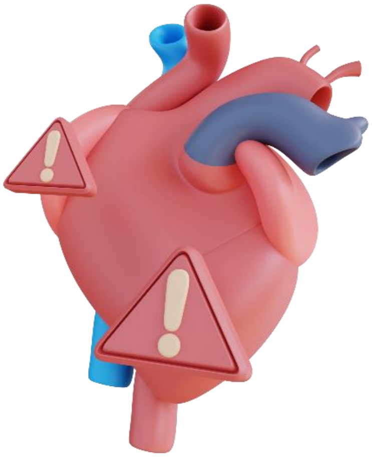
UNIVERSITE LIBANAISE

Lebanese University
Faculty of Information, Branch II

UNDERGRADUTE FINAL YEAR PROJECT

Bachelor of Data Science

Predictive Modeling for Heart Disease Risk Assessment



Presented by:	Aya EL HAJJ	7731
	Johnny CHREIM	7691

Academic Supervisors:	Dr Malak KHREISS
	Dr Rachid KHOURY
Company Supervisors:	Dr. Eng. Riad ASSAF
	Mr. Francis EL HELOU

Academic Year 2022-2023

I. ACKNOWLEDGEMENTS

We would like to express our sincere gratitude and appreciation to the following individuals who have played a significant role in the successful completion of this final year project:

First and foremost, we extend our heartfelt thanks to our project supervisors, Dr Riad Assaf, Dr Malak Khreiss & Mr. Francis El Helou, for their unwavering guidance, invaluable insights, and continuous support throughout the entire project. Their expertise and mentorship have been instrumental in shaping the direction and quality of our work.

We would like to acknowledge the faculty members of Lebanese University Faculty of Information Branch 2, for their valuable feedback, constructive suggestions, and expert advice. Their contributions have immensely enhanced the depth and rigor of our project.

We are grateful to the staff members of the Data Science Department` for their assistance and cooperation. Their promptness in addressing administrative matters and providing necessary resources has greatly facilitated the smooth progress of our project.

We would like to extend our gratitude to the participants who willingly volunteered their time and efforts in participating in our surveys, interviews, or any other data collection methods used in this study. Their valuable contributions and insights have significantly enriched our project.

Additionally, we would like to express our heartfelt gratitude to Dr. Hamssa Hasrouny for her continuous support and guidance throughout our academic journey.

Lastly, we would like to express our heartfelt thanks to our families for their constant encouragement, understanding, and belief in our abilities. Their unwavering support has been the driving force behind our academic pursuits and accomplishments.

This project would not have been possible without the support and contributions of these individuals and organizations. We are genuinely grateful to each and every one of them for their assistance, guidance, and encouragement throughout this journey.

II. ABSTRACT

Heart disease remains one of the leading causes of mortality worldwide. Early identification of individuals at high risk for developing heart disease is crucial for implementing effective prevention strategies and improving patient outcomes. This project focuses on developing a predictive modeling framework for assessing the risk of heart disease using demographic, clinical, and lifestyle factors.

The developed predictive modeling framework is then applied to a real-world dataset to assess the risk of heart disease in a population sample. The performance of the models is evaluated using metrics such as sensitivity, specificity, and area under the receiver operating characteristic curve.

The ultimate target of the project is to have a model that takes patient demographics to generate a risk assessment score that can be interpreted by the doctor to act accordingly.

The results of this project have significant implications for healthcare providers and public health interventions. Accurate risk assessment models can assist in identifying individuals at high risk for heart disease, enabling targeted interventions, lifestyle modifications, and early detection strategies. This can lead to improved patient outcomes, reduced healthcare costs, and a proactive approach to heart disease prevention.

Table of Contents

I. ACKNOWLEDGEMENTS 2

II. ABSTRACT 3

III. LIST OF TABLES 7

IV. LIST OF FIGURES 8

1. INTRODUCTION 9

1.1 PROJECT PROBLEM 9

1.2 PROJECT SCOPE AND STATEMENT 10

1.3 PROJECT OBJECTIVES 10

1.4 METHODOLOGY OVERVIEW 11

1.5 LIMITATIONS 13

2. MACHINE LEARNING MODELS 14

2.1. KNN 14

2.2. LOGISTIC REGRESSION 15

2.3. DECISION TREE 16

2.4. NAÏVE BAYES 17

2.5. RANDOM FOREST 18

2.6. SVM 19

METHODOLOGY 20

3. BUSINESS UNDERSTANDING 20

3.1. BUSINESS PROBLEM 20

3.2. BUSINESS OBJECTIVE 20

4. DATA UNDERSTANDING 22

5. DATA PREPARATION 28

5.1. HANDLING MISSING DATA 28

5.1.1. REMOVING MISSING VALUES 29
5.1.2. IMPUTING MISSING VALUES 30
5.1.2.1. Filling by Mean & Mode 30
5.1.2.2. Filling by KNN 31
5.1.2.3. Filling by Next 33
5.1.2.4. Filling by Previous 33
5.2. OUTLIERS 33
5.2.1. KEEPING THE OUTLIERS 35
5.2.2. REMOVING OUTLIERS BY STANDARD DEVIATION 35
5.2.3. REMOVING OUTLIERS BY IQR 36
5.2.4. REMOVING OUTLIERS BY DBSCAN 37
5.3. FEATURE ENGINEERING 37
5.3.1. DATA SCALING 38
5.3.2. FEATURE SELECTION 38
5.3.2.1. Select K Best Function 39
5.3.2.2. Select Percentile Function 39
5.3.2.3. Variance Threshold Function 40
5.3.3. DIENSIONALITY REDUCTION 41
5.4. BALANCING 41
5.4.1. UNDER SAMPLING 42
5.4.2. OVER SAMPLING 43
6. MODELING 45
6.1. KNN 48
6.2. LOGISTIC REGRESSION 48
6.3. DECISION TREE 49

6.4. NAIVE BAYES 49

6.5. RANDOM FOREST 50

6.6. SVM 50

7. EVALUATION 51

CHOOSING THE OPTIMAL MODEL 51

8. DEPLOYMENT 54

CONCLUSION 57

REFERENCES 59

III. LIST OF TABLES

Table 1 Variables Description 24
Table 2 data frame 25
Table 3: Best data frames for KNN 48
Table 4: Best Data frames for Logistic Regression 49
Table 5: Best Data frames for Decision Tree........ 49
Table 6: Best Data frames for Naive Bayes........ 49
Table 7: Best Data frames for Random Forest........ 50
Table 8: Best Data frames for SVM 50
Table 9: The Best Models........ 51
Table 10 Metrics of the best model........ 52
Table 11 Confusion matrix 52

IV. LIST OF FIGURES

Figure 1 Phases of CRISP-DM reference model 11

Figure 2: KNN example 15

Figure 3: Sigmoid function 15

Figure 4: Logistic Regression Example 16

Figure 5: Decision Tree Diagram 17

Figure 6: Naive Bayes Example 17

Figure 7: Random Forest Diagram 18

Figure 8: SVM Diagram 19

Figure 9 correlation Heatmap 26

Figure 10 missing values 28

Figure 11 Outliers 35

Figure 12: Bar Plot of TenYearCHD 42

Figure 13: Bar Plot of TenYearCHD after under sampling 43

Figure 14: Bar Plot of TenYearCHD after over sampling 44

Figure 15: Precision & Recall 46

Figure 16 Understanding ROC curve 47

Figure 17 ROC Curve 53

Figure 18 user-interface / subset of input fields 54

Figure 19 prediction _ heart disease risk 55

Figure 20 prediction _ No risk 55

1. INTRODUCTION

Heart disease remains a significant global health concern, accounting for a considerable number of deaths worldwide. Early identification of individuals at high risk for developing heart disease is crucial for implementing effective prevention strategies, improving patient outcomes, and reducing the burden on healthcare systems. Predictive modeling techniques offer a promising approach to assess an individual's risk of heart disease by leveraging demographic, clinical, and lifestyle factors.

As medical research and technological advancements continue to evolve, the fight against heart disease risk remains a critical endeavor. It requires constant efforts from healthcare organizations and individuals to identify and handle this disease and its serious consequences.

1.1 PROJECT PROBLEM

With the increasing prevalence of heart disease and its significant impact on individuals' health and well-being, heart disease prediction became a critical process to improve healthcare management. There is a crucial need to develop accurate and reliable methods to identify individuals at risk. Early detection of heart disease risk is the main reason to save lives, as it helps to implement preventive strategies and interventions early and to improve outcomes. Using machine learning different algorithms, we can analyze patients’ data, and predict the likelihood of developing heart disease.

The project problem is the challenge of developing an effective predictive model to accurately predict the risk of heart disease in individuals using available data. This requires overcoming several key challenges. These challenges include identifying the most significant risk factors, handling the complexity of patient data, identifying the most meaningful risk factors, and developing a reliable model that can provide early detection. By tackling this problem, we intend to empower healthcare providers with a tool that can reliably identify individuals at high risk of developing heart disease, enabling them to implement proactive measures and targeted actions with the aim of reducing the risks associated with heart disease, contributing to a healthier population.

1.2 PROJECT SCOPE AND STATEMENT

The scope of this project is to develop and evaluate a predictive modeling system for accurate heart disease risk assessment. It involves using machine learning algorithms to develop a predictive model, analyzing patient medical records to identify significant risk factors, and generating personalized risk assessment scores.

1.3 PROJECT OBJECTIVES

The primary objective of this project is improving early detection and prevention of heart disease by developing a predictive model that analyzes patient medical records to identify and highlight the most significant risk factors associated with heart disease. By considering patient demographics, lifestyle factors, and medical history, the model will generate a personalized risk assessment score for each individual, aiding in the early detection and prevention of diseases.

Project objectives include:

- Finding a comprehensive dataset of patient medical records, including demographics, lifestyle factors, and medical history to work on.
- Applying data preprocessing techniques to handle missing values, outliers, and ensure data quality.
- Performing exploratory data analysis and statistical techniques to identify the highest risk factors.
- Implementing appropriate machine learning algorithms, and training the predictive model using the preprocessed dataset.
- Evaluating the model's performance using different metrics (accuracy, recall, precision, F1-score)
- Developing a user-friendly interface that allows easy input of patient data and provides risk assessment scores.

By achieving these objectives, we will get a predictive model that detect heart disease risk for individuals by analyzing their data.

1.4 METHODOLOGY OVERVIEW

The methodology employed in this project follows the widely recognized CRISP-DM (Cross-Industry Standard Process for Data Mining) framework. The CRISP-DM framework provides a structured approach for conducting data mining projects, enabling effective and efficient data analysis and modeling. It has six sequential phases:

-Business understanding – What does the business need?

-Data understanding – What data do we have / need? Is it clean?

-Data preparation – How do we organize the data for modeling?

-Modeling – What modeling techniques should we apply?

-Evaluation – Which model best meets business objectives?

-Deployment – How do stakeholders access the results?

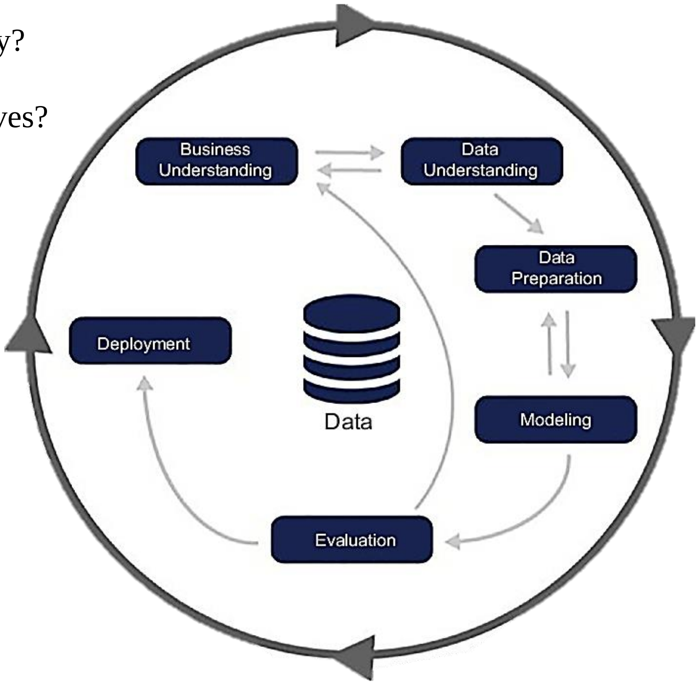


Figure 1 Phases of CRISP-DM reference model

1. Business Understanding:

This phase involves understanding the business problem of the increasing prevalence of heart disease, and defining the objectives.

