **False Positives (FP)**, and **False Negatives (FN)**. These metrics help in calculating important performance measures such as **accuracy**, **precision**, **recall**, and the **F1 score** (Fawcett, 2006).

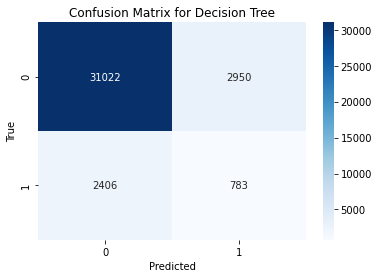
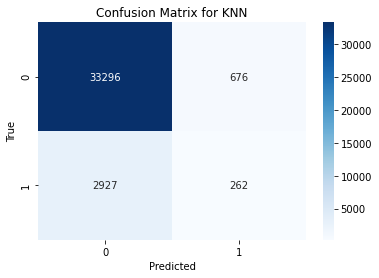
* **True Positives (TP)**: Instances where the model correctly predicted the positive class.
* **True Negatives (TN)**: Instances where the model correctly predicted the negative class.
* **False Positives (FP)**: Instances where the model incorrectly predicted the positive class (also known as a **Type I error**).
* **False Negatives (FN)**: Instances where the model incorrectly predicted the negative class (also known as a **Type II error**).

The confusion matrix is particularly useful in **imbalanced datasets**, where accuracy alone can be misleading, as it provides a more detailed picture of model performance by distinguishing between different types of errors (Sokolova & Lapalme, 2009).

Figure 5 and Figure 6 shows the confusion matrix of all four models’ Logistic regression, Random Forest and K Nearest Neighbours, Decision tree received after the fitting of the model without any kind of hyperparameter tuning.



## Figure 5. Confusion matrix of Logistic Regression and Random Forest



## Figure 6. Confusion Matrix of KNN and Decision Tree

**4.2.2. Accuracy Score:**

**Accuracy score** is a performance metric used to evaluate the effectiveness of classification models. It is defined as the ratio of correctly predicted instances (both **true positives** and **true negatives**) to the total number of predictions made by the model. In mathematical terms, it is expressed as:

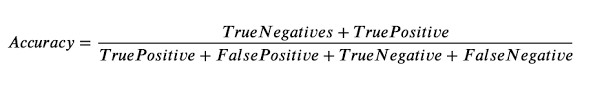


Figure 7 is a bar chart made to compare the accuracies of all the four models.



## Figure 7. Accuracy Comparison

Accuracy Comparison:

* Logistic Regression: 91.34%
* Random Forest: 91.40%
* K-Nearest Neighbors (KNN): 90.30%
* Decision Tree: 85.59%

The insights received from the bar graphs and the accuracy score are, the Random Forest model slightly outperforms Logistic Regression and KNN, the Decision Tree model on the other hand shows lower accuracy, making it less reliable for detecting Cardiovascular Diseases.

**4.2.3. Classification Report:**

Here is a summary of the results for precision, recall, and F1 score across all models:

**Logistic Regression:**

* Precision (Class 1): 0.44
* Recall (Class 1): 0.03
* F1-Score (Class 1): 0.06

The Logistic Regression model performs well for Class 0 which is no cardiovascular disease with high precision and recall, but it struggles to identify Class 1 which are individuals with cardiovascular disease risk, achieving a recall of only 3%. This indicates that very few positive cases are correctly identified.

**Random Forest:**

* Precision (Class 1): 0.48
* Recall (Class 1): 0.03
* F1-Score (Class 1): 0.06

Juts Like Logistic Regression, the Random Forest model also predicts Class 0 correctly but fails to effectively detect Class 1, although, performing slightly better in precision (0.48) for the minority class. Its performance highlights the difficulty in identifying cardiovascular disease risk.

**K-Nearest Neighbors (KNN):**

* Precision (Class 1): 0.28
* Recall (Class 1): 0.08
* F1-Score (Class 1): 0.13

The KNN model shows excellent performance for Class 0, but its effectiveness drops for Class 1, where it achieves a precision of 0.28 and a recall of 8%. This model offers a slight improvement in detecting cardiovascular diseases risk than the previous models.

Decision Tree:

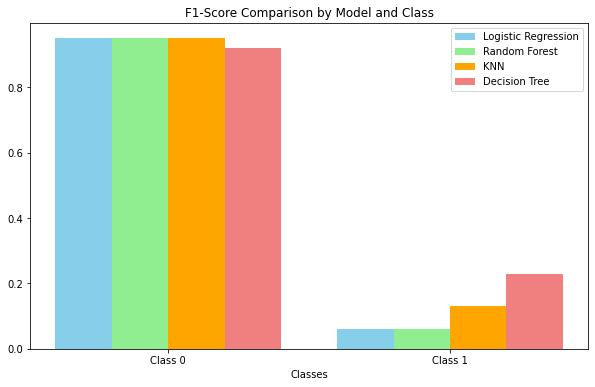
* Precision (Class 1): 0.21
* Recall (Class 1): 0.25
* F1-Score (Class 1): 0.23

Among the four models, the Decision Tree offers the best recall for detecting cardiovascular diseases at 25%. Although with a lower overall accuracy (85.59%), it demonstrates a more balanced detection capability for both classes, indicating a better ability to identify individuals at risk.

