

**Faculty of Management, Law and Social Sciences**

**DEPARTMENT: BACE**

**MODULE TITLE: Applied Machine Learning and Big Data Strategy**

**MODULE CODE: OIM7508-B**

**TITLE:**

**ANALYSIS AND IMPLEMENTATION OF MACHINE LEARNING MODELS FOR THE PREDICTION AND EARLY DETECTION OF CORONARY ARTERY DISEASE"**

**TABLE OF CONTENTS**

[1. INTRODUCTION 2](#_Toc163306235)

[2. LITERATURE REVIEW 5](#_Toc163306236)

[3. METHODOLOGY 8](#_Toc163306237)

[3.1 Data Pre-processing 9](#_Toc163306238)

[4. ANALYTICS AND FINDINGS 11](#_Toc163306239)

[4.1 Exploratory Data Analysis 11](#_Toc163306240)

[4.2 Feature Selection 19](#_Toc163306241)

[4.3 Model Training and Evaluation 20](#_Toc163306242)

[4.3.1 Parameter Tuning 20](#_Toc163306243)

[4.4 Prediction and Evaluation 23](#_Toc163306244)

[4.4.1 True Positive Rates of the models 26](#_Toc163306246)

[4.4.2 True Negative Rates of the models 27](#_Toc163306247)

[4.4.3 AUC ROC Score for all the models 28](#_Toc163306248)

[4.4.4 Confusion Matrices of Models 30](#_Toc163306249)

[4.5 Interpretation of Results 33](#_Toc163306250)

[5. DISCUSSION AND CONCLUSION 34](#_Toc163306251)

[6. Individual Contribution/Overall Reflection of Team Work 37](#_Toc163306252)

[REFERENCE LIST 40](#_Toc163306253)

[Appendix: Source Code For the Implementation On Jupyter Notebook 44](#_Toc163306254)

**LIST OF FIGURES**

Figure 4.1: Age Distribution 15

Figure 4.2: Age-Related Trends in Coronary Artery Disease Prevalence 16

Figure 4.3: Gender Distribution 17

Figure 4.4: Gender-Related Trends in Coronary Artery Disease Prevalence 18

Figure 4.5: Resting Electrocardiographic Results in Relation to Coronary Artery Disease Status 19

Figure 4.6: Maximum Flow Duration of Network Traffic 20

Figure 4.7: Distribution of Classification Labels 21

Figure 4.8: Distribution of Grid Search Cross-validation Scores of Models 24

Figure 4.9: Accuracy Scores of Models 28

Figure 4.10: True Positive Rates of Models 29

Figure 4.11: True Negative Rates of Models 30

Figure 4.12: Joint ROC Curve of Models 31

Figure 4.13: Individual ROC Curves of Models 32

Figure 4.14: Confusion Matrices of Models…………………………………………………34

**LIST OF TABLES**

*Table 4.1: Summary of Dataset Variables 12*

*Table 4.2: Evaluation Results of Binary Classification 24*

# 

# 1. INTRODUCTION

Coronary artery disease (CAD) is a prevalent and dangerous cardiovascular condition that is defined by the gradual accumulation of plaque within the coronary arteries, which ultimately results in limited blood flow to the heart muscle. CAD is among the most common and dangerous cardiovascular conditions (Aikeliyaer et al., 2023). This insidious ailment, which is frequently referred to as a silent killer, can progress covertly, with symptoms not starting to become noticeable until the advanced phases of the condition have occurred. Failure to recognize or treat coronary artery disease (CAD) can have catastrophic implications, ranging from the incapacitating condition of angina to the occurrence of life-threatening events such as myocardial infarction (also known as a heart attack) and heart failure. As the main cause of morbidity and mortality on a global scale, coronary artery disease (CAD) presents a huge challenge to public health, making it imperative to develop comprehensive strategies for the early diagnosis and intervention of this condition (Al-Ssulami et al., 2023).

One cannot overestimate the significance of early identification in terms of reducing the negative outcomes that are linked with coronary artery disease (CAD). The ability to identify a disease at an early stage enables medical professionals to initiate timely therapies with the goal of slowing the progression of the disease and lowering the risk of consequences (Bilal et al., 2023). Although the diagnostic methods that are currently available, such as electrocardiography, stress testing, and coronary angiography, are quite helpful, it is possible that they do not possess the sensitivity and specificity that are necessary for recognizing coronary artery disease in its early stages. As a consequence of this, there is an urgent want for methods that are both more sophisticated and accurate in order to identify individuals who are at risk of developing coronary artery disease (CAD) prior to the beginning of symptoms (Heo et al., 2022).

Through the examination and use of machine learning models, this research strives to answer the critical need for early detection of computer-aided design (CAD). Through the utilization of computational algorithms, the purpose of this research is to construct prediction models that are able to recognize subtle patterns and risk factors that are related with coronary artery disease (CAD) (Huang et al., 2022). In comparison to more conventional approaches, machine learning techniques provide a number of distinct advantages. These advantages include the capacity to evaluate huge datasets, recognize intricate relationships, and adaptively enhance predictions over various time periods. The purpose of this research is to demonstrate the efficacy and reliability of machine learning-based approaches in improving the prediction and early detection of coronary artery disease (CAD) through rigorous analysis and validation (Hassan et al., 2022).

The importance of this work extends beyond the area of academia, as it has the potential to provide advantages to a variety of stakeholders within the healthcare ecosystem. Patients have the potential to benefit from enhanced risk assessment and individualized preventative interventions, which will ultimately result in improved health outcomes and quality of life experiences (Orlenko et al., 2019). The insights that are obtained by machine learning models can be utilized by healthcare practitioners in order to improve the allocation of resources, expedite clinical procedures, and provide more effective patient care. In addition, policymakers and public health professionals can use the findings to inform evidence-based initiatives for the prevention and management of coronary artery disease (CAD), which will ultimately contribute to the improvement of cardiovascular health on a population scale (Özbilgin et al., 2023). The purpose of this project is to make a significant contribution to the area of cardiovascular medicine while simultaneously solving a key unmet need in the delivery of healthcare. This will be accomplished through the utilization of cutting-edge technology and innovative approaches from a variety of disciplines (Sayadi et al., 2022).

# 2. LITERATURE REVIEW

Numerous studies from fields as disparate as computer science, cardiology, and epidemiology have contributed to the body of literature on coronary artery disease (CAD) early detection and prediction (Tang et al., 2023). According to Verma et al, 2017, many biomarkers and risk factors for coronary artery disease (CAD) have been identified in the literature, including both established clinical characteristics and new molecular signs. Conventional diagnostic methods like electrocardiography and imaging modalities are still important for CAD detection, but there has been a lot of recent focus on using machine learning algorithms and other advanced computational techniques to make predictions more accurate and faster (Tang et al., 2023). The potential of these approaches to detect subtle trends and enable early intervention has been highlighted by studies investigating the use of machine learning in CAD prediction, which have shown encouraging results. Problems including data heterogeneity, model interpretability, and generalizability are still being researched in the study. The purpose of this study is to analyze the present literature in order to put current research efforts into context, find knowledge gaps, and establish a foundation for using machine learning models for CAD prediction and early diagnosis.

The vital role of the heart is emphasized by Trigka and Dritsas (2023), who stress its utmost significance to human existence. They clarify the pathophysiology of CAD, drawing attention to the fact that atherosclerotic plaques grow inside the coronary arteries, reducing blood flow and endangering cardiac function, as the disease's origin. The researchers explore the arena of long-term risk prediction for CAD, acknowledging the importance of early intervention. They investigate methods for improving the predictive accuracy of ML models using synthetic minority oversampling (SMOTE) and other similar techniques. With outstanding precision, accuracy, recall, and area under the curve (AUC), their results highlight the effectiveness of a stacking ensemble model after SMOTE.

In a similar vein, Michał Woś et al. (2023) explore the exciting new field where machine learning meets healthcare, specifically to assist doctors. With a primary focus on coronary heart disease (CHD) prediction, they analyze 16 relevant characteristics, such as body mass index (BMI) and cholesterol levels, using a dataset consisting of more than 11,000 patient records. They use a variety of ML algorithms to find that XGBoost, with the help of other model refinement strategies like backward searches and LIME-based explanations, is the best option. Their research demonstrates how machine learning can improve public health programs and clinical decision-making.

However, the difficulties caused by atherosclerosis in the diagnosis of cardiovascular disorders, especially coronary artery disease (CAD) and myocardial infarction (MI), are explained by Kumar et al. (2022). Using SVM and ANN models for prediction, they suggest using characteristics derived from electrocardiograms (ECGs) to get over diagnostic restrictions. Impressive accuracy in classifying CAD and MI was demonstrated in their study using time-domain heart rate variability (HRV) data, especially with the ANN model. This kind of automated diagnosis has the ability to provide affordable screening options, which would be helpful for doctors in identifying patients who are at risk.

Finally, the worldwide impact of coronary heart disease is discussed by Hassan et al. (2022), who highlight the use of machine learning in clinical predictive analytics. Using eleven ML classifiers, including Random Forest, multilayer perceptron, and gradient boosted trees, their work successfully predicts the occurrence of heart disease. Their research highlights the value of machine learning to enhance diagnostic capabilities, which could lead to better healthcare outcomes and better disease management solutions.