2. Data Understanding:

The data understanding stage involves identifying a dataset of patient records studying various risk factors, and conducting exploratory data analysis to discover data, and ensure its quality.

3. Data Preparation:

Performing data preprocessing and data cleaning activities. This involves addressing missing values, handling outliers, handling data imbalance, and data transformation, to prepare the data for modeling.

4. Modeling:

The modeling stage involves the development and evaluation of different machine learning algorithms using the prepared dataset. Various machine learning algorithms, such as logistic regression, decision trees, random forest, and support vector machines, are employed to create predictive models. The models are trained on a subset of the data and validated using appropriate evaluation metrics.

5. Evaluation:

The evaluation stage assesses the performance and effectiveness of the developed models. The models are evaluated using various metrics such as accuracy, precision, recall, and F1-score.

6. Deployment:

The final stage involves deploying the selected model for heart disease risk assessment. The model is integrated into a user-friendly interface that enables patients and doctors to input data and generate personalized risk assessment scores

All the mentioned phases are described in the next sections.

1.5 LIMITATIONS

However, like any approach, predictive modeling for heart disease risk assessment comes with certain limitations. It is essential to acknowledge these constraints to ensure a clear understanding of the model's capabilities and potential challenges. Here are some of the limitations:

- Data Availability and Quality: The accuracy and effectiveness of predictive models heavily rely on the availability and quality of data. The predictability of the forecasts may be impacted by limited access to thorough and up-to-date medical records, as well as potential data mistakes.

- Generalizability: Predictive models are often developed based on specific datasets, which might not fully represent the diverse global population. As a result, the model's performance across, geographic regions, and divergent people categories may differ, limiting its generalizability.

- Validation and External Testing: To ensure model dependability and resilience across different populations and situations, models should be carefully validated and externally tested on diverse datasets. Inadequate validation may lead to an overestimation of the model's performance.

- Ethical Considerations: The use of predictive models in healthcare presents ethical concerns, specifically around patient privacy and permission. Ensuring the responsible and ethical deployment of these models is crucial.

Despite these limitations, continuous efforts to refine predictive models, gather more comprehensive data, and address ethical concerns can enhance their utility in identifying individuals at high risk for heart disease and improving preventive strategies.

2. MACHINE LEARNING MODELS

2.1. KNN

The supervised machine learning algorithm K-Nearest Neighbors (KNN) is utilized for both classification and regression applications. The non-parametric approach bases its predictions on how closely the input data resembles the labeled data points in the training set.

The KNN algorithm's key features include:

- K-value:

The number of neighbors taken into account while making predictions depends on the value of the parameter k. A small k value might cause overfitting, which would capture data noise, while a big k value could cause over smoothing, which would lose local patterns. We chose a k-value of 5 in the project.

- Distance measure:

The calculation of feature vector similarities depends on the chosen distance metric. Euclidean distance is the most widely used metric, although depending on the type of data, alternative metrics such as Manhattan distance or cosine similarity can also be utilized.

- Data preprocessing:

In order to provide fair comparisons, it is crucial to normalize or standardize the features in the input data because KNN is sensitive to their size and distribution.

- Computing complexity:

Since KNN searches for nearest neighbors, its computational cost rises with the volume of training data. Several methods, like KD-trees and ball trees, can be utilized to enhance the performance of this procedure.

The aim of KNN in heart disease risk assessment is to classify an instance based on its features

Below is an explanatory figure that briefly shows how KNN operates in a machine learning model:

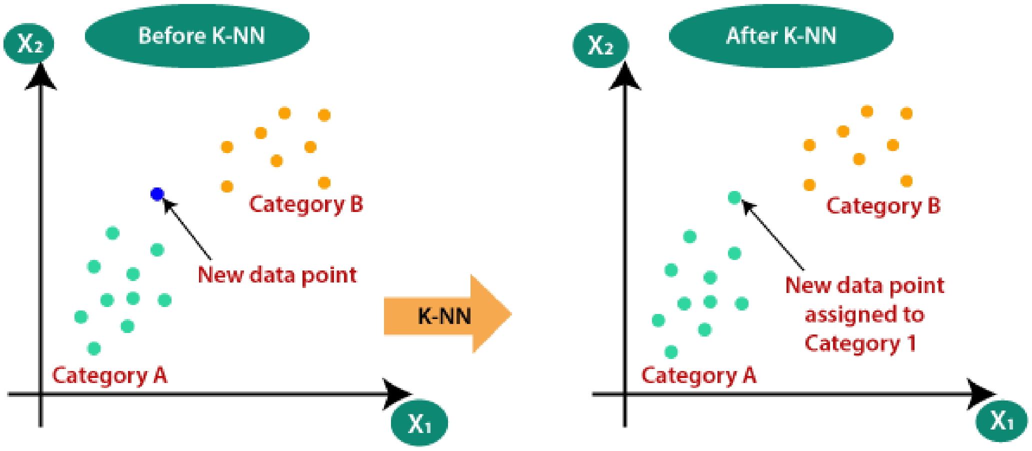


Figure 2: KNN example

In our project, Category A can be the values 0 and Category B can be the values 1 of TenYearCHD (the target variable).

2.2. LOGISTIC REGRESSION

3. A machine learning classification approach called logistic regression is used to forecast the likelihood of particular classes based on a set of dependent variables. In essence, the logistic regression model computes the sum of the input features (often with a bias term) and then determines the logistic of the outcome.
4. The algorithm for logistic regression examines the connections between variables. Using the Sigmoid function, which transforms numerical data into an expression of probability between 0 and 1.0, it assigns probabilities to discrete outcomes. Depending on whether or not the event occurs, probability ranges from 0 to 1. With a cut-off of 0.5, you can divide the population into two groups for binary predictions. Group A includes everything that is greater than 0.5, and group B includes everything that is less than 0.5.

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Figure 3: Sigmoid function

5. A hyperplane is used as a decision line to separate two categories (as far as possible) after data points have been assigned to a class using the Sigmoid function. The class of future data points can then be predicted using the decision boundary. (Jessica, 2022)

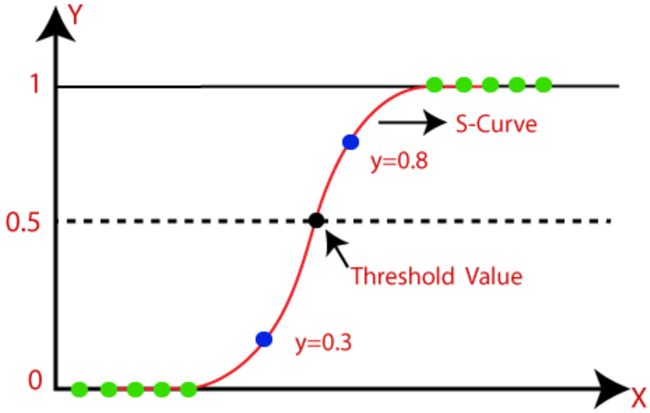


Figure 4: Logistic Regression Example

2.3. DECISION TREE

2. A decision tree is a supervised learning method that can be used to solve classification and regression problems, but it is typically favored for classification. It is a tree-structured classifier, where internal nodes stand in for a dataset's features, branches for the decision-making process, and each leaf node for the classification result.
3. The Decision Node and Leaf Node are the two nodes of a decision tree. While Leaf nodes are the results of decisions and do not have any more branches, Decision nodes are used to create decisions and have numerous branches.
4. The given dataset's features are used to execute the test or make the decisions.
5. The decision tree is a graphical depiction for obtaining all feasible answers to a choice or problem based on predetermined conditions.
6. It is known as a decision tree because, like a tree, it begins with the root node and grows on subsequent branches to form a structure resembling a tree. (Decision Tree Classification Algorithm, n.d.)
7. The CART algorithm, which stands for Classification and Regression Tree algorithm, is used to construct a tree.
8. A decision tree only poses a question and divides the tree into subtrees according to the response (Yes/No).

9. The decision tree's general structure is shown in the diagram below:

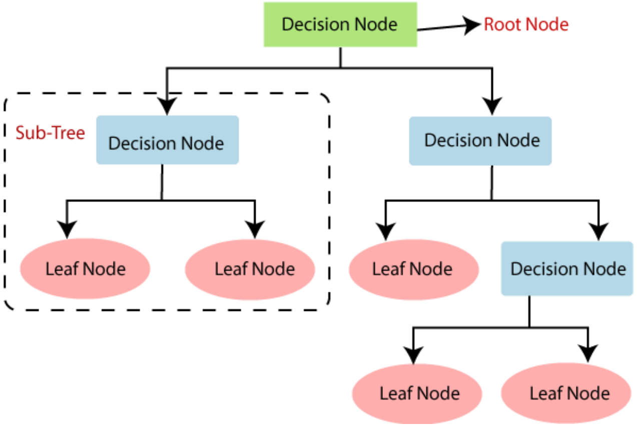


Figure 5: Decision Tree Diagram

2.4. NAÏVE BAYES

The Nave Bayes algorithm is a supervised learning method for classification based on the Bayes theorem.

It is mostly employed in text categorization with a large training set.

The Naive Bayes Classifier is one of the most straightforward and efficient classification algorithms available today. It aids in the development of quick machine learning models capable of making accurate predictions.

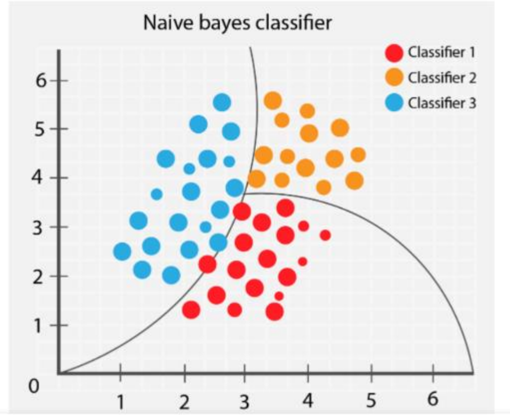


Figure 6: Naive Bayes Example

Being a probabilistic classifier, it makes predictions based on the likelihood that an object will occur.

Below are the steps to go through when using Naïve Bayes algorithm:

- Create frequency tables from the provided dataset.
- Create a likelihood table by calculating the odds of the given attributes.
- Now, determine the posterior probability using the Bayes theorem.

Advantages of Naïve Bayes Classifier:

1. Fast and easy algorithm for predictions
2. Can be used for binary and multi-class classifications
3. The most popular choice for text classification (Naïve Bayes Classifier Algorithm, n.d.)

2.5. RANDOM FOREST

3. Instead of relying on one decision tree, the random forest takes the prediction from each tree and bases its prediction of the final output on the majority votes of predictions. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.
4. Higher accuracy and overfitting are prevented by the larger number of trees in the forest.
5.