In our research of **cardiovascular disease detection**, the F1 score is crucial due to the imbalanced nature of our medical datasets, here the number of healthy individuals (negative class) may outnumber those with cardiovascular diseases risk (positive class).

1. **Prioritizing Patient Outcomes**: Accuracy can be misleading when the positive class is outnumbered. A model may achieve high accuracy by predicting the majority class correctly while failing to identify patients with cardiovascular diseases. In this scenario, a model with a high F1 score would better capture the effectiveness of detecting actual cases of disease (Davis and Goadrich, 2006).
2. **Balancing Precision and Recall**: A high F1-score indicates that the model has a good balance between precision and recall, which is very important. In CVD detection, it is very important to minimize both false positives and false negatives.
3. **Clinical Relevance**: Medical practitioners often rely on the F1 score to assess model performance comprehensively, ensuring that they can detect cardiovascular diseases effectively while also maintaining a lower rate of misdiagnoses (Fleiss, 1981). This balance is critical for improving patient outcomes and enhancing the reliability of clinical decision-making.

Figure 8 gives us a visual representation of how the models have performed based on the F1 score on both the Classes in the dataset.



**Figure 8. F1 Score Comparison**

# CHAPTER FIVE: HYPERPARAMETER TUNING

|  |
| --- |
|  |

Hyperparameter tuning refers to the process of optimizing the hyperparameters of a machine learning model to improve its performance on a specific task. Hyperparameters are parameters that are set before the learning process begins and are not learned from the data during training (Bergstra and Bengio, 2012). Unlike model parameters, which are adjusted through the learning process, hyperparameters control the learning process itself and can significantly influence the model's performance.

Hyperparameter tuning involves selecting the best combination of hyperparameters by evaluating the model's performance through methods such as grid search, random search, or more advanced techniques like Bayesian optimization. These methods systematically explore different configurations of hyperparameters using performance metrics (e.g., accuracy, F1 score) obtained from cross-validation or validation sets to identify the optimal settings that yield the best results (Hutter et al., 2011).

Effective hyperparameter tuning is crucial for developing robust machine learning models, as it helps to prevent overfitting, ensures generalization to unseen data, and ultimately enhances predictive performance (Bergstra et al., 2011).

**5.1 Logistic Regression Hyperparameter tuning:**

The below code snippet implements hyperparameter tuning for a Logistic Regression model using Grid Search Cross-Validation (GridSearchCV) from the scikit-learn library. Hyperparameter tuning is essential in machine learning as it helps improve model performance by systematically searching through combinations of parameter values (Hutter et al., 2011).



Lets understand the code better step by step:

* **Defining the Parameter Grid:**
* param\_grid is a dictionary specifying hyperparameters for tuning.
* The parameter C represents the inverse of regularization strength, where smaller values indicate stronger regularization to prevent overfitting (Hastie et al., 2009). The selected values (0.01, 0.1, 1, 10, 100) evaluate the impact of different regularization levels on model performance.
* **Setting Up Grid Search:**
* A **GridSearchCV** object named **grid\_log\_reg** is created with:
  + - **LogisticRegression(max\_iter=1000):** Initializes the model, allowing 1000 iterations for convergence.
    - **param\_grid**: Contains the hyperparameter options.
    - cv=5: Specifies 5-fold cross-validation, where the dataset is divided into five parts for training and validation (Kohavi, 1995).
    - scoring='accuracy': Optimizes the model based on accuracy on validation folds.
* **Fitting the Model:**
* The fit method on **grid\_log\_reg** begins the training process using the scaled training data **(X\_train\_scaled)** and target labels **(y\_train)**, exploring all hyperparameter combinations.
* **Extracting Best Parameters and Score:**
* **best\_logreg\_params** retrieves the hyperparameter combination that yielded the highest accuracy.
* **best\_logreg\_score** captures the best accuracy achieved with the optimal hyperparameters.

This code efficiently tunes the hyperparameters of a Logistic Regression model using Grid Search CrossValidation, aiming to identify the best combination of regularization strength and optimization algorithm. The systematic approach provided by **GridSearchCV** helps enhance model performance while ensuring robust evaluation through cross-validation.

**5.2 Random Forest Hyperparameter tuning:**

The following code snippet demonstrates how to perform hyperparameter tuning for a Random Forest Classifier using Grid Search Cross-Validation (GridSearchCV) from the scikit-learn library. This process aims to optimize the classifier’s performance by exploring various combinations of hyperparameters.