Using information on coronary artery disease (CAD) from the literature, this study intends to create and assess state-of-the-art machine learning models for CAD prediction and early diagnosis. The purpose of this research is to analyze and predict coronary artery disease (CAD) using machine learning (ML) models: Decision Tree, K-Nearest Neighbors (KNN), Linear Support Vector Machine (SVM), Logistic Regression, and Random Forest. These models was implemented through the Python scikit-learn library. The work aims to improve the accuracy and efficiency of CAD identification by using these algorithms to examine patterns and attributes within a heart disease dataset. Precision, accuracy, true positive rate (TPR), F1 score, Area under the Receiver Operating Characteristic Curve (AUC ROC), and the confusion matrix are some of the comprehensive classification metrics that will be used to objectively evaluate the ML models' performance. Improving healthcare outcomes and implementing proactive disease management techniques are the ultimate goals of this research, which employs a thorough review approach to determine the best ML algorithms and feature sets for CAD prediction (Huang et al., 2022).

# 3. METHODOLOGY

The task of predicting coronary artery disease began with a comprehensive analysis of the dataset. This initial phase involved the utilization of data analysis and visualization techniques to gain valuable insights into the relationships among various variables in the dataset. This exploratory phase provided a crucial understanding of the dataset, enabling a more informed approach to subsequent model development (Özbilgin et al., 2023).

The predictive modeling phase was characterized by a rigorous hyperparameter tuning procedure, ensuring that the machine learning models were fine-tuned for optimal performance.

This project aims to create a predictive model that can accurately identify individuals who have coronary artery disease (CAD). The study adopts a binary classification approach to categorize individuals into two groups: those who have coronary artery disease and those who do not (Sayadi et al., 2022).

To achieve this objective, the following machine learning algorithms were utilized: Decision Tree, K-Nearest Neighbors (KNN), Linear Support Vector Machine (SVM), Logistic Regression, and Random Forest. These algorithms were chosen because according to Kumar et al. (2022), these models have proven to be one of the best models in predicting and detecting various forms of diseases.

These algorithms, implemented using the Python scikit-learn library, aimed to analyze patterns and features in the heart disease dataset in order to predict and detect coronary artery disease. The performance of the models was assessed by analyzing their predictions using various classification metrics, such as accuracy, precision, true positive rate (TPR), true negative rate (TNR), F1 score, Area under the Receiver Operating Characteristic Curve (AUC ROC), and the confusion matrix (Orlenko et al., 2019).

After a comprehensive evaluation of the results for each model, particular model emerged as the optimal choice for constructing an effective detection system for coronary artery disease. The culmination of this process represents a robust predictive model tailored to accurately identify individuals with coronary artery disease, contributing to advancements in the field of cardiovascular health (Özbilgin et al., 2023).

## 3.1 Data Pre-processing

**Data pre-processing was performed to ensure the accuracy, reliability, and readiness of the dataset for subsequent analysis and model training.**

**First, the dataset was examined to identify and handle missing values (null values) and Infinity values. This is a crucial step, as missing or infinite values can negatively impact the performance of machine learning models. However, no such problematic values were present in the dataset. This reduced the need for extensive data imputation or cleaning, which can introduce biases or errors into the analysis** (Orlenko et al., 2019)**.**

**In addition, the dataset was inspected to to find columns that contained only one unique value. These kinds of columns lack variability and are not informative for predictive modeling purposes. Upon examination of this dataset, it was found that none of the columns contained only one distinct value. Every column contains a minimum of two distinct values. This indicates potential variability and relevance of the dataset features for analysis and predictive modeling.**

**Furthermore, the data types of columns were checked to ensure uniformity. Errors may occur during analysis or modelling as a result of inconsistent data types. Fortunately, it was discovered that the entries in each column of this dataset have the same type, indicating uniformity in data types throughout the dataset. This uniformity facilitates seamless data processing and analysis steps, ensuring compatibility with machine learning algorithms (**Forrest et al., 2023**).**

**Overall, based on these observations, it can be inferred that the dataset obtained for the prediction of coronary artery disease is of high quality, contains diverse and potentially informative features, and is well-prepared for further analysis and model training.**

# ****4. ANALYTICS AND FINDINGS****

This section explores and explains the analysis and implementation of the ML algorithms which begins with exploratory data analysis (EDA)

## 4.1 Exploratory Data Analysis

The dataset consists of 1,025 rows and 14 columns. During the data preprocessing stage, there was no need to remove any information from the dataset. Thus, all the original features and entries were retained in the dataset,

All of the dataset columns are numeric. Numeric data involve measurable data and can assume continuous (float) or discrete (integer) values.

Table 4.1 presents information about the columns in the dataset, including details such as the datatype, count of unique values and the range of values for each column.

**Table 4.1:** Summary of Dataset Variables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S/N** | **Variables** | **Data type** | **Unique** | **Range** |
| 1. | age | Integer | 41 | 29...77 |
| 2. | sex | Integer | 2 | 0...1 |
| 3. | cp | Integer | 4 | 0...3 |
| 5. | trestbps | Integer | 49 | 94...200 |
| 5. | chol | Integer | 152 | 126...564 |
| 6. | fbs | Integer | 2 | 0...1 |
| 7. | restecg | Integer | 3 | 0...2 |
| 8. | thalach | Integer | 91 | 71...202 |
| 9. | exang | Integer | 2 | 0...1 |
| 10. | oldpeak | Float | 40 | 0...6.2 |
| 11. | slope | Integer | 3 | 0...2 |
| 12. | ca | Integer | 5 | 0...4 |
| 13. | thal | Integer | 4 | 0...3 |
| 14. | target | Integer | 2 | 0...1 |

The ‘target’ column in the dataset is the target variable in this study. It classifies individuals into two distinct classes: 0 and 1. 0 signifies individuals without coronary artery disease, while 1 indicates individuals with coronary artery disease. This column plays a crucial role in this classification task, where the objective is to determine the occurrence or non-occurrence of coronary artery disease by relying on the information from other attributes in the dataset.

Following the initial dataset collection, the subsequent step involved the visualization and further analysis of the data to comprehend the interrelationships between different variables present in the dataset.

The dataset contains information about individuals’ ages, which have been grouped into five categories: 30s, 40s, 50s, 60s, and 70s. This grouping allows for easier analysis and interpretation of age-related trends within the dataset.

Figure 4.1 illustrates a distribution of age groups in the dataset, highlighting a normal distribution pattern with a peak in the 50s age group, indicating that this dataset contains a relatively higher number of individuals in their fifties compared to other age groups.

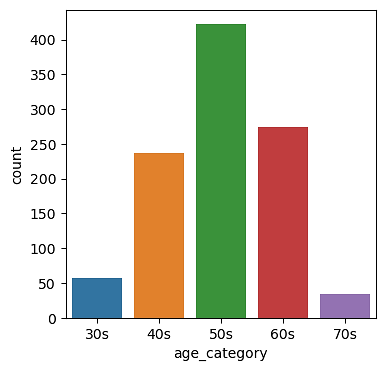


Figure 4.1: Age Distribution

Figure 4.2 highlights the relationship between age and the prevalence of coronary artery disease. Individuals were classified into discrete age groups in order to assess the prevalence of coronary artery disease across various age groups.

As depicted in the figure below, there is a decline in the prevalence of CAD from the 30s through the 60s age groups, suggesting a lower susceptibility to the disease during these years of life. However, this downward trajectory changes upon reaching the 70s age category, where a resurgence in CAD prevalence is observed. While the data suggests a relative decrease in CAD occurrence during midlife, the subsequent rise in prevalence among older individuals emphasizes the importance of age as a critical risk factor for this debilitating condition.

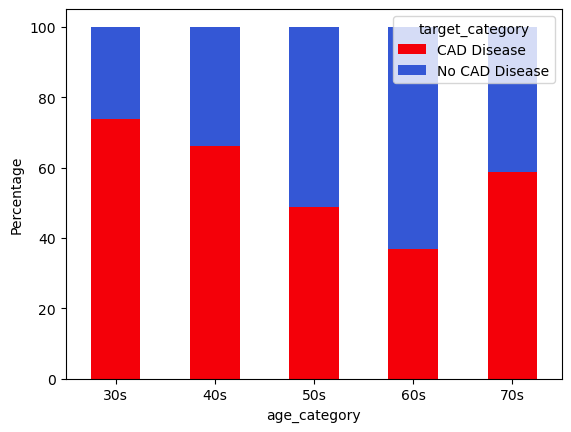


Figure 4.2: Age-Related Trends in Coronary Artery Disease Prevalence

The figure displayed below shows the distribution of genders within the dataset. The number of females appears to be significantly higher than that of males, indicating a gender imbalance within the dataset.

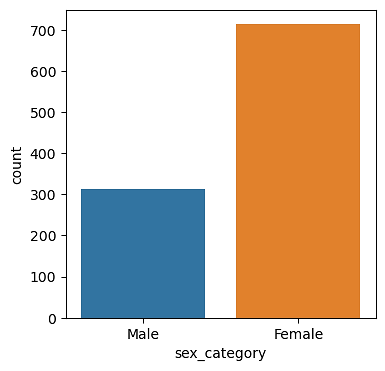


Figure 4.3: Gender Distribution

Figure 4.4 illustrates a disparity in the occurrence of coronary artery disease among females and males. Among females, a higher percentage is found to be free from CAD in comparison to those who have the disease. This suggests a lower prevalence rate of CAD among females.

Conversely, among males, the situation is noticeably different. The proportion of males with CAD surpasses the proportion of those without the disease. Specifically, the ratio of those afflicted by the disease to those free from it among males is nearly threefold, indicating a considerably higher occurrence rate of CAD among males in comparison to females in the dataset.

This gender-based discrepancy in CAD prevalence underscores the importance of understanding and addressing sex-specific risk factors.

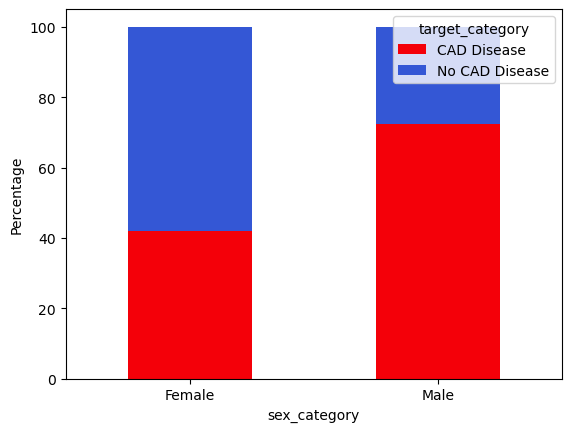


Figure 4.4: Gender-Related Trends in Coronary Artery Disease Prevalence

Figure 4.5 depicts the resting electrocardiographic (ECG) results for the two categories. A considerable proportion of persons without coronary artery disease display normal electrocardiogram findings, indicating the lack of noteworthy abnormalities.

Among individuals without coronary artery disease, a significant proportion exhibit normal ECG results, indicating the absence of notable abnormalities. Conversely, within the subset of individuals diagnosed with coronary artery disease, the predominant ECG result observed is the presence of ST-T Wave Abnormality. This anomaly suggests potential myocardial ischemia or other cardiac conditions associated with coronary artery disease.

Therefore, the information presented in Figure 4.5 underscores the diagnostic relevance of ECG in distinguishing between individuals with and without coronary artery disease.

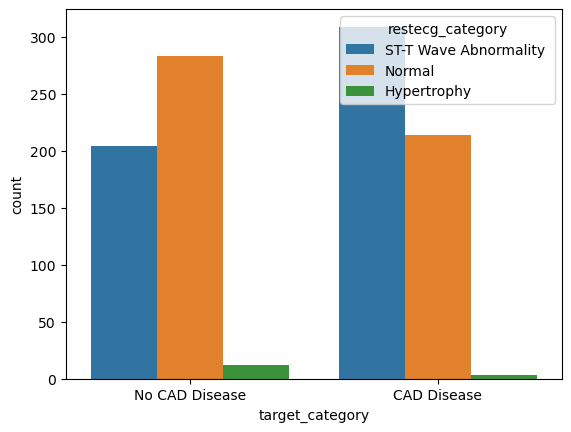


Figure 4.5: Resting Electrocardiographic Results in Relation to Coronary Artery Disease Status

The data presented in Figure 4.6 illustrates the average maximum heart rates attained during stress tests in both categories. Notably, individuals with coronary artery disease exhibit a higher average maximum heart rate during stress tests compared to their counterparts without this medical condition. This observation highlights the physiological differences in how the two groups respond under stress-inducing conditions.

The higher mean maximum heart rate observed in individuals with CAD suggests a heightened cardiovascular response to stressors, potentially reflecting the compromised nature of their cardiovascular system due to the presence of arterial blockages or other related factors.

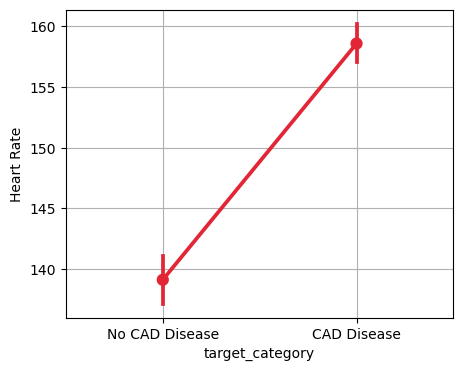


Figure 4.6: Maximum Flow Duration of Network Traffic

## 4.2 Feature Selection

The dataset utilized in this study comprises solely of numerical values. However, during the analysis stage, some categorical features were created from existing columns. These categorical features were removed from the dataset to maintain its integrity, eliminate redundancy, as the categorical columns are already encoded numerically, and to facilitate smooth execution of machine learning algorithms.

The target column, which is the target variable, consists of categorical codes 0 and 1. These codes are used to classify instances into two groups: 0 denotes the absence of coronary artery disease, and 1 represents the presence of coronary artery disease. Refer to Figure 4.7 for the number of occurrences in these two groups.

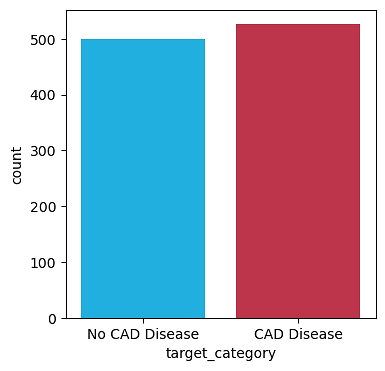


Figure 4.7: Distribution of Classification Labels

Moreover, in accordance with standard procedure for training a machine learning model, the predictor variables were separated from the target variable. For this task, the target variable is the variable to be predicted, while the predictor variables consist of all other attributes of the dataset relevant to the prediction of coronary artery disease.

## 4.3 Model Training and Evaluation

Prior to executing machine learning algorithms or any other stochastic procedures, a fixed random seed value (42) was set to ensure consistent results. This guarantees that when the code is executed with this particular seed value, the random number generator used by the models and other random processes will produce the same sequence of random numbers. As a result, the outcomes of machine learning activities, such as model training and evaluation, as well as other random processes, will remain the same every time the code is executed, guaranteeing the reproducibility of the code.

The dataset was split into two subsets, with 70% of the data allocated to the training set and the remaining 30% assigned to the testing set. The training set was utilised to train the models, whereas the test set was reserved for evaluating the models’ performance.

### 4.3.1 Parameter Tuning

Parameter tuning is necessary because hyperparameters can impact the performance of machine learning models. Therefore, tweaking these hyperparameters allows one to find the combination that maximizes the model’s performance on the given dataset. Thus, each model underwent parameter tuning to optimize its performance.

The parameters tuned for Decision Tree are ‘criterion’ for picking the split criterion, ‘splitter’ for choosing split strategies, ‘max\_depth’ for tree depth, and ‘max\_features’ for the number of features considered for the best split.

In the case of K-Nearest Neighbors, the parameters being tuned are ‘n\_neighbors’ to specify the number of neighbors, ‘weights’ to choose between uniform or distance-based weights, ‘algorithm’ for computing nearest neighbors, and ‘metric’ to measure similarity between instances.

For the Linear Support Vector Machine (Linear SVM), the parameters under consideration include ‘loss’ to determine the loss function, ‘tol’ which sets the tolerance for stopping criteria, ‘C’ for the regularization parameter, and ‘max\_iter’ to control the maximum number of iterations.

In Logistic Regression, tuning involves exploring ‘penalty’ to choose between L1 or L2 regularization, ‘C’ to set the inverse of regularization strength, ‘solver’ to specify the optimization algorithm, and ‘max\_iter’ to define the maximum number of iterations for optimization.

Finally, for Random Forest, the parameters being tuned include ‘n\_estimators’ to decide the number of trees in the forest, ‘criterion’ to select the function for measuring split quality, ‘max\_depth’ to control the maximum depth of individual trees, and ‘max\_features’ to specify the number of features considered when looking for the best split.

To systematically explore these parameter combinations, a grid search cross-validation technique was used. This searches for the optimal hyperparameters of a model by evaluating the model’s performance across different combinations of hyperparameter values using cross-validation. Cross-validation involves partitioning the training data into subsets, or folds, and then training the model on a subset of the data while using the remaining data for validation. This process is repeated multiple times.

At the end of this iterative process, the best estimator with the optimal combination of parameters from each model was selected. Figure 4.8 shows the distribution of mean cross-validation scores for each model. The KNN model exhibits the widest range of performance. The Decision Tree and Random Forest models display a narrower range of performance metrics, yet demonstrate superior overall performance within their respective ranges.

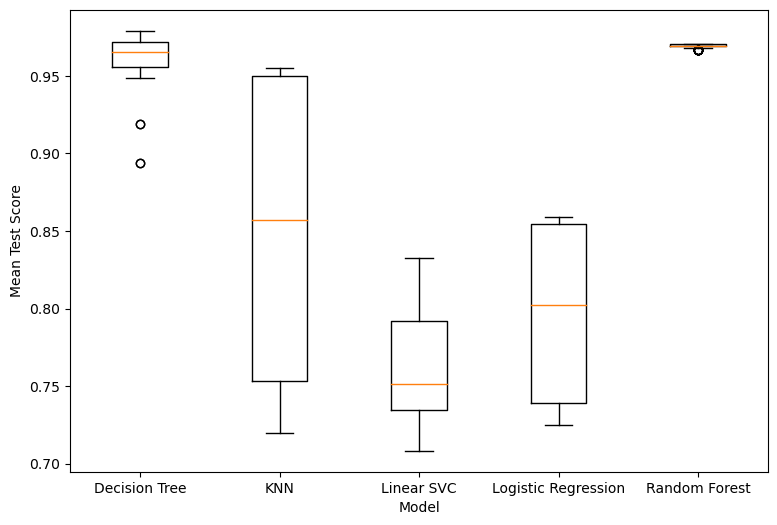


Figure 4.8: Distribution of Grid Search Cross-validation Scores of Models

## 4.4 Prediction and Evaluation

The best estimator for each model were used to predict the labels for the test set, and several metrics were used to measure their performance. The results of these evaluations are presented in the table below, allowing for a comparison of the models’ effectiveness in accomplishing the given task.

Table 4.2: Evaluation Results of Binary Classification

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **F1 Score** | **TPR** | **TNR** | **AUC ROC** |
| Decision Tree | 0.981 | 1.000 | 0.979 | 0.960 | 1.000 | 0.980 |
| KNN | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| Linear SVM | 0.799 | 0.735 | 0.814 | 0.913 | 0.692 | 0.802 |
| Logistic Regression | 0.818 | 0.775 | 0.824 | 0.879 | 0.761 | 0.820 |
| Random Forest | 0.990 | 1.000 | 0.990 | 0.980 | 1.000 | 0.990 |

The Decision Tree model exhibits exceptional accuracy of 0.981, with a flawless precision (1.000) indicating no false positive predictions, and a high F1 score of 0.979, suggesting a balanced performance between precision and recall. The TPR of 0.960 and TNR of 1.000 indicate that it can accurately detect both positive and negative cases. With an impressive AUC ROC of 0.980, the Decision Tree model shows strong discriminatory ability in distinguishing between positive and negative occurrences, making it a reliable option for coronary artery disease detection.

The KNN model achieves flawless performance of 100% across all metrics, indicating impeccable classification with no false positives or false negatives. With an AUC ROC score of 1.000, KNN demonstrates exceptional discriminatory ability, making it an outstanding choice for coronary artery disease detection tasks, particularly in scenarios where precision and recall are crucial.

Although the Linear SVM model has a slightly lower accuracy of 0.799 compared to other models, it still demonstrates reasonable precision (0.735) and a respectable F1 score (0.814). With a high TPR of 0.913, Linear SVM demonstrates its effectiveness in identifying positive cases, although its TNR is comparatively lower at 0.692. The AUC ROC score of the model, which is 0.802, indicates that it has moderate discriminatory power. This suggests that the model can be useful in tasks related to detecting coronary artery disease, while there is still room for improvement.

The Logistic Regression model achieves a moderate accuracy of 0.818, with precision (0.775) and F1 score (0.824), indicating a reasonable balance between true positives and false positives. Its true positive rate of 0.879 highlights its effectiveness in identifying positive cases, while the true negative rate of 0.761 suggests a comparatively lower ability to correctly identify negative instances. Logistic Regression exhibits a reasonable level of discriminatory power with an AUC ROC score of 0.820. This makes it a viable option for detecting coronary artery disease, however its performance could be improved through additional optimization.

The Random Forest model also emerged as a highly effective model, with an exceptional accuracy of 0.990 and flawless precision (1.000), signifying that it makes no erroneous positive predictions. Its F1 score of 0.990 highlights a well-balanced performance in terms of precision and recall. Additionally, the TPR of 0.980 and TNR of 1.000 reflect its strong capability to accurately identify both positive and negative situations. Random Forest also demonstrates outstanding discrimination ability with an impressive AUC ROC score of 0.990. This makes it an an excellent choice for coronary artery disease detection.

The KNN model exhibits the highest accuracy among the models evaluated, attaining a perfect accuracy score of 100%. The Random Forest model closely follows, with an impressive accuracy rate of 99.0%. The Decision Tree model also has a remarkable level of accuracy, with a rate of 98.1%. The Logistic Regression and Linear SVM models demonstrate relatively lower accuracies of 81.8% and 79.9% respectively.

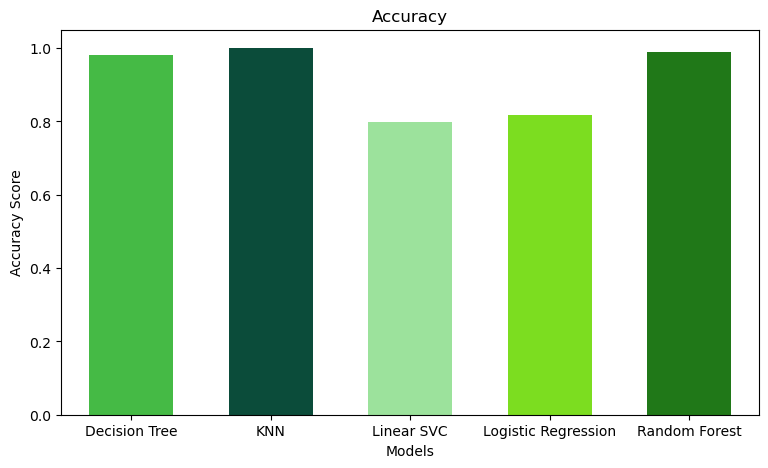


Figure 4.9: Accuracy Scores of Models

### 4.4.1 True Positive Rates of the models

Figure 4.10 depicts the True Positive Rates of the models. This metric measures the models’ ability to accurately detect CAD among those individuals who truly have the disease. The KNN model achieved the highest TPR, followed closely by Random Forest, and then the Decision Tree. The Linear SVM and Logistic Regression models exhibit slightly lower TPRs, but still showcase reasonable proficiency in CAD detection.

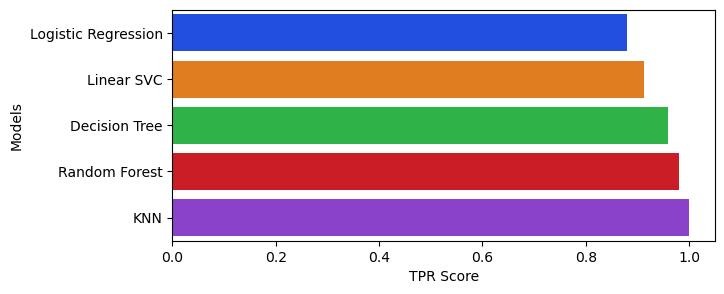


Figure 4.10: True Positive Rates of Models

### 4.4.2 True Negative Rates of the models

Figure 4.11 depicts the True Negative Rates of the models. This metric measures the models’ ability to correctly identify individuals without CAD among those individuals who truly do not have the disease. The Decision Tree, KNN, and Random Forest models achieved perfect TNR, indicating robust performance in identifying individuals without CAD. The Logistic Regression and Linear SVM models also demonstrated reasonable performance but with lower TNR values.

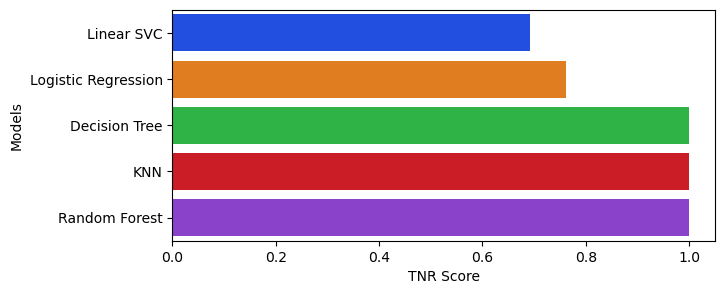


Figure 4.11: True Negative Rates of Models

### 4.4.3 AUC ROC Score for all the models

Figure 4.12 displays the Area Under the Receiver Operating Characteristic Curve (AUC ROC) for all the models. A higher AUC ROC value signifies superior discriminatory capability, indicating that the model is more proficient at differentiating between individuals with CAD and those without. Achieving a high AUC ROC score is crucial in medical diagnosis as it reflects the model’s ability to reduce both false positives and false negatives, which are critical considerations in CAD detection due to the potential severe consequences of errors.

The K-Nearest Neighbors model surpasses other models with a flawless AUC ROC score of 1.000, leveraging its ability to identify similarities among data points for accurate classification. Following closely is the Random Forest and Decision Tree models with AUC ROC scores of 0.99 and 0.98 respectively.

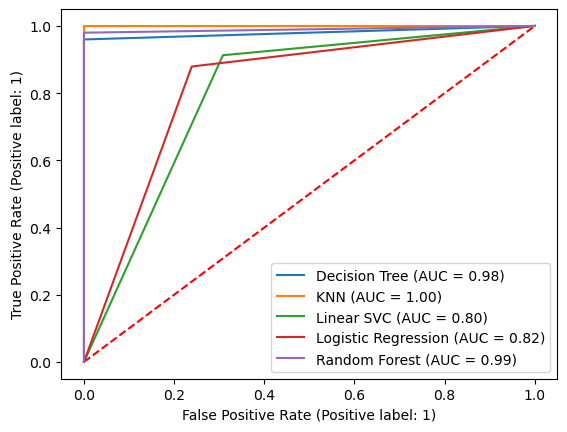
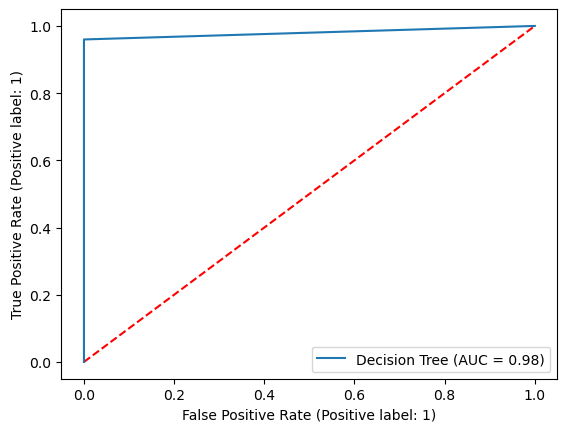
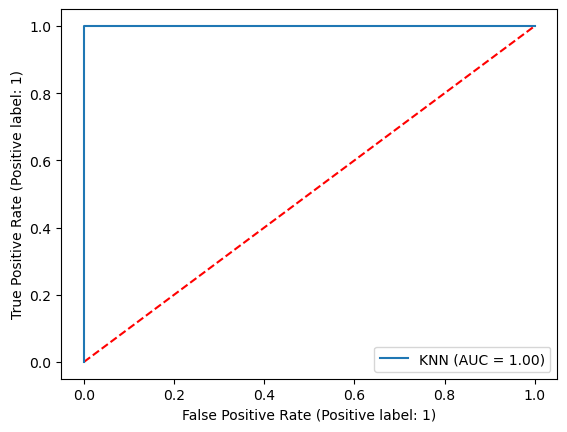
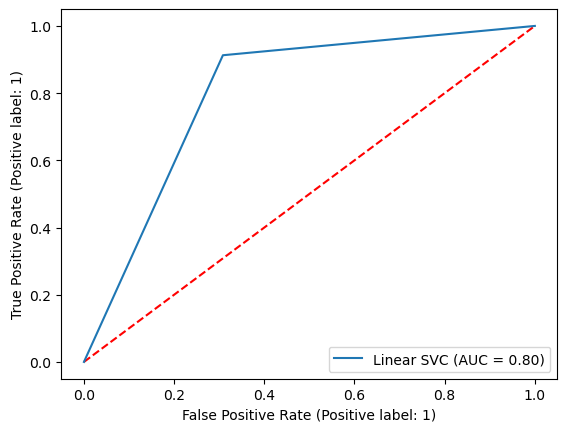
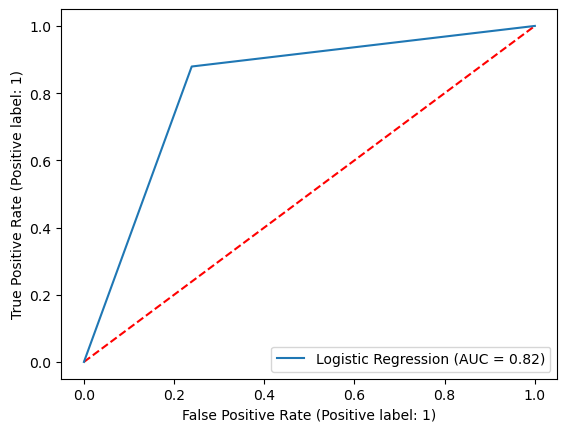


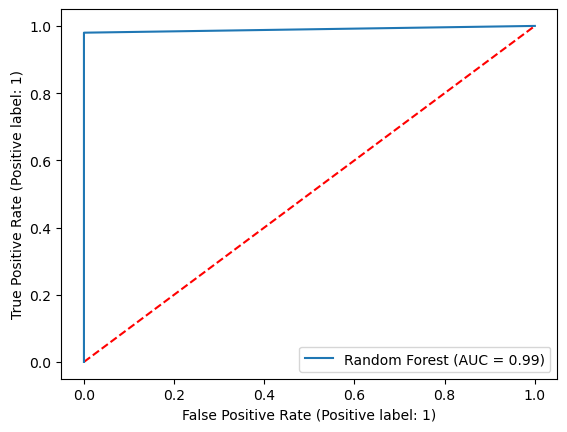
Figure 4.12: Joint ROC Curve of Models

Figures 4.13 a-e show the individual AUC ROC curve of each model.

  a) Decision Tree b) KNN

c) Linear SVM d) Logistic Regression

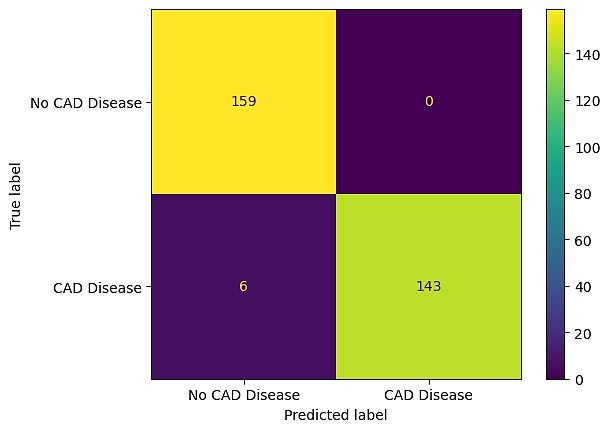
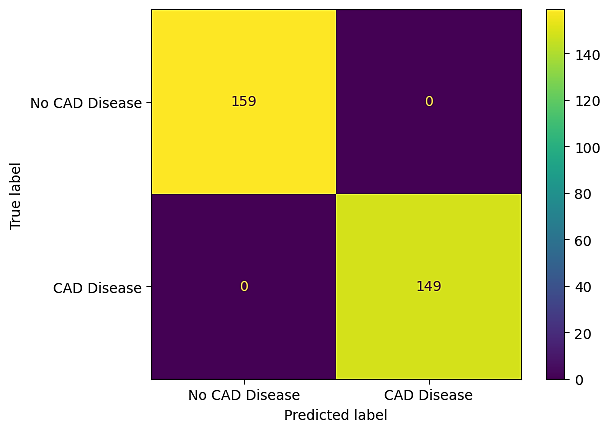


e) Random Forest

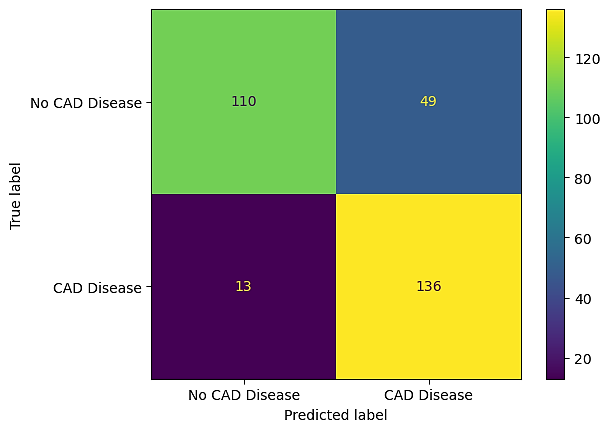
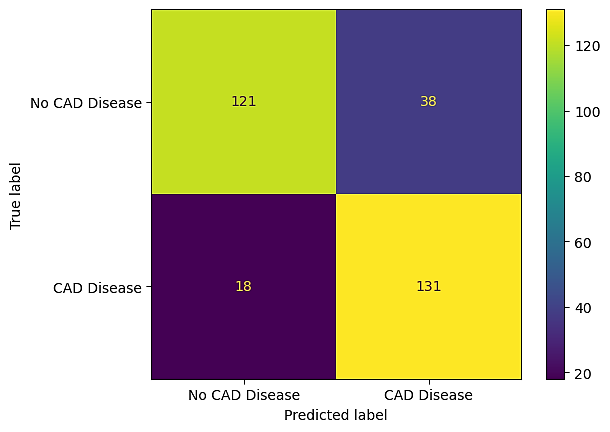
Figure 4.13: Individual ROC Curves of Models

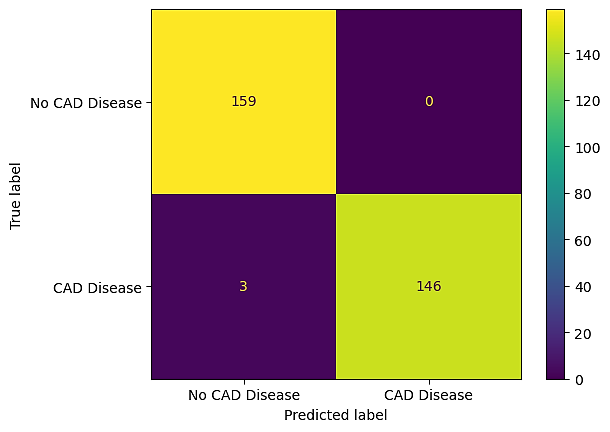
### 4.4.4 Confusion Matrices of Models

Figures 4.14 a-e illustrate confusion matrices that represent the predictive results of different models in distinguishing between those who have coronary artery disease and those who do not.

a) Decision Tree b) KNN

   
 c) Linear SVM d) Logistic Regression



e) Random Forest

Figure 4.14: Confusion Matrices of Models

The Decision Tree model, while generally effective, exhibited some limitations in accurately categorizing instances of CAD. It incorrectly classified 6 cases of individuals with coronary artery disease as negative. However, it performed well in correctly identifying all CAD-negative cases.

In contrast, the KNN model demonstrated robust performance by accurately classifying all instances of both positive and negative cases. This indicates a high level of sensitivity and specificity, suggesting its potential as a reliable tool for detecting coronary artery disease.

The Linear Support Vector Classifier exhibited the highest number of misclassifications, with 13 instances of false negatives (misclassifying individuals with coronary artery disease as negative) and 49 instances of false positives (misclassifying individuals without coronary artery disease as positive). This indicates limitations in its ability to correctly classify both positive and negative cases, possibly due to the complexity of the dataset or the linear nature of the classifier.

Similarly, the Logistic Regression model displayed errors in its predictions, misidentifying 18 instances of CAD-positive cases and misclassifying 38 instances of CAD-negative cases.

The Random Forest Classifier, despite its overall effectiveness, still made some errors. It mislabeled 3 instances of CAD-positive cases as CAD-negative. Nevertheless, it correctly classified all instances where CAD was not present. This suggests overall strong performance with few inaccuracies, highlighting its potential for coronary artery disease detection.

## 4.5 Interpretation of Results

For this prediction task, all the models demonstrated commendable performance. Three models (Decision Tree, KNN, and Random Forest) demonstrated commendable performance in CAD detection, exhibiting high accuracy, precision, F1 score, true positive rate, true negative rate, and AUC ROC score.

However, the KNN model demonstrates the highest effectiveness in predicting coronary heart disease. Among all the models evaluated, the KNN model stands out with exceptional performance, achieving flawless scores across various evaluation metrics. It demonstrated remarkable accuracy and precision, along with high sensitivity and specificity, enabling it to accurately identify individuals with CAD while minimizing false positives and negatives, thereby enhancing diagnostic accuracy. Moreover, the KNN model’s flawless predictions, devoid of errors, attest to its reliability and consistency in CAD detection. This reliability is paramount in clinical decision-making, where accuracy and precision are paramount for effective patient management.

It is important to note that the outstanding performance of the KNN model is attributed to hyperparameter tuning. The hyperparameters of this model is different from the default ones. By optimizing its hyperparameters, the model’s predictive capabilities were significantly enhanced, underscoring the importance of parameter tuning in machine learning model development. This emphasizes the necessity of fine-tuning model parameters to achieve optimal performance tailored to specific tasks, such as CAD detection.

Thus, the KNN model shows considerable promise in aiding the prediction and detection of coronary artery disease. Its exceptional performance, coupled with optimized hyperparameters, positions it as a valuable asset in healthcare settings, potentially leading to improved patient outcomes and more effective healthcare management strategies.

# 5. DISCUSSION AND CONCLUSION

Several machine learning models have shown promise in predicting and detecting coronary artery disease (CAD), however when compared to the results of the literature review, there are some key parallels and discrepancies. Decision Tree, K-Nearest Neighbors (KNN), Linear Support Vector Machine (SVM), Logistic Regression, and Random Forest are some of the models that have been studied and explored, and they all show different levels of success when it comes to CAD identification (D’Ancona et al., 2023).

The KNN model stands out in both cases, doing exceptionally well across all criteria of evaluation. This is in line with the results of earlier research, including the one by Michał Woś et al. (2023), which has also demonstrated the outstanding predictive powers of KNN in healthcare settings. With its high specificity and sensitivity, the KNN model can accurately identify patients with CAD while minimizing false positives and negatives. Its robust performance highlights its promise as a dependable tool for CAD detection (Kumar et al., 2022).

Similarly, the Random Forest model shows great accuracy, precision, and discriminatory ability in both the research and the literature. According to Hassan et al. (2022), ensemble learning methods, and Random Forest in particular, are effective at improving prediction performance and managing complicated datasets.

On the other hand, it's clear that the models tested had different levels of performance. The research's Decision Tree model achieves high precision and accuracy, in line with the literature; yet, it fails to correctly classify CAD cases, as shown by the confusion matrix's misclassifications. Whereas the Linear SVM and Logistic Regression models show lower accuracies and higher rates of misclassification, it may be difficult to properly differentiate between positive and negative cases of CAD (Aikeliyaer et al., 2023).

In conclusion, the results show that CAD prediction and detection using machine learning models, especially KNN and Random Forest, is beneficial. When it comes to clinical decision-making, the KNN model's outstanding performance highlights its consistency and reliability, which could improve patient outcomes and healthcare management techniques. The results are encouraging, but we need to validate and conduct more studies to see how well these models work in real-world clinical situations and whether they are generalizable. The findings of this study provide important information for improving diagnostic tools and developing preventative methods of disease management in cardiovascular medicine through the use of machine learning for coronary artery disease (CAD) identification.

Practical implications of identified pain points and stakeholders' demands in CAD prediction and detection highlight the need for robust and reliable machine learning algorithms in clinical practice. Healthcare professionals must weigh the pros and disadvantages of each model before integrating them into diagnostic workflows due to their different performance (Akella and Akella 2021). The KNN model is accurate and precise, but it may need more computational resources for real-time application, requiring infrastructure upgrades. Although less accurate, Logistic Regression and Linear SVM may be more computationally efficient and interpretable, making them suited for resource-constrained healthcare situations. Understanding the needs and constraints of healthcare stakeholders like clinicians, administrators, and patients can help optimize CAD prediction and detection strategies, improving patient care and clinical outcomes (Alizadehsani et al., 2018).

To address the study limitations, further analytics and validation experiments are needed to improve machine learning model dependability and generalizability. This includes trying new feature engineering methods, optimizing hyperparameter tuning, and adding datasets or external factors to increase model resilience and prediction performance. To discover and resolve issues, model performance must be monitored and evaluated in real-world clinical situations. Fostering continual progress and innovation in CAD prediction and diagnosis requires collaboration between data scientists, doctors, and healthcare administrators. Investing in data literacy and computational skills training for healthcare personnel can help integrate sophisticated analytics into clinical practice and promote data-driven healthcare delivery (Annalsmedres.org. (2023).

Looking at how the results influence the big data approach; machine learning has great potential to change cardiovascular treatment using big data analytics. The findings emphasise the necessity of using different datasets and advanced analytical methods to gain insights and influence clinical decision-making. To maximize big data's impact on patient outcomes and healthcare delivery efficiency, healthcare organizations must invest in data infrastructure, governance frameworks, and talent development. Healthcare stakeholders can use big data analytics to solve complex problems like CAD prediction and detection, improving population health and healthcare system performance, by promoting data-driven innovation and collaboration (Chen et al., 2023).

# 6. Individual Contribution/Overall Reflection of Team Work

Students: Hameed AROWORA (23024078) My key contribution to the research team was a comprehensive literature evaluation on coronary artery disease (CAD) and machine learning models for prediction and early diagnosis. This required integrating relevant studies and identifying critical insights to guide our study. Navigating the huge literature and assuring relevance to our research aims were challenges. Learnings include the need of detailed documentation and teamwork in literature review processes to match study goals and improve insights.

Student ID 22058501: Faidat Afolabi In this research project, I preprocessed and analyzed the heart disease dataset to ensure data quality and relevance for model training and evaluation. This required data cleaning, feature engineering, and exploratory data analysis to find patterns and insights. Missing data, imbalanced classes, and model training feature selection were challenges. I learned data preprocessing and data integrity in machine learning from this encounter. Teamwork improved our analysis by sharing expertise and providing helpful criticism.

Student #22070364: Oluwaseun Popoola My main research project aim was to implement and improve CAD prediction and detection machine learning models. This required algorithm coding, parameter adjustment, and model validation using relevant metrics. Optimizing model performance, controlling computational resources, and resolving overfitting were challenges. This exercise taught me how iterative refinement and hyperparameter tuning improve model correctness and generalizability. Collaboration with team members helped share expertise and solve problems, boosting project success.

Student ID 22024147: Ajayi Oladimeji My research team role included model training, evaluation, interpretation, and visualization for dissemination. Implementing machine learning methods, adjusting hyperparameters, and measuring model performance were required. Interpreting sophisticated model results and conveying conclusions were difficult. This technique revealed the pros and cons of machine learning methods for CAD prediction. Teamwork improved my analytical skills and helped me express findings clearly, boosting our research's effect.

# REFERENCE LIST

[1] Aikeliyaer Ainiwaer, Wen Qing Hou, Kaisaierjiang Kadier, Rehemuding, R., Peng Fei Liu, Halimulati Maimaiti, Qin, L., Ma, X. and Jian Guo Dai (2023). A Machine Learning Framework for Diagnosing and Predicting the Severity of Coronary Artery Disease. *Reviews in Cardiovascular Medicine*, 24(6), pp.168–168. doi:<https://doi.org/10.31083/j.rcm2406168>.

[2] Akella, A. and Akella, S. (2021). Machine learning algorithms for predicting coronary artery disease: efforts toward an open source solution. *Future Science OA*, p.FSO698. doi:<https://doi.org/10.2144/fsoa-2020-0206>.

[3] Al-Ssulami, A.M., Alsorori, R.S., Azmi, A.M. and Hatim Aboalsamh (2023). Improving Coronary Heart Disease Prediction Through Machine Learning and an Innovative Data Augmentation Technique. *Cognitive Computation*, 15(5), pp.1687–1702. doi:<https://doi.org/10.1007/s12559-023-10151-6>.

[4] Alizadehsani, R., Hosseini, M.J., Khosravi, A., Khozeimeh, F., Roshanzamir, M., Sarrafzadegan, N. and Nahavandi, S. (2018). Non-invasive detection of coronary artery disease in high-risk patients based on the stenosis prediction of separate coronary arteries. *Computer Methods and Programs in Biomedicine*, 162, pp.119–127. doi:<https://doi.org/10.1016/j.cmpb.2018.05.009>.

[5] Annalsmedres.org. (2023). *Machine learning-based forecasting of coronary artery disease risk factors and diagnostic insights*. [online] Available at: <https://www.annalsmedres.org/index.php/aomr/article/view/4527> [Accessed 5 Apr. 2024].

[6] Bilal, M., Aslam, N., Naeem, A. and Muhammad Kamran Abid (2023). Using Machine Learning Models for The Prediction of Coronary Arteries Disease. *VFAST transactions on software engineering*, 11(2), pp.149–159. doi:<https://doi.org/10.21015/vtse.v11i2.1487>.

[7] Chen, S.-F., Loguercio, S., Chen, K.-Y., Sang Eun Lee, Park, J.-B., Liu, S., Hossein Javedani Sadaei and Torkamani, A. (2023). Artificial Intelligence for Risk Assessment on Primary Prevention of Coronary Artery Disease. *Current Cardiovascular Risk Reports*, 17(12), pp.215–231. doi:<https://doi.org/10.1007/s12170-023-00731-4>.

[8] D’Ancona, G., Massussi, M., Savardi, M., Signoroni, A., Di Bacco, L., Farina, D., Metra, M., Maroldi, R., Muneretto, C., Ince, H., Costabile, D., Murero, M., Chizzola, G., Curello, S. and Benussi, S. (2023). Deep learning to detect significant coronary artery disease from plain chest radiographs AI4CAD. *International Journal of Cardiology*, [online] 370, pp.435–441. doi:<https://doi.org/10.1016/j.ijcard.2022.10.154>.

[9] Forrest, I.S., Petrazzini, B.O., Duffy, Á., Park, J.K., Marquez-Luna, C., Jordan, D.M., Rocheleau, G., Cho, J.H., Rosenson, R.S., Narula, J., Nadkarni, G.N. and Do, R. (2023). Machine learning-based marker for coronary artery disease: derivation and validation in two longitudinal cohorts. *Lancet (London, England)*, [online] 401(10372), pp.215–225. doi:<https://doi.org/10.1016/S0140-6736(22)02079-7>.

[10] G., R. and S. N., J. (2023). Prediction of Coronary Artery Disease using Artificial Intelligence – A Systematic Literature Review. *International Journal of Health Sciences and Pharmacy*, pp.1–32. doi:<https://doi.org/10.47992/ijhsp.2581.6411.0097>.