   As its name indicates, a random forest is a set of decision trees from subsets of the initial datasets; the dataset is split into subsets and each from each subset a decision tree is generated with an output. The average of the outputs of the decision trees will be the final output of the random forest model. (Random Forest Algorithm, n.d.)
6. Below is a small diagram that briefly explains how the random forest algorithm works:

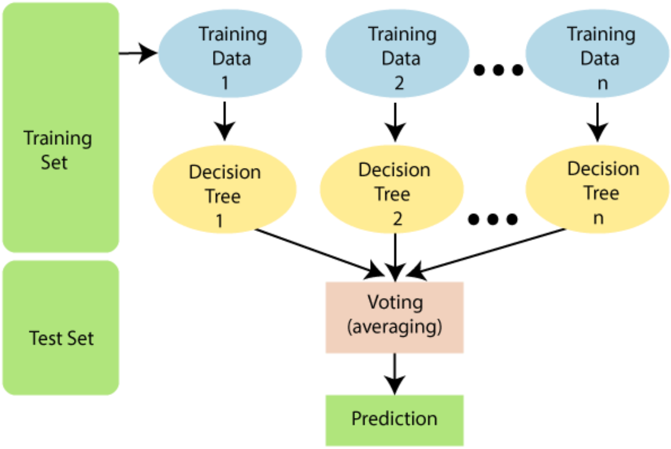


Figure 7: Random Forest Diagram

2.6. SVM

3. The SVM algorithm's objective is to establish the best line or decision boundary that can divide n-dimensional space into classes, allowing us to quickly classify fresh data points in the future. A hyperplane is the name given to this optimal decision boundary.
4. SVM selects the extreme vectors and points that aid in the creation of the hyperplane. Support vectors, which are used to represent these extreme instances, form the basis for the SVM method. (Support Vector Machine Algorithm, n.d.) Consider the diagram below, where a decision boundary or hyperplane is used to categorize two distinct categories:

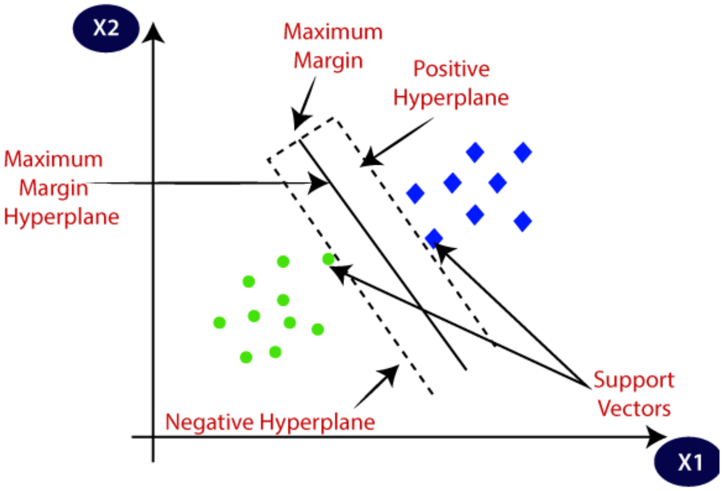


Figure 8: SVM Diagram

METHODOLOGY

3. BUSINESS UNDERSTANDING

3.1. BUSINESS PROBLEM

Heart disease can lead to serious illness, disability, and lower quality of life, it is also a leading cause of mortality worldwide, and its early detection and prevention are crucial for reducing its impact. However, healthcare providers often encounter difficulties in accurately assessing an individual's risk for developing heart disease due to the complex interplay of various risk factors.

The business problem resides in the need for an efficient reliable method of identifying individuals risk of developing heart disease. Healthcare providers can gain valuable insights from patient medical records to create personalized risk assessment scores by developing a predictive modeling solution. This approach enables early intervention and targeted preventive measures to mitigate the risk and improve patient outcomes.

3.2. BUSINESS OBJECTIVE

The project’s business objectives center on handling the issue of heart disease and its impact on healthcare organizations and individuals. These objectives include:

1. Improving early detection and prevention of heart disease: The major objective is to improve early detection and prevention of heart disease. By constructing an efficient predictive model, the aim is to identify patients at risk and employ personalized preventive strategies. This aims to lessen the effect of heart disease, and to enhance overall cardiovascular health.

2. Improving patient outcomes and life quality: the purpose of early detection is to lower the intensity of heart disease, limit consequences, and enhance patient’s overall well-being.

3. Enhancing healthcare resources: Another goal in controlling heart disease is to optimize healthcare resources. By applying effective risk assessment strategies, the objective is to

properly allocate resources, improve the overall efficiency and sustainability of healthcare organizations, and reduce healthcare costs.

Overall, these business objectives aim to improve early detection, optimize resources, enhance patient outcomes. By accomplishing these goals, the project aims to make a positive impact on individuals' lives, healthcare organizations, and the general field of heart wellness.

4. DATA UNDERSTANDING

The dataset utilized in this project is the Framingham heart study dataset. Its sourced from Kaggle, a popular online platform that provides a vast collection of datasets for exploration and analysis.

The Framingham Heart Study is a long-term, ongoing cardiovascular study conducted in Framingham, Massachusetts, USA. It began in 1948 and has provided valuable insights into the risk factors for cardiovascular disease. The dataset contains information on various risk factors such as age, gender, blood pressure, cholesterol levels, smoking status, diabetes status, and family history of heart disease. It has been instrumental in identifying and understanding the major risk factors associated with cardiovascular diseases and has contributed to the development of widely used risk assessment tools like the Framingham Risk Score.

We renamed some variables so the names become more meaningful.

Below is a table describing each variable present in the dataset (BHARDWAJ, n.d.), it gives key information such as variable names, descriptions, and data types.

Variable Name	**Description**	**Type**
gender	The gender of the participant 0: female 1: male	Integer
age	Age of the patient in years.	Integer
education	The education level of the participant. 1: Less than High School 2: High School 3: Undergraduate Degree 4: Graduate Degree	Float

currentSmoker	Indicates whether the participant is a current smoker 0: non-smoker 1: smoker	Integer
CigsPerDay	Represents the number of cigarettes smoked per day	Float
BPMeds	Indicates whether the participant is on blood pressure medication 0: no usage 1: medication usage	Float
prelevantStroke	Indicates whether the participant had a previous stroke 0: no stroke 1: stroke	Integer
prelevantHypertension	Indicates whether the participant has prevalent hypertension 0: no hypertension 1: hypertension	Integer
diabetes	Indicates whether the participant has diabetes 0: no diabetes 1: diabetes	Integer
totalCholesterolLevel	Represents the total cholesterol level in mg/dL.	Float

systolicBP	Represents the systolic blood pressure in mmHg.	Float
diastolicBP	Represents the diastolic blood pressure in mmHg.	Float
BMI	Represents the body mass index (weight in kg divided by height in meters squared).	Float
heartRate	Represents the resting heart rate in beats per minute.	Float
glucose	Represents the fasting blood sugar level in mg/dL.	Float
TenYearCHD	The target variable indicating the presence or absence of coronary heart disease within ten years 0: absence of heart disease 1: presence of heart disease	Integer

Table 1 Variables Description

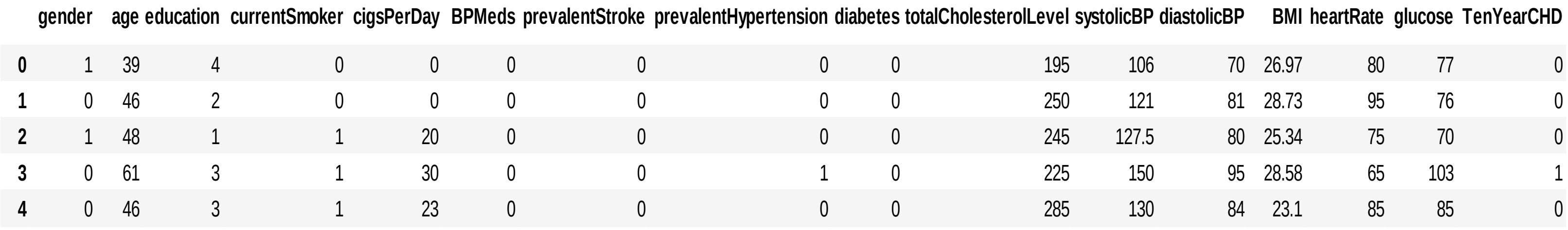
To gain a better understanding of the dataset, we conducted exploratory data analysis.

Checking the "Framingham" heart disease dataset dimensions, it contains 16 columns and 4240 rows. The goal of the dataset is to predict whether the patient has 10-year risk of future (CHD) coronary heart disease.

Below is the data structure represented by the first few rows of data:

Next, we ensured that our dataset doesn’t contain duplicates. This is an important step since duplicates can lead to inflated statistical results and distort the distribution of data, potentially resulting in biased analysis and misleading patterns.

Furthermore, we checked the correlation to quantify the relationships between the variables using a heatmap. Correlation is commonly used to analyze and understand the strength and direction of the relationship between variables, helping to identify patterns, dependencies, and potential connections in data. The correlation coefficient ranges from -1 to +1, with positive values indicating a positive correlation, negative values indicating a negative correlation, and a value of 0 indicating no correlation. A correlation close to +1 or -1 suggests a strong relationship, while a value close to 0 indicates a weak relationship between the variables.



	gender	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHypertension	diabetes	totalCholesterolLevel	systolicBP	diastolicBP	BMI	heartRate	glucose	TenYearCHD
0	1	39	4	0	0	0	0	0	0	195	106	70	26.97	80	77	0
1	0	46	2	0	0	0	0	0	0	250	121	81	28.73	95	76	0
2	1	48	1	1	20	0	0	0	0	245	127.5	80	25.34	75	70	0
3	0	61	3	1	30	0	0	1	0	225	150	95	28.58	65	103	1
4	0	46	3	1	23	0	0	0	0	285	130	84	23.1	85	85	0

Table 2 data frame

If two variables are highly correlated (correlation coefficient =0.7 or higher), it suggests a relatively strong positive relationship between the variables being analyzed. It indicates that there is a notable tendency for the variables to move together in a consistent manner, the two variables may provide similar information.

In a correlation heatmap, each variable is represented by a row and a column, and the cells show the correlation between them. The color of each cell represents the strength and direction of the correlation.

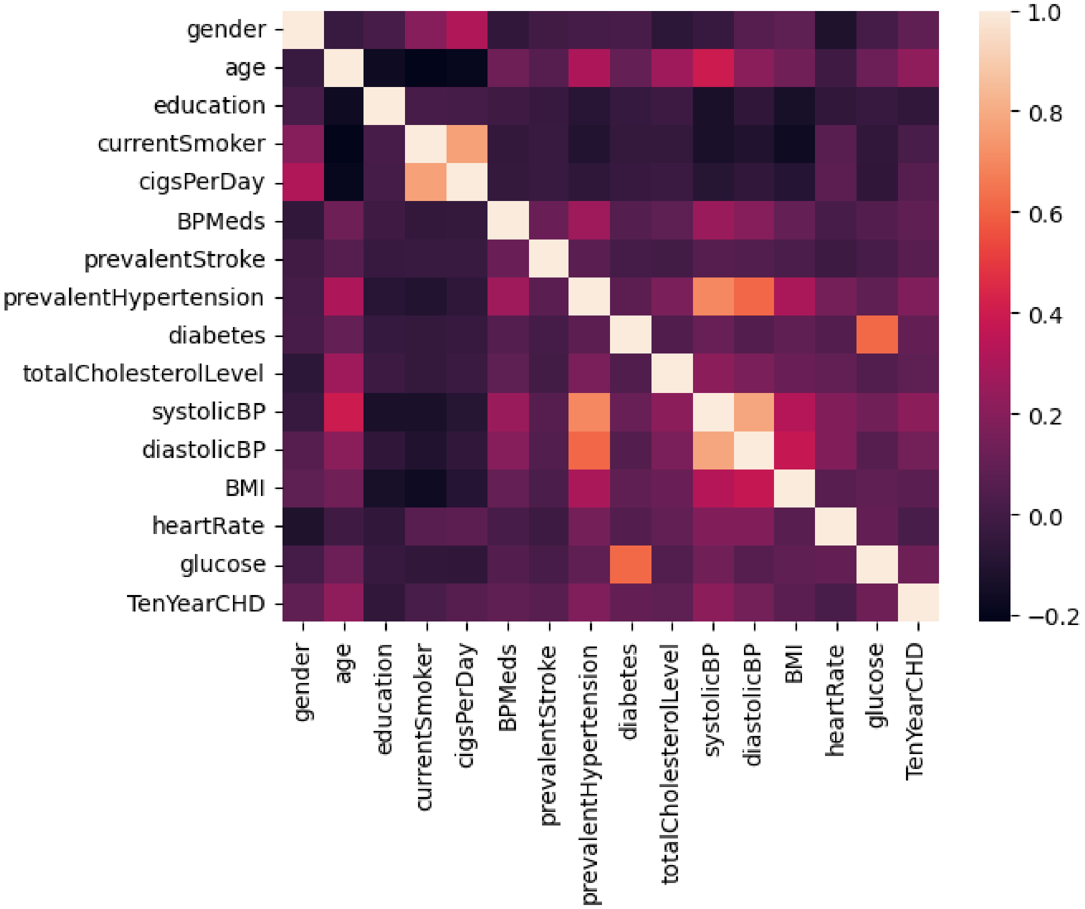


Figure 9 correlation Heatmap

The observation of a dark purple cell between the target variable TenYearCHD and the education variable indicates no correlation, which is reasonable. Thus, the education feature doesn't affect the target variable. So, we will be removing this column from the dataset.