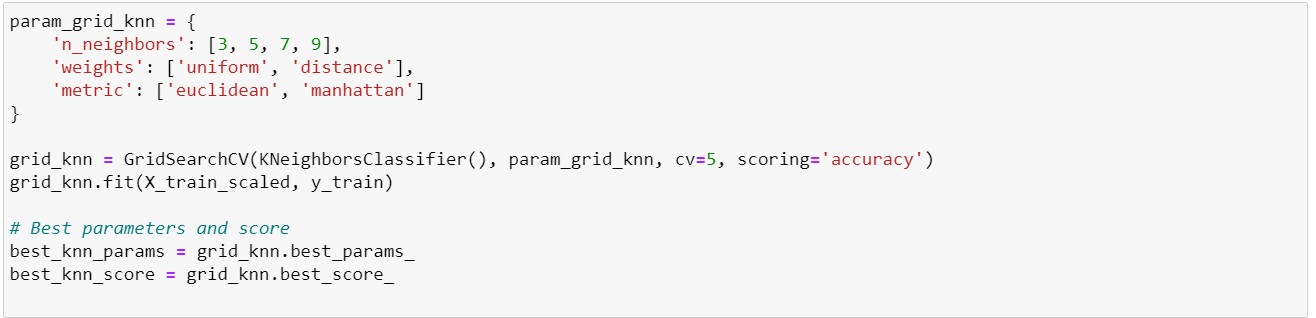


* **Defining the Parameter Grid**:
* This dictionary specifies the hyperparameters to tune:
  + - * **n\_estimators**: Number of trees in the forest (50, 100, 200). o **max\_depth**: Maximum depth of the trees (10, 20, 30, or unlimited).
      * **min\_samples\_split**: Minimum samples required to split an internal node (2, 5, 10).
      * **min\_samples\_leaf**: Minimum samples required at a leaf node (1, 2, 4).
* **Setting Up Grid Search**:
* This initializes GridSearchCV with a Random Forest model, specifying:
  + - * **Random state** for reproducibility. o **Cross-validation** (cv=5) for evaluating model performance.
      * **Scoring metric** as accuracy.
* **Fitting the Model**:
* The fit method trains the model on the training dataset, exploring all parameter combinations through cross-validation.
* **Extracting Best Parameters and Score**:
* best\_rf\_params retrieves the optimal hyperparameter combination.
* best\_rf\_score captures the highest accuracy achieved with these parameters.

This code performs hyperparameter tuning for a Random Forest Classifier using Grid Search CrossValidation, systematically exploring various configurations to enhance model performance. The approach helps ensure that the model generalizes well to unseen data and achieves optimal predictive accuracy.

**5.3 KNN Hyperparameter tuning:**

The following code snippet demonstrates how to perform hyperparameter tuning for a K-Nearest Neighbors (KNN) classifier using Grid Search Cross-Validation (GridSearchCV) from the scikit-learn library. This process aims to optimize the classifier’s performance by systematically evaluating different combinations of hyperparameters. By testing various settings for the number of neighbors, weighting methods, and distance metrics, the goal is to identify the optimal configuration that maximizes the model's accuracy in predicting outcomes.



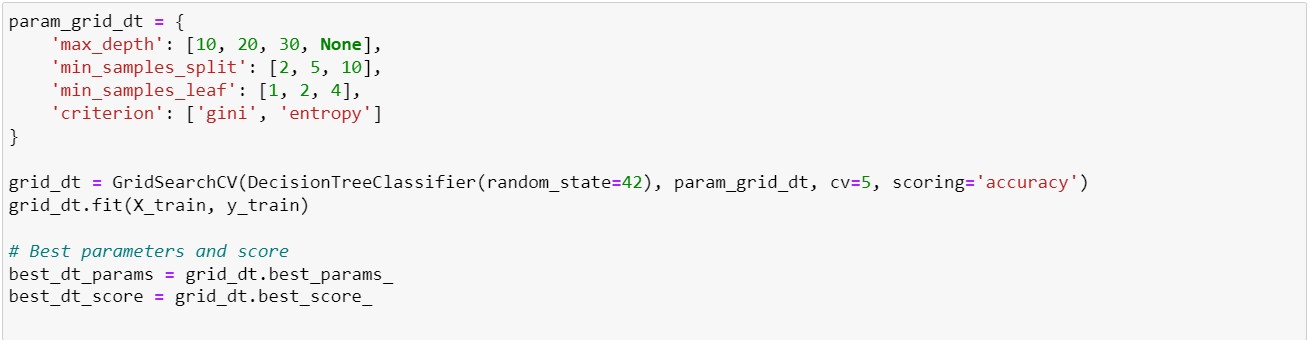
* **Defining the Parameter Grid**:
* param\_grid\_knn is a dictionary that specifies hyperparameters for tuning the **K-Nearest Neighbors (KNN)** model:
  + - * **n\_neighbors**: Specifies the number of neighbors to consider (3, 5, 7, 9).
      * **weights**: Determines how neighbors influence the prediction, with options for 'uniform' (all neighbors contribute equally) and 'distance' (closer neighbors contribute more).
      * **metric**: Specifies the distance metric to be used, with options for 'euclidean' and 'manhattan'.
* **Setting Up Grid Search**:
* A GridSearchCV object named grid\_knn is created with:
  + - * **KNeighborsClassifier()**: Initializes the KNN model.
      * **param\_grid\_knn**: Contains the hyperparameter options to be tested. o **cv=5**: Specifies 5-fold cross-validation to evaluate model performance.
      * **scoring='accuracy'**: Optimizes the model based on accuracy during cross-validation.