[11] Gangadhar, M.S., Sai, K.V.S., Kumar, S.H.S., Kumar, K.A., Kavitha, M. and Aravinth, S.S. (2023). Machine Learning and Deep Learning Techniques on Accurate Risk Prediction of Coronary Heart Disease. *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*. doi:<https://doi.org/10.1109/iccmc56507.2023.10083756>.

[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] Heo, J., Yoo, J., Lee, H., Lee, I.H., Kim, J.-S., Park, E., Kim, Y.D. and Nam, H.S. (2022). Prediction of Hidden Coronary Artery Disease Using Machine Learning in Patients With Acute Ischemic Stroke. *Neurology*, 99(1), pp.e55–e65. doi:<https://doi.org/10.1212/wnl.0000000000200576>.

[14] Huang, P.-S., Tseng, Y.-H., Tsai, C.-F., Chen, J.-J., Yang, S.-C., Chiu, F.-C., Chen, Z.-W., Hwang, J.-J., Chuang, E.Y., Wang, Y.-C. and Tsai, C.-T. (2022). An Artificial Intelligence-Enabled ECG Algorithm for the Prediction and Localization of Angiography-Proven Coronary Artery Disease. *Biomedicines*, 10(2), p.394. doi:<https://doi.org/10.3390/biomedicines10020394>.

[15] Kumar, R., Aggarwal, Y. and Vinod Kumar Nigam (2022). Autonomic Features in Prediction of Coronary Artery Disease and Myocardial Infarction. *IETE Journal of Research*, pp.1–8. doi:<https://doi.org/10.1080/03772063.2022.2069162>.

[16] Lee, Y.-H., Hsieh, M.-T., Chang, C.-C., Tsai, Y.-L., Chou, R.-H., Lu, H.H.-S. and Huang, P.-H. (2023). Improving detection of obstructive coronary artery disease with an artificial intelligence-enabled electrocardiogram algorithm. *Atherosclerosis*, [online] 381, p.117238. doi:<https://doi.org/10.1016/j.atherosclerosis.2023.117238>.

[17] Michał Woś, Drop, B. and Kiczek, B. (2023). Predicting and Detecting Coronary Heart Disease in Patients Using Machine Learning Method. *Lecture Notes in Computer Science*, pp.367–377. doi:<https://doi.org/10.1007/978-3-031-34953-9_29>.

[18] Orlenko, A., Kofink, D., Lyytikäinen, L.-P., Nikus, K., Mishra, P., Kuukasjärvi, P., Karhunen, P.J., Kähönen, M., Laurikka, J.O., Lehtimäki, T., Asselbergs, F.W. and Moore, J.H. (2019). Model selection for metabolomics: predicting diagnosis of coronary artery disease using automated machine learning. *Bioinformatics*, 36(6), pp.1772–1778. doi:<https://doi.org/10.1093/bioinformatics/btz796>.

[19] Özbilgin, F., Kurnaz, Ç. and Aydın, E. (2023). Prediction of Coronary Artery Disease Using Machine Learning Techniques with Iris Analysis. *Diagnostics*, [online] 13(6), p.1081. doi:<https://doi.org/10.3390/diagnostics13061081>.

[20] Predicting Coronary Artery Disease Utilizing Support Vector Machines: Optimizing Predictive Model. (2023). pp.21–26. doi:<https://doi.org/10.58496/mjaih/2023/004>.

[21] Sayadi, M., Varadarajan, V., Sadoughi, F., Chopannejad, S. and Langarizadeh, M. (2022). A Machine Learning Model for Detection of Coronary Artery Disease Using Noninvasive Clinical Parameters. *Life*, 12(11), p.1933. doi:<https://doi.org/10.3390/life12111933>.

[22] Tang, P., Wang, Q., Ouyang, H., Yang, S. and Hua, P. (2023). The feasibility of early detecting coronary artery disease using deep learning-based algorithm based on electrocardiography. *Aging*. doi:<https://doi.org/10.18632/aging.204688>.

[23] Trigka, M. and Dritsas, E. (2023). Long-Term Coronary Artery Disease Risk Prediction with Machine Learning Models. *Sensors*, 23(3), p.1193. doi:<https://doi.org/10.3390/s23031193>.

[24] Verma, L., Srivastava, S. and Negi, P.C. (2017). An intelligent noninvasive model for coronary artery disease detection. *Complex & Intelligent Systems*, 4(1), pp.11–18. doi:<https://doi.org/10.1007/s40747-017-0048-6>.

[25] Yang, H., Chen, Z., Yang, H. and Tian, M. (2023). Predicting coronary heart disease using an improved LightGBM model: Performance analysis and comparison. *IEEE Access*, pp.1–1. doi:<https://doi.org/10.1109/access.2023.3253885>.

# Appendix: Source Code For the Implementation On Jupyter Notebook

#!/usr/bin/env python

# coding: utf-8

**# # Required Packages, Modules and Utilities**

# In[1]:

# Modules

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

**# Models**

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import LinearSVC

from sklearn.linear\_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split, KFold, GridSearchCV

**# Evaluation Metrics**

from sklearn import metrics

from sklearn.metrics import confusion\_matrix, classification\_report, ConfusionMatrixDisplay

from sklearn.metrics import roc\_auc\_score, RocCurveDisplay

get\_ipython().run\_line\_magic('matplotlib', 'inline')

import warnings

warnings.simplefilter("ignore")

**# # Data Extraction**

# In[2]:

# Import csv file to DataFrame format

cad\_data = pd.read\_csv("cad\_dataset.csv")

# In[3]:

# Show first five rows

cad\_data.head()

**# # Data Cleaning**

# In[4]:

# Function to get dataset summary

def get\_dataset\_summary(df):

print('.' \* 100)

print(f'No. of Rows: {df.shape[0]} No. of Columns: {df.shape[1]}')

print('.' \* 100)

data\_summary = pd.DataFrame({

'DataType': df.dtypes,

'Unique Values': df.nunique(),

'Missing Values': df.isnull().sum(),

'Infinity Values': df.isin([np.inf, -np.inf]).sum()

})

print(data\_summary)

print('.' \* 100)

# In[5]:

# First view of dataset

get\_dataset\_summary(cad\_data)

**# # Exploratory Data Analysis**

# In[6]:

# Show data

cad\_data.info()

# In[7]:

# Shows the percentiles of numerical data (datatype - float & int)

# Switch index and columns

cad\_data.describe().transpose()

**# ## Visualizations**

# In[8]:

# Map values of the 'target' column to categories

cad\_data['target\_category'] = cad\_data['target'].map(lambda a: 'No CAD Disease' if a == 0 else 'CAD Disease')

print(cad\_data['target'].value\_counts(), '\n')

print(cad\_data['target\_category'].value\_counts())

plt.figure(figsize=(4, 4))

sns.countplot(data=cad\_data, x='target\_category', palette=['#00BFFF', '#D21F3C'])

plt.show()

# In[9]:

# Map values of the 'age' column to categories

# Group data by age ranges

age\_bins = [28, 40, 50, 60, 70, 80]

age\_labels = ['30s', '40s', '50s', '60s', '70s']

cad\_data['age\_category'] = pd.cut(cad\_data['age'], bins=age\_bins, labels=age\_labels, right=False)

print(cad\_data['age\_category'].value\_counts())

plt.figure(figsize=(4, 4))

sns.countplot(data=cad\_data, x='age\_category')

plt.show()

# Calculate incidence (%) of cardiovascular disease by age group

print(cad\_data.groupby('age\_category')['target\_category'].value\_counts(normalize=True))

print('\n')

age\_category\_percentages = cad\_data.groupby('age\_category')['target\_category'].value\_counts(normalize=True).unstack() \* 100

print(age\_category\_percentages)

age\_category\_percentages.plot(kind='bar', stacked=True, color=['#F40009', '#3457D5'], rot=0)

plt.ylabel('Percentage')

plt.show()

# In[10]:

# Map values of the 'sex' column to categories

cad\_data['sex\_category'] = cad\_data['sex'].map(lambda a: 'Male' if a == 0 else 'Female')

print(cad\_data['sex'].value\_counts(), '\n')

print(cad\_data['sex\_category'].value\_counts(), '\n')

plt.figure(figsize=(4, 4))

sns.countplot(data=cad\_data, x='sex\_category', order=['Male', 'Female'])

plt.show()

# Calculate incidence (%) of cardiovascular disease by gender

print(cad\_data.groupby('sex\_category')['target\_category'].value\_counts(), '\n')

gender\_percentages = cad\_data.groupby('sex\_category')['target\_category'].value\_counts(normalize=True).unstack() \* 100

print(gender\_percentages)

gender\_percentages.plot(kind='bar', stacked=True, color=['#F40009', '#3457D5'], rot=0)

plt.ylabel('Percentage')

plt.show()

# In[11]:

# Map values of the 'restecg' column to categories

rest\_ecg\_to\_cat\_func = lambda a: 'Normal' if a == 0 else 'ST-T Wave Abnormality ' if a == 1 else 'Hypertrophy'

cad\_data['restecg\_category'] = cad\_data['restecg'].map(rest\_ecg\_to\_cat\_func)

print(cad\_data['restecg'].value\_counts(), '\n')

print(cad\_data['restecg\_category'].value\_counts(), '\n')

# Count the occurrences of each value of resting electrocardiographic results for each target category

print(cad\_data.groupby('target\_category')['restecg\_category'].value\_counts())

sns.countplot(data=cad\_data, hue='restecg\_category', x='target\_category')

# In[12]:

# Show average maximum heart rate achieved during a stress test

# Error bar shows 95% confidence interval

print(cad\_data.groupby('target\_category')['thalach'].mean())

plt.figure(figsize=(5, 4))

sns.pointplot(data=cad\_data, x='target\_category', y='thalach', color='#E32636')

plt.ylabel('Heart Rate')

plt.grid()

plt.show()

**# # Feature Selection**

# In[13]:

# List of columns to remove

columns\_to\_remove = [col for col in cad\_data.columns if col.endswith('\_category')]

print(columns\_to\_remove)

# Removing rows where the column names end in '\_category'

training\_data = cad\_data.drop(columns=columns\_to\_remove)

**# ## Separation of Features and Target Variable**

# In[14]:

# X: Features for prediction

X = training\_data.drop(columns=['target']).values

print(X.shape)

# In[15]:

# y: Target to predict

y = cad\_data['target'].values

print(y.shape)

**# # Model Fitting and Evaluation**

# In[16]:

SEED = 42 # Define your seed value for reproducibiity

# Split data into 70% training set and 30% test set

# X: features; y: targets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=SEED)

print('Training Set Shape: ', X\_train.shape)

print('Test Set Shape: ', X\_test.shape)

# In[17]:

# Define parameter grids

# For Decision Tree

param\_grid\_dt = {

'criterion': ['gini', 'entropy'], # Split criterion

'splitter': ['best', 'random'], # Strategy to choose the split at each node

'max\_depth': [None, 10, 20], # Maximum depth of the tree

'max\_features': [None, 'log2', 'sqrt'] # Number of features to consider when looking for the best split

}

# For KNN

param\_grid\_knn = {

'n\_neighbors': [5, 7, 10], # Number of neighbors

'weights': ['uniform', 'distance'], # Different weight options

'algorithm': ['auto', 'ball\_tree', 'kd\_tree', 'brute'], # Algorithm to compute the nearest neighbors

'metric': ['euclidean', 'manhattan'] # Distance metrics

}

# For Linear SVC

param\_grid\_ls = {

'loss': ['hinge', 'squared\_hinge'], # Loss function

'tol': [1e-4, 1e-3, 1e-2], # Tolerance for stopping criteria

'C': [0.1, 1, 10], # Regularization parameter

'max\_iter': [1000, 2000, 3000] # Maximum number of iterations

}

# For Logistic Regression

param\_grid\_lr = {

'penalty': ['l1', 'l2'], # Regularization type

'C': [0.1, 1, 10], # Inverse of regularization strength

'solver': ['liblinear', 'saga'], # Solver for optimization

'max\_iter': [100, 200, 300] # Maximum number of iterations

}

# For Random Forest

param\_grid\_rf = {

'n\_estimators': [100, 200, 300], # Number of trees in the forest

'criterion': ['gini', 'entropy'], # Function to measure the quality of a split

'max\_depth': [None, 10, 20], # Maximum depth of the tree

'max\_features': [None, 'log2', 'sqrt'] # Number of features to consider when looking for the best split

}

# In[18]:

# Instantiate models and parameter grids

model\_and\_parameters = {

"Decision Tree": [DecisionTreeClassifier(random\_state=SEED), param\_grid\_dt],

"KNN": [KNeighborsClassifier(), param\_grid\_knn],

"Linear SVC": [LinearSVC(random\_state=SEED), param\_grid\_ls],

"Logistic Regression": [LogisticRegression(random\_state=SEED), param\_grid\_lr],

"Random Forest": [RandomForestClassifier(random\_state=SEED), param\_grid\_rf]

}

**# ## Grid Search Cross Validation**

# In[19]:

# Dictionary to store outcomes for each model

outcomes = {}

# Perform grid search for each model

for model\_name, (model, param\_grid) in model\_and\_parameters.items():

# Define cross-validation strategy

kfold = KFold(n\_splits=5, shuffle=True, random\_state=SEED)

# Define GridSearchCV

grid\_search = GridSearchCV(estimator=model, param\_grid=param\_grid, cv=kfold, scoring='accuracy')

# Perform grid search

grid\_search.fit(X\_train, y\_train)

# Store outcomes in dictionary

outcomes[model\_name] = {

'best\_params': grid\_search.best\_params\_,

'best\_score': grid\_search.best\_score\_,

'best\_estimator': grid\_search.best\_estimator\_,

'mean\_test\_score': grid\_search.cv\_results\_['mean\_test\_score']

}

# Print outcomes

for model\_name, outcome in outcomes.items():

print(f"Model: {model\_name}")

print("Best Parameters:", outcome['best\_params'])

print("Best Score:", outcome['best\_score'])

print("Best Estimator:", outcome['best\_estimator'])

print("Mean Test Score:", outcome['mean\_test\_score'])

print("Minimum Mean Test Score:", min(outcome['mean\_test\_score']))

print("Maximum Mean Test Score:", max(outcome['mean\_test\_score']))

print("\n")

# In[20]:

# Show distribution of mean test scores for each model

plt.figure(figsize=(9, 6))

# Define positions for each boxplot along the x-axis (5 models)

positions = range(1, 6)

for position, (model\_name, outcome) in zip(positions, outcomes.items()):

plt.boxplot(outcome['mean\_test\_score'], positions=[position], widths=0.4)

plt.xlabel('Model')

plt.ylabel('Mean Test Score')

plt.xticks([1, 2, 3, 4, 5], list(outcomes.keys()))

plt.show()

**# ## Prediction and Evaluation**

# In[21]:

target\_list = list(cad\_data['target\_category'].unique())

target\_list

# In[22]:

# Lists for evaluation metrics

model\_list = []

accuracy\_list = []

precision\_list = []

tpr\_list = [] # TPR = Recall = Sensitivity

f1\_score\_list = []

tnr\_list = [] # TNR = Specificity

auc\_roc\_list = []

for modelname, outcome in outcomes.items():

model\_name = modelname

model\_pred = outcome['best\_estimator'].predict(X\_test) # predict target

cm = confusion\_matrix(y\_test, model\_pred)

cm\_display = ConfusionMatrixDisplay(confusion\_matrix=cm, display\_labels=target\_list)

# Mapping of scorer name to scorer function

# Round up values to 3 decimal places

accuracy = round(metrics.accuracy\_score(y\_test, model\_pred), 3)

precision = round(metrics.precision\_score(y\_test, model\_pred), 3)

tpr = round(metrics.recall\_score(y\_test, model\_pred), 3)

f1\_score = round(metrics.f1\_score(y\_test, model\_pred), 3)

tnr = round(cm[0,0]/(cm[0,0]+cm[0,1]), 3)

roc\_auc = round(roc\_auc\_score(y\_test, model\_pred), 3)

# Append results to corresponding lists

model\_list.append(model\_name)

accuracy\_list.append(accuracy)

precision\_list.append(precision)

tpr\_list.append(tpr)

f1\_score\_list.append(f1\_score)

tnr\_list.append(tnr)

auc\_roc\_list.append(roc\_auc)

print("\nThe Metrics for {} are: ".format(model\_name))

print('\nAccuracy: {}'.format(accuracy))

print('Precision: {}'.format(precision))

print('TPR: {}'.format(tpr))

print('F1\_Score: {}'.format(f1\_score))

print('TNR: {}'.format(tnr))

print('AUC\_ROC: {} \n\n'.format(roc\_auc))

print(classification\_report(y\_test, model\_pred, target\_names=target\_list))

print(cm)

cm\_display.plot()

plt.show()

fig, ax = plt.subplots()

ax.plot([0, 1], [0, 1], 'r--')

RocCurveDisplay.from\_predictions(y\_test, model\_pred, name=str(model\_name), ax=ax)

plt.show()

print("==" \* 40)

**# ### Compilation of Binary Classification Results**

# In[23]:

# Show lists of results

print("Models: ", model\_list)

print("Accuracy: ", accuracy\_list)

print("Precision: ", precision\_list)

print("TPR: ", tpr\_list)

print("F1\_Score: ", f1\_score\_list)

print("TNR: ", tnr\_list)

print("AUC\_ROC: ", auc\_roc\_list)

# In[24]:

# Convert lists to a single DataFrame

dict\_of\_lists = {

"Model": model\_list,

"Accuracy": accuracy\_list,

"Precision": precision\_list,

"F1\_Score": f1\_score\_list,

"TPR": tpr\_list,

"TNR": tnr\_list,

"AUC\_ROC": auc\_roc\_list

}

all\_results = pd.DataFrame(dict\_of\_lists)

all\_results

# In[25]:

**# Show all ROC\_AUC curves in one chart**

ax = plt.gca()

ax.plot([0, 1], [0, 1], 'r--')

for modelname, outcome in outcomes.items():

model\_name = modelname

model\_pred = outcome['best\_estimator'].predict(X\_test)

RocCurveDisplay.from\_predictions(y\_test, model\_pred, name=str(model\_name), ax=ax)

plt.show()

# In[26]:

**# Display accuracies**

plt.figure(figsize=(9,5))

sns.barplot(data=all\_results, x='Model', y='Accuracy', width=0.6,

palette=['#32CD32', '#00573F', '#90EE90', '#7CFC00', '#138808'])

plt.title("Accuracy")

plt.xlabel("Models")

plt.ylabel("Accuracy Score")

plt.show()

# In[27]:

**# Show TPR of all models**

plt.figure(figsize=(7,3))

sns.barplot(data=all\_results, x='TPR', y='Model', palette='bright',

order= all\_results.groupby('Model')['TPR'].mean().sort\_values().index)

plt.xlabel("TPR Score")

plt.ylabel("Models")

plt.show()

# In[28]:

**# Show TNR of all models**

plt.figure(figsize=(7,3))

sns.barplot(data=all\_results, x='TNR', y='Model', palette='bright',

order= all\_results.groupby('Model')['TNR'].mean().sort\_values().index)

plt.xlabel("TNR Score")

plt.ylabel("Models")

plt.show()