Moreover, using calculations, we were able to obtain precise numerical values, and discovered a strong correlation of 0.769 between the features CigsPerDay and currentSmoker, which makes sense since the people that doesn't smoke have the same number of cigarettes per day (0).

We will be removing the currentSmoker column since it brings less information than CigsPerDay.

A strong correlation of 0.78 is also observable between diastolicBP and systolicBP. These two variables are supposed to be dependent on each other, both measurements are important in evaluating overall blood pressure levels. So, we will not drop any of these columns.

These first findings were the first step in analyzing data and identifying potential trends. More analysis is done in next sections.

The data understanding phase gave us a significant comprehension and determined the main qualities of the dataset used for our project. This understanding provides a basis for the upcoming stages of analysis and modeling.

5. DATA PREPARATION

5.1. HANDLING MISSING DATA

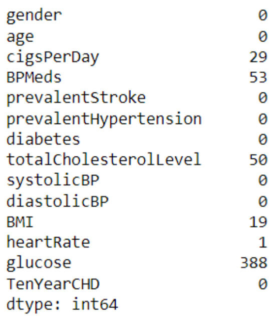
Missing data refers to the absence or lack of values for certain variables or observations in a dataset. It occurs when data points are not recorded, not available, or are incomplete for various reasons. The presence of missing data can have implications on data analysis and modeling, as it can introduce bias and affect the accuracy of results if not handled appropriately.

Missing data can occur due to several reasons, including:

1. Non-response: In surveys or questionnaires, respondents may choose not to answer specific questions, leading to missing values for those variables.
2. Data entry errors: During data collection or data entry processes, errors can occur, resulting in missing values or inconsistencies in the dataset.
3. Data loss: Technical issues, storage problems, or errors during data transfer can lead to the loss of data, resulting in missing values.
4. Incomplete records: In some cases, relevant information may not be available or may not have been collected for specific observations, leading to missing values for those observations.

The presence of missing data poses challenges for data analysis and modeling because it can lead to biased or incomplete results. Ignoring missing data or simply excluding observations with missing values can result in loss of valuable information and may introduce bias if the missingness is not random.

In the dataset used for building the model, we can notice the presence of missing data in some features;



```
gender                     0
age                        0
cigsPerDay                29
BPMeds                    53
prevalentStroke            0
prevalentHypertension      0
diabetes                   0
totalCholesterolLevel     50
systolicBP                 0
diastolicBP                0
BMI                       19
heartRate                  1
glucose                  388
TenYearCHD                 0
dtype: int64
```

Figure 10 missing values

Below are the feature names and the percentage of missing data in each feature:

- cigsPerDay: 0.68%
- BPMeds: 1.25%
- totalCholesterolLevel: 1.18%
- BMI: 0.45%
- heartRate: 0.02%
- glucose: 9.15%

Handling missing data can be done using several methods, we will explore in details each method, its advantages and disadvantages and when it should be used. By the end of Data preparation, we will be able to choose which method will be the most suitable for our dataset.

5.1.1. REMOVING MISSING VALUES

Removing NAs, which refers to removing observations or variables with missing values from a dataset, is one approach to handling missing data. This approach involves excluding the incomplete data points from the analysis. Removing NAs can be a suitable option under certain circumstances, but it should be done carefully, as it can lead to loss of information and potential bias if the missingness is not random.

Here are some key points to consider when removing NAs from a dataset:

1. Missingness pattern: Analyzing the pattern and mechanism of missingness is crucial. If missing values are completely at random (MCAR), where the missingness is unrelated to the observed or unobserved variables, removing NAs may not introduce significant bias. However, if there is a systematic pattern in missingness (e.g., missingness related to the outcome or other variables), removing NAs can lead to biased results.
2. Missing data percentage: Assess the proportion of missing values in the dataset. If the missingness is minimal and only affects a small portion of the data, removing NAs may have minimal impact on the overall analysis. However, if a substantial portion of the data is missing, removing NAs can result in a significant loss of information and potentially bias the analysis.
3. Variable importance: Consider the importance of the variable with missing values in the analysis. If the variable is critical and its missingness is expected to have a substantial

impact on the research question, removing NAs can significantly reduce the sample size and may not be appropriate. In such cases, imputation techniques or analysis methods robust to missing data should be considered.

4. Potential bias: Assess whether removing NAs may introduce bias due to the missingness mechanism. If missing values are related to specific characteristics or factors associated with the variables of interest, removing NAs can result in biased estimates or conclusions.

When removing NAs from a dataset, it is important to clearly document the missingness handling approach and justify the decision based on the above considerations. Additionally, sensitivity analyses can be conducted to examine the robustness of the results to the missing data handling approach.

It's worth noting that removing NAs should be used judiciously and after careful consideration of the dataset and research objectives. In some cases, employing imputation methods or utilizing analysis techniques robust to missing data may be more appropriate to mitigate the potential bias and loss of information associated with missing values.

In case we want to remove NAs from our dataset, the new dataset will contain 3751 instances instead of 4240, which means the loss of 489 instances (11.5% of the dataset).

5.1.2. IMPUTING MISSING VALUES

5.1.2.1. Filling by Mean & Mode

It is usual practice to handle missing data in a dataset by filling NAs with mean and mode values. It entails substituting the missing values with the corresponding variable's mean (for numerical variables) or mode (for categorical variables). The missing values are assumed to be fully random (MCAR) or to be missing at random (MAR) and to be appropriately represented by the mean or mode in this method.

While using mean and mode to fill NAs can be a quick and simple solution, there are certain drawbacks to consider:

A. Potential for bias: When missing values are associated with the variables being imputed or other unobserved factors, bias may be introduced when missing values are imputed using mean and mode. It is presumptively true that the observed values and the missing values share the same properties.

B. Reduced variance: Since all missing values will be replaced with the same value, using the mean or mode to impute missing values can minimize the volatility of the imputed variable. This decrease in variance can overestimate the level of uncertainty surrounding the imputed values.

C. Relationship distortion: Replacing NAs in a dataset with the mean or mode may cause correlations or patterns to be distorted since the imputed values may not precisely reflect the underlying values.

When considering whether to impute missing values using mean and mode, it is crucial to be aware of these restrictions as well as take the nature of the data and research goals into account. To handle missing values more skillfully while considering the relationships in the dataset, more sophisticated imputation approaches can be used, such as regression imputation, multiple imputation, or machine learning-based imputation techniques.

5.1.2.2. Filling by KNN

An alternate method for managing missing data in a dataset is to fill NAs using the k-nearest neighbors (KNN) algorithm. By estimating missing values based on the similarity between observations, KNN imputation makes use of the data that is already accessible. When the missingness is not entirely random and the correlations between the variables are crucial for precise imputation, this method is especially helpful.

Here's an overview of the process of filling NAs using the KNN imputation method:

1. Identify variables with missing values: Begin by identifying the variables in your dataset that contain missing values.

2. Prepare the dataset: If your dataset contains both numeric and categorical variables, it's important to preprocess them appropriately. Numeric variables may require scaling, while

categorical variables may need to be transformed into dummy variables or encoded numerically for distance calculation.

3. Determine the value of k: Choose a suitable value for k, which represents the number of nearest neighbors used to estimate the missing value. The optimal k value depends on the dataset size, the number of variables, and the underlying relationships between observations.
4. Calculate similarity: For each observation with missing values, calculate the similarity (distance) between that observation and all other observations in the dataset using appropriate distance metrics. Euclidean distance is commonly used for numeric variables, while other metrics like Hamming distance or Jaccard distance can be used for categorical variables.
5. Select nearest neighbors: Identify the k nearest neighbors to the observation with missing values based on the calculated distances.
6. Impute missing values: For each missing value, use the values from the k nearest neighbors to estimate the missing value. For numeric variables, impute the missing value as the average (mean or median) of the corresponding variable values in the k nearest neighbors. For categorical variables, impute the missing value as the mode (most frequent value) among the corresponding variable values in the k nearest neighbors.
7. Repeat for all missing values: Continue the imputation process for all variables with missing values in the dataset.
8. Restore the dataset: Once the missing values have been imputed, the dataset can be used for further analysis or modeling.

It's crucial to remember that KNN imputation makes the sometimes-unrealistic assumption that observations with comparable properties have similar values. It is advised to assess the imputation performance using proper assessment metrics or through cross-validation because the choice of the k value might also affect the accuracy of the imputation.

KNN imputation is a viable option for addressing missing data, particularly when there is a pattern in the missingness or where links between observations are crucial for accurate imputation. The imputed values must be carefully interpreted in light of the unique dataset and research goals while being conscious of any potential restrictions.

5.1.2.3. Filling by Next

This approach involves replacing missing values with the next available value in the variable's sequence. It is commonly used when the missing values represent a continuation of the previous observed values.

It is an easy and clear process to fill NAs with the "next" value. However, it might not always be suitable, especially if the missingness has nothing to do with the order or sequencing of the data. This technique also counts on the missing data having a constant pattern; however, if the pattern changes, bias may be introduced.

The nature of the data, the missingness mechanism, and the effects of replacing missing values with the "next" value must all be taken into account. For a thorough study of the data, it may be required to evaluate the imputation findings and take different approaches into account, such as imputation methods based on statistical modelling or other imputation algorithms.

5.1.2.4. Filling by Previous

This approach is very similar to the previous one except that in this case, missing values will be replaced by the previous available value in the variable’s sequence and then will be placed in the dataset for further analysis.

5.2. OUTLIERS

An observation in a dataset that significantly deviates from the pattern or distribution of the data as a whole is referred to as an outlier. Data analysis and statistical modeling may be significantly impacted by these data points because of their unique distance from the majority of other observations.

Here’s a brief description of the presence of outliers in a dataset:

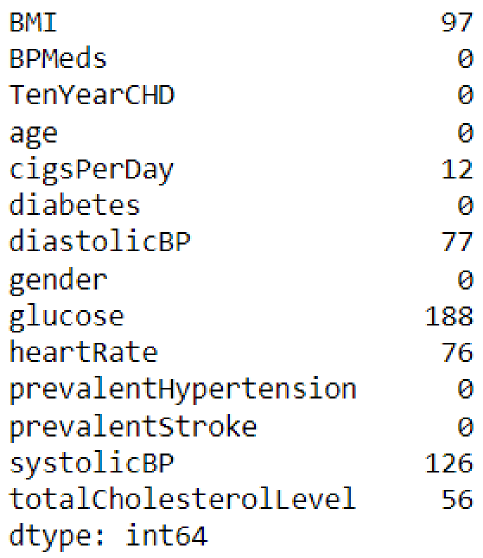
1. Definition of outliers: Outliers can be defined based on statistical methods or domain-specific knowledge. In statistical terms, outliers are often defined as observations that fall outside a certain range, such as being more than a certain number of standard deviations

away from the mean. Domain knowledge can also help identify outliers based on the specific context of the data.

2. Types of outliers: Outliers can be classified into two main types: univariate outliers and multivariate outliers. Univariate outliers are extreme values in a single variable, while multivariate outliers are observations that deviate from the overall pattern of multiple variables simultaneously.
3. Impact of outliers: Outliers can have various effects on data analysis. They can distort statistical measures, such as the mean and variance, leading to biased estimates. Outliers can also affect the results of predictive models, as they can have a disproportionate influence on the model's coefficients or decision boundaries. Additionally, outliers can impact the assumptions of statistical tests, leading to invalid conclusions.
4. Detection of outliers: Outliers can be identified through various methods, including graphical techniques, such as scatter plots, box plots, or histograms, which help visualize the distribution and identify extreme values. Statistical methods, such as the z-score, the interquartile range (IQR), or robust statistical measures, can also be employed to detect outliers.
5. Treatment of outliers: Once outliers are identified, the appropriate treatment depends on the specific context and objectives of the analysis. Outliers can be retained, transformed, or removed from the dataset. The choice of treatment should be made cautiously, considering the impact on the analysis and the underlying reasons for the outliers' presence.