* **Fitting the Model**:
* The fit method is called on grid\_knn to train the model using the scaled training data (X\_train\_scaled) and corresponding target labels (y\_train), exploring all specified hyperparameter combinations.
* **Extracting Best Parameters and Score**:
* best\_knn\_params retrieves the combination of hyperparameters that achieved the highest accuracy during the tuning process.
* best\_knn\_score captures the best accuracy score obtained with the optimal hyperparameters.

This code performs hyperparameter tuning for the K-Nearest Neighbors model using Grid Search CrossValidation. By systematically testing various combinations of hyperparameters, the code aims to identify the optimal settings that enhance model performance, specifically in terms of accuracy.

**5.4. Decision Tree Hyperparameter tuning:**

The following code snippet demonstrates how to perform hyperparameter tuning for a Decision Tree Classifier using Grid Search Cross-Validation (GridSearchCV) from the scikit-learn library. This process aims to enhance the classifier’s performance by evaluating various combinations of hyperparameters. By testing different values for maximum depth, minimum samples required to split an internal node, minimum samples required to be at a leaf node, and the splitting criterion (either 'gini' or 'entropy'), the goal is to identify the optimal configuration that maximizes the model's accuracy in predicting outcomes.



* **Defining the Parameter Grid**:
* param\_grid\_dt is a dictionary that specifies hyperparameters for tuning the **Decision Tree Classifier**:
  + - * **max\_depth**: Defines the maximum depth of the tree (values: 10, 20, 30, None), controlling how deep the tree can grow.
      * **min\_samples\_split**: Specifies the minimum number of samples required to split an internal node (values: 2, 5, 10).
      * **min\_samples\_leaf**: Indicates the minimum number of samples required to be at a leaf node (values: 1, 2, 4).
      * **criterion**: Determines the function to measure the quality of a split, with options for 'gini' and 'entropy'
* **Setting Up Grid Search**:
* A GridSearchCV object named grid\_dt is created with:
  + - * **DecisionTreeClassifier(random\_state=42)**: Initializes the Decision Tree model with a fixed random state for reproducibility.
      * **param\_grid\_dt**: Contains the hyperparameter options to be tested.
      * **cv=5**: Specifies 5-fold cross-validation to assess model performance across different data splits.
      * **scoring='accuracy'**: Optimizes the model based on accuracy during the validation process.
* **Fitting the Model**:
* The fit method is called on grid\_dt to train the model using the training data (X\_train) and corresponding target labels (y\_train), allowing the grid search to explore all specified hyperparameter combinations.
* **Extracting Best Parameters and Score**:
* best\_dt\_params retrieves the combination of hyperparameters that resulted in the highest accuracy during the tuning process.
* best\_dt\_score captures the best accuracy score achieved with the optimal hyperparameters.

# CHAPTER SIX: MODEL EVALUATION

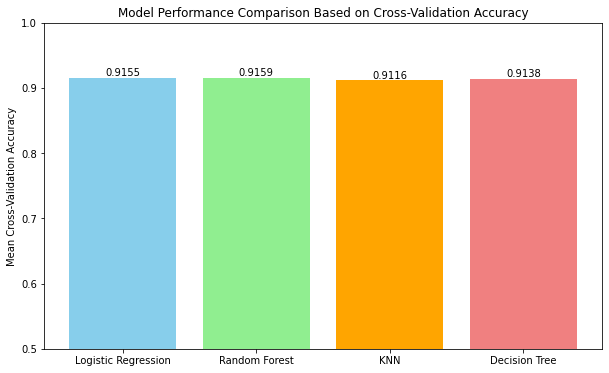
|  |
| --- |
|  |

**Cross-validation** is a statistical method used to assess the performance and generalizability of machine learning models by dividing the dataset into multiple subsets or "folds" (Hastie, Tibshirani, & Friedman, 2009). In this technique, the model is trained on some folds and tested on the remaining fold, and this process is repeated across all folds. The primary goal is to minimize overfitting, ensuring the model performs well not only on the training data but also on unseen data (James et al., 2013).

We use cross-validation because it provides a more robust evaluation of a model's performance by using all the available data for both training and validation. This approach helps avoid biased estimates that may occur when data is simply split once into training and test sets (Kohavi, 1995). By utilizing crossvalidation, particularly k-fold cross-validation, we can assess how well the model generalizes, making it particularly useful when working with smaller datasets or when trying to prevent overfitting (Friedman, Hastie, & Tibshirani, 2001).

We used Cross Validation to get the Accuracy and performance metrics such as Precision, Recall and F1 score to select which model works best for Cardiovascular disease risk detection.

The following results were seen as mentioned in the Figure 9 which shows the comparison of the accuracy achieved by all the four models using cross validation method.



