University of Reading

Department of Computer Science

Computer Science

Heart Disease Prediction with Machine Learning

In Data Science

Zohaib Ahmad

Supervisor: Nisha Singh

A report submitted in partial fulfilment of the requirements of the University of Reading for the degree of Bachelor of Science in Computer Science

April 15, 2024

Declaration

I, Zohaib Ahmad, of the Department of Computer Science, University of Reading, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly. I give consent to a copy of my report being shared with future students as an exemplar. I give consent for my work to be made available more widely to members of UoR and public with interest in teaching, learning and research.

Zohaib Ahmad

April 15 , 2024

Contents

Introduction

Background Information

Heart failure and other cardiovascular diseases contribute to around 17.9 million deaths each year worldwide as reported by the World Health Organisation [1]. This ranging category encompasses complications affecting blood vessels and other heart-related conditions. A crucial factor in addressing this health problem involves using analytics to detect potential instances of heart failure at an early age. By using various machine learning methods in healthcare, patients could receive benefits from a young age ultimately reducing fatalities related to heart failure.

There are numerous risk factors linked to these diseases, including issues such as high blood pressure, smoking habits, elevated cholesterol levels, diabetes, and lack of physical activity[2]. Family history also plays a role; individuals with a family history of CVD diagnosis have a higher susceptibility to these conditions. Furthermore, ethnic background can impact CVD risk levels. For example, Individuals from Black African or African Caribbean backgrounds in the UK are more prone to developing CVD due to higher occurrences of high blood pressure and type 2 diabetes, within these populations.

Factors that play a role in the development of disease (CVD) include age, with a higher prevalence in individuals over 50 and an increased risk as one gets older [3]. Men tend to be more prone to developing CVD at any age compared to women. Having an unhealthy diet can result in high cholesterol levels and high blood pressure and research shows that excessive alcohol consumption can lead to raised cholesterol and blood pressure levels [3] which in turn contribute to weight gain. By recognising these risk factors and integrating them into models it becomes possible to identify individuals with a heightened risk of developing heart-related problems and therefore enable targeted interventions aimed at lowering the occurrence of CVD-related health complications and mortality.

Reducing the chances of developing heart disease involves making changes to your lifestyle. A key aspect is to follow a well-rounded diet that supports heart health by including high-fibre foods, like grains, fresh fruits, and vegetables [3]. Diets such as the Mediterranean and DASH diets are often recommended for their positive impact on heart health as well as avoiding processed foods with fats, salt and sugar in order

to promote overall well-being. Being physically active is also crucial for prevention as exercise helps strengthen the heart and circulatory system and lowers cholesterol levels. Maintains blood pressure. Health professionals suggest aiming for 150 minutes [3] of moderate-intensity exercise per week to enhance heart health.

Maintaining a weight is key in preventing heart disease. A body mass index (BMI) between 20 and 25 is generally considered optimal for heart health[3]. There are various tools accessible to help individuals track their BMI and stay within this range. Cutting off unhealthy habits like smoking is imperative as it poses risks for heart and cardiovascular conditions. To significantly lower the risk of heart disease it is best to avoid the use of tobacco. Monitoring alcohol consumption is equally important; experts recommend that women limit themselves to one drink per day, while men should restrict themselves to two drinks. Additionally addressing health issues, like blood pressure, obesity and diabetes is crucial. Sufficient healthcare and regular health check-ups can help reduce the chances of complications. By prioritising these areas people can take the necessary steps to prevent heart disease thus leading a healthier lifestyle.

Machine learning is the process of computers improving their obility to carry out tasks by analysing new data, without a human needing to give instructions in the form of a program, or the study of creating and using computer systems that can do this[4]. Machine learning, which is dependent on data, needs a collection of data known as a dataset. Scientists extract information from these datasets, and machine learning algorithms are applied to find patterns in the dataset. In the context of healthcare, machine learning can be used in a variety of ways such as identifying diseases and diagnoses, creating drugs and manufacturing, medical images, personalised medicine, health records, clinical research, data collection, radiotherapy, and discovery in possible outbreaks. Traditional machine learning was used most commonly in precision medicine, where all previous data such as genes, environment and lifestyle were looked at using machine learning models and algorithms to treat individual patients accordingly.

Machine learning (ML) is transforming the healthcare sector by improving accuracy tailoring treatments to patients and presenting innovative solutions to longstanding challenges. By harnessing ML capabilities, computers are able to make connections, predictions and uncover insights from datasets that might otherwise remain undiscovered[5]. This technological advancement has an impact on health by enhancing patient outcomes and unlocking previously inaccessible medical knowledge. Moreover, ML plays a role in supporting and validating decisions. For example, when a physician prescribes a medication ML can leverage data to support the effectiveness of this treatment for patients sharing backgrounds. This process aligns with the main goal of ML to enhance outcomes and generate medical insights.

In scenarios, ML is utilised for enhancing efficiency in healthcare tasks such as managing medical records and analysing extensive clinical data sets to identify patterns. Furthermore, ML contributes to the creation of devices that assist healthcare professionals in their practices. A specialised branch within ML is called networks or artificial neural networks (ANNs) [6] which imitates the neural networks present in the human brain. ANNs serve as the foundation for deep learning systems that enable machines to learn from volumes of data. In the field of healthcare, learning techniques play a role in analysing MRI scans and other medical images to swiftly identify abnormalities. This advancement aids doctors in expediting the diagnosis and treatment process ultimately improving care. It's important to note that this technology complements rather than replaces the expertise of healthcare professionals allowing them to deliver effective diagnoses and treatments.

Abstract

The project focuses on the task of looking at heart conditions by utilizing three sets of data. Each dataset contains factors and patient details that could impact the accuracy of models. The main concern lies in

identifying the data analysis techniques for predicting heart diseases across these datasets. This requires studying how variations, in the three datasets show the effectiveness and trustworthiness of models. The project's goal was to discover the combination of datasets and machine learning models to predict heart disease accurately. Study looked at analysis of datasets, each containing patient information and features to determine the most effective pairing for precise predictions. The aim is to make cardiovascular heart disease predictions more predictable and more precise.

Objective

My primary objective was to pinpoint which dataset when paired with a machine learning model yielded reliable results for heart disease. This led to testing known models like linear regression, decision trees and random forests across diverse datasets. The project aimed to uncover how certain features in the datasets influenced the accuracy of these models. The outcomes provided insights into selecting data features and models for predicting heart disease based on available data and model performance with different data types. The focus was on enhancing prediction techniques in environments by ensuring that my chosen methods aligned well with real-world requirements.

Ethics

By utilising openly accessible datasets on platforms like Kaggle, we ensured that the data was already stripped of any confidential details thus upholding standards regarding data privacy. This approach to gathering data emphasises the handling of information in research in sensitive fields such as healthcare, where protecting patient confidentiality is crucial. Furthermore, by using datasets our study supports for transparency and reproducibility. This allows researchers to access the datasets to verify results or further investigations therefore fostering a culture of collaboration and knowledge sharing within the community. Such transparency reflects research practices and contributes to a more cooperative and reliable scientific setting.

Literature Review

This analysis played a role in selecting models for investigation and testing in our study. The literature reviews [7] covered a range of studies with some offering insights into effective models and others presenting less definitive or relevant findings. The varying quality and relevance of these studies underscored the importance of taking an approach to evaluating past research. By examining both the strengths and limitations of studies this review, helped us pinpoint which models showed better results than others.

In this research paper 1, various machine learning methods were used including K-Nearest Neighbour Algorithm, Decision Trees, Random Forest, Neural Networks Genetic Algorithm and Naïve Bayes, Logistic Regression, Gradient Boosted Trees, Support Vector Machine, and a hybrid model which included a hybrid random forest with a linear model (HRFLM)[7]. Three small datasets are used from UCI datasets of Cleveland, Switzerland and Hungarian. The UCI data set has 303 instances and out of these instances 252 records are used for training. Roughly 85% is used for training and 15% is used for testing. All the 13 attributes; age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, num were used in the machine learning model. The hybrid model scored the highest accuracy at 88.4% and the lowest accuracy score was implementing Naïve Bayes method at 75.8%[7]. Novel method such as VOTE [7] were used to balance the data despite the fairly high accuracy score with the hybrid model, the number of

instances was not a good representation when it came to machine learning models. Including the ratio of the training and testing which there were a low number of test instances, could have led to overfitting.

During the review of existing literature [8], it was noted that many studies commonly divided their data into 80% for training and 20% for testing purposes. Among the machine learning models examined Random Forest emerged as the popular choice while Logistic Regression was not as frequently favoured. The research papers explored a range of machine learning techniques focusing mainly on models and some hybrid methods. Various performance metrics like accuracy, precision, and AUC (Area Under the Curve) were utilised to assess the effectiveness of these models. The performance outcomes from the literature review revealed accuracies; Decision Trees achieved an accuracy of 86.37% with cross-validation and 86.53% without cross-validation; XGBoost attained 86.87% with cross-validation and 87.02% without; Random Forest scored 87.05% with cross-validation and 86.92% without, while Multilayer Perceptron demonstrated results of 87.28% with cross-validation and 86.94% without it[8]. Notably impressive AUC values were reported, with Decision Trees, at 0.94 and XGBoost Random Forest and Multilayer Perceptron each at a value of 0.95.

Based on these discoveries it was deduced that the Multilayer Perceptron model stood out as superior in achieving a record accuracy of 87.28%[8] especially when implemented alongside validation techniques. This finding highlighted how beneficial it is to utilise networks such as Multilayer Perceptron for accurately predicting heart disease. Additionally incorporating Body Mass Index (BMI) [8] into the models helped in boosting the model's performance. This approach to feature engineering was advantageous because BMI plays a role in the development of conditions and its inclusion significantly improved the model's prediction scores. This highlights the importance of integrating health indicators into models for heart disease. These discoveries highlight the importance of including health markers in models for heart disease and point towards avenues for additional investigation especially focusing on refining feature selection and model adjustments to enhance predictive results in the medical field.

The research paper [9] on predicting heart disease through machine learning techniques explores models with an emphasis on Support Vector Machines (SVM) due to their effectiveness in managing complex and high-dimensional datasets. The study emphasises SVMs proven reliability across applications placing it as a tool in the research. The findings of the study showcase outcomes where SVM achieved a training accuracy of 90.5% and a testing accuracy of 78.7% [9]. These results are supported by the utilisation of AUC ROC curves and confusion matrices validating the model's capability to accurately distinguish between patients with heart disease and those without. These metrics do not confirm the model's accuracy but strengthen its dependability for clinical use.

Nevertheless, employing SVM poses challenges. The model's nature and computational requirements can pose issues in environments with limited resources. Furthermore, SVM models are often criticised for their lack of interpretability [9], which is vital in diagnostics where understanding diagnostic decisions is important. Another limitation worth noting is SVMs sensitivity to parameter tuning, necessitating adjustments to prevent overfitting and underfitting and ensure model performance. When we look at SVM compared to decision trees, neural networks and logistic regression SVM stands out for its effectiveness in handling a variety of data types. This is particularly crucial in diagnosis, where the capability to identify patterns in multidimensional data can largely influence the accuracy of diagnoses and patient results.

The research in [10] discovered that Logistic Regression stood out as the model among those tested achieving an impressive accuracy level of around 92.30%. This model excelled in reducing negatives which are crucial in medical diagnosis to avoid missing any diseases. The success of Logistic Regression was

attributed to its ability to provide outcomes and manage results efficiently. On the contrary, the Decision Tree model displayed performance with an accuracy rate of approximately 85.71% [10]. Decision Trees are typically favoured for their simplicity and interpretability across data types. However, they are prone to overfitting issues especially when the trees become intricate and deep. This tendency can result in good performance on training data but poor performance on data leading to lower accuracy and potentially higher variability compared to other models. The study did not discuss the use of techniques like stopping, regularisation methods or other sophisticated enhancements for model training. Instead, it focused mainly on basic machine learning models and basic evaluation metrics without looking into more intricate strategies that could have potentially improved training efficiency and mitigated overfitting issues.

The study [11] explored how three machine learning models (K Nearest Neighbours (KNN) Naive Bayes and Random Forest) were used to predict heart disease by analysing a dataset from Kaggle. Among these models, Random Forest stood out with an accuracy rate of 95.63% in classifying heart diseases due to its handling of complexities in medical data. On the other hand, Naive Bayes fell short with an accuracy rate of 88.89% [11] as it assumes independence among features, which is often not the case in intricate medical datasets where mutuality exist. The research methodology focused on leveraging the strengths of the Random Forest algorithm to address overfitting through learning contributing to its performance. However advanced techniques, like regularisation and extensive hyperparameter tuning were not explored in depth in the paper, which could potentially enhance model performance. Additionally, the study [11]was limited to three models without exploring a range of machine learning approaches that could provide deeper insights or achieve better predictive results.

In the research paper, it was shown how machine learning models could predict heart disease by choosing features and handling missing data. To enhance the study, exploring models and developing a detailed strategy to optimise them would have been beneficial in machine learning for medical diagnostics. However, when three columns were removed [11], valuable information that could have helped predict heart disease could have been lost. This approach risked bias in the model as the remaining data may not have accurately represented all factors influencing heart disease. By ignoring this information, the study overlooked the nature of conditions potentially leading to a model that performed well in controlled environments but struggled to be effective if it was going to be used in real-world situations.

This research paper [12] thoroughly prepared the data for model training and evaluation by applying preprocessing and feature engineering techniques. This involved tasks like labelling and cleansing the data to handle values using imputation and selecting features to boost model performance. They divided the dataset into a training set (70%) and a test set (30%) [12] for an approach to training and validating models.

The Random Forest classifier stood out with a classification accuracy of 96.28% showcasing its effectiveness in handling medical diagnostic datasets. Its ensemble approach, which combines decision trees, proved beneficial in enhancing accuracy and avoiding overfitting. While the paper did not explicitly mention the performing model, simpler models like K Nearest Neighbours may struggle in datasets with numerous features without careful tuning of parameters like the number of neighbours. Compared to the performers achieving over 90% accuracy classifiers fell short of this mark [12].

The study showcased an application of methods, good at managing missing data and selecting features. Employing mean imputation preserved the dataset's integrity while utilising the info gain feature selection approach proved vital in pinpointing the features for the model. This could potentially streamline the accuracy. While the methodology displayed strengths, incorporating advanced machine-learning techniques could have enhanced the study.