Understanding the presence of outliers in a dataset is crucial for ensuring accurate and reliable data analysis. It is important to carefully examine and address outliers, considering the nature of the data, the analysis goals, and the potential implications on statistical modeling or decision-making processes.

Outliers can only be present in numerical features only as categorical variables do not follow a distribution. This is why we checked the outliers in our dataset using boxplots of numerical variables just to have an idea where the outliers reside. To be able to precisely detect the number of outliers in the dataset, we referred to the IQR method and obtained the below number of outliers in each numerical feature:



```
BMI                          97
BPMeds                        0
TenYearCHD                    0
age                           0
cigsPerDay                   12
diabetes                      0
diastolicBP                  77
gender                        0
glucose                     188
heartRate                    76
prevalentHypertension         0
prevalentStroke               0
systolicBP                  126
totalCholesterolLevel        56
dtype: int64
```

Figure 11 Outliers

Several approaches can be followed to treat outliers depending on several factors; choosing the best approach should be a very delicate procedure since outliers have a great impact on the dataset and the model.

We will explore briefly every approach and see if it can be used in our study, again the best approach will be chosen based on the highest score by the end of the study.

5.2.1. KEEPING THE OUTLIERS

Keeping outliers in a dataset means retaining the observations that are considered outliers instead of removing or transforming them. This approach recognizes the presence of extreme values and includes them in the analysis and modeling process.

Keeping outliers in a dataset can provide valuable insights and contribute to a comprehensive understanding of the data. However, it is crucial to carefully consider the impact of outliers on the analysis goals, statistical measures, and modeling techniques employed. Robust analysis methods and thoughtful interpretation can help mitigate the potential drawbacks and leverage the information provided by outliers.

5.2.2. REMOVING OUTLIERS BY STANDARD DEVIATION

Removing outliers by standard deviation is a method used to identify and remove observations that fall outside a certain range defined by the mean and standard deviation of the dataset. This approach assumes that the data follow a normal distribution and that extreme values beyond a specified threshold are considered outliers

It is simple to remove outliers by standard deviation, but there are a few things to keep in mind and certain restrictions:

1. It assumes that the data are distributed normally, which may not be true for all datasets.
2. Important data may be lost if outliers that represent legitimate extreme values or valid data points are deleted.
3. The number of observations classified as outliers and the outcomes can both be influenced by the threshold (number of standard deviations) used to define outliers.

When using this approach, it is crucial to carefully take into account the type of data, the context of the research, and any potential repercussions of deleting outliers. To manage outliers in a more robust and context-specific way, different outlier detection approaches or robust statistical techniques may also be taken into consideration.

After applying this approach in our dataset, we notice the loss of 463 instance which means the loss of around 10.9% of the original dataset.

5.2.3. REMOVING OUTLIERS BY IQR

The interquartile range (IQR) removal of outliers is a technique for locating and eliminating observations that fall outside of a given range depending on the distribution of the data. This method concentrates on the data's quartiles and is resistant to non-normal distributions.

A reliable technique that is insensitive to the particular distribution of the data is the removal of outliers using IQR. It does, however, have several restrictions and considerations:

- The number of data classified as outliers is influenced by the threshold selection (1.5 times or 3 times the IQR) and may have an impact on the outcomes.
- Important data may be lost if outliers that represent legitimate extreme values or valid data points are deleted.

When using this approach, it is crucial to carefully consider the type of data, the context of the research, and any potential repercussions of deleting outliers. To manage outliers in a more robust and context-specific way, different outlier detection approaches or robust statistical techniques may also be taken into consideration.

After using this approach in our dataset, when reassessing the updated dataset, we noticed that all the instances with a TenYearCHD (target variable) value of 1 were lost; this is why we will not be using this approach in our study because we cannot tolerate this loss.

5.2.4. REMOVING OUTLIERS BY DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise), a technique for locating and eliminating outliers based on the density and clustering patterns in a dataset, is used to find and eliminate outliers. Data points are grouped together using the density-based clustering algorithm DBSCAN depending on how close they are to one another.

DBSCAN's removal of outliers is helpful since it takes the data's density and clustering tendencies into account. It also has several restrictions and considerations, though:

- The selection of DBSCAN parameters (minPts and eps) can affect how outliers are detected. Different outlier detection outcomes could be obtained by adjusting these settings.
- Choosing the right distance metric is essential since DBSCAN is sensitive to the metric used.
- The features of the dataset and the existing clustering patterns determine how well DBSCAN removes outliers.

When we used this approach in our dataset, only 13 instances were lost (0.003% of the dataset). Although very few instances were lost but it does not mean that this is the best approach for treating outliers in the study. The best approach will be chosen based on the scores at the end.

5.3. FEATURE ENGINEERING

Feature engineering is the process of transforming raw data into a set of relevant features to improve machine learning model training, leading to better performance and greater accuracy. It involves selecting, creating, or transforming features to make them more informative or suitable for the predictive task at hand.

5.3.1. DATA SCALING

Data scaling is a preprocessing method used in machine learning and data analysis to standardize or change the numerical features of a dataset. It is sometimes referred to as feature scaling or normalization. Data scaling aims to equalize the scale of all the characteristics, preventing some features from dominating others due to their greater magnitude.

There are several ways to scale data, but two are frequently used:

- Min-Max scaling: The values are scaled using the Min-Max approach to a defined range, usually between 0 and 1. Each feature's minimum value is subtracted, and the result is divided by the range (highest value minus minimum value). This guarantees that the altered data is constrained to the given range.

- Standardization: Data are transformed using this standardization technique to have a mean of 0 and a standard deviation of 1. Each feature is given a mean value, which is subtracted, and the standard deviation is then divided. Standardization improves the data's normal distribution and reduces the impact of outliers.

We may make sure that features with various units or scales are treated equally during the learning process by scaling the data. Certain algorithms, such those that use distance calculations (like k-nearest neighbors) or gradient-based optimization (like gradient descent), may perform better as a result.

It's crucial to remember that the selection of a data scaling method is based on the details of the situation at hand as well as the characteristics of the data. Additionally, some machine learning methods, such tree-based models (like decision trees and random forests), may not need feature scaling since they are less sensitive to it.

5.3.2. FEATURE SELECTION

The objective of feature selection in machine learning and data analysis is to choose a smaller subset of pertinent features out of a larger pool of available features. By concentrating on the most informative features, the aim is to enhance interpretability, minimize complexity, and improve model performance.

5.3.2.1. Select K Best Function

To choose the K most useful features from a dataset, machine learning practitioners frequently utilize the SelectKBest function. It is frequently used in conjunction with a scoring function to order the features according to each one's significance to the desired variable.

According to a filter-based methodology, the SelectKBest function assesses each feature independently of the learning algorithm being employed. It works in two stages:

- Scoring: Based on a set of criteria, the scoring function assigns a score to each feature. For categorical variables, chi-squared, f_regression or mutual_info_regression are common scoring functions. For continuous variables in regression problems, f_classif or mutual_info_classif are common scoring functions. Higher scores are given to features that are more informative after the scoring function evaluates the link between each feature and the target variable.
- Selection: The SelectKBest function chooses the K features with the highest scores after scoring each feature individually. The user chooses the value of K, which denotes the desired number of features to be kept in the dataset.

In our dataset that contains 16 features, we chose K=10 for the Select K Best Function.

5.3.2.2. Select Percentile Function

This method is very similar to the select K Best function, instead of choosing K number of features, this model chooses a certain percentage of features. It chooses the best features based on the percentage needed inputted by the user.

This function also has two steps: scoring and then selection; scoring is done exactly like in select K Best function and selection is made based on the percentage needed. For example, if we have 100 features in our dataset and we choose a percentile of 10, the function will choose the 10 (10% of 100) features that have the highest score.

We chose a percentile of 90 in our dataset.

Due to its adaptability, the SelectPercentile function enables a more dynamic selection of characteristics based on their ratings. It automatically adjusts to the features in the dataset by using a percentile threshold and chooses a different number of features depending on their relative importance. When features are numerous or vary across multiple datasets, this can be useful.

SelectPercentile, like other filter-based approaches, simply takes into account the relevance of each characteristic individually, leaving out any interactions or combinations of features. The scoring function must be carefully chosen in order to match the challenge and the nature of the characteristics.

It is advised to examine the performance of the chosen features with the learning algorithm after using SelectPercentile to make sure they contribute to correct predictions or classifications and to determine whether the selected percentile threshold is adequate for the particular task at hand.

5.3.2.3. Variance Threshold Function

A feature selection technique used in machine learning to eliminate features with minimal variance is the VarianceThreshold function. Low-variance features may not contain much relevant information in datasets with numerical characteristics, where they are very helpful.

Here is how the VarianceThreshold function works:

- ➢ Calculate Variance: The variance of each feature in the dataset is calculated. The spread or variability of values within a feature is measured by variance. Low variance characteristics imply that the majority of values are comparable or constant, making them less useful for modeling.
- ➢ Set Threshold: You can specify a threshold value using the VarianceThreshold function. Any feature that falls below this cutoff is deemed to have low variance, and it is then taken out of the dataset. The threshold value, which is specified by the user, establishes the bare minimum of variance necessary for the retention of a feature.

The VarianceThreshold function reduces the dimensionality of the dataset and gets rid of noise or useless features by excluding features with low variance. This may result in a model that performs better, takes less time to train, and is easier to interpret.

It's crucial to remember that the VarianceThreshold function does not account for the relationships between features or their applicability to the target variable; rather, it just considers the variance within each feature individually. As a result, it might not be appropriate in all circumstances, especially when interactions or feature combinations are important.

In our dataset, for a variance threshold of 0.015, 12 features remained in the dataset after performing the Variance Threshold function.

5.3.3. DIENSIONALITY REDUCTION

Dimensionality reduction refers to the process of reducing the number of features or variables in a dataset while preserving its essential information. It aims to address the challenges posed by high-dimensional data, such as computational complexity, potential overfitting, and difficulties in interpretability.

Dimensionality reduction was performed using the Principal Component Analysis (PCA). PCA is a widely used technique that transforms the original features into a new set of linearly uncorrelated variables called principal components. These components capture the maximum amount of variance present in the data while minimizing the loss of information.

PCA was applied with the objective of retaining 95% of the variance in the data. This technique transforms the original features into a new set of linearly uncorrelated variables called principal components. These principal components capture the most important information in the data.

The purpose of dimensionality reduction was to address the curse of dimensionality, improve computational efficiency, and potentially enhance model performance by eliminating irrelevant or redundant features while retaining important information."

5.4. BALANCING

Addressing class imbalance in a dataset is referred to as balancing in the context of data cleaning. Class imbalance happens when there are significantly more or fewer instances in one class than there are in other classes. This might cause biased models and subpar performance on the minority class in machine learning tasks, especially in classification.

Here are some typical methods for class-imbalanced dataset balancing during data cleaning:

- Undersampling
- Oversampling

We will go through these 2 methods in details in the below sections.

If we plot a bar plot of the target variable, we can notice that balancing data is mandatory for the dataset as there much more instances with a target variable of 0 than 1 as per the below bar plot:

We will name the class 0 the majority class and class 1 the minority class

5.4.1. UNDER SAMPLING

In order to make the minority class equivalent to the majority class, under sampling entails lowering the number of occurrences in the majority class. A subset of examples from the majority class equal to the number of instances in the minority class are normally chosen at random to accomplish this.