## Figure 9. Accuracy of models using Cross Validation

The Figure 9 above compares the performance of all the four machine learning models namely **Logistic Regression**, **Random Forest**, **KNN**, and **Decision tree** based on their mean **cross-validation accuracy** for detection of cardiovascular disease risk. All the models perform with high accuracy, with Random Forest being the highest at **91.59%**, followed closely by Logistic Regression at **91.55%** and Decision Tree at

**91.38%**. KNN performs slightly lower than the other with **91.16%** accuracy.

As in this scenario accuracy alone should not be the primary performance metric while dealing with cardiovascular disease detection, as it may not be able to differentiate between patients at high and low risk. This is because cardiovascular diseases can typically be a class imbalanced problem, where most of the individuals may not have the disease, and a very small portion of these individuals do. In such cases, a model can achieve high accuracy simply by predicting the majority class correctly but still fail to identify those at high risk.

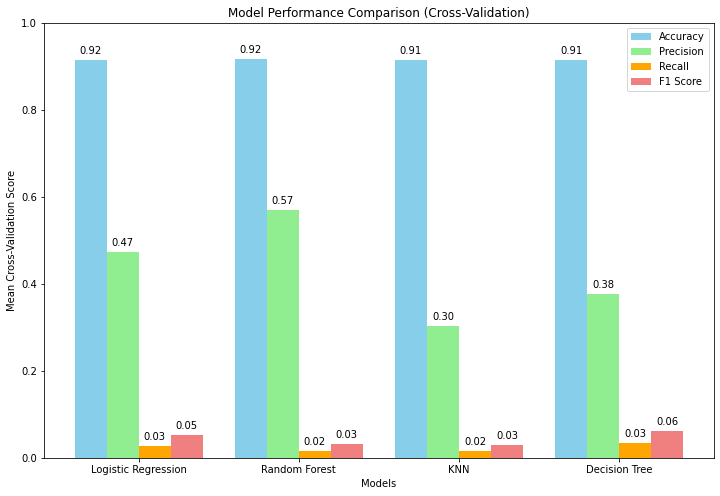
Other metrics like **precision**, **recall**, and **F1 score** become more important in this case, as they better evaluate the model’s performance for this small class, especially in this research case where correctly identifying high risk individuals is very essential.

**Precision**, **recall**, and **F1 score** are more important in this scenario, as they help in better evaluation of the model’s performance, especially in this case of healthcare application where correctly identifying high-risk patients is important. Recall for example ensures that the model predicts as many true positive cases as possible, which is crucial for cardiovascular disease detection and prevention.

In conclusion, while accuracy is important, the metrics which identify the correct classification of those with cardiovascular disease should be prioritized to ensure that the model is effective in real-world medical application where incorrect prediction can have severe consequences. Therefore, we have plotted out all the remaining performance metrics for selecting the best model for our research on cardiovascular disease risk detection.

Figure 10 contains the graph that compares the performance of four models—**Logistic Regression**,

**Random Forest**, **K-Nearest Neighbors (KNN)**, and **Decision Tree**—using **Accuracy**, **Precision**, **Recall**, and **F1 Score** for cardiovascular disease risk detection. While accuracy is high for all the models, there is a significant difference in precision, recall, and F1 Score, which are more relevant in evaluating the performance of the model in this case.



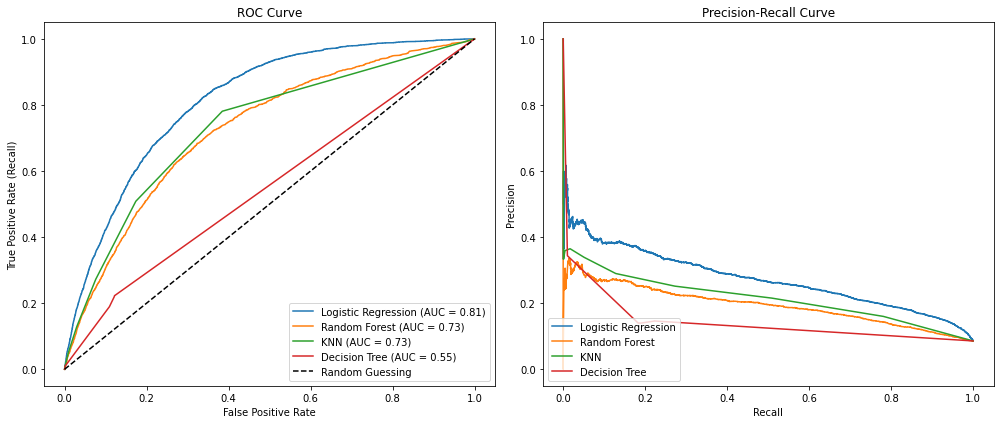
## Figure 10. Model Performance comparison using Cross validation

Using the graph above we can draw some observations which are as follows:

* **Logistic Regression** and **Random Forest** have the best accuracy around **92%.** Logistic Regression on one hand has a recall of only 0.03 This higher recall in comparison to the other models suggests it is slightly better at identifying the individuals at risk, which is very important in healthcare scenarios. On the other hand, Random forest shows an even lesser recall of 0.02 which is still low for the given scenario.
* **Decision Tree** model has an accuracy of **91%**, but very low **recall** of **0.03**, which means it also struggles to detect those at high risk for cardiovascular disease.
* **KNN** has the recall of **0.03**, it also lacks in precision. Therefore, is not ideal for our research project.

