A dissertation submitted to the **University of Greenwich**
in partial fulfilment of the requirements for the Degree of

Master of Science
in
Data Science

Comparison of numerous classification models to predict coronary heart disease

Student Name: Wahidul Alam Riyad
Student ID: 001188274

Supervisor Name: Professor Jixin Ma
Submission Date: 16th September 2022
Word Count: 14495 Words

Abstract

Early detection of coronary heart disease (CHD) can help reduce death rates because it is one of the top causes of mortality globally. It is common to hear people refer to medical analysis as a good information source. The complexity of the data and linkages makes typical methodology-based prediction problematic. This project will use historical medical data to forecast CHD with machine learning (ML) technology. By using three supervised learning techniques—Logistic Regression, K-Nearest Neighbours, and Random Forest—this study seeks to identify correlations in CHD data that might increase prediction rates. Only 14 characteristics will be utilised to forecast our target variable from the Cleveland database, which has 76 properties and is part of the UCI Machine Learning Repository.

Acknowledgements

I would also like to thank Professor Jixin Ma, supervisor of the report, who has shown great appreciation for his vital role in supporting the best result of the research work. Professor Jixin Ma has made every effort to guide me towards achieving my goal, as well as his motivation to keep moving along the right track. I am also thankful for the support and guidance I have provided on all the various matters. He does not hesitate to help in the light of his contributions and the extra time given to help me achieve my goals. His passion, motivation, and confidence helped. He was always available to answer my questions and generously gave me his time and immense knowledge. I confirm that this work is my task, and I have received an information resource acknowledgement.

Table of Contents

List of Figures

List of Abbreviations

AI: Artificial Intelligence	**ML: Machine Learning**
DL: Deep Learning	**RL: Reinforcement Learning**
LR: Logistic Regression	**KNN: K-Nearest Neighbour**
RF: Random Forest	**CHD: Coronary Heart Disease**
CAD: Coronary. Artery Disease	**CVD: Cardiovascular Disease**

1

Introduction

1.1 Background

Data has emerged as the fuel for businesses and sectors in the age of technology and digitisation. Not far behind in this regard is the healthcare sector. Patient data is now almost universally maintained electronically in hospitals and other medical facilities. Included are their medical history, symptoms experienced, diagnosis, length of sickness, recurrences, and any fatalities. As a result, there is a steady growth in the medical data collected daily. Due to a lack of efficient analytical tools, techniques, and staff, this abundance of data is frequently not fully utilised, which results in missed insights and unrecognised correlations. The health system will be significantly improved if available data are used to create screening and diagnostic models (Gonsalves et al., 2019). This will lessen the burden on the medical staff, help in early identification and rapid patient care, and minimise the workload for medical staff. Additionally, depending on their medical and family histories, it can help develop a monitoring and preventative programme for people at risk for CHD.

The vast amount of knowledge in these medical databases has recently come to the attention of researchers and industry professionals in the medical area, prompting them to analyse data for conditions including cancer, dementia, Alzheimer's, autism, and tuberculosis. Cardiovascular heart diseases (CHD), which claim an estimated 17.9 million lives annually, are the leading cause of mortality worldwide, according to the World Health Organization (WHO)(Ghosh et al., 2021). In health analysis, CHD is one of the most common diagnoses within this enormous variety. Heart disease can be treated more successfully by altering one's lifestyle, taking medicine, and, in rare situations, undergoing surgery. Heart disease therapies are discovered based on the patient's physical examinations and symptoms. Additionally, a few factors exacerbate the severity of the heart disease condition. For instance, a higher body pressure rate, obesity, a high body cholesterol level, incorrect fitness training, and regular smoking are based on family history.

As an interdisciplinary field that uses sophisticated computer methods to handle healthcare data, machine learning (ML) for health informatics has evolved. Physicians can forecast CHD and make more reliable and straightforward clinical decisions with the use of algorithms that combine the examination of clinical biomarkers with multiple credible traditional risk factors. Computers that learn from experience are employed in the machine learning (ML) process to analyse datasets. Data mining is used in healthcare because healthcare databases are sometimes enormous (Gonsalves et al., 2019). The massive amounts of data are transformed into information through data mining, which is then utilised to improve predictions and judgments. In this study, the author attempts to answer the following research issues by examining previous studies on CHD prediction: Will machine learning enhance the precision of CHD prediction? and if so, which ML algorithm is the most successful at forecasting CHD? The three supervised learning approaches of Logistic Regression, K-Nearest Neighbours, and Random Forest is the only ones the author has focused on for the sake of convenience in this article. These strategies were selected due to their adaptability to various domains and learning philosophies.

1.2 Project Aim

Early detection of coronary heart disease (CHD) can help reduce death rates because it is one of the top causes of mortality globally. The complexity of the data and linkages makes typical methodology-based prediction problematic. This project will use historical medical data to forecast CHD with machine learning (ML) technology. Using three supervised learning methods—Logistic Regression, K-Nearest Neighbours, and Random Forest—this study seeks to identify correlations in CHD data that might increase prediction rates. Only 14 out of 76 will be utilised to predict our target variable from the Cleveland database from the UCI Machine Learning Repository.

1.3 Project Objectives & Deliverables

1.3.1 To investigate the attributes responsible for increasing the rate of coronary heart disease.

- Activities required to complete the task
    - Reading journals, books, and websites
    - Preparing methodologies for the project
    - Deciding papers that will be suitable for the Literature Review
- Deliverables for the task
    - Creation of table of contents
    - Completion of Chapter 1 Introduction
    - Completion of Initial Report

1.3.2 To research the theoretical background of the supervised machine learning models for the project.

- Activities required to complete the task
    - Preparing deep research on the theoretical background
    - Providing a detailed explanation of the background
- Deliverables for the task
    - Completion of Chapter 2 Literature Review
    - Completion of Interim Report

1.3.3 To explore the dataset using exploratory data analysis and showcase the results by data visualisation.

- Activities required to complete the task
    - Collecting dataset and performing visualisation
    - Researching supervised machine learning models
- Deliverables for the task
    - Completion of exploratory data analysis
    - Completion of Chapter 3 Models Selection

1.3.4 To implement the machine learning models by hyperparameter tuning and cross-validation.

- Activities required to complete the task
    - Implementing the selected classification models for prediction
    - Comparing all the tuned models to achieve at least 85% accuracy
- Deliverables for the task
    - Completion of Chapter 4 Modelling
    - Completion of selecting the best model for further evaluation

1.3.5 To evaluate the classification models beyond accuracy and see which features contribute most to the model's outcomes.

- Activities required to complete the task
    - Evaluating the classification models with various metrics
    - Comparing features contribution of all the models
    - Determining ethics, limitations, and future work of all models
- Deliverables for the task
    - Completion of Chapter 5 Evaluation
    - Completion of Chapter 6 Conclusion & Future Work

1.4 Project Nature of Challenges

The first critical factor is access to the dataset from the Cleveland Database. If anything happens in the future that the database is unavailable, it will be challenging to find a similar dataset with the required attributes. The author downloaded the dataset and kept a few similar ones with almost the same characteristics. The hardware requirement is also essential as it will help train the models more efficiently. The author's machine can run the models efficiently but can use the lab machine if something is not working. Finding suitable models for this project is vital, as not all models can have higher accuracy than 80%. Thus, the author needs to research similar work and tune the models efficiently.

1.5 Project Development Methodology

This project's development process was conducted using an agile approach that had been modified (Figure 1). The continuous release of production software and agile development cycles are only two examples of the significant ideas that shape modern software engineering. This project aims to provide continuous delivery of project iterations rather than continuous software delivery. The project's goals cannot be achieved in a single iteration; that is obvious. Constant training and progress have been made incrementally with each successive iteration, and evaluation following each version will be necessary (Gren and Ralph, 2022). The project's covert purpose is to make errors early on so that it may learn from them and proceed with more knowledge and caution in future endeavours. As a result, obtaining the best outcomes in only one try is quite rare. They would, though, offer a starting point for the next revision.

Agile project management produces products in brief, quick work cycles that permit continuous production and review. It entails segmenting the project into many parts while ensuring continuous improvements throughout each step. Therefore, a software engineering approach must be adopted and modified to support and, even better, encourage this type of development behaviour (Gren and Shepperd, 2022). Other methods were being examined in the project's early stages, notably the research phase, to see whether they were better appropriate for the task. Ultimately, it was obvious to choose the agile methodology because of the flexibility and regular testing that would become routinely available. There are many different methods used in software engineering. Specific projects, however, cannot be completed using certain approaches, or they cannot be completed as effectively as they could.

The waterfall and agile models are the two most popular project management approaches for artificial intelligence and machine learning projects. Early in the project's development, the waterfall paradigm was a possibility. The needs and planning must be established because it is a sequential model. The

Agile technique, in contrast, is an iterative approach with a clear scope and set of requirements that can nonetheless be susceptible to change as the project moves forward. When utilising a waterfall approach, testing is often done towards the project's conclusion, but an agile model incorporates testing into every phase of the project's development. The development anticipated for this project does not fit the waterfall model's characteristics (Gren and Ralph, 2022).

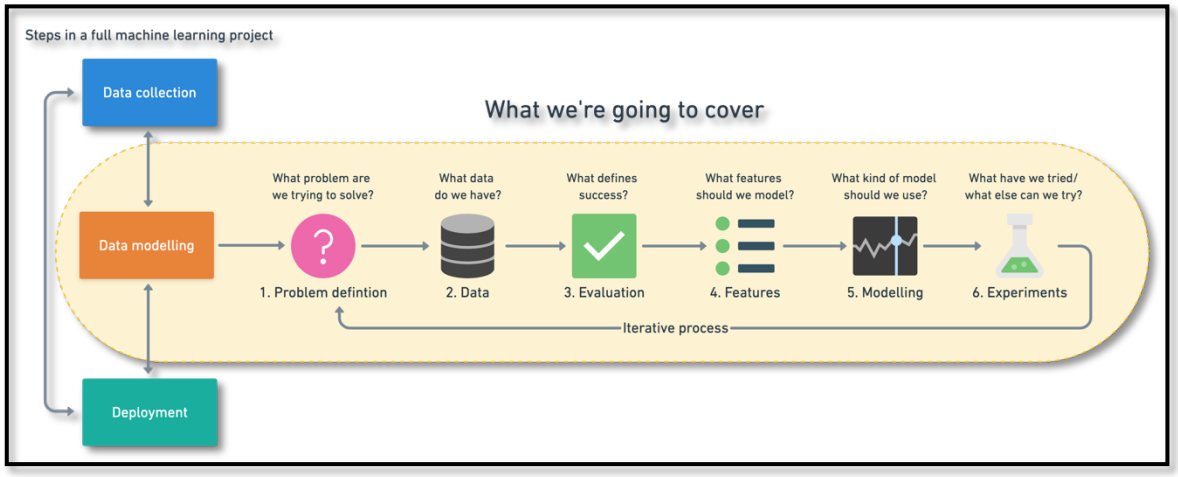


Figure 1: Adapted Agile Methodology

1.6 Project Gantt Chart

Among the several project management tools are Gantt charts. They display every task in a project in one diagram. They outline the proper sequence for completing each activity and the approximate amount of time required. Gantt charts are used by both the Waterfall and Agile project management approaches. The Agile approach allows project teams to define their objectives and revise their plans in real time based on ongoing client input (Krawczuk et al., 2021). In Agile, Gantt charts help compare an existing plan to a suggested modification and determine the amendment's impact on the entire program. A Gantt chart is made from a horizontal axis that divides the project's overall time frame into increments of days, weeks, or months. The project tasks are represented on a vertical axis. Major activities could include conducting research, selecting software, and installing software, for instance, if the project involves picking new HR software. Each job's order, timing, and duration are shown as horizontal bars of increasing lengths. Put "conduct research" at the top of the vertical axis using the same example, then on the graph, draw a bar to indicate the time expected for the research. Next, list the remaining activities below the first one, with representative bars on when they will be completed (Krawczuk et al., 2021). Bar spans could cross each other. For instance, selecting software and performing research may occur concurrently. As the project moves forward, more bars, arrowheads, or darkening bars may be added to represent tasks done in whole or part. A vertical line shows the date of the report. Please proceed to Appendix Section for Gantt Chart.

2

Literature Review

2.1 Introduction

Today for many medical practitioners, predicting heart disease is the most important responsibility, and doing so involves careful examination of the clinical and pathological data of the patients (Khateeb and Usman, 2017). Obtaining pricey medical testing for the early detection of heart disease also becomes challenging for people. Effective computerised systems for cardiac disease prediction can be beneficial in these situations. This study aims to convey in-depth knowledge of the many data mining techniques employed in developing a computerised heart disease prediction system. It also offers some recommendations for raising the standard of healthcare services generally. Researchers and medical experts are interested in developing automated systems for the low-cost prediction of cardiac illnesses to address these issues.

2.2 Coronary Heart Diseases

The arteries that provide blood and oxygen to the heart are clogged, which results in coronary heart disease (CHD) (Figure 2). Atherosclerosis-related disorders such as myocardial infarctions and angina can develop due to the build-up of atherosclerotic plaques in the walls of the coronary arteries (Song, Chen and Antoniou, 2021). It is among the most widespread cardiac conditions and has a high mortality rate globally. The likelihood of living a longer and better life can significantly increase with the prompt medical treatment of cardiac disease, which can be made possible by early identification. Due to the intricate combination of symptoms and indicators that doctors must review, diagnosing heart disease may be reasonably complicated (Song, Chen and Antoniou, 2021). An accurate diagnosis is also challenging due to the high degree of symptom overlap with those of other illnesses.

The concept of a non-invasive screening method for detecting Coronary Artery Disease (CAD) patients using the fingertip Photoplethysmogram (PPG) signal is presented in this study (Banerjee et al., 2016). To differentiate between CAD and non-CAD participants, a composite feature set that

considers both heart rate variability (HRV) and PPG waveform morphologies have been established. For categorisation, Support Vector Machine (SVM) is utilised. On a corpus of 112 cases chosen from the MIMIC II dataset, the author's technique produces sensitivity and specificity scores for diagnosing CAD patients of 0.82 and 0.88, respectively. On another dataset of 30 participants obtained from an urban hospital using a commercial oximeter device, the author also attained sensitivity and specificity values of 0.73 and 0.87 (Banerjee et al., 2016).

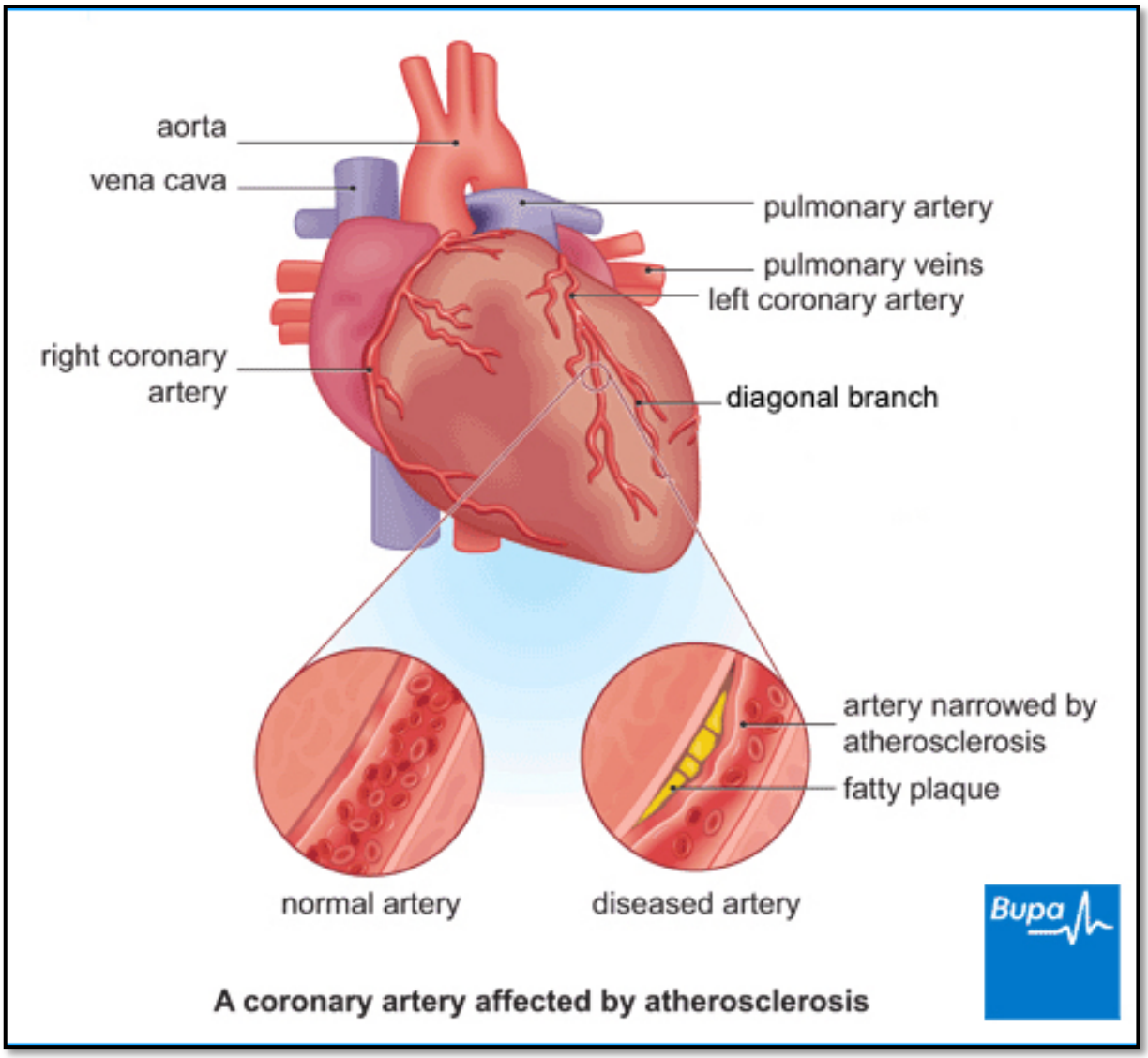


Figure 2: Coronary Artery affected by Atherosclerosis

The authors propose an automated approach for early risk categorisation of complex coronary heart disease using the Framingham scoring model in this study (Elsayed and Syed, 2017). The K-Nearest Neighbour and Random Forests algorithms were used to predict heart rate risk, and the results were compared to those produced manually to determine the level of accuracy. Compared to the human procedure, the authors' suggested automated heart rate risk prediction technique using the

Framingham model proved remarkably accurate. This paper seeks to report on the efficacy of employing K-Nearest Neighbour and Random Forests for Framingham heart and medical decision support in cardiology (Elsayed and Syed, 2017).

This study (Ara et al., 2019) aims to use machine learning and artificial intelligence algorithms to automatically detect peripheral arterial disorders suggested by the Lower Extremity Arterial Doppler (LEAD) tests. Input for the learning algorithms for illness prediction comes from the categorised waveforms. To predict normal artery function and three different types of artery diseases—aortoiliac disease, femoral-popliteal arterial disease, and trifurcation disease—the authors compare the performance of two conventional machine learning techniques and two neural networks. The hierarchical neural network model (HNN) is examined to handle an unbalanced data collection. The HNN's initial level predicts what is expected and what is ill. Other illnesses are predicted from the rest using the two remaining neural networks. High F1 scores were attained by HNN using 10-fold cross-validation. With scores for the average case being 99 per cent, the aortoiliac disease being 97 per cent, the femoral-popliteal arterial disease being 94 per cent, and the trifurcation disease being 89 per cent. According to the comparison, HNN outperforms multilayer perceptron, random forests, and SVM (Ara et al., 2019).

2.2.1 Major Attributes

According to this study (Hajar, 2017), numerous risk factors for CAD can be managed, but not all of them. Age, sex, family history, and race are all uncontrollable factors. High blood pressure, high cholesterol, smoking, diabetes, being overweight or obese, not exercising enough, eating poorly, and stress are the risk factors that may be managed. One of the dangers of CHD development is hypertension. Myocardial infarction, angina pectoris, and sudden death are only a few of CHD's clinical symptoms predisposed by hypertension. A higher risk of CVD is linked to even high regular BP readings. The significant risk in those with mild to moderate hypertension was localised in people who also had diabetes, dyslipidaemia, and left ventricular hypertrophy (Hajar, 2017). It was frequently discovered that elderly hypertensives already had target organ damage, such as reduced renal function, silent myocardial infarction, strokes, transient ischemic episodes, retinopathy, or peripheral artery disease. According to the Framingham research, one or more of these disorders affected at least 60% of older men and 50% of senior women with hypertension.

Cholesterol poses a significant additional risk for CVD. It was established that variations in the incidence rate of CVD were related to changes in cholesterol levels (Anon, 1984). These results were well-received by epidemiologists and clinicians, who concurred that total plasma cholesterol was an effective diagnostic for predicting CVD. It was discovered that its constituent, low-density lipoprotein

cholesterol (LDL-C), the primary lipoprotein carrying cholesterol in the blood, was closely related to CVD (Klag et al., 1993). LDL cholesterol levels predict the development of CVD later in life in young adulthood, and it was also discovered. It has been demonstrated that these advantages of lowering serum cholesterol for CHD risk are age-related. A 10 per cent drop in blood cholesterol results in a 50% reduction in CHD risk at age 40. A 40% reduction at age 50, a 30% reduction at age 60, and a 20% reduction at age 70 (Law, Wald and Thompson, 1994).

According to Framingham research, smoking increases a person's risk of myocardial infarction (MI) or sudden death, which is correlated with daily cigarette use (Doyle et al., 1962). Compared to a lifetime of not smoking, smoking doubles the risk of morbidity and death from ischemic heart disease, which correlates with the frequency and intensity of smoking (Campbell et al., 1998). There is proof that quitting smoking lowers the risk of nonfatal MI and all-cause death in people with CHD. After an acute MI, around 20% of patients decide to quit smoking, which results in a 40% decrease in death rates and infarct recurrences (Critchley and Capewell, 2003). Because smoking is a significant risk factor for both fatal and nonfatal recurrences of ischemic heart disease and a first MI, all individuals with the condition should be counselled to discontinue doing so (Jolly et al., 1999).

Before 1979, Kannel et al. did not recognise diabetes as a significant cardiovascular risk factor; instead, they relied on information from the Framingham heart study to determine its significance in the aetiology of CVD. An increased risk of two- to thrice developing the clinical atherosclerotic disease was noted after 20 years of monitoring the Framingham group. The study was also among the first to show that women with diabetes had a greater risk of cardiovascular disease (CVD) than men do. Much other research has confirmed these findings (Kannel, 1979). The Kannel essay transformed the way doctors viewed diabetes. As a significant cardiovascular risk factor, it is now acknowledged. Diabetes and CVD have a specific causal link.

2.2.2 Prediction of Other Diseases

A chronic condition that can impact every system in the body is diabetes mellitus (DM). The quality of a patient's health or the reduction of risk factors can be improved by an early diagnosis of diabetes. This study's (Daanouni, Cherradi and Tmiri, 2020) primary goal is to assess the effectiveness of a few machine learning algorithms applied to predicting diabetes diseases. To this end, we use and assess four machine learning algorithms (Decision Tree, K-Nearest Neighbours, Artificial Neural Network, and Deep Neural Network) to predict diabetes mellitus. The Pima Indian dataset has been used to train and evaluate these approaches. After reducing noisy data and employing feature selection using neighbourhood components, the experimental methods' performances have been assessed. The learning process is sped up, and data interpretation is improved by analysis to minimise the number of

features and lessen the complexity of dimensionality. Accuracy, Sensitivity, and Specificity are a few similarity measures used to compare model performance (Daanouni, Cherradi and Tmiri, 2020).

Alternative strategies for predicting the likelihood of severe liver fibrosis caused by chronic hepatitis C include machine-learning techniques. To stage chronic liver illnesses without the limitations of biopsy, machine learning techniques have lately been applied as non-invasive alternatives. By integrating serum biomarkers with clinical data to create classification models, this study (Hashem et al., 2018) seeks to examine several machine learning approaches to predicting advanced fibrosis. According to the METAVIR score, a prospective cohort of 39,567 chronic hepatitis C patients was split into two groups: one group was classified as having mild to moderate fibrosis (F0-F2) and the other as having advanced fibrosis (F3-F4). Multi-linear regression, decision tree, genetic algorithm, and particle swarm optimisation models were created for cutting-edge fibrosis risk prediction. The effectiveness of the suggested models was assessed using receiver operating characteristic curve analysis. Advanced fibrosis was significantly associated with age, platelet count, AST, and albumin. With an AUROC of 0.73 to 0.76 and an accuracy of 66.3 to 84.4 per cent, the machine learning algorithms under investigation successfully predicted advanced fibrosis in patients with HCC (Hashem et al., 2018).

Presently, several scientists and agricultural specialists are interested in predicting plant diseases. When employing machine learning algorithms to identify plant illnesses early on efficiently, the prediction of plant diseases serves as the cornerstone of the process (Bhatia et al., 2021). The unbalanced dataset presents a problem for this field of agriculture research, in any case. Machine learning models' findings may be skewed by unbalanced datasets in favour of the significant class with the greatest number of dataset samples. Resampling strategies that balance the dataset to increase the effectiveness of machine learning models can be used to solve this issue. Therefore, in the current study (Bhatia et al., 2021), the effects of resampling methods have been assessed on imbalanced plant disease datasets using various machine-learning classifiers, i.e., Random Forest, Nave Bayes, Multinomial Logistic Regression, and Bagged Class. According to the evaluation's findings, Random Over Sampling performed the best among all resampling techniques for the Tomato Powdery Mildew Disease dataset for Random Forest Classifier, scoring 99.24 per cent accuracy. While for the Soybean Large dataset for Bagged Classification and Regression Tree Classifier, Synthetic Minority Over-sampling Technique achieved 98.53 per cent accuracy (Bhatia et al., 2021).

Several learning strategies have been used to find genes linked to diseases. Early on, they frequently used a binary classification method to solve this issue, using training sets made up of both positive and negative data (Le and Nguyen, 2015). Positive samples are created using known illness genes, whilst negative samples are the remainder that is not known to be linked to diseases. The negative

training set should comprise non-disease genes but building such set-in biomedical research is impossible. This is the limitation of binary classification-based methods. To lessen this ambiguity, more practical classification-based strategies have been suggested. This research (Le and Nguyen, 2015)often represents data sources by kernel matrices for unary and PU learning classifiers and vectorial forms for binary classifiers. Since disease genes are predicted using vectorial representations of data, the authors of this study examined several categorisation algorithms. The simulation result demonstrated that the one-class SVM-based method performed poorly compared to the unary classification strategy, which combines class probability estimation and density estimation techniques. It is interesting to note that the performance of the best binary classification approach is on par with that of binary semi-supervised classification techniques and biased SVM-based PU learning. Additionally, they are all superior to the SVM-based multi-level one (Le and Nguyen, 2015).

2.3 Machine Learning

Artificial intelligence is a subset of machine learning. A field of computing algorithms called machine learning is constantly developing and aims to replicate human intelligence by learning from the environment. Machine learning aims to create or use algorithms to automate the learning process for machines without human intervention using contextualised data input. Machine learning enables the creation of models for prediction, pattern recognition, decision-making, and simulating other cognitive activities. A machine learning model's output largely relies on the data input.

Computational sustainability is a rapidly expanding field of technology. By proposing assessment measures for research subject ideas, (Wagstaff, 2012) furthers the argument for the need for more substantive machine learning research. Additionally, (Gomes, 2011) examines connecting machine learning to the problems that are presented in the real world concerning the economy, society, and environment. In addition to the conventional performance indicators, they emphasise that we may also track money saved, lives saved, time saved, the effort saved, quality of life improved, etc. Our measures' emphasis on effect will encourage an upstream reorganisation of research activities.

With new technologies and techniques being invented, improved, and optimised regularly, various approaches and methods may be used to tackle an issue utilising machine learning. However, such methods have significant drawbacks, including the requirement for substantial amounts of high-quality data, a lot of computing power, and essential engineering work (Holzinger, 2018). The (Wagstaff, 2012) study topic evaluation criteria inspired the project's methodology when relating this data to the project's issue area.

It gives epidemiologists new tools to address issues for which conventional approaches are inadequate, especially considering the increased emphasis on "Big Data" (Bi et al., 2019). *Big Data* is a term used to describe an enormous volume that is exponentially rising and includes too much information for a human data analyst to process. However, massive data management has improved over time, thanks to machine learning. Machine learning is used to teach computers using datasets and algorithms that enable automated decision-making and problem-solving. Considering the algorithm's expected output, machine learning algorithms are classified into taxonomies (Zhang, 2010).

2.3.1 Supervised Learning

The biomedical community, healthcare issues, and patient care may all benefit from cutting-edge technology like machine learning and big data analytics. By correctly interpreting medical data, they also aid in the early diagnosis of diseases. By creating classification models, machine learning techniques may be used to forecast chronic illnesses like renal and heart problems. This study (Krishnani et al., 2019) suggests an extended pre-processing approach to forecast coronary heart diseases (CHD). By utilising machine learning techniques, including K-Nearest Neighbours, Decision Trees, and Random Forests, this research seeks to predict the risk of coronary heart disease (CHD). Comparing these algorithms is also done based on how accurately they anticipate outcomes. Several methods are being tested on the "Framingham Heart Study" dataset, which has 4240 records. Random Forest, Decision Tree, and K-Nearest Neighbour all had accuracy rates of 96.8%, 92.7 per cent, and 92.89 per cent in our experimental research. Because of this, the Random Forest classification method outperforms other machine learning algorithms in terms of accuracy by incorporating our pre-processing processes (Krishnani et al., 2019).

Jupyter notebook, which has a variety of libraries and header files for precise and accurate work, is the most incredible tool for implementing Python programming. This study (Gupta et al., 2022) uses a dataset that was gathered from the repository of the University of California, Irvine. Predict cardiac disease using various supervised machine-learning techniques, such as K-Nearest Neighbour, Decision Tree, Logistic Regression, Nave Bayes, and Support Vector Machine (SVM) model (UCI). According to the performance measures, the findings show that Logistic Regression outperformed all other supervised classifiers. The confusion matrix of all the models indicates that the model is less dangerous because it has fewer false negatives than other models. The classifier's accuracy can also be increased by using ensemble approaches (Gupta et al., 2022).

A precise and early prediction of heart failure (HF) risks crucial since it has been established that HF is one of the primary causes of mortality. The best and most accurate method of diagnosing HF is through clinical procedures, such as angiography. However, studies reveal that this procedure is not

only expensive but also has adverse effects. Recently, the stated goal has been achieved using machine learning approaches. In 35 journal publications published since 2012, state-of-the-art machine learning classification algorithms have been applied to datasets related to heart disease. This survey (Khan et al., 2019) study seeks a thorough literature review based on those articles. This study, which conducts a critical analysis of the chosen publications and identifies gaps in the literature, will be helpful to researchers looking to use machine learning in the medical field, especially when working with datasets related to heart disease. According to the report, ensemble classifiers, neural networks, and support vector machines are the most often used classification approaches (Khan et al., 2019).

Chest pain or irritation is a symptom of ischaemic heart disease (IHD). According to the World Health Organization, Pakistan's leading cause of death is coronary heart disease. To prevent casualties, a model must be highly accurate and precise (Shehzadi et al., 2022). Multiple models with various properties were previously explored to improve the detection accuracy, but they were unsuccessful. The present stage of cardiac disease is classified in this research study using an artificial system. The authors' model predicts an accurate diagnosis of chronic disorders. Training and test datasets are used to train and evaluate the scenario. Machine learning techniques like LR, NB, and RF predict an illness's progression. With a maximum accuracy of 99% for RF, 97% for NB, and 98% for LR, the experimental results of this study have demonstrated that the authors' technique has outperformed previous procedures. This level of precision will result in a slight reduction in the annual mortality toll from ischemic heart disease (Shehzadi et al., 2022).

2.3.2 Unsupervised Learning

The capacity to use reliable patient data to customise preventive care for specific populations is part of the promise of precision population health. The automatic discovery of therapeutically beneficial subgroups that consider clinical, genetic, and environmental heterogeneity may be possible with advanced analytics. To identify clinically significant subgroups of coronary artery disease, this study (Flores et al., 2021) tested whether unsupervised machine learning techniques could understand diverse and missing clinical data. Individuals with newly diagnosed and symptomatic coronary artery disease are included in the Genetic Determinants of Peripheral Arterial Disease research, which is a prospective cohort. One hundred fifty-five phenotypic and genetic characteristics from 1329 subjects were used in generalised low-rank modelling and K-means cluster analysis. To investigate links between clusters and major adverse cardiovascular and cerebrovascular events, as well as all-cause mortality, Cox proportional hazard models were utilised. After that, we evaluated how well the American College of Cardiology Heart Association pooled cohort equations and cluster-based risk classification performed (Flores et al., 2021).

Four phenotypically and prognostically different groups were found via unsupervised analysis (Flores et al., 2021). The risk of significant adverse cardiovascular and cerebrovascular events was highest in cluster 2 (youngest/multi-ethnic), while all-cause mortality was highest in cluster 1 (oldest/most comorbid; 26%). Despite having reportedly identical risk factors and lifestyle profiles, cluster 4 (middle-aged/healthiest behaviours) suffered a more significant incidence of adverse cardiovascular and cerebrovascular events (30%) than cluster 3 (middle-aged/lowest medication adherence; 23%). Cluster membership provided more helpful information for determining the risk of myocardial infarction, stroke, and death than the pooled cohort equations. Unsupervised clustering revealed four categories of coronary artery disease with different clinical trajectories. Sharper insights into illness characterisation and risk assessment may be obtained by processing heterogeneous patient data with the help of flexible unsupervised machine learning methods (Flores et al., 2021).

Chronic illnesses including Alzheimer's, diabetes, and the chronic obstructive pulmonary disease typically take a long time to develop and advance slowly, placing a growing strain on patients, their families, and the healthcare system (Wang, Sontag and Wang, 2014). A better knowledge of their course greatly aids early diagnosis and individualised therapy. Due to the incompleteness and irregularity of the observations and the variety of the patient situations, modelling illness development based on empirical information is very difficult. We provide a probabilistic illness development model that considers these issues in this work. Our model has three advantages over current disease progression models. Using discrete-time data with non-equal intervals develops a continuous time progression model. It derives the whole progression trajectory from a collection of incomplete data that only includes sparse progression coverage. It learns a small set of medical concepts to bridge the gap between the hidden progression process and the typically minimal and noisy visible medical data. We use a real-world COPD patient cohort to apply our model and gain some intriguing clinical insights, which helps to show off its possibilities (Wang, Sontag and Wang, 2014).

3

Models Selection

3.1 Logistic Regression

As a classification algorithm, Logistic Regression is used. It forecasts a binary result based on several unrelated inputs. A result that can only have two outcomes—the event occurs (1) or does not—is referred to as a binary outcome (0) (Kutrani, Eltalhi and Ashleik, 2021). The variables or factors that have no direct relationship to the result are known as independent variables. A binary event's probability is determined using logistic regression, which is also used to address categorisation problems. Incoming email spam detection, for instance, or determining if a credit card transaction is fraudulent or not. A tumour's benignity or malignancy can be determined using logistic regression in a medical setting. It may be used in marketing to foretell whether a specific user would purchase a specific product. If a student will finish their course on time or not, a firm offering online education may utilise logistic regression to make that prediction (Kutrani, Eltalhi and Ashleik, 2021).

3.1.1 Advantages

In machine learning, logistic regression is significantly more straightforward than other approaches: A machine learning model is a mathematical representation of an actual process. A machine learning model must be trained and tested before implementation (Figure 3). For the model to transfer a particular input to some output, like a label, it must first be trained, which is the process of identifying patterns in the input data (Radovanović et al., 2018). Comparatively speaking, logistic regression is simpler to train and use than other approaches. When the dataset can be linearly separated, logistic regression is compelling: If a straight line can be used to divide two groups of data from one another, then the dataset is said to be linearly separable. When your Y variable can only take two values, logistic regression is utilised because it is more effective to divide the data into two distinct groups if it is linearly separable (Radovanović et al., 2018).

A valuable observation from logistic regression is: With the help of logistic regression, we can determine the direction of the connection and how relevant an independent variable is (i.e., the coefficient size) (positive or negative). When a change in one variable's value likewise changes the other variable's value, this is referred to as a positive connection between the two variables (Radovanović et al., 2018). For instance, you get better at a specific sport with more training hours. Although correlation does not always imply causality, it is vital to understand this! In other words, logistic regression may demonstrate a positive connection between outside temperature and sales, but this does not imply that sales are increasing due to the temperature (Radovanović et al., 2018).

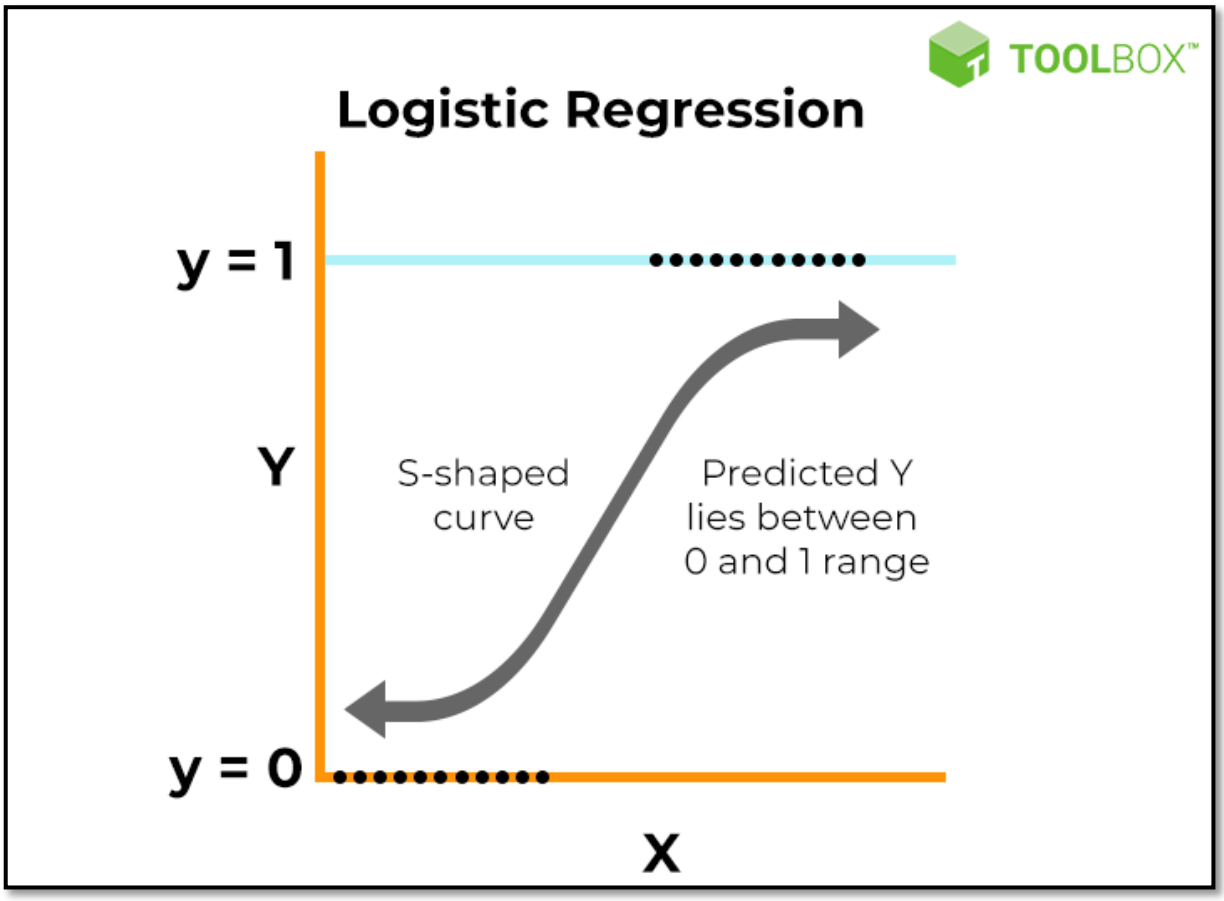


Figure 3 Logistic Regression

3.1.2 Disadvantages

A continuous result cannot be predicted using logistic regression. To comprehend this constraint more fully, let us look at an example. Logistic regression cannot be utilised in medical applications to forecast the maximum fever increase in a pneumonia patient. The continuous measuring scale is the reason behind this (Kutrani, Eltalhi and Ashleik, 2021). The predictor (independent) and predictor (dependent) variables must be linear for the logic regression to work. Why does this pose a constraint? The observations' linear separability is highly implausible in the real world. Think about dividing the

iris plant into the Sentosa or Versicolor families. Petal and sepal sizes are the determining factors in the distinction between the two groups. A petal size of 2 cm might qualify the plant for both the Sentosa and Versicolor groups, making it impossible to draw a clear distinction when describing the iris plant (Kutrani, Eltalhi and Ashleik, 2021).

Therefore, even though the logistic regression model assumes linearly separable data, this assumption is not necessarily valid. If the sample size is too tiny, logistic regression could not be reliable. A lower number of actual observations form the basis of the model created by logistic regression when the sample size is small (Kutrani, Eltalhi and Ashleik, 2021). Overfitting may occur as a result. When a model is fitted too close to a small data collection due to a lack of training data, overfitting is a statistical modelling mistake. Alternatively, to put it another way, the amount of input data is insufficient for the model to detect patterns. In this situation, the model cannot accurately forecast the results of a new or upcoming dataset (Kutrani, Eltalhi and Ashleik, 2021).

3.2 K-Nearest Neighbour

One of the fundamental machine learning algorithms, K Nearest Neighbours (KNN), belongs to the supervised learning category (Figure 4). It is a popular, straightforward, and compelling illustration of a lazy learner's non-parametric classification method. Unclassified data points are based on how closely related and different from other accessible data points; the KNN algorithm classifies them (Irbaz et al., 2020). This approach is predicated on the notion that comparable data points can be located close to one another. The nearest neighbour count is represented by K. The closest neighbour algorithm is what is used when K is equal to 1. The algorithm may predict the label of an unlabelled position X in this case by locating the nearest labelled point to X and giving that point the label. This is the most straightforward case (Irbaz et al., 2020).

The model's K value must be carefully chosen; if it is, the model may over- or underfit the data. If K is set too high, the prediction will be computationally expensive. In contrast, a low K number will result in the forecast being highly influenced by the data noise (Irbaz et al., 2020). By calculating the square root of N, where N is the total number of samples, it is customary in the industry to determine the best value of K. As it differs from issue to issue, naturally, take this with a grain of salt. Due to the ease with which it may be utilised, the ease with which classification and regression problems can be applied, and the simplicity with which the results it produces can be interpreted, it is frequently used to address issues in various sectors (Irbaz et al., 2020).

3.2.1 Advantages

Both understanding and using the K-NN technique are reasonably straightforward. K-NN technique searches the entire dataset for K nearest neighbours to categorise the new data point. In the closest neighbour, the majority class would be assigned to new data entries. As a non-parametric method, K-NN must adhere to certain presumptions to be implemented (Anton et al., 2018). K-NN does not require as many data assumptions as other parametric models, such as linear regression before it can be used. K-NN tags new data entries based on learning from past data; it does not explicitly create any models. The fact that it uses instance-based learning makes k-NN a memory-based strategy. As we add more training examples, the classifier quickly modifies. When used in real time, it enables the algorithm to react swiftly to adjustments in the input. When dealing with binary issues, most classifier algorithms are simple to construct; however, when dealing with multi-class problems, K-NN adapts naturally. One of K-NN's benefits is that it can be applied to classification and regression issues. When choosing the initial hyperparameter, K-NN may take some time, but the other parameters are aligned once it has been done. While creating a K-NN model, the K-NN method allows the user to select the distance (Anton et al., 2018).

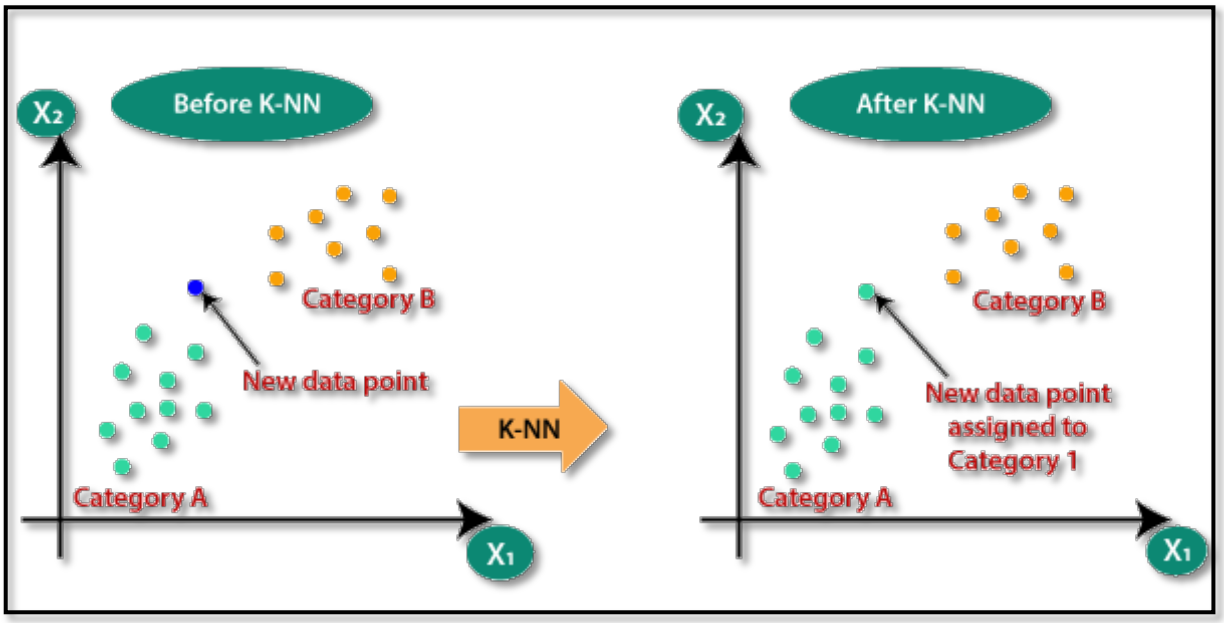


Figure 4 K-Nearest Neighbour

3.2.2 Disadvantages

Although K-NN may be reasonably simple to use, its effectiveness or speed rapidly decreases as the size of the dataset increases. KNN performs well when there are few input variables. However, as the number of variables increases, the K-NN algorithm finds it more difficult to anticipate the results of additional data points (Irbaz et al., 2020). It is very required that features have the same scale if you

choose to create k-NN using a standard distance, such as the Euclidean or Manhattan distances because absolute differences between features weigh equally. For example, a given distance in feature one must mean the same for feature two. Choosing the ideal number of neighbours to consider while categorising new data entries is one of the main challenges with K-NN. On variable data, k-NN does not work well. If we compare two classes, A and B, and most of the training data are classified as A, the model will ultimately favour A heavily. This might lead to the incorrect classification of the less prevalent class B (Irbaz et al., 2020). The K-NN method is susceptible to outliers because it only uses the distance criterion to choose the neighbours, and the missing value problem is fundamentally unsolvable by K-NN.

3.3 Random Forest

The random forest classifier is a supervised learning approach for classification and regression issues. Given its great degree of versatility and simplicity of use, it is one of the most widely used machine learning algorithms (Figure 5). These algorithms create decision trees based on a random selection of data samples, and each tree yields a prediction (Joshi et al., 2018). As a forest comprises numerous trees, so do several decision trees. Furthermore, it employs randomisation to improve its precision and prevent overfitting, which may be a significant problem for such a complex algorithm. Feature selectors, recommender systems, and image classifiers are just a few of its everyday uses. Detecting fraud, classifying loan applications, and predicting sickness are a few of its real-world uses. The Boruta method, which chooses essential characteristics in a dataset, is built on this principle. A vote then chooses a possible proposal (Joshi et al., 2018).

The random forest will randomly choose "k" features from your dataset's "m" features, supposing that k<m. The method will now choose a node with the maximum information gain among the k characteristics and calculate the root node from there (Joshi et al., 2018). The method then divides the node into child nodes and repeats this step "n" times. There are n trees in a forest, and the findings of all the decision trees in the forest will be combined. Given that it relies on decision trees' features, it is unquestionably one of the most complex algorithms. It is an ensemble algorithm in the technical sense. The algorithm uses an indicator of the attribute selection to produce the individual decision trees. A separate random sample is used for each tree. Every tree cast’s a vote in a classification challenge, and the solution comes down to the most popular class (Joshi et al., 2018). The average of all the tree outputs will be calculated in a regression issue, on the other hand, and that will be the outcome.

3.3.1 Advantages

The adaptability of random forests is one of its main features. It can do both regression and classification tasks, making it simple to see the relative weights that it gives the input characteristics (Alsouda, Pllana and Kurti, 2019). Because the default hyperparameters used by Random Forest frequently yield good prediction results, it is also a beneficial approach. Since not many of them, understanding the hyperparameters is relatively simple. Overfitting is one of the most significant issues with machine learning. The classifier will not overfit the model if the forest has enough trees. However, most of the time, the random forest classifier will prevent this (Alsouda, Pllana and Kurti, 2019).

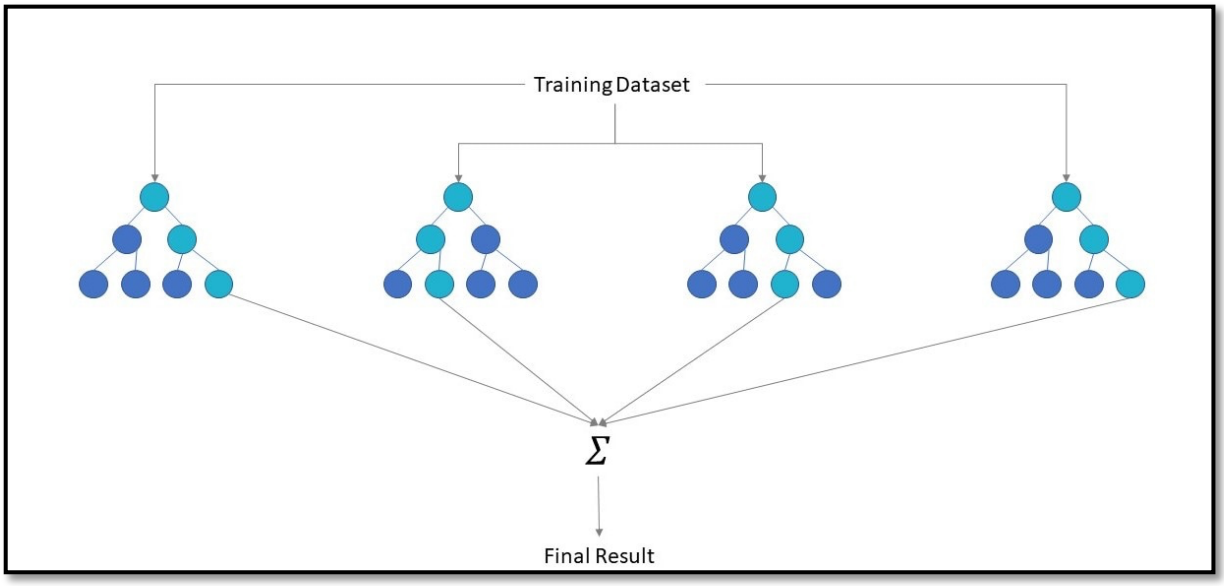


Figure 5 Random Forest

3.3.2 Disadvantages

The biggest drawback of the random forest approach is that it may become too sluggish and inefficient to generate predictions in real time when there are many trees in the model. These algorithms are quick to train but take a while to make predictions after training (Joshi et al., 2018). A slower model is produced due to the need for more trees for a more precise forecast. Although the random forest technique is adequate for most real-world applications, there may be some circumstances in which run-time efficiency is crucial and alternative methods are preferable. Random forest is a predictive modelling tool rather than a descriptive one also means that alternative strategies would be preferable if you wanted to describe the relationships in your data (Joshi et al., 2018).

4

Modelling

4.1 Programming Language & Libraries

We will utilise Scikit-Learn for machine learning and modelling tasks as we move through the following subjects, along with pandas, matplotlib, and numpy for data analysis. Exploratory Data Analysis (EDA) is the procedure of looking at a dataset and learning more about it. Model training is developing a model to learn how to forecast a target variable based on other factors. Model assessment is the process of analysing a model's forecasts using measures unique to the situation.

The process of comparing various models to determine which is the best. When a decent model is identified, how can it be improved? This is known as model fine-tuning. Importance of Feature – Are certain features more crucial for prediction, given that we are trying to forecast the presence of heart disease? Cross-validation: Can we be sure that it will function on data we have not seen before if we create a decent model? Reporting: What would we display to a potential employer if we had to exhibit our work?

Given their clinical data, can we estimate a patient's likelihood of having heart disease? In this instance, binary classification is the issue we will be looking at (a sample can only be one of two things). This is because, to determine whether a person has heart disease or not, we will consider several various qualities (i.e., bits of information) about them.

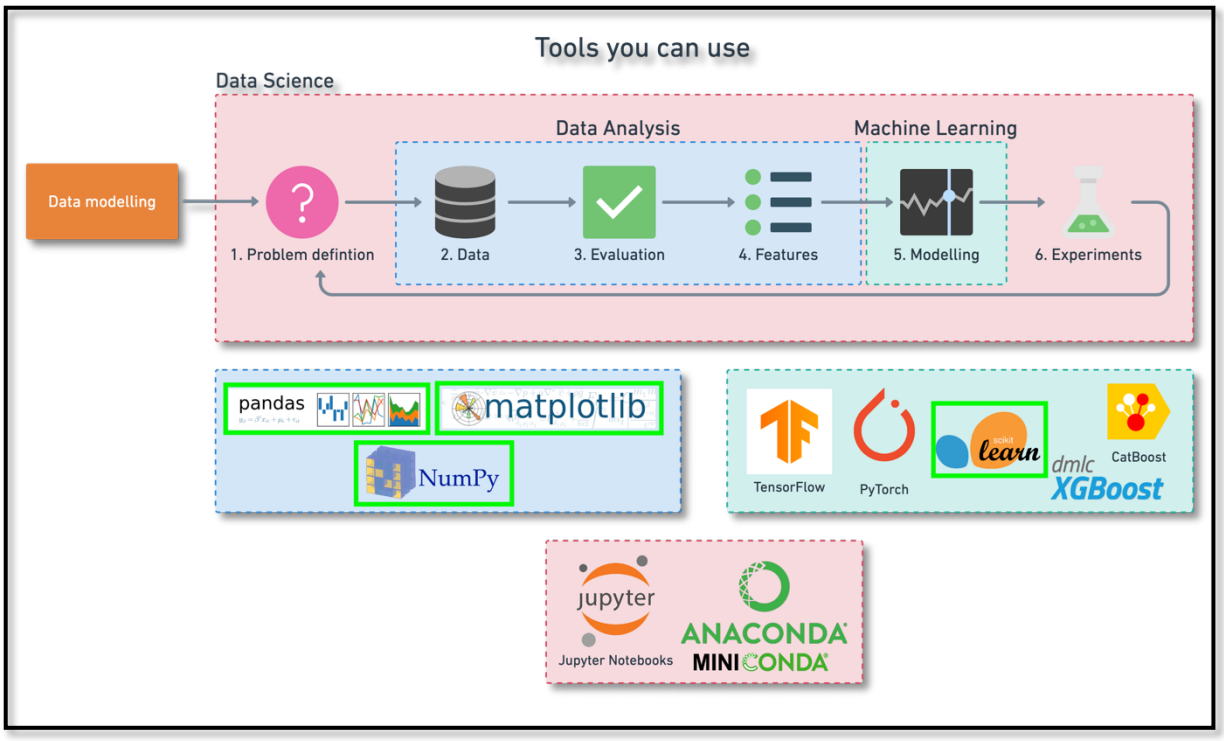


Figure 6 Programming Language and Libraries

Python - Python is a popular programming language for computers that are used to create software and websites, automate processes, and analyse data. One of the most popular programming languages today is a result of its versatility and beginner-friendliness. Python is a general-purpose language that may be used to make many applications and is not tailored for any issues (Kohn et al., 2020). Data scientists and other experts may use Python to do intricate statistical computations, design machine learning algorithms, generate data visualisations, manage, and analyse data, and perform other operations involving data. Python has established itself as a standard in the field of data science. Line and bar graphs, pie charts, histograms, and 3D plots are just a few of the many types of data visualisations that Python can create. Several libraries, like TensorFlow and Keras, are available in Python, which helps programmers create data analysis and machine learning applications more rapidly and effectively (Kohn et al., 2020).

Scikit-Learn - A crucial Python package that is frequently used in machine learning applications is called Scikit-learn. The machine learning tools that Scikit-learn is focused on include general-purpose, mathematical, statistical, and statistical algorithms (Zhang and Urbanowicz, 2020). These algorithms serve as the foundation for many machine learning technologies. Developing various algorithms for machine learning and associated technologies dramatically benefits from using Scikit-learn, a free tool. Classification, regression, and clustering algorithms are some of the significant Scikit-learn components that are helpful for machine learning. Scikit-learn, for instance, allows work

on random forests, in which specific digital trees store node information integrated with various tree topologies to produce a forest approach. Another way to put it is that each tree has clustered nodes in a tree topology. Analysis from different trees is combined to generate a global strategy that more precisely processes data to provide results (Zhang and Urbanowicz, 2020).

Any project will often begin with many necessary libraries imported, as seen below. As we work on our projects, though, libraries may be imported. We will probably want to perform some cleaning up once we have worked on our problem for a couple of hours. At the top of our notebook, we might wish to group all our utilised libraries.

```
# Regular EDA and plotting libraries
import numpy as np # np is short for numpy
import pandas as pd # pandas is so commonly used, it's shortened to pd
import matplotlib.pyplot as plt
import seaborn as sns # seaborn gets shortened to sns

# We want our plots to appear in the notebook
%matplotlib inline

## Models
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier

## Model evaluators
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import plot_roc_curve
```

Figure 7 Importing Libraries and Models

4.2 Obtaining Dataset

The facts that our problem description is built on will be what we will want to examine in this situation. This may entail sourcing, establishing various parameters, talking to specialists about it, and learning what to anticipate. The original information was sourced from the UCI Machine Learning Repository's Cleveland database. Our download of it from Kaggle was formatted, in any case. Seventy-six properties are included in the initial database. However, only 14 are utilised in this case. The elements we will employ to forecast our target variable are known as attributes (sometimes referred to as features).

The terms "independent variables" and "dependent variables" describe attributes, characteristics, and target variables. To predict our dependent variable, we employ the independent variables. Alternatively, in this instance, the patient's various medical characteristics would serve as the independent variables, and the presence or absence of heart disease would serve as the dependent variable. Your dealings with data are described in a data dictionary. Since not every dataset contains them, we might need to conduct more research or consult a subject matter expert (an individual with knowledge of the data) in this situation. The traits we will employ to forecast our target variable are as follows: (heart disease or no heart disease).

The following are the features we'll use to predict our target variable (heart disease or no heart disease).

1. age - age in years
2. sex - (1 = male; 0 = female)
3. cp - chest pain type
    - 0: Typical angina: chest pain related decrease blood supply to the heart
    - 1: Atypical angina: chest pain not related to heart
    - 2: Non-anginal pain: typically esophageal spasms (non heart related)
    - 3: Asymptomatic: chest pain not showing signs of disease
4. trestbps - resting blood pressure (in mm Hg on admission to the hospital)
    - anything above 130-140 is typically cause for concern
5. chol - serum cholestoral in mg/dl
    - serum = LDL + HDL + .2 * triglycerides
    - above 200 is cause for concern
6. fbs - (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
    - '>126' mg/dL signals diabetes
7. restecg - resting electrocardiographic results
    - 0: Nothing to note
    - 1: ST-T Wave abnormality
        - can range from mild symptoms to severe problems
        - signals non-normal heart beat
    - 2: Possible or definite left ventricular hypertrophy
        - Enlarged heart's main pumping chamber
8. thalach - maximum heart rate achieved
9. exang - exercise induced angina (1 = yes; 0 = no)
10. oldpeak - ST depression induced by exercise relative to rest
    - looks at stress of heart during excercise
    - unhealthy heart will stress more
11. slope - the slope of the peak exercise ST segment
    - 0: Upsloping: better heart rate with excercise (uncommon)
    - 1: Flatsloping: minimal change (typical healthy heart)
    - 2: Downslopins: signs of unhealthy heart
12. ca - number of major vessels (0-3) colored by flourosopy
    - colored vessel means the doctor can see the blood passing through
    - the more blood movement the better (no clots)
13. thal - thalium stress result
    - 1,3: normal
    - 6: fixed defect: used to be defect but ok now
    - 7: reversable defect: no proper blood movement when excercising
14. target - have disease or not (1=yes, 0=no) (= the predicted attribute)

Figure 8 Data Dictionary

4.3 Exploratory Data Analysis

The following step is to investigate after importing a dataset. There is no one right method to go about this. However, we ought to strive to familiarise ourselves with the dataset regularly. Compare the target variable and several columns against one another. Recall the definitions of the various columns by consulting our data dictionary. On the dataset we are using, our goal is to become authorities. As a result, if someone has a question about it, we can answer it, and when we start creating models, we can sound-check them to make sure they are not doing too well (overfitting) or to figure out why they might be performing poorly (underfitting). The following is a quick checklist you might wish to go through because there is not an exact established technique for EDA. Which question or questions are we attempting to answer or disprove? How do we handle various data kinds, and what sort of data do we have? What information from the data is missing, and how should we take it? What are the outliers, and why should we be concerned with them? To make the most of our data, how can we add, modify, or delete features?

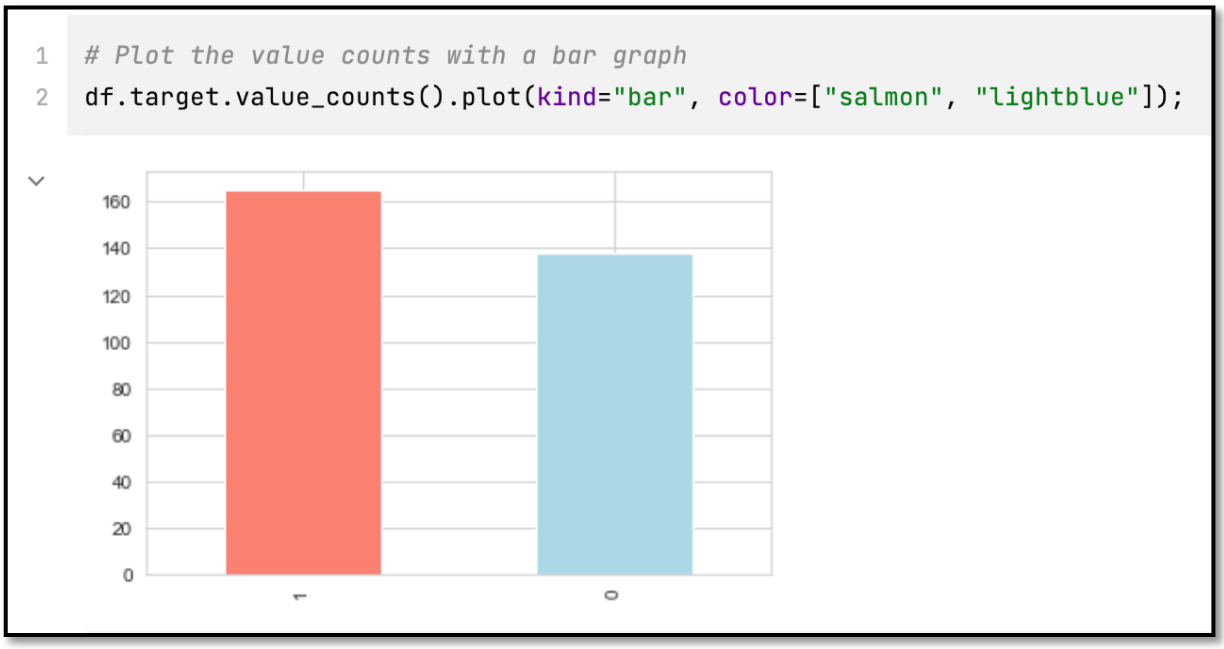


Figure 9 Male to Female Ratio

Our objective column may be balanced because these two numbers are almost equal. It can be more challenging to model a set with an uneven target column, where certain classes contain much more data. The ideal sample size for each of our target classes is one. Value counts() have an option called normalise that may be set to true if we want these numbers to be expressed as percentages. By invoking the plot() method and specifying the desired kind of plot, we may plot the target column value counts; in this instance, the bar is a reasonable choice.df.info() shows a quick insight into the

number of missing values we have and what type of data we are working with. In our case, there are no missing values, and all our columns are numerical in nature.

```
1  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
```

Figure 10 Null Figures on the Dataset

With the help of the method pd.crosstab(column 1, column 2), we may compare two columns. This is beneficial if we understand how our independent variables interact with our dependent variables. Compared to the sex column, let us look at our goal column. In case you forgot, according to our data dictionary, the target column's values are 1 for present heart illness and 0 for no heart disease. In terms of sex, 1 indicates a man and 0 a woman. From this, what can we conclude? Make a straightforward heuristic now.

Given that there are around 100 women in the study and that heart disease is present in 72 of them, we can deduce from this one variable that there is a 75% likelihood that a participant who is a woman will have heart disease. About half of the 200 guys have heart disease, which is the majority. The likelihood that a participant will develop heart disease is therefore predicted to be 50% if he is male.

According to the average of these two numbers, there is a 62.5% likelihood that the individual in question has heart disease, regardless of any other factors. We will aim to outperform this with machine learning, using this as our very basic foundation.

```
# Create a plot
pd.crosstab(df.target, df.sex).plot(kind="bar", figsize=(10,6), color=["salmon", "lightblue"])

# Add some attributes to it
plt.title("Heart Disease Frequency for Sex")
plt.xlabel("0 = No Disease, 1 = Disease")
plt.ylabel("Amount")
plt.legend(["Female", "Male"])
plt.xticks(rotation=0); # keep the labels on the x-axis vertical
```

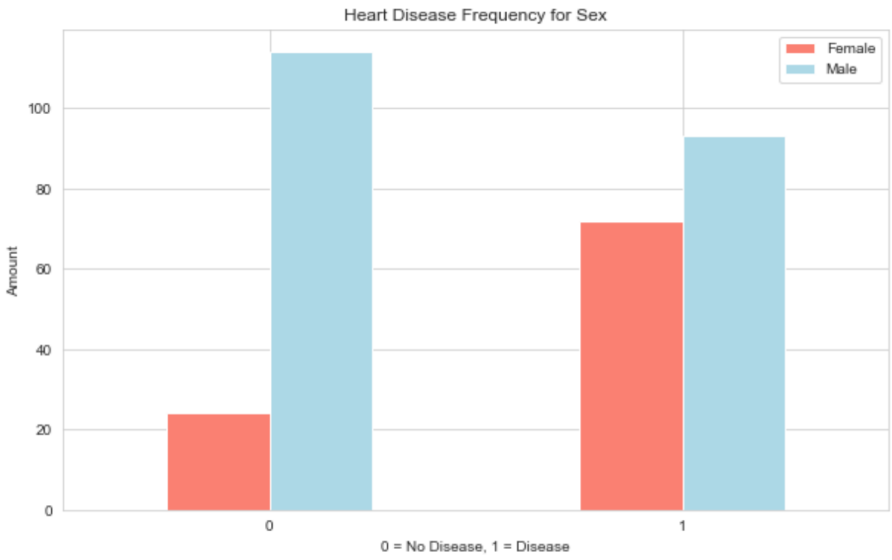


Figure 11 Heart Disease Frequency according to Gender

Let us combine a few independent factors, such as age and thalach (maximum heart rate), and compare the results to our goal factor, heart disease. We will use a scatter plot because the values for age and thalach are so varied. From this, what can we conclude? Generally, a person's maximum heart rate increases with age (older people have greener dots, which are higher on the graph's left side). However, this might be the result of the right side of the graph having a more significant number of total dots (older participants). Although both studies are observational, we aim to develop a comprehension of the data.

```python
# Create another figure
plt.figure(figsize=(10,6))

# Start with positve examples
plt.scatter(df.age[df.target==1],
            df.thalach[df.target==1],
            c="salmon") # define it as a scatter figure

# Now for negative examples, we want them on the same plot, so we call plt again
plt.scatter(df.age[df.target==0],
            df.thalach[df.target==0],
            c="lightblue") # axis always come as (x, y)

# Add some helpful info
plt.title("Heart Disease in function of Age and Max Heart Rate")
plt.xlabel("Age")
plt.legend(["Disease", "No Disease"])
plt.ylabel("Max Heart Rate");
```

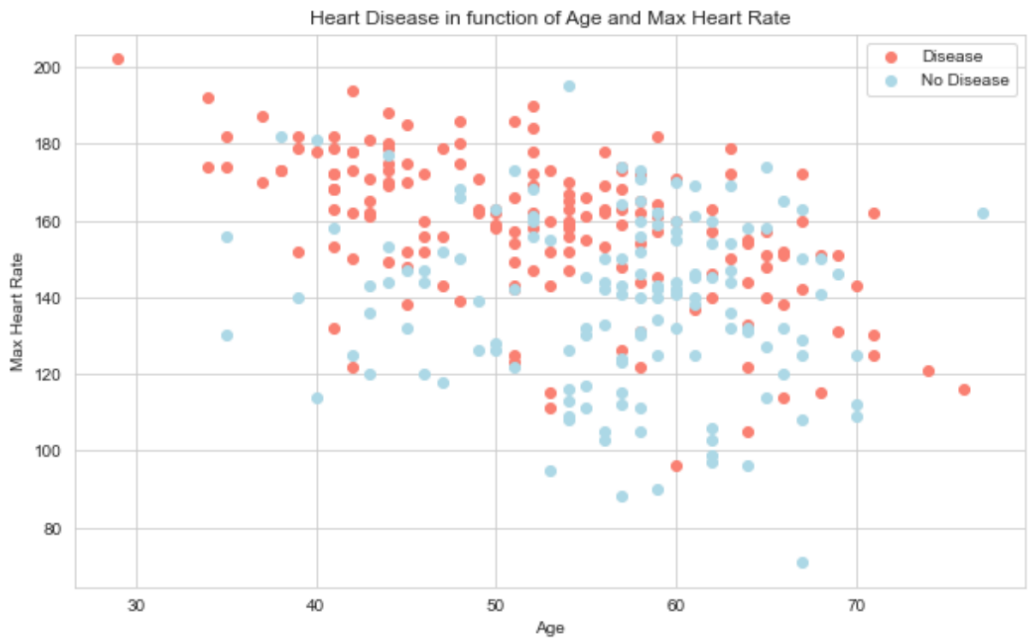


Figure 12 Age vs Max Heart rate for Heart Disease

From this, what can we conclude? Though atypical angina (value 1) claims it is unrelated to the heart, there seems to be a more significant proportion of people with heart illness than without. Now is a crucial time to keep in mind that you may want to conduct more study on your values if your data dictionary is not giving you enough details. Asking a specialist or doing a Google search to learn more might be part of this study. However, its significance is still ambiguous. The phrase appears in the titles of a few publications, although the articles themselves do not define or mention the term. In

other articles, the word is used to describe causes of chest discomfort that are not cardiac. The graph above provides a glimpse at the definitional ambiguity evident in the data; however, it is inconclusive.

```
# Create a new crosstab and base plot
pd.crosstab(df.cp, df.target).plot(kind="bar",
                                   figsize=(10,6),
                                   color=["lightblue", "salmon"])

# Add attributes to the plot to make it more readable
plt.title("Heart Disease Frequency Per Chest Pain Type")
plt.xlabel("Chest Pain Type")
plt.ylabel("Frequency")
plt.legend(["No Disease", "Disease"])
plt.xticks(rotation = 0);
```

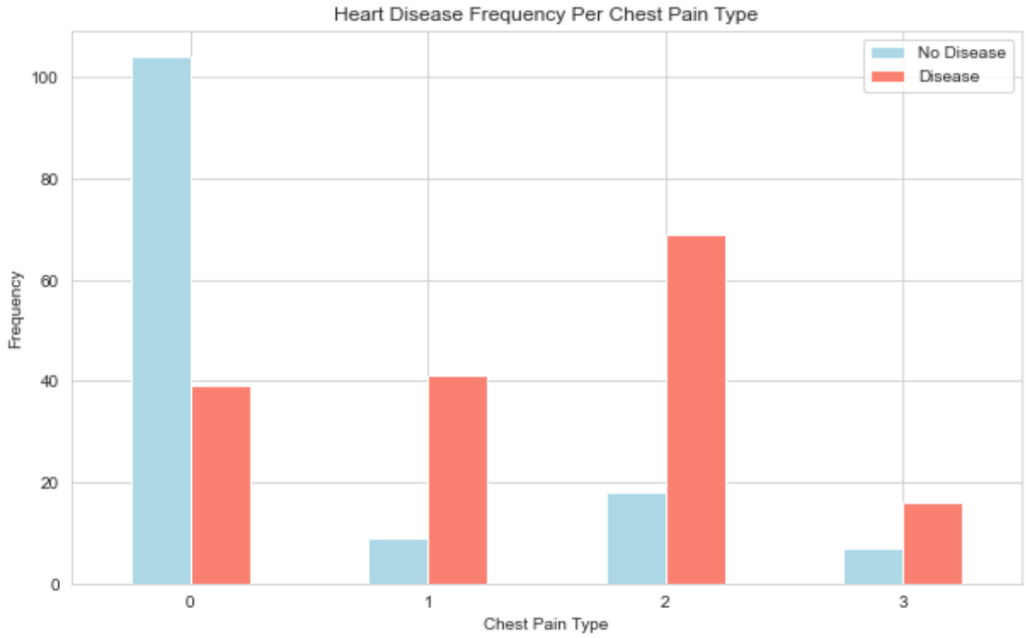


Figure 13 Heart Disease Frequency per Chest Pain Type

4.4 Training & Splitting

Here we introduce the training/test split, one of the fundamental ideas in machine learning. Our data will now be divided into training and test sets. Our model is tested using our test set after being trained on our training set. The test set and training set must be kept apart. Why not build a model from all the data? Consider a scenario in which we wished to deploy our model on patients while visiting a hospital. How would we be able to tell how well our model performs on a new patient who was not a part of the initial full dataset we had? The test set enters the picture here. As far as practical,

it is employed to simulate bringing our model into a realistic setting. Because of this, it is essential that our model never learns from the test set and that it only be judged using it. We may use Scikit-Learns train test split() and pass our independent and dependent variables to divide our data into a training and test set (X & y).

To specify how much of our data we want in the test set, we utilise the test size option in the train test split() method. Examine the training data we have. 80% of your data should be used for training and 20% for testing, according to a general guideline. A train and test sets will be enough for our problem. However, if the problem was with another thing, you could also use cross-validation (which we will see in a second) or a validation (train/validation/test) set. Each issue will be unique, though, as I said. For additional information, check out Rachel Thomas' blog article titled How (and why) to design a solid validation set.

```
# Random seed for reproducibility
np.random.seed(42)

# Split into train & test set
X_train, X_test, y_train, y_test = train_test_split(X, # independent variables
                                                    y, # dependent variable
                                                    test_size = 0.2) # percentage of data to use for test set
```

Figure 14 Train & Test Split Data

4.5 Building Models

Having examined the data, we will now attempt to forecast our target variable using machine learning and the 13 independent factors. Do you still recall our assessment metric? Remember the issue we had? Can we determine a patient's likelihood of having heart disease based on their clinical characteristics? We will be attempting to address it. We will proceed with this research if we can determine whether a patient has a cardiac disease with an accuracy of 85% during the proof of concept. That is the objective that we will pursue. However, our dataset must be prepared before developing a model. We can now begin fitting models with the data we have prepared. The following will be used, and their outcomes will be compared.

1. Logistic Regression - LogisticRegression()
2. K-Nearest Neighbors - KNeighboursClassifier()
3. RandomForest - RandomForestClassifier()

We are working on a classification issue, and these are the methods that the Scikit-Learn algorithm cheat sheet proposes using (plus a few more). At this time, it is not necessary to be thoroughly familiar with these algorithms. Why not use LinearSVC instead, as I do not see Logistic Regression? Because Logistic Regression is a categorisation model according to the Scikit-Learn documentation, I was also perplexed when I noticed it was not included in the list. Let us say that after trying LinearSVC, we find it to be ineffective and go on to the other possibilities on the map.

Iterative processes are used in data science and machine learning. You have a toolkit full of these algorithms. Understanding our challenges (such as classification versus regression) and being aware of the tools at our disposal to address them is crucial at the beginning of our path to becoming practitioners. We may test several algorithms to see which performs best since our dataset is not that large. The identical model.fit(X train, y train) and model.score() methods are used by all algorithms in the Scikit-Learn module. (X test, Y test) score Score() displays the percentage of accurate predictions (1.0 = 100% accurate). Let us put the algorithms in a dictionary and develop a fit that scores them since the selected algorithms use the same techniques for assessing them and fitting them to the data.

```python
# Put models in a dictionary
models = {"KNN": KNeighborsClassifier(),
          "Logistic Regression": LogisticRegression(),
          "Random Forest": RandomForestClassifier()}

# Create function to fit and score models
def fit_and_score(models, X_train, X_test, y_train, y_test):
    """
    Fits and evaluates given machine learning models.
    models : a dict of different Scikit-Learn machine learning models
    X_train : training data
    X_test : testing data
    y_train : labels assosciated with training data
    y_test : labels assosciated with test data
    """
    # Random seed for reproducible results
    np.random.seed(42)
    # Make a list to keep model scores
    model_scores = {}
    # Loop through models
    for name, model in models.items():
        # Fit the model to the data
        model.fit(X_train, y_train)
        # Evaluate the model and append its score to model_scores
        model_scores[name] = model.score(X_test, y_test)
    return model_scores
```

```python
model_scores = fit_and_score(models=models,
                             X_train=X_train,
                             X_test=X_test,
                             y_train=y_train,
                             y_test=y_test)
model_scores
```

Figure 15 Implementing Models in a Dictionary

4.6 Model Comparison

We can visualise our model scores since we have stored them in a dictionary by first converting them to a DataFrame.

```
{'KNN': 0.6885245901639344,
 'Logistic Regression': 0.8852459016393442,
 'Random Forest': 0.8360655737704918}
```

Figure 16 Model Comparison Part 1

```
1  model_compare = pd.DataFrame(model_scores, index=['accuracy'])
2  model_compare.T.plot.bar();
```

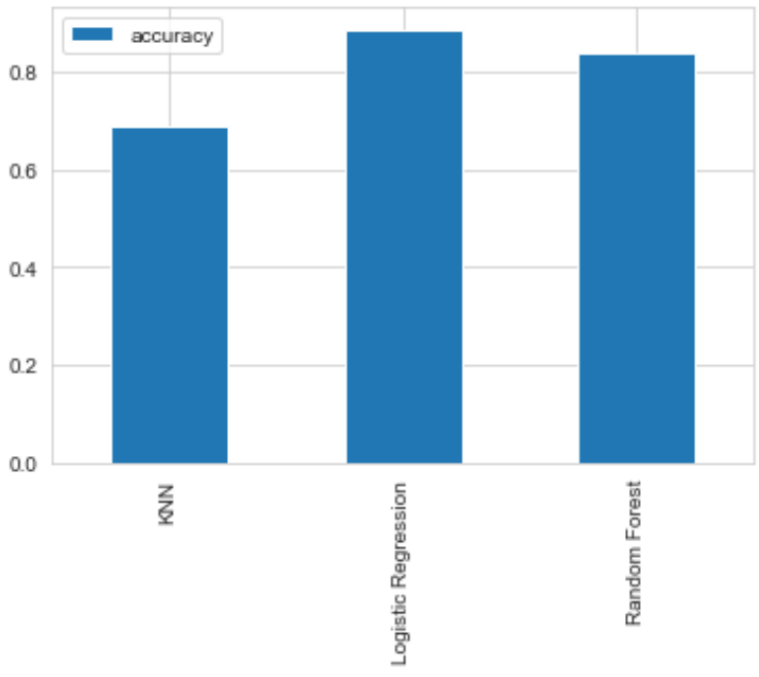


Figure 17 Model Comparison 2

4.7 Hyperparameter Tuning & Cross-Validation

We know the proper oven temperature and grill setting for cooking our favourite cuisine. However, when our roommate prepared their preferred food, they set the oven to 200 degrees on the fan-forced setting—different settings and resulted in the same oven. Machine learning algorithms may be used in the same way. By altering the settings (hyperparameters), we may employ the same algorithms and obtain various outcomes. However, misusing machine learning algorithms might result in food being burned, as if the oven were set too hot. It fits the data too well (overfits) when the parameters are changed since it works so well.

The Goldilocks model is what we are seeking. A model that performs well on our dataset and cases that have not been seen before. Although a validation set might be used to test various hyperparameters, we'll utilise cross-validation instead because there isn't much data. K-fold cross-validation is by far the most used form. After dividing it into component parts, we test a model on each k-fold of our data. Take the case of five folds (k = five) as an illustration. It may appear like this—5-fold cross-validation instead of the usual train-test split. We will analyse several models after tuning their hyperparameters using this configuration. In addition, we will simultaneously receive a few more metrics, including ROC, F1-score, recall, and accuracy.

For the K-Nearest Neighbors (KNN) method, the number of neighbours is the primary hyperparameter that may be adjusted. N neighbours are set to 5 by default. Neighbours are what? Think of a graph containing all our various samples, similar to the scatter graph shown above. KNN operates under the presumption that dots with more excellent spatial proximity belong to the same class. If n neighbors=5, a dot is considered in the same class if the five nearest surrounding dots are also in that class. We should look up the information we have omitted, such as what constitutes being close or how distance is determined.

Let us test a couple of other numbers for n neighbours for the time being. The graph suggests that the optimal value for n neighbours is 11. Despite this, the RandomForestClassifier or LogisticRegression outperformed the KNN model in terms of performance. As a result, we will ignore KNN and concentrate on the other two. We manually tweaked KNN, but now let us check how LogisticsRegression and RandomForestClassifier perform when we use RandomizedSearchCV. RandomizedSearchCV explores several possible combinations, assesses them, and stores the best ones rather than requiring us to experiment with various hyperparameters manually.

```python
# Create a list of train scores
train_scores = []

# Create a list of test scores
test_scores = []

# Create a list of different values for n_neighbors
neighbors = range(1, 21) # 1 to 20

# Setup algorithm
knn = KNeighborsClassifier()

# Loop through different neighbors values
for i in neighbors:
    knn.set_params(n_neighbors = i) # set neighbors value

    # Fit the algorithm
    knn.fit(X_train, y_train)

    # Update the training scores
    train_scores.append(knn.score(X_train, y_train))

    # Update the test scores
    test_scores.append(knn.score(X_test, y_test))
```

Figure 18 Tuning K-Nearest Neighbour by Hand

```
plt.plot(neighbors, train_scores, label="Train score")
plt.plot(neighbors, test_scores, label="Test score")
plt.xticks(np.arange(1, 21, 1))
plt.xlabel("Number of neighbors")
plt.ylabel("Model score")
plt.legend()

print(f"Maximum KNN score on the test data: {max(test_scores)*100:.2f}%")
```

```
Maximum KNN score on the test data: 75.41%
```

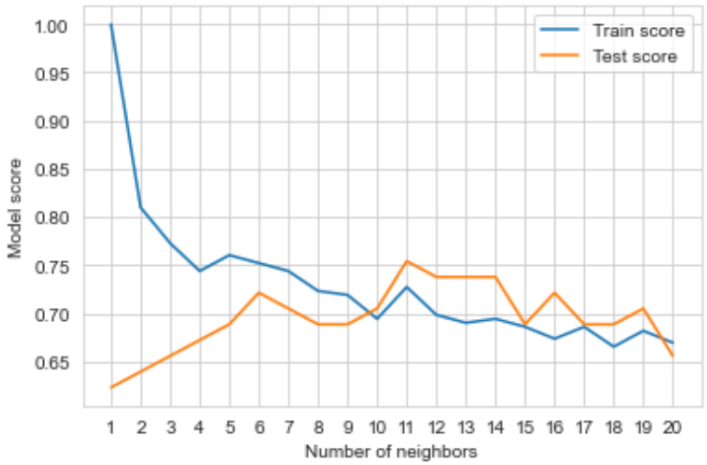


Figure 19 K-NN Accuracy Score after Hand Tuning

4.7.1 Tuning Models with RandomisedSearchCV

We learn that we may modify many different hyperparameters by reading the Scikit-Learn documentation for LogisticRegression. For RandomForestClassifier, the same holds. Let us build a hyperparameter grid (a list of several hyperparameters) for each and then put them to the test. Let us try tuning our LogisticRegression model using RandomizedSearchCV now. We will set n iter to 20 and provide the various log reg grid hyperparameters. As a result, 20 possible combinations of the hyperparameters from the log reg grid will be tested using RandomizedSearchCV, and the best ones will be saved. Excellent! In both the RandomForestClassifier and LogisticRegression, tuning the hyperparameters for each model resulted in a minor performance improvement. Similar to fine-tuning our oven's settings to prepare our preferred food to perfection, this. We will attempt fine-tuning it further with GridSearchCV, but since LogisticRegression is edging ahead, we will use it instead.

```
# Different LogisticRegression hyperparameters
log_reg_grid = {"C": np.logspace(-4, 4, 20),
                "solver": ["liblinear"]}

# Different RandomForestClassifier hyperparameters
rf_grid = {"n_estimators": np.arange(10, 1000, 50),
           "max_depth": [None, 3, 5, 10],
           "min_samples_split": np.arange(2, 20, 2),
           "min_samples_leaf": np.arange(1, 20, 2)}
```

Figure 20 Logistic Regression and Random Forest Hyperparameters

```
# Setup random seed
np.random.seed(42)

# Setup random hyperparameter search for LogisticRegression
rs_log_reg = RandomizedSearchCV(LogisticRegression(),
                                param_distributions=log_reg_grid,
                                cv=5,
                                n_iter=20,
                                verbose=True)

# Fit random hyperparameter search model
rs_log_reg.fit(X_train, y_train);
```

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
```

Figure 21 Logistic Regression RandomSearchCV Hyperparameters

```
rs_log_reg.score(X_test, y_test)
```

```
0.8852459016393442
```

Figure 22 Logistic Regression RandomSearchCV Accuracy

```
# Setup random seed
np.random.seed(42)

# Setup random hyperparameter search for RandomForestClassifier
rs_rf = RandomizedSearchCV(RandomForestClassifier(),
                           param_distributions=rf_grid,
                           cv=5,
                           n_iter=20,
                           verbose=True)

# Fit random hyperparameter search model
rs_rf.fit(X_train, y_train);
```

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
```

Figure 23 Random Forest RandomSearchCV Hyperparameters

```
# Evaluate the randomized search random forest model
rs_rf.score(X_test, y_test)
```

```
0.8688524590163934
```

Figure 24 Random Forest RandomSearchCV Accuracy

4.7.2 Tuning Models with GridSearchCV

GridSearchCV will try every conceivable combination, in contrast to RandomizedSearchCV, which performs n iter combinations across a grid of hyperparameters. Given that our grid only supports a maximum of 20 possible hyperparameter combinations, we obtain the same outcomes as before in this situation. GridSearchCV may take a while to run if our grid has many different hyperparameter combinations. Therefore, it is a good idea to begin with, RandomizedSearchCV, attempt a specific number of possibilities, and then use GridSearchCV to hone them.

```
# Different LogisticRegression hyperparameters
log_reg_grid = {"C": np.logspace(-4, 4, 20),
                "solver": ["liblinear"]}

# Setup grid hyperparameter search for LogisticRegression
gs_log_reg = GridSearchCV(LogisticRegression(),
                          param_grid=log_reg_grid,
                          cv=5,
                          verbose=True)

# Fit grid hyperparameter search model
gs_log_reg.fit(X_train, y_train);
```

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
```

```
# Check the best parameters
gs_log_reg.best_params_
```

```
{'C': 0.23357214690901212, 'solver': 'liblinear'}
```

```
# Evaluate the model
gs_log_reg.score(X_test, y_test)
```

```
0.8852459016393442
```

Figure 25 Logistic Regression GridSearchCV Accuracy

5

Evaluation

5.1 Models Evaluation

Okay, a few phrases would have come out as made-up to someone who is not an aspiring data scientist like us. Before we see each one in use, let us quickly review it. We will need to create predictions based on the test set using our model to access them. Using a trained model and the data we want to forecast, we can create predictions by executing predict() on the model. Based on test results, we will make forecasts.

We may adjust the number of knobs on each model we employ to control how it behaves through hyperparameter tuning. Model performance may improve or worsen if these settings are changed. Importance of features - If we are utilising a variety of characteristics to create predictions, are any more important than others? Which factor—sex or age—is more significant for predicting heart disease? Comparison of expected and actual values using a confusion matrix in a tabular format. All matrix values will be top left to bottom right if 100% of the assumptions are valid (diagonal line). The process of cross-validation divides our dataset into several pieces, trains and tests our model on each piece, and then averages the results to determine performance.

Precision is the genuine positive percentage over the entire sample count. Sklearn provides an internal classification report() method that returns key classification metrics, including precision, recall, and f1-score. False positives decrease as accuracy increases. The recall is the ratio of true positives to all true positives and all false negatives. Less false negatives are caused by higher recall. Precision and recall are combined into one statistic, the F1 score. The best value is 1, while the poorest value is 0. An actual positive rate vs false positive rate diagram is called a ROC curve. The ROC curve's undersurface is the area under the curve (AUC). A score of 1.0 corresponds to an ideal model.

5.1.1 ROC Curve & AUC Scores

A ROC curve: what is it? Comparing the actual positive rate to the false positive rate is a technique for analysing the performance of our model. Consider a diagnostic procedure to ascertain a person's presence of a specific disease to acquire a practical example in a real-world issue. When a person tests positive but does not indeed have the disease, it is called a false positive in this situation. In contrast, a false negative is when a person tests negative but has the condition, leading others to believe they are healthy.

The plot roc curve function in Scikit-Learn enables us to build ROC curves and determine the area under the curve (AUC) measure. As we can see from the plot roc curve function's documentation, it accepts (estimator, X, and y) as arguments. Where X and Y are the data we want to evaluate the estimator on, an estimator is a fitted machine learning model. In this instance, we will utilise the test data, X test and y test, as well as the GridSearchCV version of our LogisticRegression estimator, gs log reg. This is fantastic because our model outperforms guessing, which would be represented as a line with an AUC of 0.5 and running from the bottom left to the top right corner. There is still an opportunity for improvement because a perfect model would have an AUC score of 1.0.

```
1  # Import ROC curve function from metrics module
2  from sklearn.metrics import plot_roc_curve
3
4  # Plot ROC curve and calculate AUC metric
5  plot_roc_curve(gs_log_reg, X_test, y_test);
```

Figure 26 ROC and AUC Curve Function

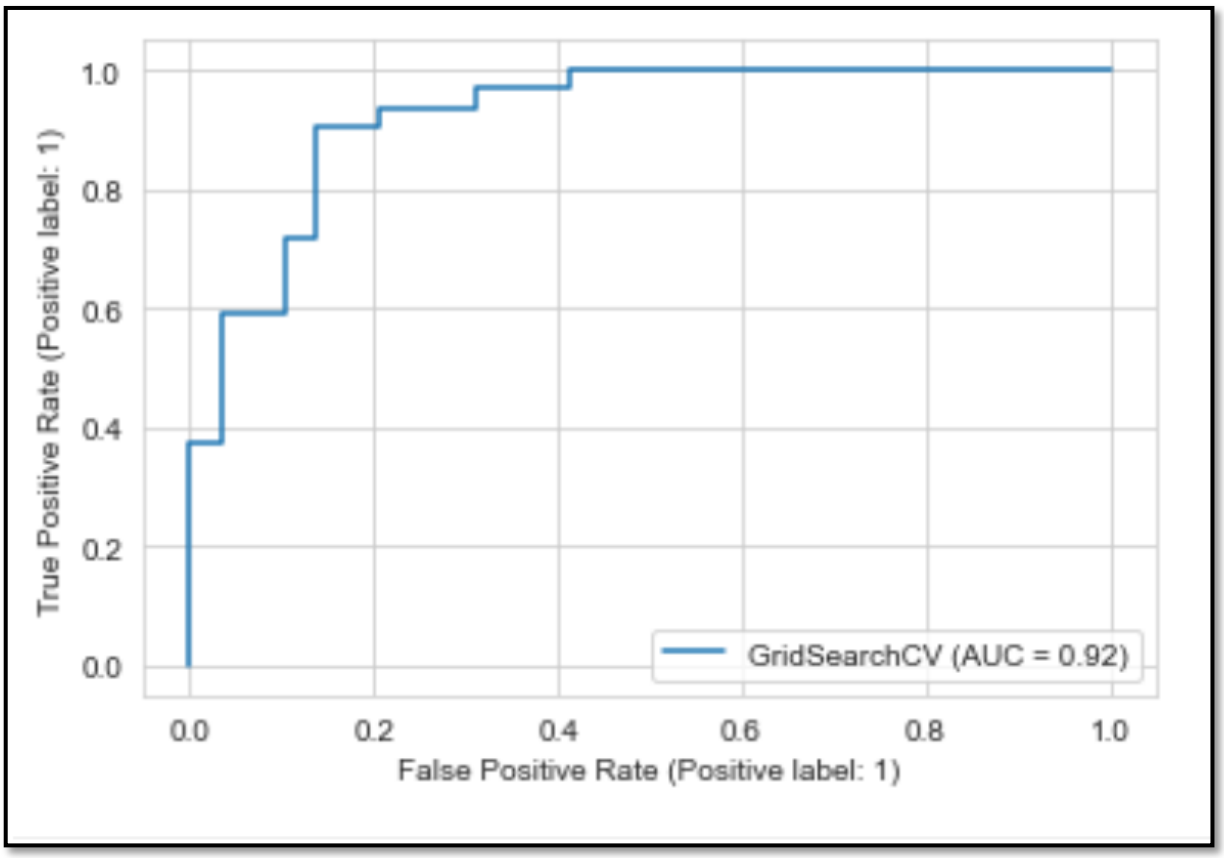


Figure 27 ROC and AUC Curve

5.1.2 Confusion Matrix

Where our model correctly predicted outcomes and where it mispredicted them may be seen visually using a confusion matrix (or, in other words, got confused). With Scikit-Learn, we can construct a confusion matrix and send it along to the actual labels and predicted labels using the confusion matrix(). As can be seen, the default confusion matrix provided by Scikit-Learn is a little dull. We would likely want to make it visual for a presentation. Let us build a method that accomplishes this using Seaborn's heatmap() function. In both groups, we can observe that the model consistently predicts the incorrect label. In essence, the model predicted 0 on four instances when it should have predicted 1 (false negative) and projected one on three instead of 0 on each occasion (false positive).

```python
# Import Seaborn
import seaborn as sns
sns.set(font_scale=1.5) # Increase font size

def plot_conf_mat(y_test, y_preds):
    """
    Plots a confusion matrix using Seaborn's heatmap().
    """
    fig, ax = plt.subplots(figsize=(3, 3))
    ax = sns.heatmap(confusion_matrix(y_test, y_preds),
                     annot=True, # Annotate the boxes
                     cbar=False)
    plt.xlabel("true label")
    plt.ylabel("predicted label")

plot_conf_mat(y_test, y_preds)
```

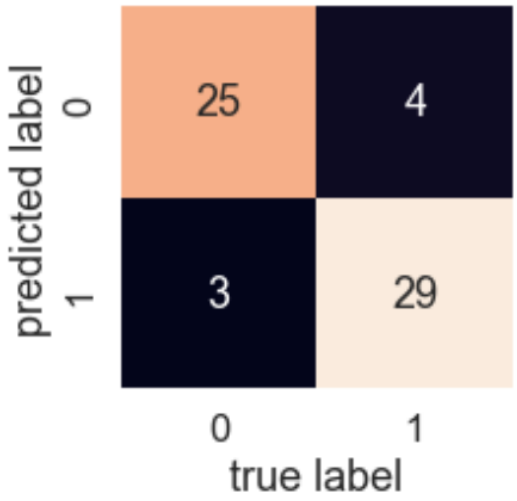


Figure 28 Confusion Matrix Predictions

5.1.3 Classification Report

Now that we have a deeper understanding of our model, let us move on. With the help of a classification report(), we can create a report that we can provide to the actual labels, models, and anticipated labels. The accuracy and recall of our model for each class will also be detailed in a classification report. However, a single training and test set was used to compute each of them. We will compute them using cross-validation in order to give them additional stability. Using cross val score() and various scoring parameter values, we will employ the best model, the best hyperparameters, and the best hyperparameter values. When using cross val score(), data, labels, and an estimator (a machine learning model) are all inputs. After that, a predetermined score parameter and cross-validation are used to test the machine learning model using the data and labels. Let us review the best hyperparameters before putting them to use.

```
# Show classification report
print(classification_report(y_test, y_preds))
```

```
              precision    recall  f1-score   support

           0       0.89      0.86      0.88        29
           1       0.88      0.91      0.89        32

    accuracy                           0.89        61
   macro avg       0.89      0.88      0.88        61
weighted avg       0.89      0.89      0.89        61
```

Figure 29 Classification Report

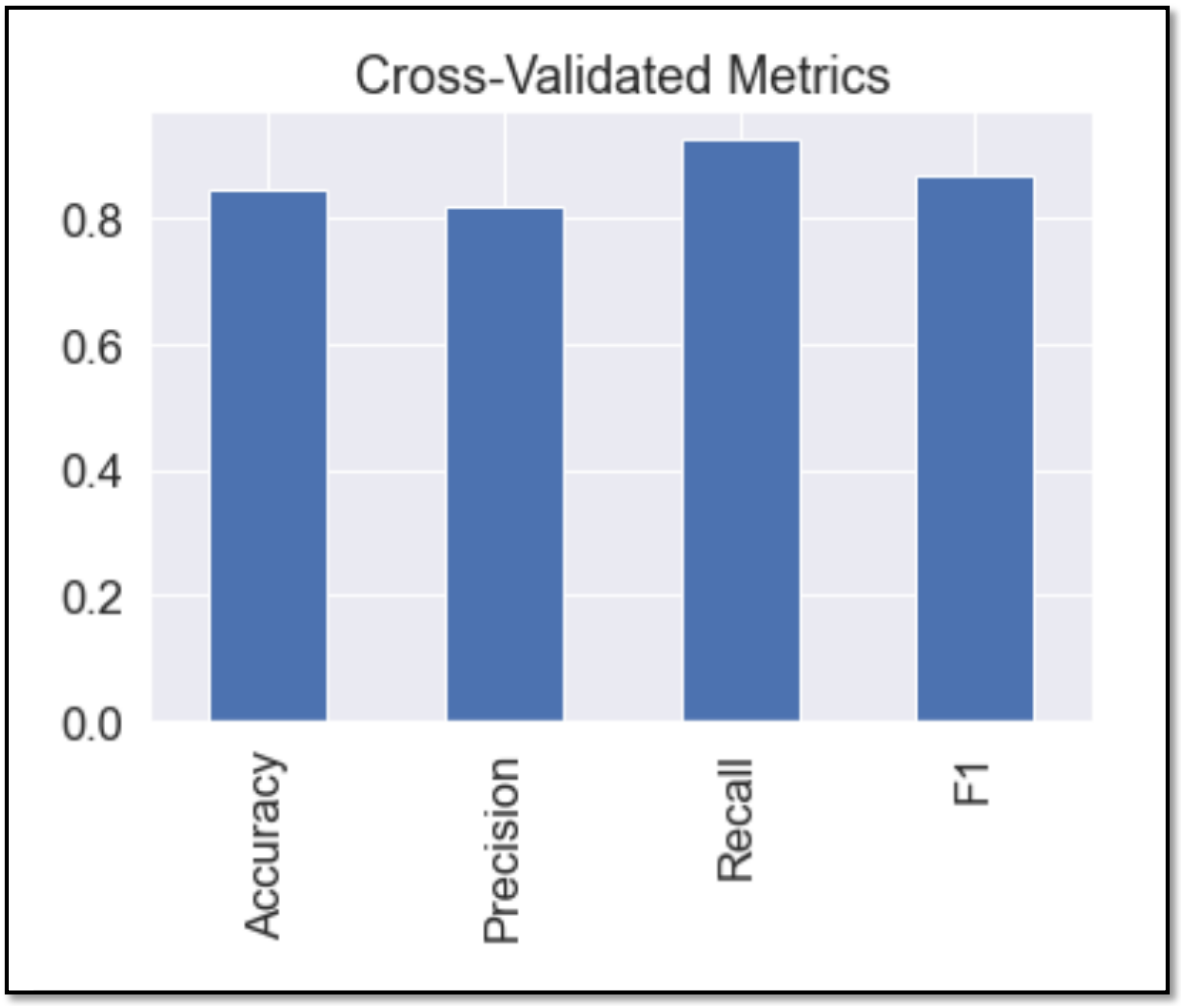


Figure 30 Cross-Validated Metrics

5.2 Feature Importance

The question "which features contribute most to the results of the model?" may also be asked in terms of feature significance. Or, in the case of our issue, trying to determine whether heart disease can be predicted using a patient's medical characteristics, which characteristics are most important in creating a model that can determine whether someone has heart disease or not? Because each model's method for identifying patterns in data varies slightly from one another, unlike some of the other functions we have examined, each model's method for determining the significance of those patterns also varies. As a result, determining which attributes were most crucial varied slightly depending on the model. Usually, we may locate an example by searching for "[MODEL TYPE] feature importance," such as "random forest feature importance," in the Scikit-Learn documentation or by looking through the material directly.

We will look at how we can determine the feature relevance for LogisticRegression since that is what we will be doing. We will take advantage of the coef_ property for this. The coefficient of the features in the decision function is what the coef_ attribute of the LogisticRegression attribute in Scikit-Learn represents. After fitting a LogisticRegression object, we may access the coef_ attribute. Some are unfavourable, while others are favourable, as we shall see. The feature's contribution to the model's decision increases with increasing value (more fabulous bar). If the value is negative, a negative correlation exists. Positive values work the other way around. As an illustration, the sex attribute has a negative value of -0.904, which indicates that the target value lowers as the sex attribute's value rises. The sex column and the target column can be compared to demonstrate this.

As can be seen, there are roughly three times as many patients with heart disease (goal = 1) when sex is 0 (female) as without (72 vs 24). The ratio of those with heart disease to those without it eventually decreases to virtually 1 to 1 (114 vs 93) when the number of genders climbs to 1 (male). What does this indicate? It suggests the model has discovered a pattern that accurately represents the data. These numbers and this particular dataset show that female patients are more likely to develop heart disease. A strong association, perhaps?

The model predicts a positive correlation of 0.470, higher than zero but not quite as high as the sex-to-target correlation. This indicates that our model detects the trend that the goal value rises as the slope climbs, according to the positive correlation. As we can see from the comparison (pd.crosstab(df["slope"], df["target"]), it is. Target increases about slope. With this knowledge, what can we do? We ought to consult an authority on the issue about this. They may be curious about the areas where the machine learning model finds the most patterns (areas with the highest correlation) and the areas where it does not (lowest correlation).

```
# Visualize feature importance
features_df = pd.DataFrame(features_dict, index=[0])
features_df.T.plot.bar(title="Feature Importance", legend=False);
```

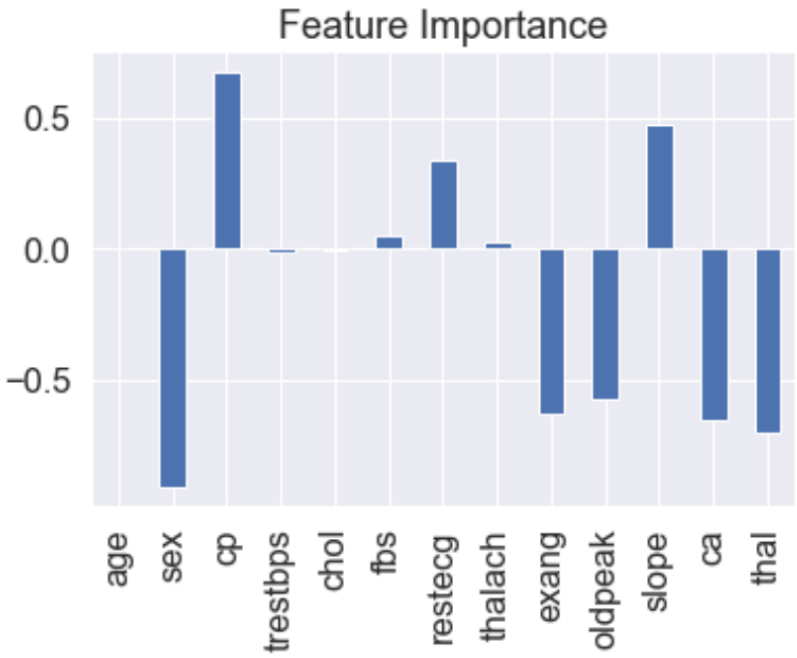


Figure 31 Feature Importance of the Models

```
pd.crosstab(df["sex"], df["target"])
```

target sex	0	1
0	24	72
1	114	93

Figure 32 Heart Disease Frequency according to Gender using Models

```
# Contrast slope (positive coefficient) with target
pd.crosstab(df["slope"], df["target"])
```

target slope	0	1
0	12	9
1	91	49
2	35	107

Figure 33 Heart Disease based on Slope Factor

5.3 Legal, Social, Ethical & Professional Issues

For this project, the original data came from the Cleveland database from UCI Machine Learning Repository. The UCI Machine Learning Repository is a collection of datasets, domain theories, and data generators that the machine learning community uses to test machine learning algorithms empirically. Since then, it has been widely utilised as a critical source of machine learning datasets by students, instructors, and researchers worldwide. As the dataset is available to the public, I do not require any resources or responses from a third party. I will use this dataset to build and train the models to predict the attributes contributing to coronary heart disease. Thus, I do not require any approval from the Ethics department. This project will use libraries like pandas, NumPy, matplotlib, Scikit-Learn, etc. Python is the primary programming language, and macOS is the operating system I will use for this project. The whole project will run on a Jupyter notebook which will be locally installed in the machine. To train the model efficiently, I need access to 6- 8 cores of CPU that can reduce the runtime and increase the accuracy of the models. The software I will use for this project is free to download and ready to use. I also require access to high-quality journal articles for literature review and theoretical background, which can be accessed from the university portal.

6

Conclusion & Future Work

6.1 Conclusion

It is common to hear people refer to medical analysis as a good information source. Early detection of coronary heart disease (CHD) can help reduce death rates because it is one of the top causes of mortality globally. The complexity of the data and linkages makes typical methodology-based prediction problematic. This project will use historical medical data to forecast CHD with machine learning (ML) technology. Using three supervised learning methods—Logistic Regression, K-Nearest Neighbours, and Random Forest—this study seeks to identify correlations in CHD data that might increase prediction rates. Only 14 out of 76 features will be utilised to predict our target variable from the Cleveland database from the UCI Machine Learning Repository. Finally, the most accurate model for predicting coronary heart disease is logistic regression.

6.2 Future Work

Trying out various models, fine-tuning existing models, and determining the ideal hyperparameters were all done. This has been a set of experiments that we have been working through. We could go on, in actuality. However, nothing lasts forever, as we all know. To debate or explore our many possibilities for moving forward with our team would be a significant next step. Could we gather additional data? A better model could be worth a try. Consider CatBoost or XGBoost if we are working with structured data. Can we go beyond what we have already done to improve the models that are now in use? What exporting and sharing options are available if our model is good enough? Here, it is essential to remember that time will be our biggest constraint. Keeping the distance between tests as short as possible is crucial. We will learn more about what does not work as we experiment and begin to understand what does as we do.

Bibliography

Alsouda, Y., Pllana, S. and Kurti, A. (2019) IoT-based Urban Noise Identification Using Machine Learning, In *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, New York, NY, USA, ACM, pp. 62–67.

Anon (1984) The Lipid Research Clinics Coronary Primary Prevention Trial Results, *JAMA*, 251(3), p. 351.

Anton, S. D., Kanoor, S., Fraunholz, D. and Schotten, H. D. (2018) Evaluation of Machine Learning-based Anomaly Detection Algorithms on an Industrial Modbus/TCP Data Set, In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, New York, NY, USA, ACM, pp. 1–9.

Ara, L., Luo, X., Sawchuk, A. and Rollins, D. (2019) Automate the Peripheral Arterial Disease Prediction in Lower Extremity Arterial Doppler Study using Machine Learning and Neural Networks, In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, New York, NY, USA, ACM, pp. 130–135.

Banerjee, R., Vempada, R., Mandana, K. M., Choudhury, A. D. and Pal, A. (2016) Identifying coronary artery disease from photoplethysmogram, In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, New York, NY, USA, ACM, pp. 1084–1088.

Bhatia, A., Chug, A., Prakash Singh, A. and Singh, D. (2021) Investigate the Impact of Resampling Techniques on Imbalanced Datasets: A Case Study in Plant Disease Prediction, In *2021 Thirteenth International Conference on Contemporary Computing (IC3-2021)*, New York, NY, USA, ACM, pp. 278–285.

Bi, Q., Goodman, K. E., Kaminsky, J. and Lessler, J. (2019) What is Machine Learning? A Primer for the Epidemiologist, *American Journal of Epidemiology*.

Campbell, N. C., Thain, J., Deans, H. G., Ritchie, L. D. and Rawles, J. M. (1998) Secondary prevention in coronary heart disease: baseline survey of provision in general practice, *BMJ*, 316(7142), pp. 1430–1434.

Critchley, J. A. and Capewell, S. (2003) Mortality Risk Reduction Associated With Smoking Cessation in Patients With Coronary Heart Disease, *JAMA*, 290(1), p. 86.

Daanouni, O., Cherradi, B. and Tmiri, A. (2020) Diabetes Diseases Prediction Using Supervised Machine Learning and Neighbourhood Components Analysis, In *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*, New York, NY, USA, ACM, pp. 1–5.

Doyle, J. T., Dawber, T. R., Kannel, W. B., Heslin, A. S. and Kahn, H. A. (1962) Cigarette Smoking and Coronary Heart Disease, *New England Journal of Medicine*, 266(16), pp. 796–801.

Elsayed, H. A. G. and Syed, L. (2017) An automatic early risk classification of hard coronary heart diseases using framingham scoring model, In *Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing*, New York, NY, USA, ACM, pp. 1–8.

Flores, A. M., Schuler, A., Eberhard, A. V., Olin, J. W., Cooke, J. P., Leeper, N. J., Shah, N. H. and Ross, E. G. (2021) Unsupervised Learning for Automated Detection of Coronary Artery Disease Subgroups, *Journal of the American Heart Association*, 10(23).

Ghosh, P., Azam, S., Karim, A., Jonkman, M. and Hasan, MD. Z. (2021) Use of Efficient Machine Learning Techniques in the Identification of Patients with Heart Diseases, In *2021 the 5th International Conference on Information System and Data Mining*, New York, NY, USA, ACM, pp. 14–20.

Gomes, C. P. (2011) Computational Sustainability, In pp. 8–8.

Gonsalves, A. H., Thabtah, F., Mohammad, R. M. A. and Singh, G. (2019) Prediction of Coronary Heart Disease using Machine Learning, In *Proceedings of the 2019 3rd International Conference on Deep Learning Technologies - ICDLT 2019*, New York, New York, USA, ACM Press, pp. 51–56.

Gren, L. and Ralph, P. (2022) What makes effective leadership in agile software development teams?, In *Proceedings of the 44th International Conference on Software Engineering*, New York, NY, USA, ACM, pp. 2402–2414.

Gren, L. and Shepperd, M. (2022) Problem reports and team maturity in Agile automotive software development, In *Proceedings of the 15th International Conference on Cooperative and Human Aspects of Software Engineering*, New York, NY, USA, ACM, pp. 41–45.

Gupta, C., Saha, A., Subba Reddy, N. v and Dinesh Acharya, U. (2022) Cardiac Disease Prediction using Supervised Machine Learning Techniques., *Journal of Physics: Conference Series*, 2161(1), p. 012013.

Hajar, R. (2017) Risk factors for coronary artery disease: Historical perspectives, *Heart Views*, 18(3), p. 109.

Hashem, S., Esmat, G., Elakel, W., Habashy, S., Raouf, S. A., Elhefnawi, M., Eladawy, M. I. and ElHefnawi, M. (2018) Comparison of Machine Learning Approaches for Prediction of Advanced Liver Fibrosis in Chronic Hepatitis C Patients, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(3), pp. 861–868.

Holzinger, A. (2018) From Machine Learning to Explainable AI, In *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, IEEE, pp. 55–66.

Irbaz, M. S., Azad, A., Sathi, T. A. and Lota, L. N. (2020) Nurse care activity recognition based on machine learning techniques using accelerometer data, In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, New York, NY, USA, ACM, pp. 402–407.

Jolly, K., Bradley, F., Sharp, S., Smith, H., Thompson, S., Kinmonth, A.-L. and Mant, D. (1999) Randomised controlled trial of follow up care in general practice of patients with myocardial infarction and angina: final results of the Southampton heart integrated care project (SHIP), *BMJ*, 318(7185), pp. 706–711.

Joshi, S., Upadhyay, H., Lagos, L., Akkipeddi, N. S. and Guerra, V. (2018) Machine Learning Approach for Malware Detection Using Random Forest Classifier on Process List Data Structure, In *Proceedings of the 2nd International Conference on Information System and Data Mining - ICISDM '18*, New York, New York, USA, ACM Press, pp. 98–102.

Kannel, W. B. (1979) Diabetes and cardiovascular disease. The Framingham study, *JAMA: The Journal of the American Medical Association*, 241(19), pp. 2035–2038.

Khan, Y., Qamar, U., Yousaf, N. and Khan, A. (2019) Machine Learning Techniques for Heart Disease Datasets, In *Proceedings of the 2019 11th International Conference on Machine Learning and Computing - ICMLC '19*, New York, New York, USA, ACM Press, pp. 27–35.

Khateeb, N. and Usman, M. (2017) Efficient Heart Disease Prediction System using K-Nearest Neighbor Classification Technique, In *Proceedings of the International Conference on Big Data and Internet of Thing - BDIOT2017*, New York, New York, USA, ACM Press, pp. 21–26.

Klag, M. J., Ford, D. E., Mead, L. A., He, J., Whelton, P. K., Liang, K.-Y. and Levine, D. M. (1993) Serum Cholesterol in Young Men and Subsequent Cardiovascular Disease, *New England Journal of Medicine*, 328(5), pp. 313–318.

Kohn, T., van Rossum, G., Bucher II, G. B., Talin and Levkivskyi, I. (2020) Dynamic pattern matching with Python, In *Proceedings of the 16th ACM SIGPLAN International Symposium on Dynamic Languages*, New York, NY, USA, ACM, pp. 85–98.

Krawczuk, P., Papadimitriou, G., Nagarkar, S., Kiran, M., Mandal, A. and Deelman, E. (2021) Anomaly Detection in Scientific Workflows using End-to-End Execution Gantt Charts and Convolutional Neural Networks, In *Practice and Experience in Advanced Research Computing*, New York, NY, USA, ACM, pp. 1–5.

Krishnani, D., Kumari, A., Dewangan, A., Singh, A. and Naik, N. S. (2019) Prediction of Coronary Heart Disease using Supervised Machine Learning Algorithms, In *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, IEEE, pp. 367–372.

Kutrani, H., Eltalhi, S. and Ashleik, N. (2021) Predicting factors influencing survival of breast cancer patients using logistic regression of machine learning, In *The 7th International Conference on Engineering & MIS 2021*, New York, NY, USA, ACM, pp. 1–6.

Law, M. R., Wald, N. J. and Thompson, S. G. (1994) By how much and how quickly does reduction in serum cholesterol concentration lower risk of ischaemic heart disease?, *BMJ*, 308(6925), pp. 367–372.

Le, D.-H. and Nguyen, M.-H. (2015) Towards more realistic machine learning techniques for prediction of disease-associated genes, In *Proceedings of the Sixth International Symposium on Information and Communication Technology*, New York, NY, USA, ACM, pp. 116–120.

Radovanović, S., Delibašić, B., Jovanović, M., Vukićević, M. and Suknović, M. (2018) Framework for integration of domain knowledge into logistic regression, In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, New York, NY, USA, ACM, pp. 1–8.

Shehzadi, S., Hassan, M. A., Rizwan, M., Kryvinska, N. and Vincent, K. (2022) Diagnosis of Chronic Ischemic Heart Disease Using Machine Learning Techniques, *Computational Intelligence and Neuroscience*, 2022, pp. 1–9.

Song, S., Chen, T. and Antoniou, G. (2021) ANFIS Models for Heart Disease Prediction, In *2021 the 5th International Conference on Innovation in Artificial Intelligence*, New York, NY, USA, ACM, pp. 32–35.

Wagstaff, K. (2012) Machine Learning that Matters, *CoRR*, abs/1206.4656.

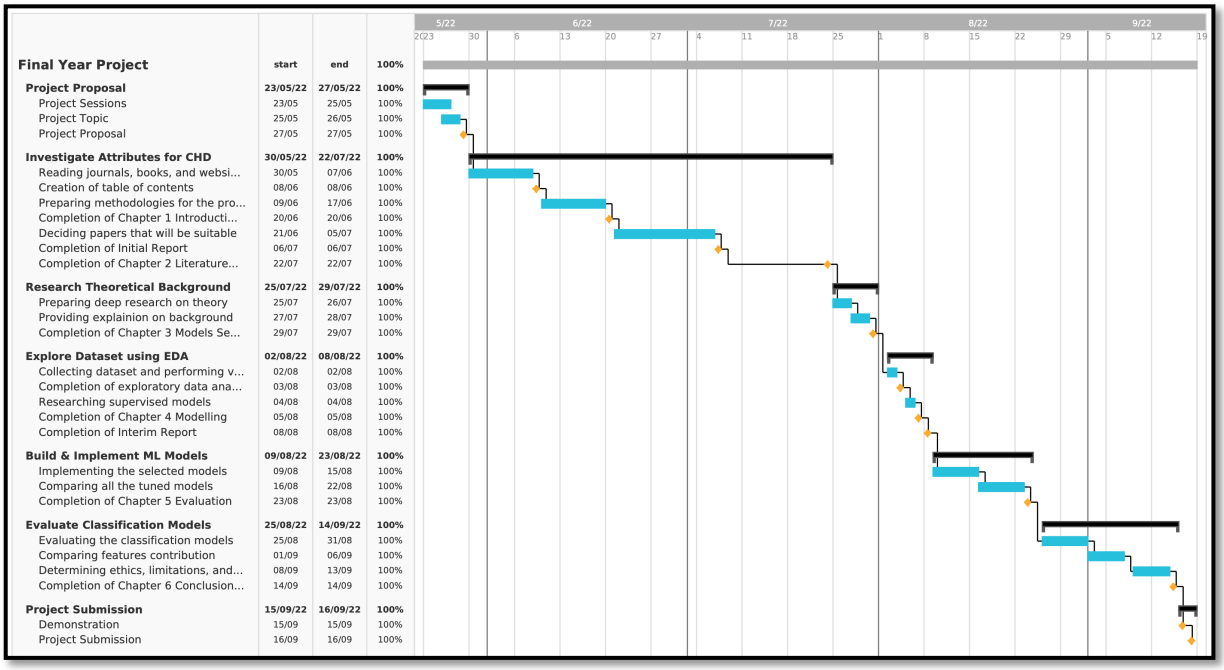
Wang, X., Sontag, D. and Wang, F. (2014) Unsupervised learning of disease progression models, In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, ACM, pp. 85–94.

Zhang, R. F. and Urbanowicz, R. J. (2020) A scikit-learn compatible learning classifier system, In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, New York, NY, USA, ACM, pp. 1816–1823.

Zhang, Y. (2010) *New Advances in Machine Learning*, InTech.

Appendix

Gantt Chart



Final Year Project	start	end	100%
Project Proposal	**23/05/22**	**27/05/22**	**100%**
Project Sessions	23/05	25/05	100%
Project Topic	25/05	26/05	100%
Project Proposal	27/05	27/05	100%
Investigate Attributes for CHD	**30/05/22**	**22/07/22**	**100%**
Reading journals, books, and websi...	30/05	07/06	100%
Creation of table of contents	08/06	08/06	100%
Preparing methodologies for the pro...	09/06	17/06	100%
Completion of Chapter 1 Introducti...	20/06	20/06	100%
Deciding papers that will be suitable	21/06	05/07	100%
Completion of Initial Report	06/07	06/07	100%
Completion of Chapter 2 Literature...	22/07	22/07	100%
Research Theoretical Background	**25/07/22**	**29/07/22**	**100%**
Preparing deep research on theory	25/07	26/07	100%
Providing explainion on background	27/07	28/07	100%
Completion of Chapter 3 Models Se...	29/07	29/07	100%
Explore Dataset using EDA	**02/08/22**	**08/08/22**	**100%**
Collecting dataset and performing v...	02/08	02/08	100%
Completion of exploratory data ana...	03/08	03/08	100%
Researching supervised models	04/08	04/08	100%
Completion of Chapter 4 Modelling	05/08	05/08	100%
Completion of Interim Report	08/08	08/08	100%
Build & Implement ML Models	**09/08/22**	**23/08/22**	**100%**
Implementing the selected models	09/08	15/08	100%
Comparing all the tuned models	16/08	22/08	100%
Completion of Chapter 5 Evaluation	23/08	23/08	100%
Evaluate Classification Models	**25/08/22**	**14/09/22**	**100%**
Evaluating the classification models	25/08	31/08	100%
Comparing features contribution	01/09	06/09	100%
Determining ethics, limitations, and...	08/09	13/09	100%
Completion of Chapter 6 Conclusion...	14/09	14/09	100%
Project Submission	**15/09/22**	**16/09/22**	**100%**
Demonstration	15/09	15/09	100%
Project Submission	16/09	16/09	100%