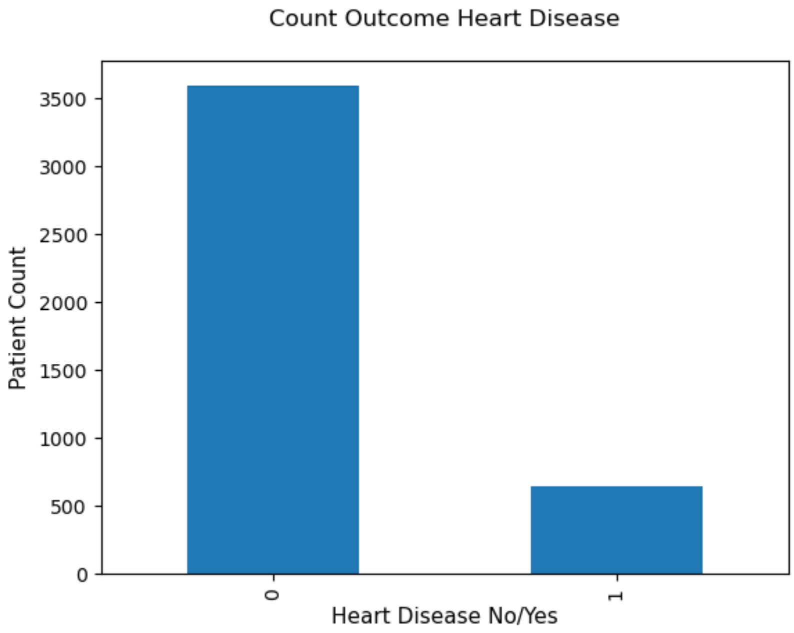


Figure 12: Bar Plot of TenYearCHD

Under sampling is used to produce a more balanced dataset so that the machine learning model can equally benefit from both types of data. Under sampling, however, can lead to the loss of information from the majority class, and thus might not be appropriate if the amount of data already available is little.

It's important to keep in mind that under sampling is just one method for addressing class imbalance; there are also methods like over sampling, creating synthetic data, or employing algorithms created particularly to handle unbalanced data. The dataset's unique properties and the needs of the current challenge will determine which technique is used.

Below is a bar plot of the target variable after executing under sampling:

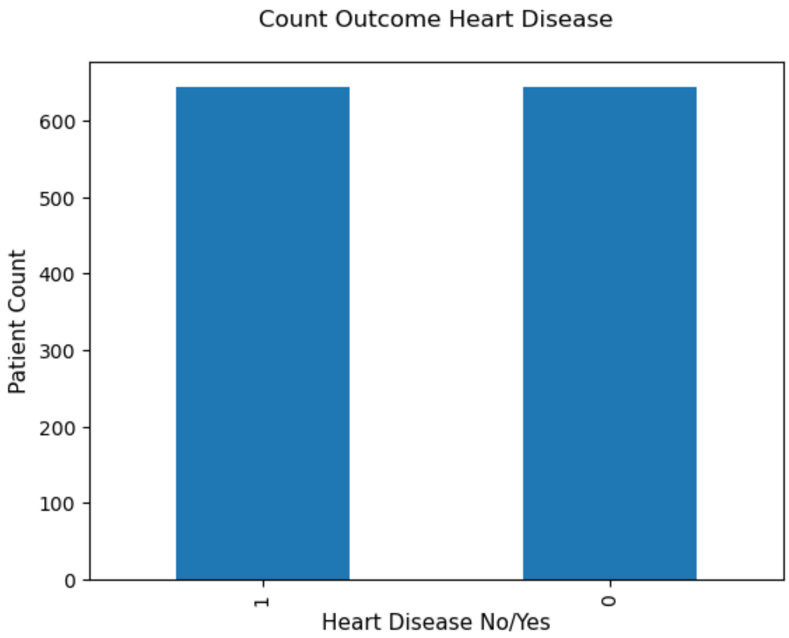


Figure 13: Bar Plot of TenYearCHD after under sampling

After under sampling, we can clearly notice that the shape of the dataset was reduced to 1288 instances instead of 4240, a loss of 2952 instances (70% of the data was lost).

5.4.2. OVER SAMPLING

Over sampling comprises increasing the number of occurrences in the minority class in order to bring the minority class into parity with the majority class. This is sometimes done by replicating or duplicating already existing instances from that group in order to raise the representation of the minority class.

A more balanced dataset is created through oversampling so that the machine learning model can equally profit from both types of data. By increasing the proportion of examples in that class, the model can more accurately depict the patterns and characteristics of the minority class.

Oversampling, however, carries the risk of overfitting, in which the model underperforms on unobserved data due to becoming overly specialized to the training data.

Below is a bar plot of the target variable after executing over sampling:

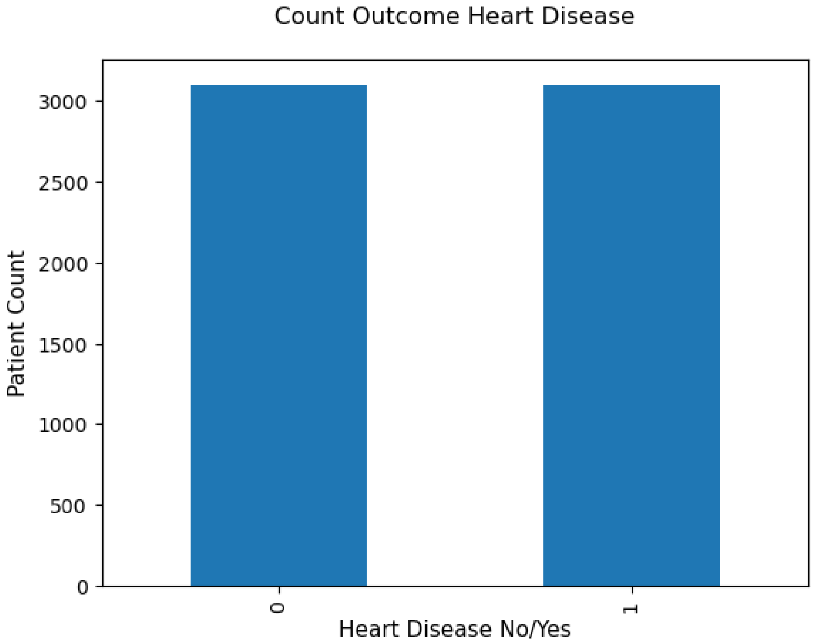


Figure 14: Bar Plot of TenYearCHD after over sampling

After under sampling, we can clearly notice that the shape of the dataset has increased to 6202 instances instead of 4240, an addition of 1962 instances.

6. MODELING

In this part of the project, we will be testing several supervised learning models:

1. **KNN**
2. **Logisitic Regression**
3. **Decision tree**
4. **Naïve Bayes**
5. **Random Forest**
6. **Support Vector Machine (SVM)**

We will write a function for each model that takes 30% of the dataset as test size and 70% for training. This function tries the model on all the possible datasets with all the possibilities for:

a. Dealing with missing values
b. Dealing with outliers
c. Features selection
d. Balancing

After executing the model and all the possible datasets and generating the recall, precision, accuracy and F1-score, the function will generate the top 5 datasets with the highest scores.

This function will be executed for each model to finally know which model and which dataset will be the best when it comes to high recall, precision, accuracy and F1-score.

Let's briefly explain each metric in a machine learning model and how to interpret it:

i. Recall

For each prediction in the context of binary classification, there are four possible outcomes:

1. True Positive (TP): When the model predicts a positive instance as expected.
2. False Positive (FP): The model predicts erroneously that a positive instance will occur when a negative class would.
3. True Negative (TN): The model predicts a negative event with accuracy.

4. False Negative (FN): The model predicts erroneously that a given instance will be negative when in fact the class will be positive.

The ratio of true positives to the total of true positives and false negatives is used to determine recall, also known as sensitivity or true positive rate: **Recall = TP / (TP + FN)**

A recall value of 1 would be the perfect case which means that there are no false negatives.

ii. Precision

The ratio of true positives to the total of true positives and false positives is used to calculate precision: **Precision = TP / (TP + FP)**

In other words, precision is the percentage of the model's positive predictions that come true. It focuses on reducing false positives because, depending on the application, mistakenly classifying negative cases as positive can have detrimental effects.

A greater precision score means that the model is more adept at avoiding false positives, which means that the majority of its positive predictions are accurate. This could, however, result in more false negatives (FN), which are positive cases that are mistakenly categorized as negative.

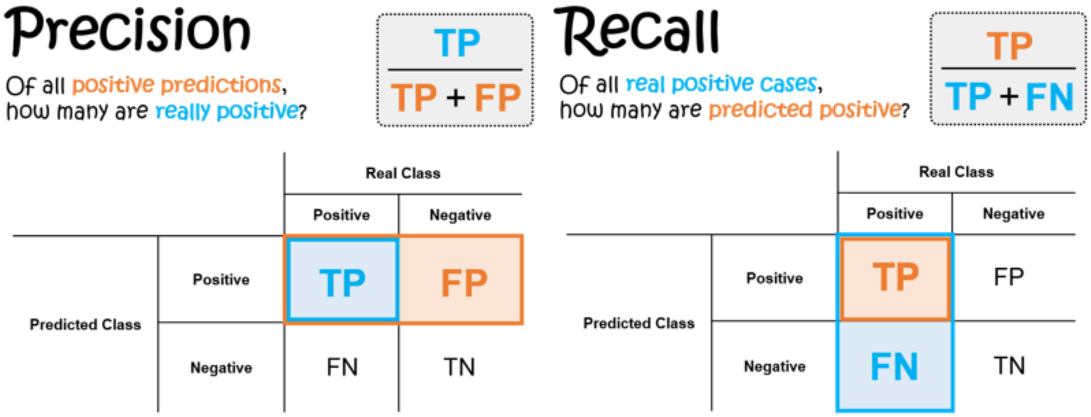


Figure 15: Precision & Recall

iii. Accuracy

In machine learning, accuracy is a frequently used metric to assess a classification model's overall correctness. Out of all the examples, it calculates the percentage of instances that were correctly categorised.

Accuracy is calculated as the ratio of the sum of true positives and true negatives to the total number of instances: **Accuracy = (TP + TN) / (TP + TN + FP + FN)**

A higher accuracy value means that a greater percentage of the model's predictions were accurate. However, accuracy alone might not give a whole view of the model's performance, particularly in datasets with imbalances where the proportion of cases in various classes vary greatly.

iv. F1-score

A popular performance indicator in machine learning, particularly for binary classification problems, is the F1 score. In order to provide a fair assessment of a model's performance, it combines the precision and recall measurements.

Recall is the percentage of accurately anticipated positive occurrences out of all positively predicted instances, whereas precision measures the percentage of accurately predicted positive instances out of all really occurring positive instances. The harmonic mean of these two metrics, which accounts for both precision and recall, is used to produce the F1 score:

F1 score = 2 * (precision * recall) / (precision + recall)

The F1 score is a number between 0 and 1, where a value of 1 indicates perfect precision and recall and signifies that the model successfully balanced the two criteria. A better-performing model is one with a higher F1 score, while a model with a lower score may have precision and recall biases.

ROC Curve

Another metric used in model evaluation is ROC curve. An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve displays the relationship between the two parameters:

True Positive Rate

False Positive Rate

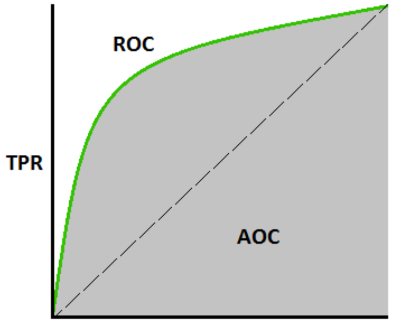


Figure 16 Understanding ROC curve

The ROC curve is widely used in evaluating binary classifiers and their ability to discriminate between classes. It provides a comprehensive view of the model's performance across various decision thresholds, allowing us to assess its sensitivity and specificity trade-off. The more that the ROC curve hugs the top left corner of the plot, the better the model does at classifying the data into categories.

The Area Under the ROC Curve (AUC) served as a summary metric for the overall performance of our model. A higher AUC value indicates better discrimination ability, with 1.0 representing a perfect classifier and 0.5 indicating a random classifier. ROC-AUC does not work well under severe imbalance in the dataset.