Best-Suited Model for Cardiovascular Disease Detection:

In cardiovascular disease risk detection, recall and F1 score are more important than accuracy, as we prioritize identifying all potential high-risk patients (Pedregosa et al., 2011). Based on our findings, **Logistic Regression** model seems to be the most suitable model due to its higher recall and F1 Score, which balances both precision and the accuracy. This model is better suited as our research is based on healthcare application where missing a diagnosis can have severe consequences.



## Figure 11. ROC Curve and Precision Recall Curve

The image presents two key evaluation metrics: the **ROC (Receiver Operating Characteristic) Curve** on the left and the **Precision-Recall Curve** on the right. Both plots are used to evaluate the performance of four machine learning models—Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbors (KNN), and Decision Tree (DT)—in predicting cardiovascular disease risk.

**ROC Curve:**

The ROC curve illustrates the trade-off between the **True Positive Rate** (Recall) and **False Positive Rate**. A key metric here is the **Area Under the Curve (AUC)**, which represents how well the model distinguishes between classes.

* **Logistic Regression (AUC = 0.81)** outperforms the other models with the highest AUC, suggesting that it is the most effective in distinguishing between patients at risk and those not at risk of cardiovascular diseases.
* **Random Forest and KNN (AUC = 0.73)** have comparable AUCs, both performing moderately well but not as robust as Logistic Regression.
* **Decision Tree (AUC = 0.55)** has the lowest AUC, indicating that it struggles with distinguishing between the two classes, likely performing close to random guessing.

**Precision-Recall Curve:**

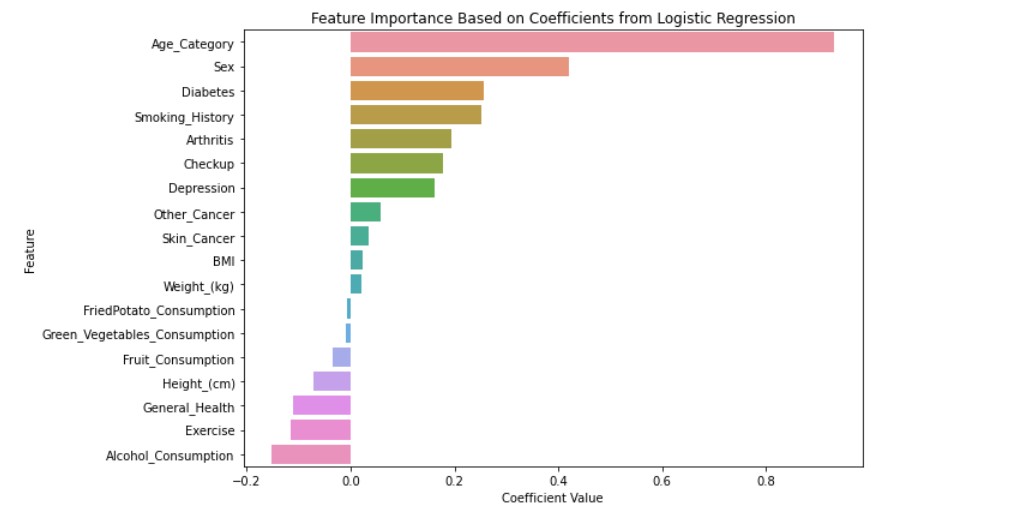
This curve focuses on the trade-off between **Precision** (the proportion of true positives out of all positive predictions) and **Recall** (the proportion of true positives out of actual positives), which is particularly important in imbalanced datasets such as health risk prediction.

* **Logistic Regression** again performs the best, maintaining the highest precision at various levels of recall. This suggests that Logistic Regression is more reliable in identifying actual cases of cardiovascular disease with fewer false positives.
* **Random Forest** shows moderate precision but a noticeable decline as recall increases, indicating a higher number of false positives at higher recall levels.
* **KNN** and **Decision Tree** exhibit a steeper drop in precision with increasing recall, meaning they are less reliable in correctly predicting patients at risk, especially at higher recall levels.

**Insights:**

1. **Logistic Regression** consistently demonstrates the best performance in both the ROC and Precision-Recall curves, making it the most reliable model for detecting cardiovascular disease risks in this study.
2. **Random Forest** performs decently, but its precision decreases sharply with higher recall, suggesting potential issues with false positives in predicting high-risk patients.
3. **KNN** and **Decision Tree** perform poorly in comparison, particularly in the Precision-Recall curve, indicating that these models may struggle with the imbalanced nature of the dataset where true positives (patients with CVD) are relatively few.

This analysis underscores the importance of selecting models that balance recall and precision in health risk predictions, where false negatives and false positives can have significant consequences (Han, Kamber & Pei, 2011).



## Figure 11. Feature importance of Logistic Regression Model

The Figure 11 contains a graph **feature importance chart**, where different features are ranked based on their contribution to a model's predictive power. Here is a summary:

* **Top Features**: The highest-ranked features are Age, sex and diabetes having the largest bar, indicating that they are the most important factor in the model's predictions. The top three features appear to contribute the most, as their bars are significantly larger than the others.
* **Middle Features**: The features in the middle of the chart namely depression, checkup and arthritis still have some influence but are less impactful compared to the top features.
* **Lower Features**: The lower-ranked features have very short bars, indicating that their contribution to the model's predictions is minimal or negligible. These features might have a weaker or no correlation with the target variable.

In a cardiovascular disease risk detection context, this chart can be helpful to identify which features (like age, diabetes, smoking history, etc.) have the most influence on the risk prediction. **Focusing on these top features** may improve model performance, while the least important features could potentially be dropped to simplify the model without sacrificing much predictive power.

For instance, a high-ranking feature like "age" or "Diabetes" might be critical in predicting cardiovascular disease risk, whereas low-ranking features might contribute little to the final prediction.

If this chart was generated from a logistic regression model, these are the **coefficients** representing the importance of each feature, with larger absolute values corresponding to more significant contributions. Alternatively, if this was derived from a tree-based model (e.g., Random Forest), this could represent the **decrease in impurity** across splits associated with each feature.

# CONCLUDING REMARKS

|  |
| --- |
|  |

This study has demonstrated the power of leveraging machine learning techniques in predicting cardiovascular disease risk, using models such as Logistic Regression, K-Nearest Neighbours (KNN), Decision Tree, and Random Forest. The dataset used in this research, included wide range of healthrelated factors, using these factors we were able to analyse the relationship between these features and the likelihood of developing cardiovascular diseases.

The models were evaluated using key performance metrics such as accuracy, precision, recall, F1 Score and lastly ROC-AUC curves. These metrics provided valuable insights into how each model predicts the risk of cardiovascular diseases in individuals and how these models balance the need to minimize false negatives and false positives. Given the critical nature of CVD risk detection, where early intervention can be very crucial in saving lives, this balance was the most important factor in assessing model’s performance.

**Key Findings:**

1. The Logistic Regression model emerged as the most reliable model in this study, achieving a high ROC-AUC score of 0.81, which indicates a strong ability to differentiate between individuals at risk and individuals not at risk of CVDs. The model’s precision-recall curve further showed that it performs well in identifying individuals who truly have cardiovascular risks, with an emphasis on high precision across recall thresholds. Logistic regression model showed comparatively higher recall and F1 score suggesting lesser false negatives and more effective than other models, which was very important for cardiovascular risk detection.
2. The Random Forest model also demonstrated a strong performance, achieving AUC of 0.73. However, the recall remained lower than Logistic Regression model, indicating that it misses some individuals who are at risk of developing cardiovascular diseases. It had the ability to capture non-linear interactions between features but the higher false negative rate made it less ideal for application, where failing to detect an individual at risk could lead to serious health consequences.
3. The K-Nearest Neighbours (KNN) model showed lower overall performance compared to the other two models, with a moderate AUC of 0.73 and a low precision-recall curve. While KNN can effectively model complex interactions between health factors, its higher false positives, as seen in the precision-recall curve, indicates it is not ideal for healthcare application.
4. Decision Tree, struggles the most with differentiating between the target classes, as seen by its AUC of 0.55. This model approach random guessing mostly due to the imbalanced dataset like this one, where the majority of individuals do not have CVDs. Although Decision Tree is good at decision making process, its higher rate of both false positives and false negatives make it not ideal for using in healthcare scenario of Cardiovascular disease risk detection.

**Insights from Feature Importance:**

As Logistic regression model was considered the best among the models build, we decided to check the feature that contribute the most to the prediction of these cardiovascular disease risks.

* **Age**: As expected, age was one of the most important predictors, with older individuals exhibiting a higher likelihood of developing cardiovascular diseases. This finding is also been seen in medical literatures; the risk of cardiovascular diseases increases with the increase in age.
* **Diabetes Status**: The presence of Diabetes was a major factor in predicting CVD risk, showing the strong link between metabolic disorders and heart health.
* **Smoking History**: Smoking also remains a risk factor for cardiovascular diseases, and this was clearly seen in the model’s predictions. Reducing smoking habit can have a great impact on CVD preventions.

**Implications for Healthcare:**

These findings from the research give us an idea of the potential for machine learning models to assist in early detection of cardiovascular risks, making it easier for the healthcare providers to take preventative measure. However, there are important considerations we get from this research:

* **Logistic Regression** is the strongest candidate among the models we prepared for practical application in the real-world due to its **higher recall** and ability to provide interpretable results, which can be easily use by healthcare services to help an individual at risk. Its balance performance across **precision**, **recall**, and **F1 score** ensures it capture most individuals at risk.
* The **precision-recall curves** and **ROC-AUC scores** show that these models, particularly **Logistic Regression**, perform well in balancing the need for accurate detection without overwhelming the healthcare system with false positives.

**Limitations and Future Work:**

Despite the findings, this research faced certain limitations:

* The class imbalance in the dataset presents a challenge for model performance, particularly in terms of recall. While precision-recall curves provide some insight into this, future work should consider using techniques like oversampling (SMOTE) or cost-sensitive learning to improve the detection of minority classes.

* The models used in this study rely on structured data. Use of additional sources of data such as unstructured medical records, genetic information or even wearable device data could further improve the accuracy and application of the models in real world scenarios.
* Lastly, the **interpretability** of models like Random Forest and Logistic Regression should be further explored, ensuring that clinicians can understand and trust the models’ outputs, especially in critical healthcare decisions.

**Conclusion:**

In conclusion, this research shows the effectiveness of machine learning models in predicting cardiovascular disease risk, with Logistic Regression emerging as the top performing model. This model, when applied in real-world settings, have a potential to assist in the early detection of cardiovascular diseases leading to personalized treatment plans, resulting in reducing global burden of cardiovascular diseases. The ability to identify high risk individuals based on the key health factors such as age, diabetes, and smoking habits offers a valuable tool for healthcare providers. Further, refining these models, particularly by addressing class imbalances and incorporating more diverse type of data can be critical to their successful implementation in real-world medical application.

# REFERENCES

|  |
| --- |
|  |

* Alaa, A.M., Bolton, T., Di Angelantonio, E., Rudd, J.H.F., & van der Schaar, M. (2019).

Cardiovascular disease risk prediction using automated machine learning: A prospective study of 423,604 UK Biobank participants. *PLOS ONE*, 14(3), e0213653. Available at: [https://doi.org/10.1371/journal.pone.0213653.](https://doi.org/10.1371/journal.pone.0213653)

* Altman, N.S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), pp. 175-185.
* Bergstra, J. & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, pp. 281-305.
* Bergstra, J., Yamins, D. & Cox, D.D. (2011). Hyperparameter optimization with Bayesian optimization. *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1-9.
* Bishop, C.M. (2006). *Pattern recognition and machine learning*. Springer.
* Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), pp. 5-32.
* Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. (1984). *Classification and regression trees*. Wadsworth International Group.
* Chen, C., Liaw, A. & Breiman, L. (2004). Using random forest to learn imbalanced data. *University of California, Berkeley*.
* Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), pp. 21-27.
* Davis, J. & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233-240.
* Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), pp. 861874.
* Fleiss, J.L. (1981). *Statistical methods for rates and proportions*. 2nd ed. New York: Wiley.
* Friedman, J., Hastie, T. & Tibshirani, R. (2001). *The elements of statistical learning*. New York: Springer.
* Han, J., Kamber, M. & Pei, J. (2011). *Data mining: Concepts and techniques*. 3rd ed. San Francisco: Morgan Kaufmann.
* Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. 2nd ed. New York: Springer.
* Ho, T.K. (1995). Random decision forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pp. 278-282.
* Hosmer, D.W., Lemeshow, S. & Sturdivant, R.X. (2013). *Applied logistic regression*. 3rd ed. New York: John Wiley & Sons.
* Hutter, F., Hoos, H.H. & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. *Learning and Intelligent Optimization*, pp. 1-13.
* Izenman, A.J. (2008). *Modern multivariate statistical techniques*. New York: Springer.
* James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). *An introduction to statistical learning with applications in R*. New York: Springer.
* Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 1137-1145.
* Kotsiantis, S.B., Kanellopoulos, D. & Pintelas, P.E. (2006). Handling imbalanced datasets: A review. *GECAD Intelligent Systems and Control* [online]. Available at: [https://www.gecad.isc.org/publications/handling\_imbalanced\_datasets.pdf.](https://www.gecad.isc.org/publications/handling_imbalanced_datasets.pdf)
* Kotsiantis, S.B., Zaharakis, I. & Pintelas, P. (2006). Machine learning: A review of classification and combining techniques. *Artificial Intelligence Review*, 26(3), pp. 159-190.
* Kuhn, M. & Johnson, K. (2013). *Applied predictive modeling*. Springer.
* McKinney, W. (2017). *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. 2nd ed. Sebastopol: O’Reilly Media.
* Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, pp. 28252830.
* Rousseeuw, P.J. & Leroy, A.M. (1987). *Robust regression and outlier detection*. John Wiley & Sons.
* Shishehbori, F. & Pourpanah, F. (2024). Enhancing cardiovascular disease risk prediction with machine learning models. *arXiv*, 2401.17328. Available at: [https://doi.org/10.48550/arXiv.2401.17328.](https://doi.org/10.48550/arXiv.2401.17328)
* Subramani, S., Varshney, N., Anand, M.V. et al. (2023). Cardiovascular diseases prediction by machine learning incorporation with deep learning. *Frontiers in Cardiovascular Medicine*

[online]. Available at: [https://www.frontiersin.org/articles/10.3389/fcvm.2023.01000/full.](https://www.frontiersin.org/articles/10.3389/fcvm.2023.01000/full)

* Tukey, J.W. (1977). *Exploratory data analysis*. Reading, MA: Addison-Wesley.
* 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. Available at: [https://doi.org/10.1371/journal.pone.0174944.](https://doi.org/10.1371/journal.pone.0174944)