In this research [13], we thoroughly analysed a dataset related to heart disease using machine learning techniques such as Multilayer Perceptron (MLP) K Nearest Neighbours (KNN) Decision Tree (DT) Random Forest (RF) Logistic Regression (LR) and AdaboostM1 (ABM1) [13]. These methods went through testing through a 10-fold cross-validation approach to validate the consistency of outcomes across diverse segments of the dataset. The main objective was to determine which technique excelled in predicting the occurrence of heart disease.

The standout performers among the classifiers were KNN, DT and RF all achieving flawless scores in sensitivity, specificity and accuracy with each hitting 100%. This suggests that these models were finely tuned for the dataset and adept at categorising all test instances without any mistakes. The adoption of 10 cross-validation [13] proved to be a strategy for assessing model performance by reducing the risk of overfitting and confirming that the models' accuracy extended to different unseen subsets of data. This approach improved the models' ability to perform across scenarios. While some classifiers showed performance the study could have delved into exploring advanced or alternative machine learning methods that could offer fresh insights or better predictive capabilities. For instance, trying out techniques like Gradient Boosting or XGBoost could have been beneficial in comparison to the models used. Moreover, considering deep learning strategies may have helped spot patterns in the data that conventional algorithms might have overlooked.

To sum up, the study effectively showcased using machine learning classifiers to predict heart disease accurately using a prepared dataset. Employing validation techniques enhanced result reliability indicating that KNN, DT and RF exhibited impressive accuracy levels [13]. However, achieving scores raised concerns about overfitting issues. Overfitting arises when a model fits closely to the training data it was exposed to hampering its ability to predict new or unseen data accurately. Attaining 100% accuracy could suggest that these models simply memorised the data rather than truly understanding how to generalise from it.

The ML HDPM [14] is a blend of machine learning and deep predictive modelling utilised techniques, like selecting features balancing data and optimising learning. Impressively the ML HDPM achieved a 95.5% accuracy in training and an 89.1% accuracy [14] in testing showing its effectiveness in predicting heart disease. However, the results also hint at overfitting issues since the testing accuracy is slightly lower than the training accuracy. A notable concern highlighted in the research is that while the ML HDPM shows performance metrics with a F score of 91.5% in training and 89.6% in testing it fails to explicitly address or mitigate risks of overfitting beyond its current methods. This lack of attention could lead to vulnerabilities when applying the model to datasets or real-world scenarios that differ from the training data.

The study[14] implemented validation techniques such, as K cross-validation to get a better result for accuracy which is a common approach used to combat overfitting and enhance model generalisability. Despite this, the results found during training indicate that there might be potential, for researching into regularisation methods[15] or more thorough cross-validation approaches to ensure the model maintains its effectiveness across different clinical settings. In general, the paper offers a strategy for predicting heart disease using machine learning methods showcasing notable advancements in model accuracy and performance. However, adding fine-tune the methodologies related to data quality feature selection and optimisation strategies will improve the accuracy and practicality of the predictive model.

The K nearest neighbours (KNN) algorithm showcased performance achieving an accuracy rate of 88.52% [16]. This accuracy highlighted KNNs effectiveness in analysing a dataset containing attributes like age, gender, chest pain and blood pressure. On the other hand, specific performance metrics for Logistic Regression and Random Forest Classifier were not given which suggested they might have been less successful compared to KNN. The emphasis on the performance of the KNN algorithm implies that other

techniques may not have matched its level of accuracy indicating potential areas for enhancement in those models.

Each algorithm employed in the study comes with its set of advantages and limitations. Logistic Regression stands out for its capacity to offer outcome probabilities for decision-making but its assumption of a linear relationship between variables could restrict its efficacy when dealing with linear data [17]. In contrast, KNN is known for its adaptability as it does not make assumptions about data distribution thereby enabling insights from datasets [18]. Nonetheless, it is resource intensive and sensitive to features and data scaling issues. On the other hand, Random Forest Classifier excelled in mitigating overfitting concerns with datasets and can handle diverse data types without requiring scaling. However, it may favour characteristics with complexity and the machine learning model demanded substantial computing power. By using these techniques in conjunction, with data preparation and dividing data into training and testing groups we were able to create a model that boosts predictive analysis in healthcare, particularly for heart conditions. This strategy did not only enhanced precision but also helped lower healthcare expenses by allowing early detection and effective treatment strategies.

In the heart disease prediction study [19] using machine learning algorithms, the Decision Tree classifier showed a performance with an accuracy of 92%. Its structured approach resembled decision making it suitable for the data relationships in the dataset. Conversely, both Logistic Regression and Random Forest classifiers achieved accuracies of around 80% [19]. These findings indicate that while these models are generally strong, they may have faced challenges with the dataset's complexity or feature distributions. The Random Forest model was effective against overfitting but did not surpass Logistic Regression suggesting issues with how ensemble strategies were applied, or model parameters were adjusted. The study focused on basic machine learning techniques like model training and testing assessing models through confusion matrices and classification reports and simple train test data splits. Common algorithms such as Logistic Regression, Decision Trees, Random Forests and SVM were used without methods, like cross-validation, early stopping or systematic hyperparameter tuning.

Introducing cross-validation methods could enhance model generalisation across diverse data subsets. Moreover, implementing stopping could be beneficial for models to prevent overfitting by stopping training when there is no improvement in validation results. Precise methods for adjusting hyperparameters like grid search or randomised search were not used which would have had the potential to improve model performance. These advanced strategies are particularly valuable, in healthcare settings where the precision and dependability of models can greatly influence patient outcomes. [] did not implement any advanced data preprocessing or optimisation techniques, hence reducing the accuracy of the models and in some cases lowering the validity of the dataset as there may have been missing pieces of data or data which was not highly relevant to heart disease prediction.

Four different machine learning models were utilised in [20]; XGBoost, Support Vector Machines (SVM) Naïve Bayes and Logistic Regression. Each model displayed strengths and weaknesses when measured against performance metrics such as: accuracy, F1 score, precision and recall. The XGBoost model emerged as the performer showcasing an accuracy of 92% an F1 score of 91% precision at 93% and recall at 92% [20]. This exceptional performance was credited to the models' learning strategy, which combines learners to create a stronger predictive model with increased accuracy. XGBoost stands out for its ability to effectively handle data features and reduce errors sequentially and therefore enhancing prediction capabilities.

In contrast, the SVM model exhibited an accuracy rate of 64% despite a recall rate of 95%. With a precision of 60%, it excelled in identifying positive cases but it also misclassified a notable number of negatives as

positives. This challenge might be linked to SVMs' sensitivity to kernel selection and data feature scaling indicating a need for refined hyperparameter tuning or data normalisation [20]. Furthermore, advanced machine learning techniques were ingrained into the study prominently featuring the use of XGBoost. This model uses gradient boosting frameworks which are more complex versions of boosting techniques that improve model predictions and computational efficiency. XGBoost can manage data irregularities such as missing values and outliers making it less prone to overfitting.

One notable limitation of the research was the removal of four columns: oldpeak, slope, number of major vessels and output. These omissions could have introduced biases and restricted the models' ability to fully understand the complexities of heart disease symptoms and indicators. For example, oldpeak, which represents ST depression caused by exercise compared to rest, is an indicator for diagnosing ischemia. Slope, number of vessels and output were also essential, for evaluating heart disease risk. By removing these columns the models were at the cost of losing predictive capability and potentially increasing bias or variance when applied to real-world data that includes these variables. The study effectively utilised both traditional and advanced machine learning methods to build models excluding data features which may have limited the models' performance in a clinical environment.

Various techniques were explored in the study ranging from Random Undersampling and Oversampling to advanced methods, like SMOTE and its variations such as Borderline SMOTE, SVM SMOTE, Adasyn and SMOTE Tomek [21]. Among these methods, SMOTE (Synthetic Minority Over Sampling Technique) stood out as an approach for addressing imbalanced datasets. This method involves creating instances of the minority class by synthesising data points between existing ones [21]. Doing this helps to achieve a balanced dataset and enhances the classifier's performance when predicting minority class outcomes. Unlike oversampling techniques that replicate existing examples, SMOTE generates instances thereby adding more generalisation capacity to the model and reducing the risk of overfitting.

The study highlighted the benefits of using SMOTE for one dataset with class imbalance. Conventional oversampling or under sampling strategies would have struggled to model this dataset because of its distribution. By applying SMOTE the goal was not to balance out the classes but to introduce diversity within the minority class to improve the learning process. This approach effectively boosted the representation of the minority class in a way that contributed to a resilient predictive model. One of the datasets in this project had an imbalance issue so we included SMOTE to address it. By using SMOTE we achieved a balance in class examples, which significantly boosted the accuracy and trustworthiness of the models we created.

This approach [22] involved a procedure, for converting the unbalanced dataset into several balanced sets using methods like random partitioning or grouping. This is then followed by applying classifiers and a combined rule for merging adding layers of complexity. In comparison, SMOTE created samples by interpolating between existing instances in the minority group making it easier to grasp and put into action. Furthermore, the new technique significantly altered the class distribution by forming subsets potentially impacting the integrity and accuracy of the initial data. On the other hand, SMOTE maintained the properties of the data better by creating synthetic minority samples while also altering the dataset through interpolation among existing minority samples.

Both approaches rely on parameters, such as determining how many synthetic samples [22] are generated for SMOTE or deciding on the number of splits or clusters in the technique. However, achieving results with the method may depend more on these settings due to its intricate data manipulations involving multiple balancing steps followed by classifier combinations. Moreover, SMOTE has been widely proven effective across fields and scenarios. Its consistent efficacy has been demonstrated in studies establishing it as a

choice for managing imbalanced datasets [23]. The track record of versatility and proven success often positions SMOTE as a favoured approach as used in multiple research papers.

In summary, the research examined provided an overview of how machine learning models were utilised in predicting heart disease. Each study employed different approaches and datasets to improve the accuracy and reliability of predictions. The analysis emphasised models such as Random Forest, Multilayer Perceptron, and Support Vector Machines, highlighting Random Forest's strength in effectively handling medical data and achieving high accuracy levels.

A common theme across these studies was the emphasis on data management and model validation techniques to prevent overfitting and ensure the models' applicability. Techniques such as cross-validation and the incorporation of health metrics like BMI demonstrated enhancements in model performance. Additionally, advanced methods like SMOTE addressed class imbalances, showcasing the evolving strategies for enhancing machine learning applications in healthcare.

Methodology

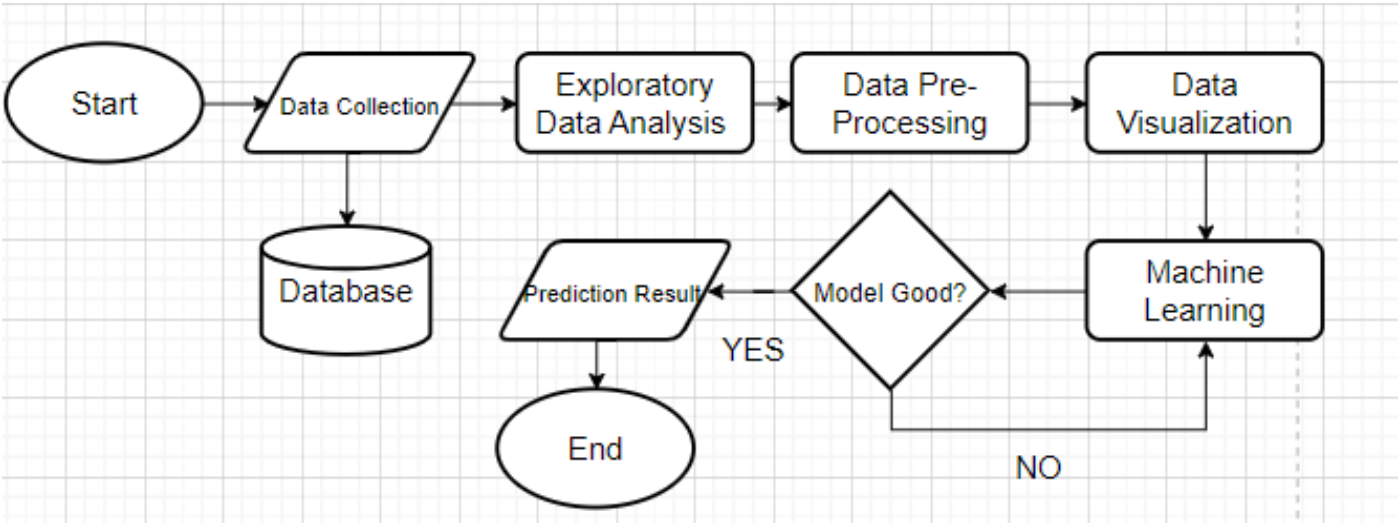

Figure 1: Methodology Flowchart

Data Exploration

The dataset related to diseases sourced from Kaggle comprises information collected from over 70,000 individuals who participated in a large-scale health survey. This survey aimed to assess well-being across demographics and was conducted using a sampling method to ensure inclusivity and representativeness of the data. It contains details about histories, lifestyle choices and results from regular health check-ups.

The initial step in exploring the data involves importing the "C_disease.csv" dataset into a Pandas DataFrame using the Python library for data manipulation. This dataset consisted of around 70,000 entries spread across 13 columns demographics, health parameters and disease markers. The first task post-data import is to identify and count rows to uphold data integrity and precision during analysis. Addressing duplicates can strengthen result reliability in analyses or machine learning applications.

After identifying duplicates, looked at detecting values within the dataset. Each column is scrutinised for NaN (Not a Number) instances to determine their presence in each column. This evaluation helps in assessing the quality of data and determining preprocessing steps, such as filling in or removing values as needed. Another key aspect of exploration involves analysing the distribution of values within the 'cardio' column of the dataset. By calculating the frequencies of '0' and '1' values insights can be gained into their prevalence, which is essential for classification tasks or analyses related to conditions within the dataset.

The dataset "heart_RP.csv" contains around 300 instances and 14 columns focusing on cardiovascular metrics like chest pain type and blood pressure that are crucial for diagnosing heart conditions. On the other hand, the "h_disease.csv" dataset expands on this with over 250,000 instances spread across 22 columns encompassing a large set of health indicators including lifestyle behaviour and healthcare access. These datasets are meticulously checked for duplicate and missing values to ensure a dataset that is ready for comprehensive analysis.

Furthermore, part of the process involves examining all datasets to detect any values providing an overview of data quality. This information is displayed in a DataFrame where 'True' indicates the presence of a missing value. Additionally, to dive deeper into metrics calculations are made for parameters like middle value and most common value across all numerical columns in the datasets. These factors help us to grasp the significance and range of the data, which is beneficial for data processing and examination.

By combining information from these three sets of data a thorough investigation can be carried out to delve into the connections, between lifestyle decisions, medical histories, and the occurrence of diseases. This method helped the understanding of the factors influencing health among the individuals surveyed.

Dataset 1

C_disease.csv

Columns:

id: Unique identifier for each record.

age: Individual's age, days

gender: Numerically encoded gender.

height: Height in centimetres.

weight: Weight in kilograms.

ap_hi: Systolic blood pressure.

ap_lo: Diastolic blood pressure.

cholesterol: Categorised or measured cholesterol level.

gluc: Categorised or measured glucose level.

smoke: Binary indicator of smoking status (0 = non-smoker, 1 = smoker).

alco: Binary indicator of alcohol intake.

active: Binary indicator of physical activity level.

cardio: Binary indicator of cardiovascular disease presence (0 = no, 1 = yes).

Dataset 2

H_disease.csv

Columns:

HeartDiseaseorAttack: Whether the individual has heart disease or has had an attack (0 = no, 1 = yes).

HighBP: High blood pressure (0 = no, 1 = yes).

HighChol: High cholesterol (0 = no, 1 = yes).

CholCheck: Whether the individual has had a cholesterol check in the last five years (0 = no, 1 = yes).

BMI: Body Mass Index.

Smoker: Smoking status (0 = non-smoker, 1 = smoker).

Stroke: Whether the individual has had a stroke (0 = no, 1 = yes).

Diabetes: Diabetes status (0 = no, 1 = yes; 2 = borderline).

PhysActivity: Physical activity status (0 = no, 1 = yes).

Fruits: Fruit consumption (0 = no, 1 = yes).

Veggies: Vegetable consumption (0 = no, 1 = yes).

HvyAlcoholConsump: Heavy alcohol consumption (0 = no, 1 = yes).

AnyHealthcare: Access to healthcare (0 = no, 1 = yes).

NoDocbcCost: Not visiting doctor due to cost (0 = no, 1 = yes).

GenHlth: General health status (scale of 1 to 5).

MentHlth: Mental health status (physical activity or exercise during the past 30 days).

PhysHlth: Physical health status (physical activity or exercise during the past 30 days).

DiffWalk: Difficulty walking (0 = no, 1 = yes).

Sex: Numerically encoded gender (0 = female, 1 = male).

Age: Age category (Fourteen-level age category).

Education: Education level (encoded numerically).

Income: Income category (encoded numerically).

Dataset 3

heart_RP.csv

Columns:

age: Age of the individual.

sex: Sex of the individual (1 = male, 0 = female).

cp: Chest pain type (value ranging from 0 to 3).

trestbps: Resting blood pressure (in mm Hg on admission to the hospital).

chol: Serum cholesterolic (mg/dl).

fbs: Fasting blood sugar > 120 mg/dl (1 = true; 0 = false).

restecg: Resting electrocardiographic results (values 0,1,2).

thalach: Maximum heart rate achieved.

exang: Exercise induced angina (1 = yes; 0 = no).

oldpeak: ST depression induced by exercise relative to rest.

slope: The slope of the peak exercise ST segment.

ca: Number of major vessels (0-3) colored by flourosopy.

thal: A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect).

condition: Presence of heart disease (1 = yes; 0 = no).

Exploration of Columns

High blood pressure can contribute to the development of diseases by putting strain on the heart making it work harder and pump with force due, to increased blood flow [24]. This prolonged pressure can lead to the heart enlarging. therefore weakening over time. Another effect of blood pressure is the damage it causes to artery walls reducing their flexibility and compromising blood circulation. Furthermore, the formation of plaques as a result of blood pressure can further restrict blood flow to the heart.

Engaging in physical activity is another way to lower the chances of developing heart disease. Exercise helps strengthen the heart muscles promoting blood circulation throughout the body [25]. Additionally, mental well-being plays a role. Conditions such as PTSD, anxiety or stress can raise blood pressure levels by increasing blood flow potentially leading to calcium deposits in arteries and ultimately contributing to heart disease. The aim is to improve accuracy and precision in predicting diseases. It's crucial not to overlook any health concerns that could pose a risk.

Data Cleaning and Preprocessing

In this stage of the project, it was focused on cleaning up the data. This important process involved standardising the age data from days to years for understanding eliminating any entries to keep the data unique and handling any missing information. Depending on their importance and impact on our analysis we. Filled in missing values. Removed them to maintain the accuracy and relevance of our dataset.

After completing the data cleaning phase, moved on to enhancing our datasets through feature engineering to boost the precision of our models. We computed individuals' Body Mass Index (BMI) based on their height and weight a measure for evaluating health conditions related to body weight. Additionally,

we developed a health status column by combining mental health ratings to offer a perspective on each patient's well-being. To effectively manage variables, we utilised one hot encoding to convert them into a format suitable for machine learning algorithms. This method was particularly beneficial for data like types of chest pain and thalassemia which were transformed into binary columns for easier numerical analysis.

The final phase, in our process, involved reviewing each dataset to validate the appropriateness of data transformations made and ensure that high data quality was maintained. This laid the foundation for the exploration of data and the use of machine learning techniques. Following this organised method not only improved the clarity and usefulness of the data. Also guaranteed that the datasets were well prepared for meaningful analyses and accurate predictions allowing us to reach significant conclusions, about the connections, among lifestyle choices, medical backgrounds and health results.

Data Visualisation

The study utilised a range of tools to analyse the three datasets. Initially, we crafted histograms and density plots to examine factors such, as age and Body Mass Index (BMI). These visual representations, with age represented on one axis and frequency on the other allowed us to see trends and pinpoint any outliers. When it came to data analysis bar charts played a role in illustrating the prevalence of conditions like cholesterol levels across age groups. By mapping age categories on one axis against the number of cholesterol cases on the other these charts facilitated comparisons between age brackets shedding light on health patterns linked to varying ages.

We also made use of correlation heatmaps to explore interrelationships between variables. These heatmaps depicted variables along both axes with colours indicating the strength of correlation between each pair of variables. This approach helped to identify connections for exploration. Furthermore, box plots were employed to explore into distributions within subgroups like age categories grouped by cholesterol levels. These visualisations showcased quartiles and medians providing insights into both trends and variations, within each subgroup.

By incorporating these visualisation techniques, we ensured that the graphical representations were not only informative but user-friendly and easier to visualise making it easier for scientists to comprehend the data insights and make informed decisions based on them. Models and evaluation metrics models were built to predict the risk of cardiovascular diseases, including logistic regression and random forests. Model evaluation was rigorously performed using metrics such as accuracy, ROC-AUC, and confusion matrices to validate the predictive power and reliability of the models.

Machine Learning Models

Logistic Regression

Logistic Regression was a used model primarily used for binary classification tasks but it could also be adapted for multi-class classification, with adjustments like multinomial logistic regression. The main concept of regression was to establish a connection between features and the likelihood of a result. In classification, the goal was to predict the probability of an observation belonging to one of two categories making it suitable for scenarios with two outcomes such as pass/fail, win/lose, or alive/dead [26].

Logistic regression was highly valued for its clarity as each coefficient directly reflected the odds ratio of the feature making it easier to grasp the impact of each variable. It was known for its stability, reducing the risk of overfitting especially when there were more observations than features in the dataset. Additionally, its

capability to provide probabilities as a measure of event likelihood made it a valuable model for tasks requiring risk evaluation.

However, logistic regression had limitations [26] due to assuming linear relationship between the logit of the outcome and predictor variables. This assumption could oversimplify data relationships that were often non-linear, impacting its performance on datasets with complex interactions. Furthermore, logistic regression was sensitive to outliers and influential data points that could skew estimates disproportionately, affecting model accuracy and reliability.

XGBoost

XGBoost, also known as eXtreme Gradient Boosting [27] is a version of the gradient boosting algorithm created to enhance speed and model effectiveness. It has gained a reputation for excelling in Kaggle competitions that prioritise predictive precision. This algorithm uses tactics to enhance both efficiency and memory usage. Essentially, XGBoost functions as a method, by constructing decision trees with each subsequent tree fixing mistakes from the preceding ones.

XGBoost stands out as a flexible machine learning tool that excels in tackling regression and classification tasks. It boosts the ability to cater to optimisation objectives and evaluation criteria set by the users. Known for its accuracy compared to traditional algorithms XGBoosts [27] advanced structure and efficient data management contribute to its success. Notably, it simplifies handling missing data by determining the approach during training eliminating the need for manual input but despite its advantages [27] XGBoost does also comes with challenges. Without tuning, it can become complex and vulnerable to overfitting requiring configuration efforts. Moreover, being resource intensive it demands power and energy consumption which may pose limitations in environments with restricted resources.

Decision Tree

The decision tree is a user friendly in data science and machine learning for handling classification and regression tasks. This model aids decision making by showing the results that arise from a series of choices. Within a decision tree each internal node serves as a decision point each branch symbolises the outcome of a decision and each leaf node signifies a classification or prediction.

In decision trees, entropy is computed using a formula to measure the uncertainty or disorder in a dataset. Entropy assists in determining how to divide the data at each node within the decision tree [7].

$$Entropy = -\sum_{j=1}^{m} p_{ij} \log_2 p_{ij}$$

Figure 2 : Entropy Equation

Decision trees have an ability to select features by pinpointing and utilising the informative ones at each decision point. This allows for prioritising and ranking key elements in a dataset leading to decision making. They are known for their flexibility in handling missing data through splits or scenarios well as their strength in dealing with outliers without major complications. Decision trees are also effective in tasks involving regression.

One of the advantages of decision trees is their interpretability. They visually represent decision-making processes clearly through their structure; each leaf node signifies a decision or classification branches show potential outcomes and internal nodes indicate decision points based on features [28]. This transparency makes it easy for both experts and non-experts to understand how decisions are reached and the reasoning

behind them. Decision trees are user-friendly as they can process types of data including numerical inputs without requiring extensive preprocessing like normalisation or scaling. This simplifies the preparation process and saves time by dividing the feature space decision trees excel at capturing linear relationships which can improve accuracy in complex data scenarios. Furthermore, decision trees are known for their training and prediction capabilities using algorithms to swiftly determine the best splits.

On the other hand, decision trees come with their set of limitations. They are sensitive to changes in data which can result in tree structures and create instability in the model. When the tree becomes overly complex it may overfit the training data by capturing fluctuations of the true data patterns leading to poor performance on new data [28]. Decision trees typically split along feature space axes, which may not always be optimal for types of data distributions especially when dealing with correlated features or situations where diagonal splits would be more effective. Furthermore, they might show a bias towards features with categories while potentially overlooking attributes with fewer categories. In tasks, decision trees may not perform as effectively as advanced models like ensemble methods because of their precise data partitioning which could hinder generalisation.

In summary despite these limitations, decision trees remain an interpretable tool in machine learning for regression and classification tasks. They offer simplicity, speed and versatility by handling data types and missing information. However, their tendency to overfit and potential challenges in scenarios can be alleviated by using methods to enhance accuracy and robustness.

Random Forest

By using the results from decision trees, the Random Forest technique (a used ensemble learning approach) [7] in machine learning boosts anticipated precision and consistency in both categorisation and prediction tasks. It is seen as one of the algorithms in this domain it is known for its versatility and capacity to manage intricate data. Random Forest proves to be a flexible algorithm across machine learning scenarios. Numerous data experts and machine learning professionals opt for it due to its amalgamation of decision trees, with randomisation thereby enhancing accuracy and robustness.

Random Forest outperforms a single decision tree by merging several trees, offering superior generalisation and accuracy. It is strong against overfitting due to the variety introduced by bagging and random feature selection.[29] Random Forest ranks features according to their impact in the model, providing insights into the most influential aspects. It efficiently handles missing data, often making imputation unnecessary and shows versatility across various tasks like feature selection, regression, and classification. This method can effectively manage both continuous and categorical variables.

Random Forest can be challenging to interpret and explain due to its complexity especially when compared to a decision tree [29]. Dealing with datasets can also pose challenges as training multiple trees can be time consuming and costly. Achieving performance with Random Forest requires tuning of various hyperparameters, including the number of trees, maximum tree depth and the selection of features at each split.

In summary Random Forest stands out as an adaptable method in the realm of machine learning elevating accuracy through ensemble learning techniques. By combining decision trees and incorporating randomness via tactics such as bootstrap sampling and random feature selection it mitigates overfitting issues and enhances generalisation while optimising hyperparameters for Random Forest demands attention to detail. Navigating challenges related to complexity and interpretability its reliability, precision and flexibility in accommodating diverse datasets have cemented its popularity among data experts and machine learning professionals.

Multi-Layer Perceptron

A Multilayer Perceptron (MLP) is a type of network that comprises multiple layers of connected neurones organised in a forward flow pattern. It consists of a layer that takes in data hidden layers for handling information and a final layer for generating forecasts [30]. MLPs compute outputs through forward propagation, where input data passes through the network and is transformed by weights and activation functions to generate predictions. These predictions are compared to the actual targets using a loss function to evaluate the network's performance.

When going through back-propagation the network modifies its weights by considering the calculated error and passing this error back, through the layers to reduce it. This repeated procedure frequently uses optimisation techniques such as descent persists until the network reaches the intended performance level. MLPs, known for their versatility are frequently utsed in learning for tasks like regression and classification. They are good at recognising patterns in data, however they may suffer from overfitting particularly when dealing with networks and scant training data. To solve these issues and enhance MLP performance methods like regularisation and hyperparameter optimisation come into play.

One of the benefits of this is its flexibility allowing it to be used with types of input data. MLP can be applied in scenarios as it has the ability to adapt well to unseen data after effective training. Its flexibility enables it to effectively model performance in tasks and capture nonlinear relationships between inputs and outputs. There are, however, some downsides to using MLPs [30]. For example they can behave like a "box model," making it hard to understand how the network makes decisions or predictions. Another issue is the risk of overfitting especially if the model is very complex or lacks training data. Training an MLP can also be computationally intensive especially when dealing with datasets or deep networks. Additionally tuning MLP hyperparameters is necessary for performance. Despite these challenges MLPs can be a tool, in machine learning with a range of potential applications if used thoughtfully and diligently.

To summarise Multilayer Perceptron stands out as a versatile category of networks that excel at recognising complex patterns in data. They are ideal for machine learning assignments thanks to their ability to capture linear connections and autonomously acquire features. Despite some drawbacks, such as the risk of overfitting and the need for hyperparameter tuning, MLP's remain a choice in the realm of machine learning due to their effectiveness and scalability.

Balanced Random Forest

Balanced Random Forest method offers advantages in enhancing the performance of the minority class. By ensuring weight for both classes and balancing the representation of majority and minority classes in each decision trees training set it enhances predictions for the minority class [31]. This approach mitigates bias towards the majority class resulting in fair models for imbalanced data. Similar to standard Random Forest BRF uses trees to enhance generalisation and resilience against data noise. Its flexibility across data types and ability to fine tune settings for scenarios make it a favourable and versatile option for a variety of applications.

However, Balanced Random Forest encounters challenges like handling datasets can prolong the training time due to the need to balance the data for each decision tree as well as adjusting parameters to achieve this balance makes this method more complex compared to Random Forest. Furthermore, choosing parameters and balancing methods such as down sampling could result in overfitting issues especially when dealing with limited training data.

Cross Validation

In machine learning and data science cross validation assesses how well a model can handle unseen data. It helps in measuring the effectiveness of models and serves as a defence against issues like overfitting and

under-fitting. The main goal of cross validation is to estimate how well a model performs on data by dividing the dataset into subsets or "folds."[32] This process involves training and testing the model on combinations of these folds. One used technique is k cross validation, where the data is split into k equal sized folds for training and testing purposes. The model is trained on k 1 folds in each iteration, then evaluated on the remaining fold to determine its performance. Another approach to evaluating performance is leave one out validation (LOOCV) [32]which involves testing each data point once while using all points for training purposes.

This technique can be resource intensive due to its demands. To offer flexibility in evaluating model robustness leave p out cross-validation reserves p samples for testing while utilising the rest of the data for training purposes. When it comes to time series cross-validation this method is designed specifically for time-related data to ensure that information from the past does not influence the present. To assess the model effectively various performance metrics such as error (for regression models) accuracy, precision, recall and F1 score are calculated in each iteration and then averaged out. Although cross-validation may take up some time its thorough approach offers an evaluation of model performance and can handle large datasets and complex models efficiently.

Cross validation plays a role in minimising both overfitting and under-fitting [32] by testing the model on data subsets repeatedly. This approach helps detect and address these issues effectively. It offers an evaluation of the models performance by averaging results from trials giving a more accurate prediction of how well the model will perform on new and unseen data. By using training and testing sets cross validation provides an assessment of the models performance, which is especially useful for small datasets as it optimises data usage for training and testing purposes thereby improving the models reliability in scenarios with limited data.

Ethics

There is a concern about the risk of data exposure, where details from the test dataset could indirectly impact the training dataset if not managed correctly and it could pose challenges with correlated data necessitating tailored methods for datasets with dependencies such as time series or geographical information. To sum up, cross-validation is a method for building efficient machine learning models. By offering an unbiased assessment of model efficiency it assists data analysts, in making informed choices regarding model selection and parameter optimisation. While implementing cross-validation may demand resources the advantages of establishing reliable and high-performing models far exceed any potential drawbacks.

Early Stopping

Early stopping is a method used to prevent overfitting when training a model, in networks or models like gradient boosting machines that require iterative training [33]. This approach involves evaluating the model’s performance on a validation set during the training process and halting the training if the models performance begins to decline or ceases to improve. To apply early stopping, a portion of the training data is set aside as a validation set. Throughout the training phase, the model is assessed against this validation set at the conclusion of each epoch If there is progress in performance on the validation as the set training continues but if there is no improvement after a specified number of epochs the training process is halted. In some cases, it is common for the model parameters to be reset to those that exhibited performance on the validation set.

This approach ensures that the model does not just learn from the noise in the training data allowing it to perform better on data. Early stopping also makes the training time more efficient by cutting out training cycles that do not boost model performance.[33] It also eliminates the need to preset the number of

training cycles by using validation set performance to determine when to stop training. However, the effectiveness of early stopping relies on the validation set used. If the validation set does not accurately mirror the data distribution it can result in delayed stopping impacting the model's performance. This method calls for adjustment of the 'patience' parameter, which is responsible for how many epochs to wait without improvement before halting. Setting this parameter low might end training prematurely while setting it high may not effectively prevent overfitting. Fluctuations in a validation set performance can also complicate deciding when to stop training and see if there is an improvement or just a performance dip.

SMOTE

In machine learning, if in the dataset there is an imbalance occurs when one class has more instances compared to the other categories. This issue can be tackled by utilising the SMOTE (Synthetic Minority Over Sampling Technique) approach. SMOTE works by creating instances of the minority class by interpolating between existing minority class samples aiming to balance out the classes [33]. It identifies the k neighbours for each minority class sample (, with k=5) selects a neighbour and generates a new artificial example at a random point in the feature space, between the two samples. This process is repeated for all minority class samples until there is a balance. SMOTE is a very useful method when looking at datasets which might have a large imbalance as it helps to reduce bias, furthermore, it helps to reduce the integrity of the dataset as one may delete the majority class to make it balanced hence a large reduction in the dataset. SMOTE can be used in many machine-learning models. However, again there is a problem of overfitting as well as problems such as outliers which SMOTE cannot detect or disregard.

One-Hot Encoding

In machine learning, one hot encoding serves as a technique to transform data into a binary form. Every distinct category, within a variable, is allocated its binary column where a value of 1 signifies its existence and 0 denotes its absence. This approach enhances the precision of data analysis and modelling by translating data into a structure that machine learning algorithms can process easily.

Accuracy

In the machine learning section accuracy was a measure used to evaluate how models performed in different classification tasks. It was a metric for assessing model performance during online evaluations. Accuracy was widely recognised as an aspect to understand in the field of machine learning which contributed to its appeal. Calculating AI accuracy involved determining the ratio of predictions to the number of predictions made across all categories. Typically expressed on a scale from 0 to 1 or as a percentage an accuracy score of zero indicated misclassification by the classifier while a score of one or 100% signalled labelling. The foundation for evaluating machine learning encompassed four components: positives (TP) true negatives (TN) false positives (FP) and false negatives (FN). The formula for computing accuracy was defined as TP + TN / TP + TN + FN + FP [34]. Accuracy played a role by providing insights, into model performance and errors with error being linked to accuracy.

Precision

In the research precision was a measure often utilised by researchers working with curated datasets to focus on refining algorithmic techniques. Precision represented the ratio of positives to all predictions providing an insight into the accuracy of data points [35]. While precision proved effective in scenarios where positive predictions had consequences it tended to overlook negatives resulting in a loss of data. It played a role in evaluating the dependability of machine learning models. A model with recall and low precision could accurately identify samples but might also misclassify negative samples.

Recall

Recall estimated a model's capacity to accurately recognise positives (TP/(TP + FN))[36] indicating its ability to pinpoint data. It proved beneficial in addressing issues related to imbalanced classes and situations where missing positives incurred costs. However, recall did not consider the impacts of positives.

F1 Score

The F1 score alongside metrics like recall and precision was employed for evaluating tasks involving classes. [37] and derived from the confusion matrix. Operated in such a way that enhancing one metric would typically lead to a reduction in another striking a delicate balance between recall and precision. Hence it was practised assessing them based on their average known as the F1 score often described as "accuracy for classes."

AUC

AUC stands for the area under the ROC curve providing an assessment of performance across classification thresholds. Its significance lies in measuring prediction accuracy rather than their exact values (scale-invariant) and being unaffected [38], by classification thresholds (threshold thus evaluating prediction quality regardless of threshold selection. However, there were scenarios where these benefits might not be ideal like AUC may not have signalled when calibrated probability outputs were required.

Implementation and Experimentation

In this section, we discuss how we implement the data processing and machine learning techniques. We begin by tidying up the data, such as converting age from days to years and eliminating duplicates and missing values to enhance the dataset's accuracy. Following that we proceed to feature engineering, where we compute BMI based on height and weight and apply one hot encoding which was extremely uncommon in any of the research papers found, for variables like chest pain categories and thalassemia. Subsequently, models using algorithms such as MLP, Decision Trees, Logistic Regression, XGBoost and Balanced Random Forest were used as they perform well with medical data and they perform well according to the research papers. Especially machine learning models such as MLP and Logistic Regression and look at models which were not very common in the research papers such as Random Forest. To assess these models' performance employed cross-validation to ensure their effectiveness and measure their accuracy, precision, f1, recall and AUC metrics.

Data Exploration

```
import pandas as pd # loads the dataaset which saved in the same folder as this jupyter ipynb file

df = pd.read_csv('C_disease.csv')

print(df)  # prints out the the first 5 rows of each column
```

Figure 3

```
import pandas as pd

df = pd.read_csv('h_disease.csv')

print(df)
```

Figure 4

```
import pandas as pd

df = pd.read_csv('heart_RP.csv')

print(df)
```

Figure 5

```
       index     id  age  gender  height  weight  ap_hi  ap_lo  cholesterol
0          0      0   50       2     168    62.0    110     80            1
1          1      1   55       1     156    85.0    140     90            3
2          2      2   51       1     165    64.0    130     70            3
3          3      3   48       2     169    82.0    150    100            1
4          4      4   47       1     156    56.0    100     60            1
...      ...    ...  ...     ...     ...     ...    ...    ...          ...
69995  69995  99993   52       2     168    76.0    120     80            1
69996  69996  99995   61       1     158   126.0    140     90            2
69997  69997  99996   52       2     183   105.0    180     90            3
69998  69998  99998   61       1     163    72.0    135     80            1
69999  69999  99999   56       1     170    72.0    120     80            2

       gluc  smoke  alco  active  cardio
0         1      0     0       1       0
1         1      0     0       1       1
2         1      0     0       0       1
3         1      0     0       1       1
4         1      0     0       0       0
...     ...    ...   ...     ...     ...
69995     1      1     0       1       0
69996     2      0     0       1       1
69997     1      0     1       0       1
69998     2      0     0       0       1
69999     1      0     0       1       0

[70000 rows x 14 columns]
```

Figure 6 : Output of Figure 3

A shown in Figure 3, 4, 5 these pieces of code imported a dataset called C_disease.csv, h_disease.csv and heart_RP.csv , into a DataFrame (df) using pandas, which is a data handling tool, in Python after that it displays the DataFrame. This allowed to see if the dataset was correctly being uploaded onto Jupyter. Figure 6 as shown above showed the first five rows and the last five rows of the dataset. The dataset in Figure 3 had 70,000 rows and 14 columns, where each row represents an individual's medical measurements and lifestyle choices.

```
import pandas as pd

df = pd.read_csv('C_disease.csv')
 # checks the dataset for any duplicated values
duplicated_rows = df.duplicated()
df = duplicated_rows.sum()  # calculates the total sum of the number of duplicated values

print(f'There are {df} duplicated rows.')
```

```
There are 0 duplicated rows.
```

Figure 7 : Duplicated Rows

```
import pandas as pd

df = pd.read_csv('h_disease.csv')

duplicated_rows = df.duplicated()
df = duplicated_rows.sum()                # prints the number of duplicated rows

print(f'There are {df} duplicated rows.')
```

```
There are 23899 duplicated rows.
```

Figure 8: Num of Duplicated Rows

In Figure 7 the code reloads the dataset and searches for any duplicated rows. It uses the duplicated () function [39] to detect duplicates. Then consolidates them to determine a count. This is a very important step in data cleansing to maintain accuracy in the analysis by removing repeated records as this helps the machine learning models output better results due to the training and testing which occurs. The output as shown in Figure 7 indicates the count of duplicated rows present, in the dataset which was 0. However, in Figure 8 there were 23899 duplicated rows which are removed in the preprocessing section.

```
import pandas as pd

df = pd.read_csv('C_disease.csv')

print(df.isna().any()) # prints if there is any null values in the columns
```

```
index          False
id             False
age            False
gender         False
height         False
weight         False
ap_hi          False
ap_lo          False
cholesterol    False
gluc           False
smoke          False
alco           False
active         False
cardio         False
dtype: bool
```

Figure 9: Missing Values

The isna().any() function is used in Figure 9 is used to detect any missing data in all columns. This function provides a false value for each column with true signifying the presence of missing data and false indicating no missing values. If there were any missing values it would have impacted the machine learning models further on the line. All three datasets had zero missing data which implied that the dataset may have already been cleansed before put on Kaggle where all three datasets where found.

```
import pandas as pd

df = pd.read_csv('C_disease.csv')

numof0 = df['cardio'].value_counts().get(0, 0)  # counts the number of values
numof1 = df['cardio'].value_counts().get(1, 0)  # counts the number of values

print("Number of 0s:", numof0)
print("Number of 1s:", numof1)
```

```
Number of 0s: 35021
Number of 1s: 34979
```

Figure 10: Number of Instances with heart issue and without

```
import pandas as pd

df = pd.read_csv('h_disease.csv')

numof0 = df['HeartDiseaseorAttack'].value_counts().get(0, 0)  # uses .value.count function
numof1 = df['HeartDiseaseorAttack'].value_counts().get(1, 0)

print("Number of 0s:", numof0)     # prints out the number of patients with HeartDiseaseorAttack
print("Number of 1s:", numof1)      # prints out the number of patients with HeartDiseaseorAttack
```

```
Number of 0s: 229773
Number of 1s: 23888
```

Figure 11: Number of Instances with heart issue and without

In Figure 10 it tallied the number of zeros and ones, in the 'cardio' column of the dataset potentially indicating a classification goal as having or not having heart disease. It is important to grasp the class distribution to evaluate evenness and guide modelling choices. This was done in order to see whether SMOTE would be needed to be implemented but as shown in Figure 10 the number of instances with the value of zero was very similar to the value of ones. but the ones with the dataset shown in Figure 11 there was a large difference between the patients with the condition and those without the condition hence why SMOTE was used later on in order to balance the dataset.

```
import pandas as pd

df = pd.read_csv('h_disease.csv')

print(df.mean(numeric_only=True))  # prints the means of each column
```

```
HeartDiseaseorAttack     0.094173
HighBP                   0.428990
HighChol                 0.424113
CholCheck                0.962667
BMI                     28.382475
Smoker                   0.443186
Stroke                   0.040570
Diabetes                 0.296904
PhysActivity             0.756577
Fruits                   0.634264
Veggies                  0.811437
HvyAlcoholConsump        0.056201
AnyHealthcare            0.951049
NoDocbcCost              0.084164
GenHlth                  2.511379
MentHlth                 3.184778
PhysHlth                 4.242028
DiffWalk                 0.168221
Sex                      0.440348
Age                      8.032197
Education                5.050461
Income                   6.054052
dtype: float64
```

Figure 12: Mean of Dataset Columns

In Figure 12 the code uses the pandas library to conduct an assessment of the dataset 'h_disease.csv'. By employing the df.mean () function on the DataFrame it computes the value for each numerical column. This step is crucial as it aids in grasping the tendency of each feature distribution in the dataset a key aspect of exploratory data analysis. The output is a list of mean values for each numerical column in the dataset. These columns represented various medical or demographic variables associated with heart disease, such as cholesterol levels. The mean age of individuals in the dataset is 8.032197 years, which gave an insight into the dataset itself as 8 is usually a very small number suggesting the age column might be encoded. Including these averages helped to provide a quick overview of the dataset's characteristics. Mean, median and mode were tested in all three of the datasets to give hindsight such as looking into box-plots.

```
import pandas as pd

df = pd.read_csv('C_disease.csv')

female_smokers = df[(df['gender'] == 1) & (df['smoke'] == 1)].shape[0]   # prints out the number of female smokers

male_smokers = df[(df['gender'] == 2) & (df['smoke'] == 1)].shape[0]

print("Number of female smokers:", female_smokers)
print("Number of male smokers:", male_smokers) # prints out the number of male smokers
```

```
Number of female smokers: 813
Number of male smokers: 5356
```

Figure 13: Male and Female Smokers

```
import pandas as pd

df = pd.read_csv('C_disease.csv')

female_drinkers = df[(df['gender'] == 1) & (df['alco'] == 1)].shape[0]

male_drinkers = df[(df['gender'] == 2) & (df['alco'] == 1)].shape[0]

print("Number of female drinkers:", female_drinkers)   # prints out the number of female drinkers/alcoholics
print("Number of male drinkers:", male_drinkers) # prints out the number of male drinkers/alcoholics
```

```
Number of female drinkers: 1161
Number of male drinkers: 2603
```

Figure 14: Male and Female Alcohol Drinkers

Figure 13 shows how many men and women smoke in the dataset. Females are represented as 1 and males as 2 while smokers are labelled as 1. The analysis revealed 813 female smokers and 5356 male smokers indicating a higher smoking rate, among males, in this group. When it comes to understanding why people develop diseases or heart conditions smoking plays a role. Smoking is responsible for around 8 million deaths with one in every five smoking-related deaths linked to heart disease[40]. The chemicals found in cigarettes and tobacco products can lead to thickening of artery walls and therefore impairs blood flow. As well as this, carbon monoxide from smoking reduces oxygen levels in the body and unlike oxygen, carbon monoxide binds strongly with haemoglobin in blood cells resulting in decreased oxygen saturation levels that should ideally be at 95%. This decrease in oxygen supply weakens the heart muscles, thickens the viscosity of blood making efficient oxygen transport throughout the body and vital organs such as the heart more difficult.

Figure 14 showed the counts of individuals who consume alcohol while also split by gender. According to the dataset, there are 1161 female drinkers and 2623 male drinkers. Alcohol consumption is another factor linked to heart disease. Heavy drinking can have an impact on arteries and heart function. Alcohol intake can lead to effects like cardiomyopathy, arrhythmias, stroke and obesity [41]. Excessive alcohol consumption may weaken the heart muscles hindering blood circulation and possibly causing heart failure. Arrhythmias can interfere with the coordination between the chambers (atria) and lower chambers (ventricles) of the heart. Additionally, obesity is associated with heart disease as alcoholic drinks are high in calories. The combination of excessive drinking can lead to heart attacks and other cardiovascular issues and it is known that alcohol directly affects the heart by making the cells highly toxic. In addition to known factors contributing to disease risk, there are measures such as including fruits and vegetables in your diet to lower the risk of getting such diseases. Many fruits and vegetables contain nutrients like vitamin C, which act as antioxidants in our bodies. These antioxidants help slow down or prevent the buildup of plaque from cholesterol in our arteries, in turn reducing the risk of heart disease over time.

Data Preprocessing and Cleaning

```
import pandas as pd
import numpy as np

df = pd.read_csv('C_disease.csv')
df['age'] = df['age'] // 365   #round down age from days to years


print(df)
df.to_csv('C_disease.csv', index=False) #saves the changes
```

Figure 15: Conversion of Numbers

The code shown in Figure 15 converts the 'age' feature from days to years in the 'c_disease.csv' dataset to make the data more interpretable. It uses integer division to ensure that the age is represented as whole years as if it were written in decimals it would have had a negative effect on the machine learning models. In Figure 15 the DataFrame is printed after it adjusts the 'age' column to show ages in years making the dataset easier to understand and to read which is really important in data science. Next, the CSV file was overwritten, with the DataFrame to accurately reflect the updated age information in the dataset.

```
import pandas as pd

df = pd.read_csv('C_disease.csv')

df['height'] = df['height'] / 100
df['BMI'] = df['weight'] / (df['height'] ** 2)
df['BMI'] = df['BMI'].round(1)              # calculates the bmi by using the height and weight in the dataset

print(df.head())


df.to_csv('updated1_dataset.csv', index=False)  #saves it to a new file
```

Figure 16: BMI Calculation

In Figure 16 the program computes the Body Mass Index (BMI) for every person, in the dataset. Rounds off the outcome to one decimal point and it converts the 'height' from centimetres to meters by dividing by 100 since BMI is determined using meters.

```
import pandas as pd

df = pd.read_csv('h_disease.csv')

df_duplicatedrow = df.drop_duplicates()  # finds all the duplicated values and removes them using the df.drop function

print(df_duplicatedrow)

df_duplicatedrow.to_csv('updated_dataset2.csv', index=False)   # is saved to a new file
```

Figure 17: Removal of Duplicated Rows

```
import pandas as pd

df = pd.read_csv('updated_dataset2.csv')

duplicated_rows = df.duplicated()
df = duplicated_rows.sum()      # prints the number of duplicated rows

print(f'There are {df} duplicated rows.')
```

```
There are 0 duplicated rows.
```

Figure 18: Sum of Duplicated Rows

In Figure 17 and 18 the code found all the duplicated rows in the dataset and removed them. This is evident in Figure 8 where there were 23899 duplicated rows. Later Figure 18 showed that there were 0 duplicated rows. Removal of duplicated rows is extremely important not only for a more accurate prediction but the values in the dataset should be as unique as possible. Another reason is to reduce potential overfitting as when the model is training or learning it may learn the repeated data making the model less reliable or could even make it biased.

```
import pandas as pd

df = pd.read_csv('heart_RP.csv')
df_end = pd.get_dummies(df, columns=['cp', 'slope', 'restecg', 'thal'], prefix=['cp', 'slope', 'restecg', 'thal'])
# one hot encoding of 4 columns which has multiple values
df_end.to_csv('updated3_dataset.csv', index=False)
print(df_end)
```

```
     age  sex  trestbps  chol  fbs  thalach  exang  oldpeak  ca  condition  \
0     69    1       160   234    1      131      0      0.1   1          0
1     69    0       140   239    0      151      0      1.8   2          0
2     66    0       150   226    0      114      0      2.6   0          0
3     65    1       138   282    1      174      0      1.4   1          1
4     64    1       110   211    0      144      1      1.8   0          0
..   ...  ...       ...   ...  ...      ...    ...      ...  ..        ...
292   40    1       152   223    0      181      0      0.0   0          1
293   39    1       118   219    0      140      0      1.2   0          1
294   35    1       120   198    0      130      1      1.6   0          1
295   35    0       138   183    0      182      0      1.4   0          0
296   35    1       126   282    0      156      1      0.0   0          1

     ...  cp_3  slope_0  slope_1  slope_2  restecg_0  restecg_1  restecg_2  \
0    ...     0        0        1        0          0          0          1
1    ...     0        1        0        0          1          0          0
2    ...     0        0        0        1          1          0          0
3    ...     0        0        1        0          0          0          1
```

Figure 19: One-Hot Encoding Experimentation

The code in Figure 19 performs one-hot encoding [42] on categorical variables in the dataset, which is necessary for preparing the data for many machine-learning algorithms that require numerical input. There were few columns such as thal: A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversible defect), cp: Chest pain type (value ranging from 0 to 3), restecg: Resting electrocardiographic results (values 0,1,2), which all had multiple values of three each. In 'thal' column the numbers 3, 6 and 7 represent forms of thalassemia without any ranking implying one number may be better than the other. One hot encoding eliminates any apparent classification that a machine learning model could deduce from the numbers. By one-hot encoding the 'thal' column as well as the other three columns, each type of thalassemia becomes its own feature in the dataset, allowing models to treat each type independently. This has a positive effect when it comes to the machine learning model.

Machine Learning Modelling

```
continuous_cols = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']

# Standardises only the continuous columns
scaler = StandardScaler()
X_train[continuous_cols] = scaler.fit_transform(X_train[continuous_cols])
X_val[continuous_cols] = scaler.transform(X_val[continuous_cols])
X_test[continuous_cols] = scaler.transform(X_test[continuous_cols])
```

Figure 20: Standardisation Experiment

Standardising the columns as shown in Figure 1 of your dataset is done by using the StandardScaler ()[44] to adjust these columns to have an average of zero and a standard deviation of one. The 'age', 'trestbps', 'chol', 'thalach', 'oldpeak' were continuous columns and by applying the standardisation in part of the MLP machine learning model resulted in a large increase in the performance methods. This customised method different from the other standardised methods in the other MLP models scored higher.

```
# early stoppping to prevent overfitting
early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_val, y_val), callbacks=[early_stopping])
```

Figure 21: Early Stopping Method

The code shown in Figure 21 was the process of training the network model, the MLP machine learning model. This included early stopping, a technique used to prevent overfitting. Overfitting happens when the model learns detail and noise from the training data, which can lead to a decrease in performance on new data, such as the validation set. If the result does not improve for five epochs (patience=5) [44][45, training is halted. When the model stops running, the model's weights are rolled back to the state where it had the best validation loss.

```
# cross_validation happens
clf = KerasClassifier(model)
scores = cross_val_score(clf, X_train, y_train, cv=5)
```

Figure 22: Cross-Validation Method

Cross validation assesses the models performance by splitting the training data into k folds with a 5 fold cross validation being employed in this scenario. This method is advantageous as it enhances both the

quantity and diversity of data utilised for training and assessing the model resulting in an evaluation of its performance, on new data. The 'KerasClassifier' function covers the Keras model, enabling to a scikit learn classifier. Through 'cross_val_score' [45] the model's performance is assessed across validation folds showcasing its ability to generalise effectively across diverse dataset subsets.

Results

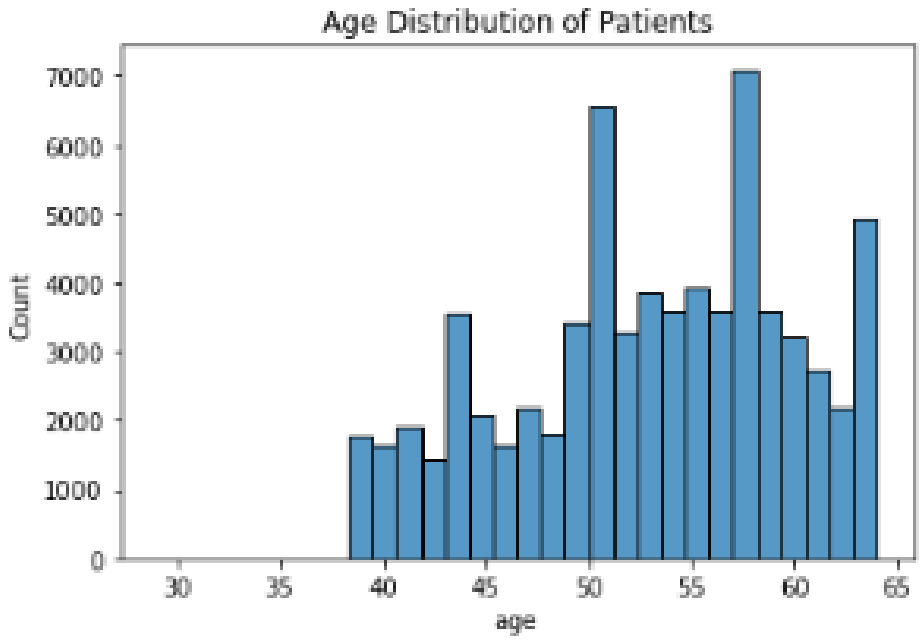

Figure 23: Age Distribution of Patients

in the age distribution of patients shown in Figure 23 there is an increase, in numbers from ages 39 to 56 with patient numbers rising from about 1800 to 4000. After age 56 there is a decline with the count dropping to around 2500. Certain age ranges stand out for deviating from the trend for example at age 44 there is a spike in numbers to 3500 and at the age of 50 there is an increase in numbers reaching around 6500 surpassing what was expected based on the trend. Following this peak another spike occurs at age 57 reaching a peak of 7000 patients before decreasing more. Age 63 also shows a rise to 5000 patients deviating from the expected decrease. The distribution displayed does not match the normal distribution, which usually shows a symmetrical bell curve. Rather the peaks noted at ages 44, 50, 57 and 63 indicate that there could be influences or unique population traits causing these anomalies.

```
At Age Group: 35-40,The Total Cholesterol Cases: 2111
At Age Group: 40-45,The Total Cholesterol Cases: 10253
At Age Group: 45-50,The Total Cholesterol Cases: 14247
At Age Group: 50-55,The Total Cholesterol Cases: 23106
At Age Group: 55-60,The Total Cholesterol Cases: 26262
At Age Group: 60-65,The Total Cholesterol Cases: 19698
```

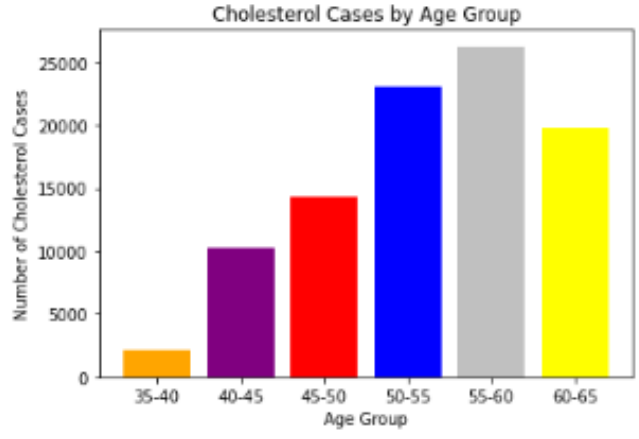

Figure 24: Cholesterol Cases by Age Group Bar Graph

The chart shows how cholesterol problems are spread out among age groups showing a rise in the number of cases as people get older peaking between the ages of 55 and 60. Starting at around 2000 cases for those aged 35-40 there is an increase to around 10000 cases for individuals between 40- 45. This trend continues upward reaching around 13000 cases in the 45-50 age group. The number reaches its point with around 23,000 cases for individuals between the ages of 50 and 55 followed by a peak of around 26,000 cases among those aged 55 to 60. Next, the graph shows a decline with the number of cases dropping to around 20,000 for the age group of 60 to 65. This dataset implies that while cholesterol issues worsen as people age but there is a decrease in reported instances once they surpass the age of 60.

Cholesterol is a type of fat present in cells that helps with food digestion. The World Health Federation stated back in 2008 that 39% of adults worldwide had cholesterol with a quarter of those cases connected to diseases. The buildup of low-density cholesterol in artery walls forms plaques that can block blood flow potentially leading to heart failure. Cholesterol primarily impacts arteries by causing plaque accumulation or inflammation which raises blood pressure and strains the heart. This can result in decreased oxygen supply to the heart muscles contributing to disease.

```
Age Group: <30
 Q1:167.0,  Median:175.0,  Q3:175.0
Age Group: 30-40
 Q1:160.0,  Median:165.0,  Q3:170.0
Age Group: 40-50
 Q1:160.0,  Median:165.0,  Q3:170.0
Age Group: 50-60
 Q1:159.0,  Median:165.0,  Q3:169.0
Age Group: 60+
 Q1:158.0,  Median:164.0,  Q3:169.0
```

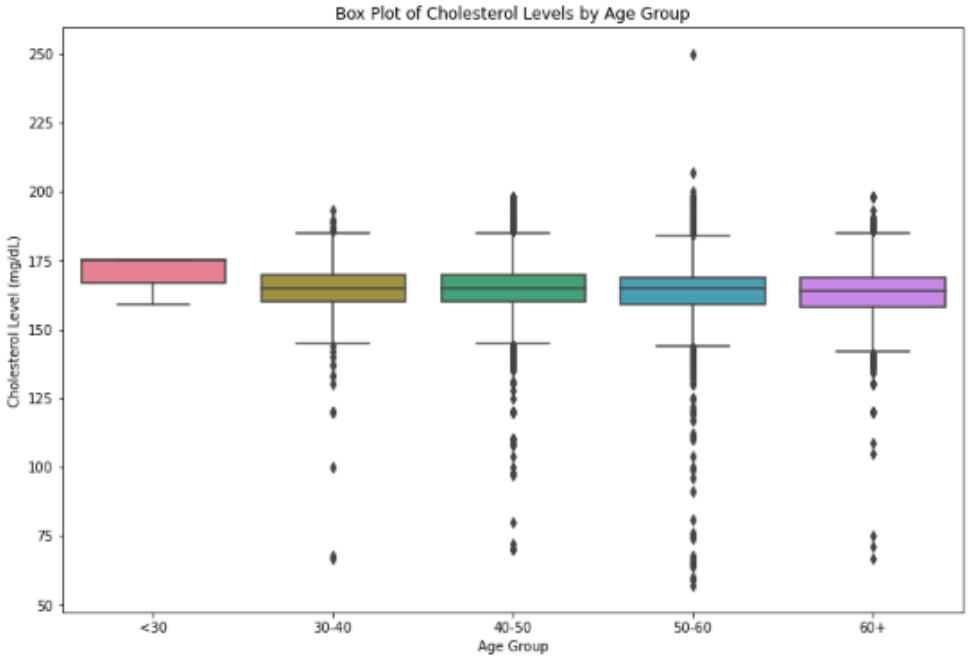

Figure 25: Box Plot of Cholesterol Levels by Age Group

The box plot in Figure 25 shows the changes in cholesterol levels as people age. Generally, individuals under 30 median have a cholesterol level at around 175 mg/dL. As people move from their 30s to 40s, the average level tends to decrease to 165 mg/dL. In the age range of 50-60 the cholesterol levels are quite diverse with a number of individuals exhibiting levels below the average for this group. Individuals aged over 60 have a level lower at around 165 mg/dL showcasing a wide distribution of levels with many falling below this median mark. This indicates that older individuals typically display a spectrum of cholesterol levels making it more frequent to encounter than average levels within the age bracket.

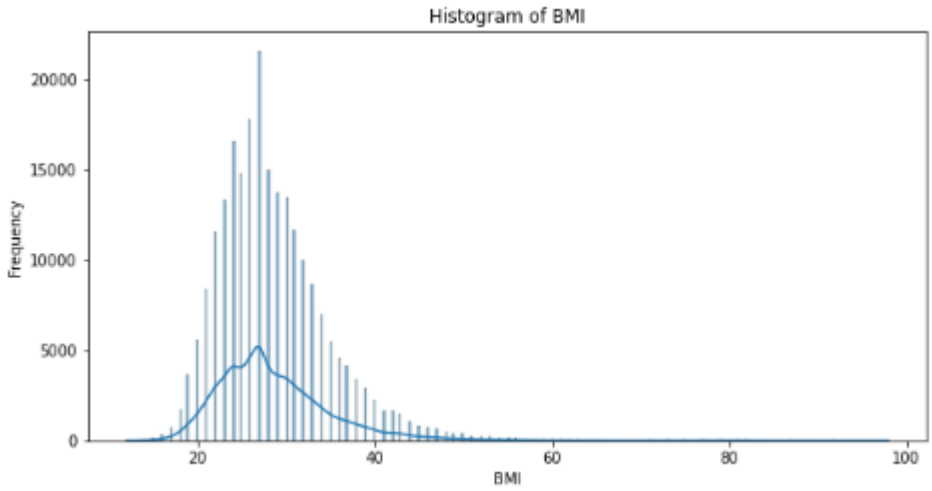

Figure 26: Histogram

In Figure 26, the histogram shows how often different BMI values appear in a group of people according to the instances in the dataset. The shape looks like a bell-shaped curve hinting at a trend towards a spread that is a bit lopsided, showing it is not perfectly even. The most common BMI value is at 25, with 22,000 occurrences. The ascent to this peak is steep, with the frequency rapidly increasing as BMI approaches 25, indicating a high concentration of individuals within this BMI category. Following the peak, the descent is more gradual, showing a slower decline in frequency as BMI values exceed 25. There are also changes in the peak frequency region. These differences might indicate the existence of subgroups within the population with varying BMI distributions or maybe external factors impacting an individual's BMI. Despite these inconsistencies, the average focus is clearly around a BMI of 25.

In Figure 27, from the graph we can clearly see a pattern where the number of people, with heart disease or heart attacks increases as age group codes go up. It starts at an age group code of 1 with a number of around 0.005 and steadily climbs to about 0.245 at age group code 12. The error bars, showing the variation or uncertainty within each age group stay consistently sized throughout all age groups indicating accuracy in measurements across age groups.

The second chart, which displays the prevalence of strokes exhibits a rise as age group codes increase. The error bars slightly expand in size as the age group codes increase suggesting a bit of variation in the age groups. Particularly noticeable is that as the encoded age group codes increased in stroke cases the proportions became more apparent.

The third graph in Figure 27, showed that the number of diabetes cases increased as the age group codes went up until code 11 where it reached a peak of around 0.49. After that, there is a drop by code 13 with

the proportion decreasing to about 0.41. The error bars on the graph remain fairly steady suggesting accuracy across the age groups.

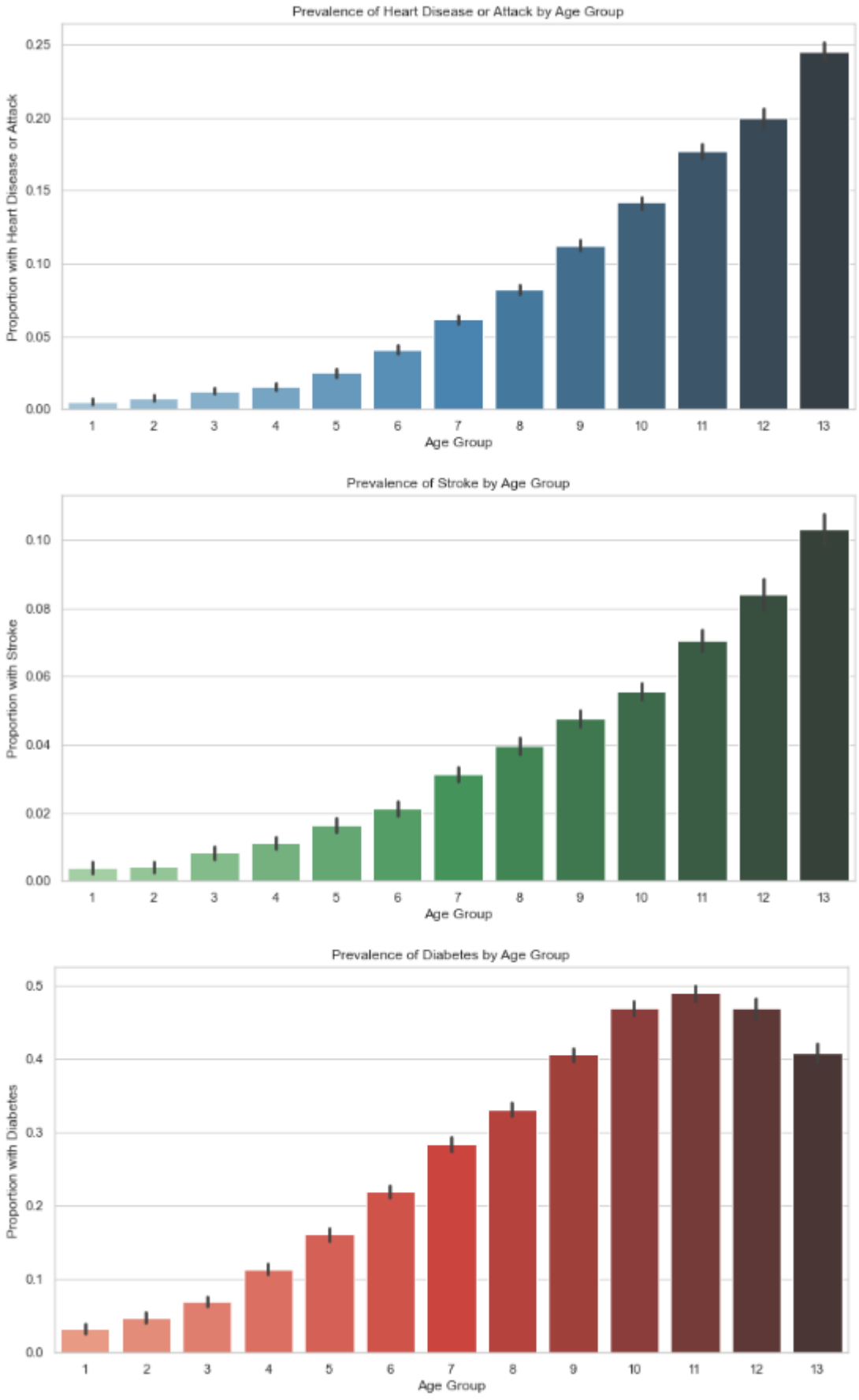

Figure 27: HeartDisease, Stroke and Diabetes Prevalence by Age Group

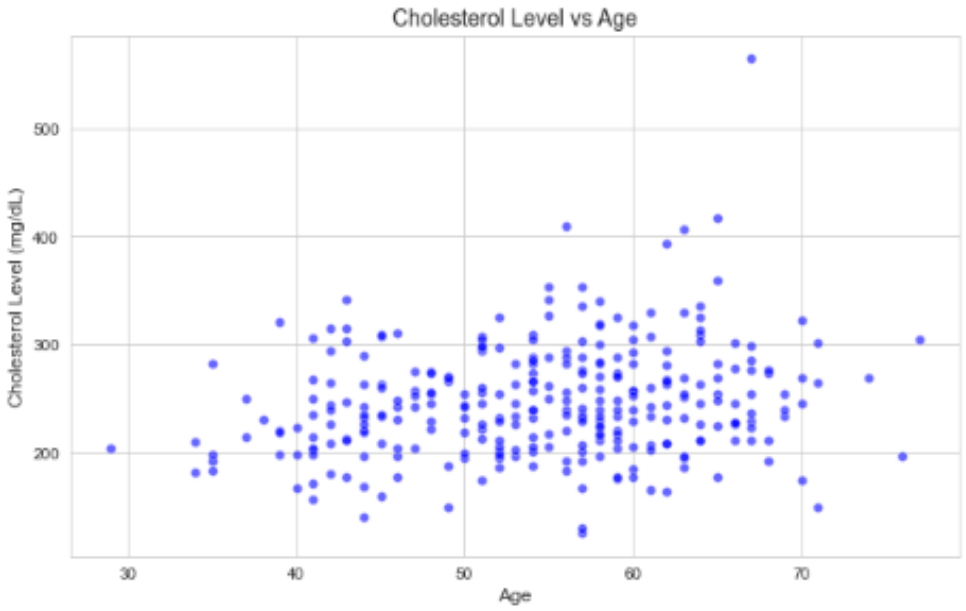

Figure 28: Cholesterol vs Age

The graph illustrates the cholesterol levels observed among age brackets ranging from individuals aged 30 to those over 70. It is clear that there is a range of cholesterol levels across all age groups with no pattern of increase or decrease as individuals age. The majority of cholesterol measurements lie within the 200 to 300 mg/dL range. It seems like there is not a connection between age and cholesterol levels. Rather the levels of cholesterol seem to stay fairly steady of age. However, there are a couple of exceptions among individuals, where some people have much higher or lower cholesterol levels compared to the rest.

One of the standout findings in Figure 29 is the moderately strong positive correlation between the 'condition' variable and factors such as 'cp' (chest pain type), 'thalach' (maximum heart rate achieved), and 'ca' (number of major vessels colored by fluoroscopy), all having correlation values around 0.5. This suggests that these particular attributes are commonly found in patients with heart disease.

Furthermore, factors linked to how the body reacts to activity such as 'oldpeak' (the decrease, in ST segment during exercise compared to rest) and 'exang' (angina triggered by exercise) exhibit a positive connection with the 'condition'. This suggests a link where people who encounter these exercise-induced symptoms during exercise are at a risk of having heart issues.

When it comes to age related discoveries a trend is noticed where 'thalach' and 'age' have a connection. As people age their highest heart rate attained during activity usually decreases, aligning with medical knowledge. On the other hand the 'restecg' (resting results) factor shows a minimal link to heart disease suggesting that based on this data resting ECG outcomes alone may not be a strong indicator of heart issues.

Finally, there is a correlation between 'oldpeak' and the 'slope' of the peak exercise ST segment. This indicates that a reduction in ST depression during exercise is associated with a slope of the recovery phase ST segment. These discoveries offer perspectives on interpreting exercise stress test outcomes in relation, to heart disease diagnosis.

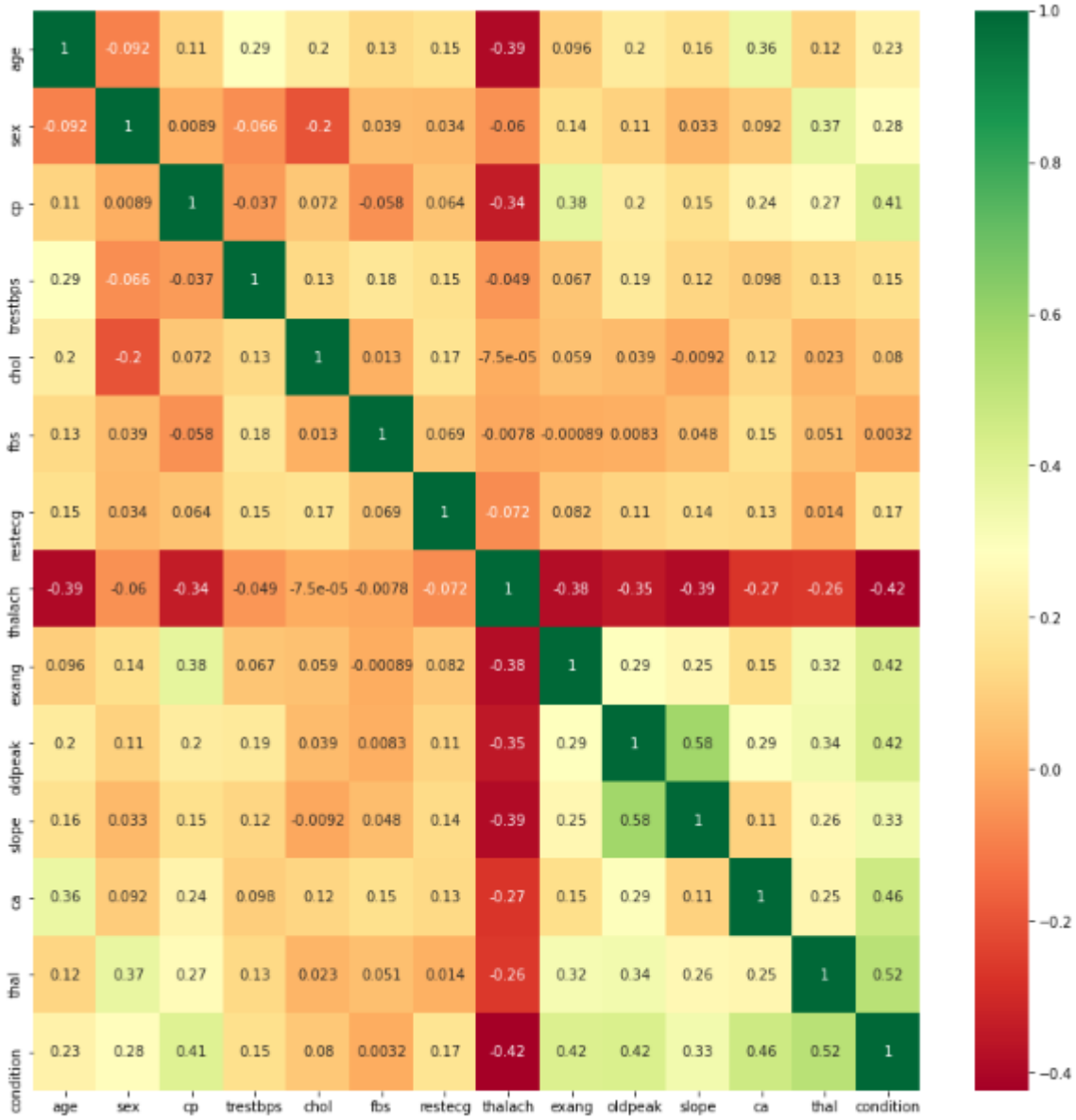

Figure 29: Heatmap of Dataset 3

Table 1: Dataset 1

Model	Accuracy	Precision	Recall	F1 Score	AUC Score
Decision Tree	0.63	0.63	0.63	0.63	0.63
Logistic Regression	0.71	0.74	0.65	0.69	0.71
XGBoost	0.74	0.75	0.71	0.73	0.74
Random Forest	0.72	0.73	0.71	0.72	0.50
MLP	0.72	0.72	0.72	0.72	0.72

In Table 1, the Decision Tree model performance was steady, across metrics scoring 0.63 for accuracy, precision, recall, F1 score and AUC (Area Under the Curve). This indicates a rounded strategy, in predicting classes and minimising false positives although there is still room to enhance the overall accuracy of the model.

The Logistic Regression model outperformed the Decision Tree with higher scores, notably achieving an accuracy of 0.71 and an AUC score to match. Precision at 0.74 indicates a higher reliability in its positive predictions. Nevertheless, the recall at 0.65 suggests that it missed a fair number of positive cases.

XGBoost had the highest scores in all categories except for AUC, where it matched Logistic Regression. An accuracy of 0.74 and precision of 0.75 show its strong predictive capabilities and a recall of 0.71 alongside an F1 score of 0.73 demonstrates its effectiveness in identifying true positives while maintaining the balance between precision and recall.

The Random Forest model displayed a decrease in performance compared to XGBoost and performed similarly to Logistic Regression regarding accuracy and AUC both at 0.72. The precision and recall scores suggest a model although there is a decline in AUC, to 0.50.

the MLP (Multi-Layer Perceptron) classifier generated uniform scores of 0.72 across all metrics, indicating a consistent and balanced classification performance.

Model	Accuracy	Precision	Recall	F1 Score	AUC
Decision Tree	0.83	0.23	0.28	0.25	0.58
Decision Tree with SMOTE	0.76	0.18	0.37	0.24	0.59
Logistic Regression	0.90	0.53	0.13	0.20	0.56
Logistic Regression with SMOTE	0.71	0.22	0.73	0.33	0.72
XGBoost	0.90	0.51	0.12	0.20	0.55
XGBoost with SMOTE	0.72	0.21	0.66	0.32	0.69
Random Forest	0.89	0.40	0.12	0.17	0.54
Balanced Random Forest	0.72	0.24	0.80	0.36	0.75
MLP	0.90	0.73	0.52	0.52	0.52
MLP With Smote	0.77	0.61	0.71	0.62	0.71

The Decision Tree model has an accuracy of 0.83 which was quite high. in spite of that, its precision stands at 0.23 indicating that it often misidentified non cases as cases. The recall rate of 0.28 is also on the side suggesting that it misses a number of actual cases. The F1 score, which strikes a balance between precision

and recall is 0.25 while the AUC score is 0.58 indicating that the model performs better than chance in distinguishing between classes.

Then when SMOTE was applied the Decision Tree model the accuracy decreased to 0.76. The recall rises to 0.37 indicating that it identified true cases while sacrificing overall accuracy. Precision saw a decline and the AUC score slightly improved to 0.59 suggesting an enhancement in the model's capability to differentiate between classes after balancing the dataset.

Logistic Regression model results indicated an accuracy of 0.90 but a limited recall of 0.13 showing that it missed true positive cases. The precision stood at a level of 0.53 suggesting that, over half of its predictions are accurate but the low recall significantly impacted the F1 score bringing it down to 0.20. The AUC score was at 0.56, which was close to random chance. This showed that the model may not be effectively capturing all the complexities in the dataset. With SMOTE the model's accuracy decreased to 0.71 but its recall jumps to 0.73, showing a significant improvement in identifying positive cases. The AUC also increased to 0.72, indicating better model performance in distinguishing between classes.

XGBoost showed a similar pattern to Logistic Regression, with high accuracy of 0.90 but low recall of 0.12 and the precision of 0.51 indicated that about half of the positive predictions made by the model are correct. The AUC score is 0.55, suggesting limited ability in differentiating classes. With SMOTE, the recall improves to 0.66, and the AUC to 0.69, highlighting the benefits of balancing the dataset.

The Random Forest model showed an accuracy at 0.89. The model struggled with recall at 0.12 and had a precision of 0.40 indicated it was not as precise as Logistic Regression or XGBoost. The AUC score was the lowest, standing at 0.54. The Balanced Random Forest addressed class imbalances by enhancing recall to 0.80 and increasing the AUC to 0.75 showing performance in distinguishing between classes.

The Multi-Layer Perceptron (MLP) model consistently delivered results with all scores matching at 0.72 indicating a rounded performance across metrics. When SMOTE was used with the MLP model, the outcome stayed consistent showing that SMOTE did not significantly impact the performance.

Table 3: Dataset 3

Model	Accuracy	Precision	Recall	F1 Score	AUC
Decision Tree	0.79	0.84	0.76	0.80	0.80
Logistic Regression	0.79	0.84	0.76	0.80	0.80
XGBoost	0.83	0.91	0.78	0.84	0.84
Random Forest	0.83	0.91	0.78	0.84	0.84
MLP	0.93	0.92	0.94	0.93	0.94

The Decision Tree model showed an accuracy of 0.79 with precision at 0.84 suggesting a high chance that the positive predictions were accurate. The recall rate was 0.76 indicating an ability of the model to identify positive instances. An F1 score of 0.80 reflected a trade off between precision and recall while an AUC score of 0.80 demonstrated a capacity to distinguish between the different classes.

The results of the Logistic Regression model closely matched those of the Decision Tree model in terms of metrics such as; accuracy, precision, recall, F1 score and AUC. This similarity in performance indicates that both models demonstrated proficiency in dealing with the dataset.

XGBoost and Random Forest models showed performance compared to Decision Tree and Logistic Regression in terms of accuracy of 0.83, precision of 0.91 and AUC of 0.84. Their higher precision suggested that more of the predictions were accurate while the superior AUC indicates an ability to differentiate between classes. Both models had a recall rate of 0.78 than the previous ones and an F1 score of 0.84 indicating a strong balance between precision and recall.

The MLP model showed performance achieving an accuracy of 0.93, which was the highest among all models tested. Its precision was also high at 0.92 indicating the models ability to make predictions. With a recall rate of 0.94 the model effectively identified positive cases. The high F1 score of 0.93 highlighted a balance between precision and recall. Additionally, the AUC score excelled at 0.94 showcasing the models capability to differentiate between classes.

Table 4: Cross-Validation Accuracy with MLP

Dataset	Accuracy
Dataset 1	0.82
Dataset 2	0.90
Dataset 3	0.82

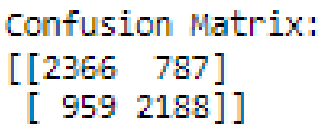

Figure 30: Dataset 1 MLP

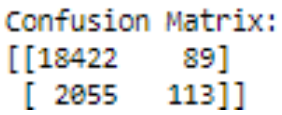

Figure 31: Dataset 2 without Smote MLP

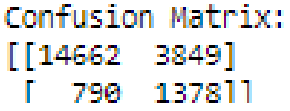

Figure 32: Dataset 2 with SMOTE MLP

```
Confusion Matrix:
[[11  0]
 [ 2 14]]
```

Figure 33: Dataset 3 MLP

Discussion and Analysis

Datasets 1 and 3, both had balanced class distributions and had relatively even performance across various evaluation metrics. This balance allowed the models to learn features from both classes without bias as shown in Figure 10, leading to more general and consistent predictions. In Dataset 1, models like; Decision Tree, Logistic Regression, XGBoost, Random Forest and MLP showed similar performance, with each scoring around the 0.63 to 0.74 range on the accuracy, precision, recall, F1 score, and AUC, indicating a balanced learning as shown in Table 1.

In all the literature reviews there were varying sizes of datasets, with some studies having smaller datasets similar to Dataset 3. [10] using KNN, Naive Bayes and Random Forest with high accuracy levels also suggested a focused and possibly robust approach to model training and validation very similar but using MLP and Logistic Regression to add newness to the study and to see differences of how different preprocessing techniques, even if less complicated, can score a higher score. In the [11] study the Random Forest model demonstrated an accuracy rate of 95.63% showing its ability to effectively handle data patterns. Nonetheless, excluding three columns from the dataset could lead to bias by eliminating predictors of the result that may distort the models' ability of the underlying connections. Different dataset sizes affect model training and performance can be linked to the [10].

At first, Dataset 2 had an imbalance issue as shown by the output of affecting the models' performance until SMOTE was used. The biased dataset initially provided accuracy but sacrificed precision and recall since the models tended to predict the majority class. The Logistic Regression model achieved a high accuracy of 0.90 but had a low recall and F1 scores at 0.13 and 0.20. After implementing SMOTE, recall was notably improved by identifying instances of the minority class causing a decrease in precision due to an increase in false positives.

Logistic Regression with SMOTE saw an increase in recall to 0.73 but a lower precision at 0.22. Dataset 3's smaller size had more effective training of complex models like Multi-Layer Perceptron (MLP). The smaller dataset instances allowed for more detailed pattern recognition by the models as demonstrated in Figure 29 with few clear correlations, thus leading to a higher performance. Dataset 3 achieved the highest scores among all models, with an accuracy of 0.93, precision of 0.92, recall of 0.94, F1 score of 0.93 and AUC of 0.94.

Preprocessing techniques played an important role in enhancing model performance. In Dataset 3, standardisation and one-hot encoding were important in achieving high performance. Standardisation helped to alleviate the risk of certain features taking over the model due to their common occurrences. One-hot encoding ensured that categorical variables were properly represented, allowing models to interpret these features correctly. In the research paper [13] there was use of sophisticated data handling techniques, such as hybrid models.

Even though the hybrid method for MLP used in [MDPI] still scored a lower result of 88.4% accuracy compared to 93%. The decision to incorporate BMI as a feature in Dataset 1 initially was eventually

removed due to a decrease in model performance showing the impact of feature engineering as irrelevant and misleading features can decrease model accuracy.

The MLP model showed best results in Dataset 3, indicating that the dataset was well-suited to leverage the complexity and flexibility of neural networks. The hybrid model (HRFLM) from [8] achieved an 88.4% accuracy, which was fairly high but still lower than the advanced MLP used in the study. In contrast, Datasets 1 and 2 might not have fully benefited from this complexity due to their size or class imbalance. Random Forest and XGBoost models showed robustness in all datasets as also shown in [13] where the use of ensemble methods like Random Forest and AdaboostM1 were used. Dataset 3 proved that careful preprocessing can further enhance the performance of metrics.

The MLP struggled to identify the instances of heart disease in Dataset 2 in Figure 32. Following the implementation of SMOTE the number of positives increased to 3,849. Model's accuracy in detecting cases saw an increase, with 14,662 true positives. MLP model detection leaned heavily on confirming heart disease with 18,422 true positives in Figure 31, but 2,855 actual cases of heart disease and evidently the model struggled with false negatives. Before SMOTE model recognised fewer true positives, 14,662, but SMOTE balanced out the detection better despite a rise in false positives to 3,849. In Dataset 3 Figure 33 the model perfectly identified all without heart disease and this may be due to the reason that the number of instances was largely smaller than the other two datasets. The MLP model aslo only missed 2 actual cases, exhibiting precise detection despite the small sample size.

The model in Table 4 showed an accuracy rate of 82% meaning it correctly diagnoses heart disease in patients 82% of the time. This model gets the diagnosis right 90% of the time which was interesting as Dataset 2 was the most unbalanced dataset and extremely biased in one side. Dataset 3 had the same cross validation accuracy rate as Dataset 1. One factor which may have caused this was the similarity in the type of data Dataset 1 and Dataset 3 had. However, a higher result was expected for Dataset 3 as it scored the highest in performance metrics for MLP.

Conclusion and Future Work

The conclude the study conducted offers perspectives on utilising machine learning models to predict heart disease. Through the use of data preprocessing methods, like SMOTE, one hot encoding and standardisation and cross validation, I managed to improve the models' effectiveness. Employing a range of algorithms such as: MLP, Decision Trees, Logistic Regression, XGBoost, Random Forest and Balanced Random Forest Model, which was unique as it was not mentioned in any of the literature reviews. These models helped give an insight into which model was able to predict heart disease the best.

The addition of SMOTE played a role in handling class imbalance for Dataset 2 where there was a significant enhancement in recall after applying the method. This highlighted the significance of tackling biases in datasets to illustrate the reliability of models. MLP stood out as the best performer for Dataset 3 showed the most promising capabilities of neural networks when datasets were properly implemented and features are carefully chosen.

This study revealed that simpler models might not fully develop the complexity of neural networks unless the data is appropriately pre-processed and feature engineered. The models' varying performances across the three datasets emphasised the need for tailored approaches looking at dataset characteristics to achieve the best results. Furthermore, it was very interesting to see the differences between the dataset which was pre-processed but performed worse than when it was not pre-processed. This was by adding another column such as BMI into Dataset 1, however, I decided to remove it as it was used as a base dataset for comparison due to the very limited research papers on the dataset. Hence allowing all Dataset

1, Dataset 2 and Dataset 3 to be unique as Dataset 2 was imbalanced but had 200,000 + instances and then Dataset 3 which had 303 instances. A deliberate decision was made to refine the datasets to ensure each one offered a unique perspective and contribution to the study of heart disease prediction using machine learning models.

Future work would be to explore using deep learning methods which could enhance the ability to predict heart diseases, in particular, deep neural network structures such as Convolutional Neural Networks (CNN) which might result in even better outcomes. Furthermore, having access to more varied sets of data is also important. When models are trained using a range of information it can enhance their accuracy and make them even more robust, like in the real world where databases can be in the millions or even billions such like the NHS in the UK has around 65 million health records. [46] Also, for future work might be looking at not only heart disease but other areas which fall under the category of precision medicine. An example of this would be further researching topics of cancer as there 'are more than 200 types of cancer' [47].

Reflection

In reviewing the project there were many aspects that stood out in terms of development and obstacles. Notably, an insight into machine learning, especially neural networks and Multi Layer Perceptrons (MLP). This progress was furthered looked into by incorporating advanced methods such as standardisation, hyperparameter tuning and cross validation. These methods played a key role in improving model reliability.

In addition, exploring into a range of research materials highlighted the fact that data preprocessing and data exploration play a huge role when it comes to the employment of the machine learning models. This step showed how important and refining the data is key, to building machine learning models. Furthermore, visualising data was found to be a technique for simplifying outcomes for better results in the performance metrics.

One of the obstacles faced involved preparing Dataset 1. The datasets had constraints in the instances restricting the choices, making it harder to preprocess in the dataset. This problem underscored the balance between data quality and model effectiveness. For machine learning projects to succeed it is essential to highlight the significance of choosing datasets which can be experimented properly and it does not multiple limitations.

While the project mainly focused on tuning machine learning models for outcomes the early data processing stages played a vital role in achieving overall success. Further use of enhancing methods might lead to significant enhancements in model precision and reliability compared to modifications made during or later machine learning phases, which is also significant.

Appendix

GitLab Link

https://csgitlab.reading.ac.uk/gz021457/cardiovascular-disease-prediction-using-data-science.git

Datasets Used

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset?resource=download

https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset

https://www.kaggle.com/datasets/alexteboul/heart-disease-health-indicators-dataset

References

[1] WHO (2022). *Cardiovascular Diseases*. [online] www.who.int. Available at: https://www.who.int/health-topics/cardiovascular-diseases.

[2] Ding, L., Liang, Y., Tan, E.C.K., Hu, Y., Zhang, C., Liu, Y., Xue, F. and Wang, R. (2020). Smoking, heavy drinking, physical inactivity, and obesity among middle-aged and older adults in China: cross-sectional findings from the baseline survey of CHARLS 2011–2012. *BMC Public Health*, 20(1). doi:https://doi.org/10.1186/s12889-020-08625-5.

[3] NHS (2017). *Cardiovascular disease*. [online] nhs.uk. Available at: https://www.nhs.uk/conditions/cardiovascular-disease/#:~:text=age%20%E2%80%93%20CVD%20is%20most%20common.

[4] IBM (2023). *What is Machine Learning?* [online] IBM. Available at: https://www.ibm.com/topics/machine-learning.

[5] www.linkedin.com. (n.d.). *Harnessing the Power of AI and Machine Learning in Modern Data Architectures*. [online] Available at: https://www.linkedin.com/pulse/harnessing-power-ai-machine-learning-modern-data-nicholas-vscbc/

[6] IBM (2023). *What Are Neural Networks?* [online] www.ibm.com. Available at: https://www.ibm.com/topics/neural-networks.

[7] Mohan, S., Thirumalai, C. and Srivastava, G. (2019). Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques. *IEEE Access*, 7, pp.81542–81554. doi:https://doi.org/10.1109/

[8] Bhatt, C.M., Patel, P., Ghetia, T. and Mazzeo, P.L. (2023). Effective Heart Disease Prediction Using Machine Learning Techniques. *Algorithms*, 16(2), p.88. doi:https://doi.org/10.3390/a16020088.

[9] Sabri Elmitwally, N. (n.d.). *Heart Disease Prediction Using Machine Learning Method*. [online] Available at:

https://www.openaccess.bcu.ac.uk/14128/1/Heart_Disease_Prediction_Using_Machine_Learning_Method.pdf.

[10] https://iopscience.iop.org/article/10.1088/1742-6596/2161/1/012013/pdf

[11] https://www.researchgate.net/publication/365887966_Prediction_of_Heart_Disease_Using_Machine_Learning_Algorithms

[12] Hassan, Ch.A. ul, Iqbal, J., Irfan, R., Hussain, S., Algarni, A.D., Bukhari, S.S.H., Alturki, N. and Ullah, S.S. (2022). Effectively Predicting the Presence of Coronary Heart Disease Using Machine Learning Classifiers. *Sensors*, 22(19), p.7227. doi: https://doi.org/10.3390/s22197227.

[13] Ali, M.M., Paul, B.K., Ahmed, K., Bui, F.M., Quinn, J.M.W. and Moni, M.A. (2021). Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison. *Computers in Biology and Medicine*, 136, p.104672. doi: https://doi.org/10.1016/j.compbiomed.2021.104672.

[14] Al-Alshaikh, H.A., P, P., Poonia, R.C., Saudagar, A.K.J., Yadav, M., AlSagri, H.S. and AlSanad, A.A. (2024). Comprehensive evaluation and performance analysis of machine learning in heart disease prediction. *Scientific Reports*, [online] 14(1), p.7819. doi:https://doi.org/10.1038/s41598-024-58489-7.

[15] Simplilearn.com. (n.d.). *Regularization in Machine Learning || Simplilearn*. [online] Available at: https://www.simplilearn.com/tutorials/machine-learning-tutorial/regularization-in-machine-learning#:~:text=There%20are%20two%20main%20types

[16] https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012072/meta

[17] Edgar, T.W. and Manz, D.O. (2017). *Logistic Regression - an overview | ScienceDirect Topics*. [online] www.sciencedirect.com. Available at: https://www.sciencedirect.com/topics/computer-science/logistic-regression.

[18] www.dremio.com. (n.d.). *K-Nearest Neighbors | Dremio*. [online] Available at: https://www.dremio.com/wiki/k-nearest-neighbors/#:~:text=It%20is%20a%20non%2Dparametric

[19] RAJKUMAR, V. (2022). *Heart Disease Prediction using Machine learning*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2022/02/heart-disease-prediction-using-machine-learning-2/#h-train-test-split

[20] Nandal, N., Goel, L. and Tanwar, R. (2022). *Machine learning-based heart attack prediction: A symptomatic heart attack prediction method and exploratory analysis*. [online] f1000research.com. Available at: https://f1000research.com/articles/11-1126.

[21] https://ieeexplore.ieee.org/document/10126733/references#references

[22] Sun, Z., Song, Q., Zhu, X., Sun, H., Xu, B. and Zhou, Y. (2015). A novel ensemble method for classifying imbalanced data. *Pattern Recognition*, 48(5), pp.1623–1637. doi:https://doi.org/10.1016/j.patcog.2014.11.014.

[23] https://www.researchgate.net/publication/325789071_SMOTE_for_Learning_from_Imbalanced_Data_Progress_and_Challenges_Marking_the_15-year_Anniversary

[24] Mayo Clinic. (n.d.). *How high blood pressure can affect the body*. [online] Available at: https://www.mayoclinic.org/diseases-conditions/high-blood-pressure/in-depth/high-blood-pressure/art-20045868#:~:text=High%20blood%20pressure%20can%20narrow.

[25] NHS (2018). *Coronary heart disease - prevention*. [online] nhs.uk. Available at: https://www.nhs.uk/conditions/coronary-heart-disease/prevention/#:~:text=Regular%20exercise%20will%20make%20your.

[26] encord.com. (n.d.). *Logistic Regression: Definition, Use Cases, Implementation*. [online] Available at: https://encord.com/blog/what-is-logistic-regression/.

[27] kharkar, D. (2023). *Unravelling the Power of XGBoost: Boosting Performance with Extreme Gradient Boosting*. [online] Medium. Available at: https://medium.com/@dishantkharkar9/unravelling-the-power-of-xgboost-boosting-performance-with-extreme-gradient-boosting-302e1c00e555.

[28] CORP-MIDS1 (MDS). (n.d.). *Decision Tree*. [online] Available at: https://www.mastersindatascience.org/learning/machine-learning-algorithms/decision-tree/.

[29] Sharma, A. (2020). *Decision Tree vs. Random Forest - Which Algorithm Should you Use?* [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/.

[30] Banoula, M. (2021). *An Overview on Multilayer Perceptron (MLP)*. [online] Simplilearn.com. Available at: https://www.simplilearn.com/tutorials/deep-learning-tutorial/multilayer-perceptron.

[31] Chen, C. and Liaw, A. (n.d.). *Using Random Forest to Learn Imbalanced Data*. [online] Available at: https://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf.

[32] Sharma, A. (2017). *Cross Validation in Machine Learning*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/cross-validation-machine-learning/.

[33] Galli, S. (2023). *Overcoming Class Imbalance with SMOTE: How to Tackle Imbalanced Datasets in Machine Learning*. [online] Train in Data's Blog. Available at: https://www.blog.trainindata.com/overcoming-class-imbalance-with-smote/#:~:text=SMOTE%20is%20a%20type%20of

[34]
Educative. (n.d.). *What is accuracy in machine learning?* [online] Available at: https://www.educative.io/answers/what-is-accuracy-in-machine-learning.

[35] Deepchecks. (n.d.). *What is Precision in Machine Learning*. [online] Available at: https://deepchecks.com/glossary/precision-in-machine-learning/.

[36] www.evidentlyai.com. (n.d.). *Accuracy vs. precision vs. recall in machine learning: what's the difference?* [online] Available at: https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall.

[37] Arize AI. (n.d.). *Understanding and Applying F1 Score: A Deep Dive with Hands-On Coding*. [online] Available at: https://arize.com/blog-course/f1-score/.

[38] Google (2019). *Classification: ROC Curve and AUC | Machine Learning Crash Course*. [online] Google Developers. Available at: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

[39] Mage. (n.d.). *Data Cleaning - Remove duplicates*. [online] Available at: https://www.mage.ai/blog/data-cleaning-remove-duplicates.

[40] CDC (2014). *SMOKING AND CARDIOVASCULAR DISEASE*. [online] *https://www.cdc.gov/tobacco/sgr/50th-anniversary/pdfs/fs_smoking_cvd_508.pdf*. CDC. Available at: https://www.cdc.gov/tobacco/sgr/50th-anniversary/pdfs/fs_smoking_CVD_508.pdf.

[41] www.bhf.org.uk. (n.d.). *Effects of alcohol on your heart*. [online] Available at: https://www.bhf.org.uk/informationsupport/heart-matters-magazine/medical/effects-of-alcohol-on-your-heart#:~:text=Drinking%20alcohol%20to%20excess%20can.

[42] Sharma, N. (2023). *How to do One Hot Encoding? Transform Your Categorical Data!* [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2023/12/how-to-do-one-hot-encoding/#:~:text=One%2Dhot%20encoding%20is%20a.

[43] Scikit-Learn (2019). *sklearn.preprocessing.StandardScaler — scikit-learn 0.21.2 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

[44] Brownlee, J. (2018). *Use Early Stopping to Halt the Training of Neural Networks At the Right Time*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/.

[45] www.w3schools.com. (n.d.). *Python Machine Learning - Cross Validation*. [online] Available at: https://www.w3schools.com/python/python_ml_cross_validation.asp.

[46] Kollewe, J. (2019). *NHS data is worth billions – but who should have access to it?* [online] the Guardian. Available at: https://www.theguardian.com/society/2019/jun/10/nhs-data-google-alphabet-tech-drug-firms#:~:text=The%20NHS%20database%20holds%20the.

[47] Cancer Research UK (2014). *Types of cancer*. [online] Cancer Research UK. Available at: https://www.cancerresearchuk.org/about-cancer/what-is-cancer/how-cancer-starts/types-of-cancer#:~:text=For%20example%2C%20nerves%20and%20muscles.