6.1. KNN

After deploying KNN to our database, we can notice the below are the top 5 data frames with the best metrics:

Rank	Missing values	Outliers	Features Selection	Balancing
1	Fill by Next	Remove by Standard Deviation	Keep all the features	Oversampling
2	Fill by Previous	Remove by Standard Deviation	Select Percentile	Oversampling
3	Remove NAs	Remove by Standard Deviation	Keep all the features	Oversampling
4	Remove NAs	Remove by Standard Deviation	Select Percentile	Oversampling
5	Fill by Previous	Remove by Standard Deviation	Keep all the features	Oversampling

Table 3: Best data frames for KNN

6.2. LOGISTIC REGRESSION

After deploying Logistic Regression to our database, we can notice the below are the top 5 data frames with the best metrics:

Rank	Missing values	Outliers	Features Selection	Balancing
1	Fill by Previous	Remove by Standard Deviation	Dim Reduction	Oversampling
2	Remove NAs	Remove by Standard Deviation	Select Percentile	Oversampling
3	Remove NAs	Remove by Standard Deviation	Variance Threshold	Oversampling
4	Remove NAs	Remove by Standard Deviation	Dim Reduction	Oversampling

5	Remove NAs	Remove by Standard Deviation	Keep all the features	Oversampling

Table 4: Best Data frames for Logistic Regression

6.3. DECISION TREE

In our heart disease risk assessment project, the final decision will be 0 or 1 (0: no heart disease, 1: heart disease) based on the features.

After deploying Decision Tree to our database, we can notice the below are the top 5 data frames with the best metrics:

Rank	Missing values	Outliers	Features Selection	Balancing
1	Remove NAs	Remove by Standard Deviation	Select Percentile	Oversampling
2	Fill by Previous	Remove by Standard Deviation	Keep all the features	Oversampling
3	Fill by KNN	Remove by Standard Deviation	Keep all the features	Oversampling
4	Fill by Mean and Mode	Remove by Standard Deviation	Dim Reduction	Oversampling
5	Fill by KNN	Remove by Standard Deviation	Select Percentile	Oversampling

Table 5: Best Data frames for Decision Tree

6.4. NAIVE BAYES

After deploying Naïve Bayes to our database, we can notice the below are the top 5 data frames with the best metrics:

Rank	Missing values	Outliers	Features Selection	Balancing
1	Fill by Mean and Mode	Remove by Standard Deviation	Dim Reduction	Oversampling
2	Fill by Previous	Remove by Standard Deviation	Dim Reduction	Oversampling
3	Remove NAs	Remove by Standard Deviation	Dim Reduction	Oversampling
4	Fill by KNN	Remove by Standard Deviation	Dim Reduction	Oversampling
5	Fill by KNN	Remove by Standard Deviation	Dim Reduction	Oversampling

Table 6: Best Data frames for Naive Bayes

6.5. RANDOM FOREST

After deploying Random Forest algorithm to our database, we chose n=100 (100 decision trees were generated), we can notice the below are the top 5 data frames with the best metrics:

Rank	Missing values	Outliers	Features Selection	Balancing
1	Fill by Mean and Mode	Remove by Standard Deviation	Keep all the features	Oversampling
2	Fill by Next	Remove by Standard Deviation	Keep all the features	Oversampling
3	Fill by KNN	Remove by Standard Deviation	Keep all features	Oversampling
4	Fill by Previous	Remove by Standard Deviation	Keep all the features	Oversampling
5	Fill by Mean and Mode	Remove by Standard Deviation	Select K best	Oversampling

Table 7: Best Data frames for Random Forest

6.6. SVM

After deploying SVM algorithm to our database, we can notice the below are the top 5 data frames with the best metrics:

Rank	Missing values	Outliers	Features Selection	Balancing
1	Fill by Previous	Remove by Standard Deviation	Dim Reduction	Oversampling
2	Remove NAs	Remove by Standard Deviation	Dim Reduction	Oversampling
3	Fill by Mean and Node	Remove by Standard Deviation	Dim Reduction	Oversampling
4	Remove NAs	Remove by Standard Deviation	Select Percentile	Oversampling
5	Remove NAs	Remove by Standard Deviation	Variance Threshold	Oversampling

Table 8: Best Data frames for SVM

7. EVALUATION

In the previous chapter, we were able to recognize which are the best data frames for each algorithm, but the ultimate objective of the problem is to find the best model and the best data frame. We will see how in the below section.

CHOOSING THE OPTIMAL MODEL

In the previous chapter, we generated the 5 best data frames for each model. We chose 6 models so in total we generated 30 data frames.

We will now create a function that will generate the top 3 models and data frames among these 30 data frames. Below ate the steps we followed to do this function:

1. Store all the data frames into one single list
2. Sort this list according to the metrics
3. Filter the first 3 models and their respective data frame

As a result, we will obtain the below model(s) and their respective data frames:

Rank	Model	Missing values	Outliers	Features Selection	Balancing
1	Random Forest	Fill by Mean and Mode	Remove by Standard Deviation	Keep all the features	Oversampling
2	Random Forest	Fill by Next	Remove by Standard Deviation	Keep all the features	Oversampling
3	Random Forest	Fill by KNN	Remove by Standard Deviation	Keep all the features	Oversampling

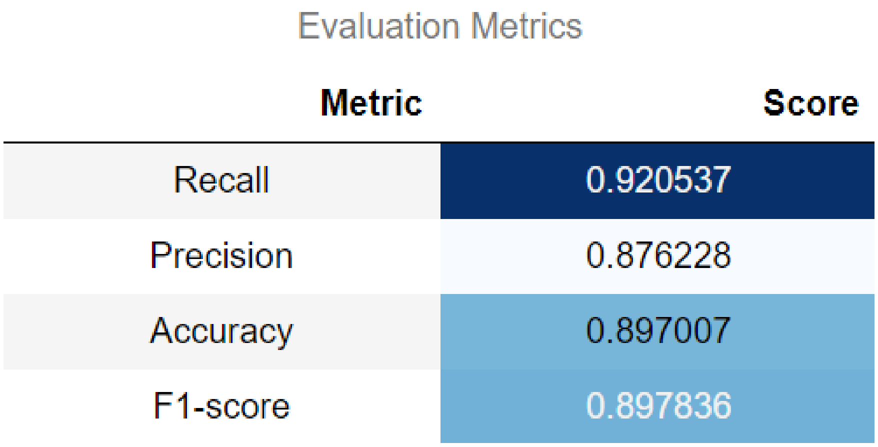
Table 9: The Best Models

We notice that Random Forest is the best model for this problem but the best data frame is the one in which we should:

1. Fill the missing values by Mean and Median
2. Remove outliers by Standard Deviation
3. Keep all the features
4. Perform oversampling to balance the data

The shape of the best data frame is 6570 instances and 14 features (currentSmoker and education were initially removed since they are not significant).

Below are the metrics of the random forest algorithm for the best data frame:

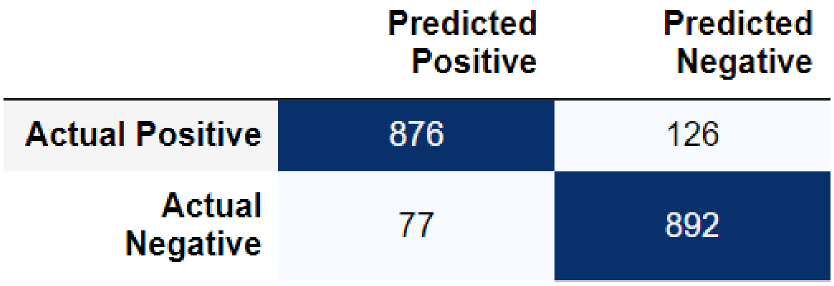


Evaluation Metrics

Metric	Score
Recall	0.920537
Precision	0.876228
Accuracy	0.897007
F1-score	0.897836

Table 10 Metrics of the best model

Below is the confusion matrix that shows the number of TP, FP, TN and FN:



	Predicted Positive	Predicted Negative
Actual Positive	876	126
Actual Negative	77	892

Table 11 Confusion matrix

The following graph presents the ROC curve illustrating the performance of our classifier in distinguishing between positive and negative classes:

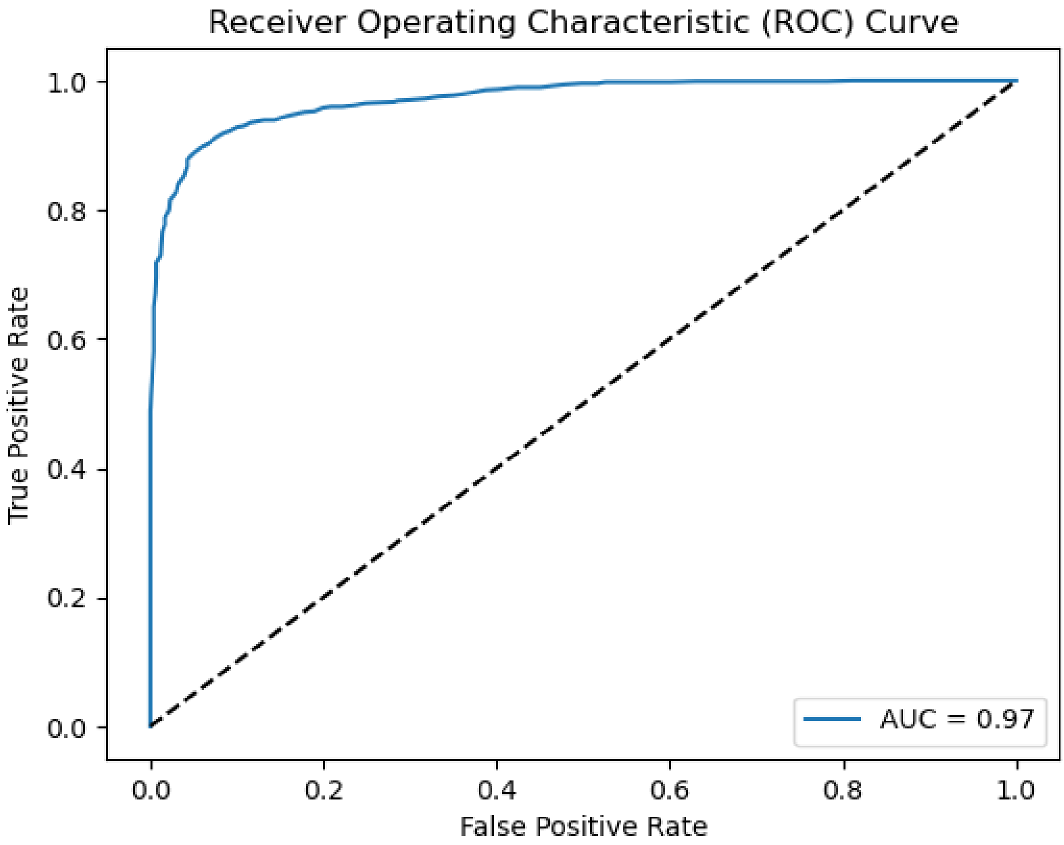


Figure 17 ROC Curve

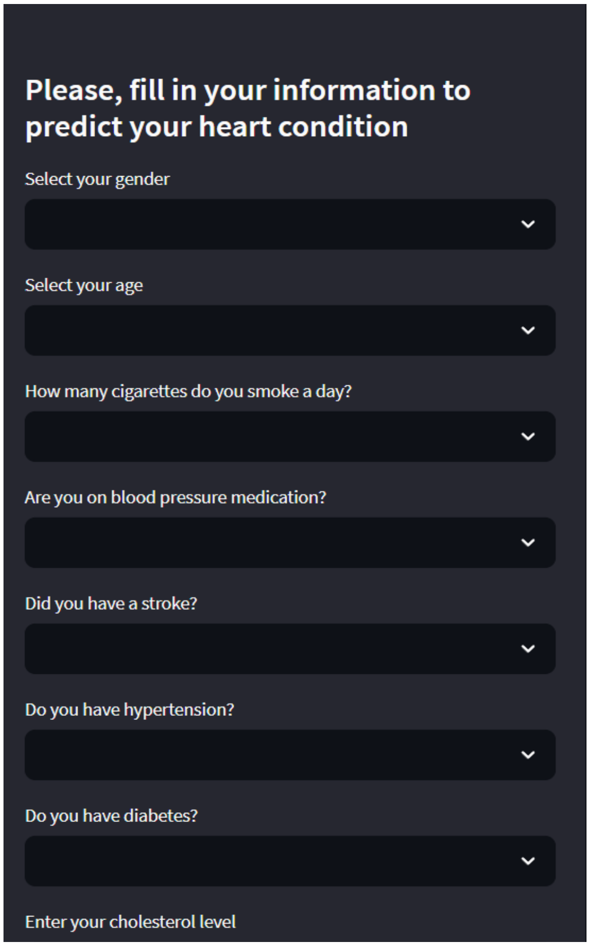
The Area Under the ROC Curve (AUC) for our classifier is 0.97. This signifies that our model has a high discriminatory power and performs significantly better than random chance. An AUC of 0.97 indicates strong predictive capabilities, as it demonstrates a high true positive rate while maintaining a low false positive rate.

8. DEPLOYMENT

In this final phase of the CRISP-DM process, the deployment phase, the heart disease prediction model was integrated into a user-friendly web application using Streamlit. Streamlit is an open-source python framework that simplifies the process of creating interactive web applications for data science and machine learning projects. It enables the development of an app in the same way of writing a Python code.

The goal of the deployment stage is to create a user interface that accepts personal information from individuals as input, and returns prediction results for their risk of heart disease. When the patients visit a clinic or a health organization, and get their different health tests' results, they can insert these metrics into the input fields, and they will get a personalized risk score of developing heart disease.

The application's user interface was designed to be simple and user-friendly. It consists of a single page with input fields for relevant information, such as age, gender, blood pressure, cholesterol levels, and the other key features used while training the model.



Please, fill in your information to predict your heart condition

Select your gender

Select your age

How many cigarettes do you smoke a day?

Are you on blood pressure medication?

Did you have a stroke?

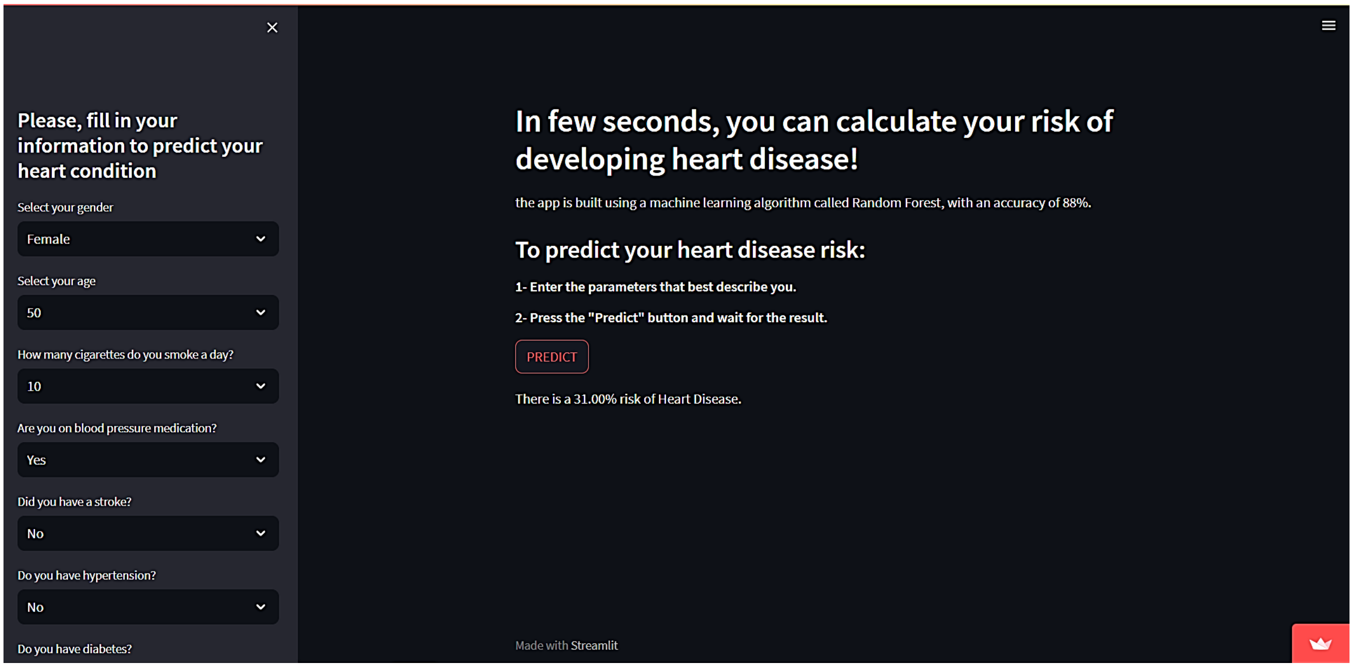
Do you have hypertension?

Do you have diabetes?

Enter your cholesterol level

Figure 18 user-interface / subset of input fields

After the input is validated, it is passed to the heart disease prediction model, previously developed and evaluated during the model development phase. The model processes the input features and generates a prediction result. The prediction result is then displayed to the patient in a user-friendly format, indicating "No Heart Disease Risk." or "There is a (…%) risk of Heart Disease." with a percentage representing the estimated risk level based on the input information.



Please, fill in your information to predict your heart condition

Select your gender: Female

Select your age: 50

How many cigarettes do you smoke a day?: 10

Are you on blood pressure medication?: Yes

Did you have a stroke?: No

Do you have hypertension?: No

Do you have diabetes?

In few seconds, you can calculate your risk of developing heart disease!

the app is built using a machine learning algorithm called Random Forest, with an accuracy of 88%.

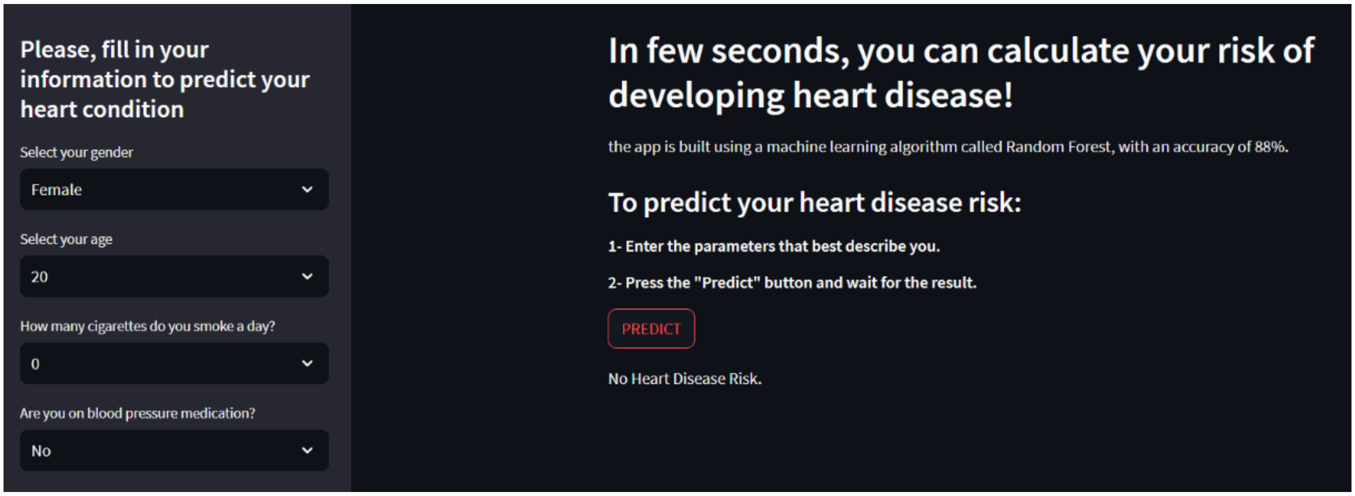
To predict your heart disease risk:

1- Enter the parameters that best describe you.

2- Press the "Predict" button and wait for the result.

There is a 31.00% risk of Heart Disease.

Figure 19 prediction _ heart disease risk



Please, fill in your information to predict your heart condition

Select your gender: Female

Select your age: 20

How many cigarettes do you smoke a day?: 0

Are you on blood pressure medication?: No

In few seconds, you can calculate your risk of developing heart disease!

the app is built using a machine learning algorithm called Random Forest, with an accuracy of 88%.

To predict your heart disease risk:

1- Enter the parameters that best describe you.

2- Press the "Predict" button and wait for the result.

No Heart Disease Risk.

Figure 20 prediction _ No risk

The application went through testing to ensure its functionality and accuracy. Then, it was hosted on Streamlit Share to make it accessible to users. Streamlit Share is a platform provided by Streamlit that simplifies the process of deployment and sharing of Streamlit applications with others.

The deployment phase successfully resulted in a user-friendly application using Streamlit for heart disease risk prediction, providing a platform where users could input their information and receive personalized risk predictions. By following a carefully designed user interface, incorporating a reliable machine learning model, the deployed application offers an accessible and valuable tool for individuals to assess their heart disease risk.

Click here to try the web application!

CONCLUSION

In conclusion, we were able to reach the project's main purpose which is to develop a predictive model that analyzed patient medical records in order to generate a risk score of heart disease.

All this was done through several steps: finding the right database with reasonable features and a good size, cleaning the database by deploying the best methods, trying all the possible modeling algorithms and then choosing the best one: Random Forest.

Before evaluating the models, we were certain that Random Forest algorithm would perfectly suit the project and its objectives since it has a very high accuracy, is efficient for large datasets like the one used in the project, does not overfit with more features.

After finding the right model, we were able to develop a simple web application that takes all the medical records as features to generate a risk score. Choosing the right model is never enough for any project to succeed if it is not deployed in real life to help human beings. This was the main idea behind the app.

It is well known that heart disease is one of the top reasons behind deaths worldwide; in poor and wealthy countries. This is why the objective of the project is not only scientific but also humanitarian as it helps doctors and patients detect the possibility of heart disease at early stages, to avoid it.

Several challenges were encountered during this project but we were thankfully able to face them and surpass them. The first challenge was in the first phase of the project: finding the right database. As we wanted an accurate database that contains patients' medical records, it was really challenging to find the right one but, after several weeks of intensive research, we were able to find the Framingham database from Kaggle.com which was the perfect fit to our project.

Another challenge encountered during the project was during the data cleaning phase; choosing the right methods and functions to clean the data (outliers, missing values, features and balancing) was very difficult as we were not able to compare them and did not know which method would best suit our database. This is why we searched further and got the idea of choosing all the possible methods and generate several databases to finally choose the one with the highest metrics for modeling.

Another challenge was in the deployment phase of our project; as data scientists, we did not previously have enough background in web development, this is why this phase of the project was challenging but we insisted to do it as it was the cherry on top of the project. With intensive research during several days, we were able to find the Streamlit library in Python that enabled us to develop a simple website that would do the job.

In today's rapidly advancing technological landscape, AI has emerged as a powerful tool with transformative potential in various industries. The medical field is no exception, as it stands to benefit significantly from the integration of AI technologies. The website being referred to can be seen as a tangible manifestation of this collaboration, acting as a platform that bridges the gap between medical professionals and AI experts.

One of the main ideas in the assertion is that data science and machine learning can be used for beneficial purposes. These fields have typically been linked to commercial or business uses, such as enhancing marketing tactics or streamlining corporate procedures. This website, however, demonstrates that its applicability goes beyond profit-driven undertakings and may be used to address important issues in healthcare.

REFERENCES

BHARDWAJ, A. (n.d.). *Framingham heart study dataset*. Retrieved from Kaggle: https://www.kaggle.com/datasets/aasheesh200/framingham-heart-study-dataset?resource=download

Decision Tree Classification Algorithm. (n.d.). Retrieved from javaTpoint: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

Jessica, S. (2022, January 15). *How Does Logistic Regression Work?* Retrieved from KDNuggets: https://www.kdnuggets.com/2022/07/logistic-regression-work.html

Naïve Bayes Classifier Algorithm. (n.d.). Retrieved from javaTpoint: https://www.javatpoint.com/machine-learning-naive-bayes-classifier

Random Forest Algorithm. (n.d.). Retrieved from javaTpoint: https://www.javatpoint.com/machine-learning-random-forest-algorithm

Support Vector Machine Algorithm. (n.d.). Retrieved from javaTpoint: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm