Heart Disease Classification

CIND 820: Capstone Project

Project by: Sherriza Khan

Student ID: 501145108

Table of Contents

[Introduction 3](#_Toc152310223)

[Data Description 6](#_Toc152310224)

[Methodology 7](#_Toc152310225)

[Stage 1: Study Replication 8](#_Toc152310226)

[Stage 2: Improved Classification 10](#_Toc152310227)

[Classifiers Developed 12](#_Toc152310228)

[Modeling Results 13](#_Toc152310229)

[Cleveland Dataset Stage 1 Results 13](#_Toc152310230)

[IEEE Dataset Stage 1 Results 15](#_Toc152310231)

[Mendeley Dataset Stage 1 Results 16](#_Toc152310232)

[Cleveland Dataset Stage 2 Results 17](#_Toc152310233)

[Feature Selection: Simple Filters 17](#_Toc152310234)

[Feature Selection: Embedded Methods 19](#_Toc152310235)

[Feature Selection: Backward Elimination 22](#_Toc152310236)

[IEEE Dataset Stage 2 Results 23](#_Toc152310237)

[Feature Selection: Simple Filters 23](#_Toc152310238)

[Feature Selection: Embedded Methods 26](#_Toc152310239)

[Feature Selection: Backwards Elimination 28](#_Toc152310240)

[Mendeley Dataset Stage 2 Results 30](#_Toc152310241)

[Feature Selection: Simple Filters 30](#_Toc152310242)

[Feature Selection: Embedded Methods 32](#_Toc152310243)

[Feature Selection: Backwards Elimination 34](#_Toc152310244)

[Conclusions & Recommendations 36](#_Toc152310245)

[GitHub Repository Link: 37](#_Toc152310246)

[Bibliography 37](#_Toc152310247)

# Introduction

Heart disease is second most common cause of death in Canadians and the primary cause of hospitalisations in Canada (Government of Canada, 2017). The term heart disease broadly relates to a variety of heart conditions. Coronary heart disease also referred to as ischemic heart disease is the most frequently diagnosed form of heart disease (Government of Canada, 2017). Early detection of coronary heart disease as well as identifying primary risk factors may reduce the fatality and severity of heart disease and improve patient quality of life.

This project seeks to answer two research questions. The first is to replicate the heart disease predictive models and performance measures in the study, *A Reliable Machine Intelligence Model for Accurate Identification of Cardiovascular Diseases Using Ensemble Techniques* (2022). 3 commonly used heart disease datasets were under consideration in this study. The first dataset is the Cleveland dataset with 303 patient records collected from patients undergoing angiography at the Cleveland Clinic in Cleveland, Ohio between May 1981 and September 1984 (Detrano, et al., 1989). The dataset is accessible through the UCI Machine Learning Repository[[1]](#footnote-1) and contains 13 demographic and clinical features. 7 features within the Cleveland dataset are categorical while the remaining are numeric features (Doppala, Bhattacharyya, Janarthanan, & Baik, A Reliable Machine Intelligence Model for Accurate, 2022). Under consideration in the study is the Mendeley dataset consisting of 1000 patient records and 12 demographic and clinical features collected from patients in Indian hospitals and published on the Mendeley Data Centre in 2021[[2]](#footnote-2). 6 features within the Mendeley dataset are categorical while the remaining are numeric features. The third dataset referred to as the Comprehensive dataset is an aggregate of 5 widely used datasets consisting of 1190 angiography patient records and 11 demographic and clinical features publicly available through the IEEEDataPort[[3]](#footnote-3). 6 features within the Comprehensive dataset are categorical while the remaining are numeric features. The dataset consists of 294 patient records from the Hungarian Institute of Cardiology collected between 1983 and 1987, 120 patient records from the VA Medical Centre in Long Beach, California, USA collected between 1984 and 1987, 123 patient records collected from University Hospitals in Zurich, Switzerland and Basel, Switzerland collected in 1985 (Siddhartha, 2023). The dataset also contains the Cleveland dataset with 303 patient records and 270 patients records from the Statlog heart disease dataset (Siddhartha, 2023). 6 features within the Comprehensive dataset are categorical while the remaining are numeric features. There were 11 common demographic and clinical features across the three selected datasets.

The authors developed an ensemble model using Naïve Bayes, Random Forest, Support Vector Machine (SVM) and Gradient Boosting classifiers. The accuracy of the proposed ensemble model was compared to Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, Support Vector Machine (SVM), Gradient Boosting and Extreme Gradient Boosting (XGBoost) classification models. The performance of each model was evaluated on the basis of specificity, precision, recall, F1-score and Matthew’s Correlation Coefficient (Doppala, Bhattacharyya, Janarthanan, & Baik, A Reliable Machine Intelligence Model for Accurate, 2022).

Secondly, this study will seek to improve the classification accuracy of heart disease across the 3 datasets. Feature selection will be performed using filter, wrapper and embedded methods. Filter methods under investigation include the application of chi-squared, odds-ratio and basic t-tests (Kuhn & Johnson, 2019). Wrapper methods applied will include backward elimination (Kuhn & Johnson, 2019). Embedded methods under consideration will include permutation-based feature importances (Kuhn & Johnson, 2019). Feature importance metrics will be evaluated for only Random Forest, Gradient and XGBoost classifiers (scikit-learn developers, 2023). The proposed study will also include an ensemble model consisting of only the Random Forest Classifier and XGBoost classifiers. Hyperparameter tuning and K-fold cross validation of the best classifier from each feature selection method will be applied for model validation and stability analysis. The computational speed and memory usage of each classification model will be used to evaluate model efficiency.

This project will use Python to preprocess the data and develop the classifiers. Classifiers will be developed in Scikit-learn using the following classes: sklearn.naive\_bayes.GaussianNB for Gaussian Naïve Bayes, sklearn.tree.DecisionTreeClassifier for Decision Tree, sklearn.ensemble.RandomForestClassifier for Random Forest, sklearn.linear\_model.LogisticRegression for Logistic Regression, sklearn.svm.SVC for Support Vector Machine (SVM), sklearn.ensemble.GradientBoostingClassifier for Gradient Boosting, xgboost.XGBClassifier for Extreme Gradient Boosting (XGBoost) Classifier and sklearn.ensemble.VotingClassifier for Ensemble Models (scikit-learn developers, 2023). The sklearn.metrics module will be used for calculating performance metrics, the psutil and timeit libraries will be used for memory usage and computational speed respectively (2023 Python Software Foundation, 2023).

# Data Description

For ease of use, the features in each dataset are listed in the table below along with a brief description.

**Table 1:Features and Description across datasets** (Doppala, Bhattacharyya, Janarthanan, & Baik, A Reliable Machine Intelligence Model for Accurate Identification of Cardiovascular Diseases Using Ensemble Techniques, 2022)

|  |  |  |  |
| --- | --- | --- | --- |
| **Cleveland Features** | **Comprehensive Dataset Features** | **Mendeley Dataset Features** | **Description** |
| Age | Age | Age | Age of patient in years |
| Sex | Sex | Gender | 1: Female  0: Male |
| cp | Chest Pain Type | Chest Pain | Categorizes chest pain type  0: typical angina  1: atypical angina  2: non-anginal pain  3: asymptomatic |
| trestbps | Resting BP | Resting BP | Resting blood pressure at the time of hospital admission reported in mmHg |
| Chol | Cholesterol | Serum Cholesterol | Serum cholesterol reported in mg/dl |
| fbs | Fasting Blood Sugar | Fasting Blood Sugar | Fasting blood sugar in excess of 120 mg/dl?  1: True  0: False |
| restecg | Resting ECG | Resting Electro | Summarizes the resting state electrocardiographic results  0: Normal results  1: Indicates an abnormality in the ST-T segment such as T-wave inversion, ST elevation or ST depression in excess of 0.05 mV  2: Results most likely indicate left ventricular hypertrophy measured according to Estes’ criteria |
| thalach | Max Heart Rate | Max Heart Rate | Maximum heart rate achieved |
| exang | Exercise Angina | Exercise Angina | Exercise induced angina?  1: True  0: False |
| OldPeak | Oldpeak | Oldpeak | Degree of depression of ST segment within electrocardiogram induced by exercise |
| Slope | ST slope | Slope | Categorizes the slope at the ST segment during peak exercise  With regards to Cleveland and IEEE Dataset:  Level 1: Upsloping  Level 2: Flat  Level 3: Downsloping  With regards to Mendeley Dataset:  Level 0: Normal  Level 1: Upsloping  Level 2: Flat  Level 3: Downsloping |
| ca |  | No. of Major Vessels | The discrete number of major blood vessels that appear coloured after a fluoroscopy procedure ranges from 0-3 |
| thal |  |  | Presence of the blood disorder thalassemia  1: Fixed defects characterized by the absence of blood flow within some areas of the heart  2: Normal blood flow  3: Reversible defect characterized by the presence of abnormal blood flow |
| Target | Target | Target | 1: Heart disease present  0: No heart disease |

# Methodology

This study will be performed in 2 stages. Stage 1 is replication of the study by Doopala et al., and stage 2 is development of improved classifiers using the study datasets. A workflow has been produced for both techniques to be completed at each stage. The stages will be completed in Python.

Each dataset under consideration in the study has been saved in a separate Python notebook as follows:

Initial Codes – Cleveland

* Documents code relevant to the Cleveland Clinic Dataset obtained from the UCI Machine Learning Repository

Initial Codes – IEEE

* Documents code relevant to the Comprehensive Heart Disease Dataset obtained from the IEEE Datamart

Initial Codes – Mendeley

* Documents relevant to the Cardiovascular Disease Dataset obtained from the Mendeley Data Center

## Stage 1: Study Replication

1. Remove blanks and duplicates from each dataset
2. Perform one-hot encoding to categorical nominal features with more than 2 levels
3. Perform 60:40 test, training split
4. Fit min-max scaler to numerical training data use results to transform testing data
5. Use training set to develop the following algorithms:
   1. Decision Tree
   2. Random Forest
   3. Naive Bayes
   4. Logistic regression
   5. SVM
   6. Gradient boosting
   7. XGBoost
   8. Ensemble algorithm consisting of
6. Naive Bayes
7. Random Forest
8. SVM
9. Gradient Boosting
10. Apply algorithms to test set
11. Evaluate the following performance metrics:
    1. Specificity
    2. Precision
    3. Recall
    4. F1-score
    5. Matthew’s Correlation Coefficient (MCC)

Hyperparameters will be tuned if necessary to achieve an accuracy as close as possible to the

study authors.

Start

Data Preprocessing: Eliminate blanks and duplicates

Perform 60:40 test, training split

Perform min-max scaling on training set and transform testing set

Use training set to develop classification algorithms:

Decision Tree

Random Forest

Naive Bayes

Logistic regression

SVM

Gradient boosting

XGBoost

Ensemble algorithm

Apply developed classification algorithms to test set

Calculate performance metrics: Accuracy, Precision, Recall, Specificity, F1 Score, MCC

Figure : Stage 1 study replication workflow

Classifier performance was evaluated using Matthew’s Correlation Coefficient (MCC) and ranges between -1 to 1. A value of -1 indicates a classification wherein there is complete reversal between the predicted and actual class labels (Statology, 2023). An MCC of zero corresponds to a scenario similar to random guessing and 1 indicates a perfect classification. The MCC is best used to evaluate the performance of classification models wherein there is an imbalance of target class labels (Statology, 2023). The formula for the MCC equation is presented below (Statology, 2023).

Specificity provides a measure of the number of true negatives as a proportion of the number of to true negatives and false positives. In other words, it provides a measure of evaluating a classifier’s ability to capture true negatives.

Hyperparameters were tuned when necessary in an attempt to replicate the study results as closely as possible.

### Stage 2: Improved Classification

Stage 2 of the project applies a number of different techniques to improve the performance of the classification. The workflow is presented below. The RobustScaler() in Python was applied as opposed to the MinMaxScaler(). The datasets have a degree of skewness particularly in the cholesterol feature. The advantage of the RobustScaler() is that it is less sensitive to outliers than the MinMaxScaler() (Brownlee, 2020). Robust scaling involves subtracting the median value from each data point and dividing by the interquartile range (Brownlee, 2020).

1. Remove blanks and duplicates from each dataset
2. Perform one-hot encoding to categorical nominal features with more than 2 levels
3. Perform 60:40 test, training split
4. Fit RobustScaler() scaler to numerical training data use results to transform testing data
5. Perform feature selection to training set
   1. First Round: Simple Filter methods
      1. Apply chi-squared tests for categorical features with 3 or more levels
      2. Apply odds ratio tests for categorical features with 2 levels
      3. Apply basic t-tests and for integer features
   2. Second Round: Wrapper methods
      1. Apply backward elimination using mlxtend SequentialFeatureSelector
   3. Third Round: Embedded methods
      1. Use impurity-based feature importance to perform feature selection for Random Forest, Gradient Boosting and XGBoost classifiers
6. Use training set subjected to feature selection to develop and tune hyperparameters the following algorithms:
   1. Decision Tree
   2. Random Forest
   3. Naive Bayes
   4. Logistic regression
   5. SVM
   6. Gradient boosting
   7. XGBoost
   8. Ensemble algorithm consisting of Random Forest and Gradient Boosting
7. Apply algorithms to test set and tune hyperparameters of best performing classifier
8. Perform k-folds cross-validation (where k=10) on consistent dataset for best classifier
9. Evaluate the following performance metrics:
   1. Specificity
   2. Precision – relate it to target class (diseased may be more important)
   3. Recall – related it to target class
   4. F1-score
   5. Matthew’s Correlation Coefficient
   6. Computational Speed
   7. Memory Consumption
10. Evaluate classifier stability by analyzing the variance between folds

Feature selection was performed using simple filters, wrapper methods and embedded methods. Simple filters evaluate the relationship between a single predictor and a target (Johnson & Kuhn, 2019). Filter methods vary depending on variable type. While basic t-tests can be used for numerical features and binary targets, chi-squared and odds ratios are used to evaluated the relationship between categorical and binary features and binary targets respectively (Johnson & Kuhn, 2019). Wrapper methods perform feature selection and model fitting in an iterative process wherein a model is fitted with a specified number of features and a specified performance metric to be optimized is calculated (Johnson & Kuhn, 2019). The results of the performance metric will guide the addition or removal of further features in subsequent iterations. Unlike simple filter and wrapper methods, embedded methods are intrinsic to the modeling process (Johnson & Kuhn, 2019).

Embedded feature selection is applicable to tree-based classifiers such as Random Forest, Gradient Boosting and XGBoost classifiers applied within this project. The mean permutation-based feature importance was calculated for each of the aforementioned classifiers. Features that provide a greater degree split within the decisions trees are assigned a higher importance and is calculated for each feature in the dataset. More specifically, feature importance is calculated for each decision tree within tree-based methods. The calculation takes into account the degree to which the feature split improves the model performance and is weighted by the total instances impacted by the node. While impurity-based methods are biased to high cardinality features, permutation feature importances are not. The permutation method evaluates the change in the model’s performance that occurs when dataset is randomly shuffled ( scikit-learn, 2023).

### Classifiers Developed

Random Forest, Gradient Boosting and XGBoost are all tree-based classifiers. Random Forest consists of multiple decision trees whereas gradient boosting consists of a series of poorly performing decision trees that sequentially correct the errors of the prior tree (sci-kit learn, 2023). XGBoost is an optimized form of a gradient boosted tree that includes pruning where portions of decision trees that do not contribute significantly to model improvement are removed from the model entirely (Brownlee, Jason, 2021). An ensemble model applies a collection of poorly performing classifiers, otherwise known as weak learners to develop a classification model. Each individual model, though weak, compensates for the performance of the remaining models. For example, instances correctly classified by one model are incorrectly classified by the remaining models (Jason Brownlee, 2021).

# Modeling Results

## Cleveland Dataset Stage 1 Results

The table presented below details the results of the stage 1 modeling replicating the procedure applied by the study authors. For reference the study results are presented in the figure below. False negatives are of particular importance for disease classification and this is summarized by the recall metric. The F1 score approximates the trade off between recall and precision a measure of accuracy of positive predictions.

Table : Cleveland Dataset Study Replication Performance Statistics

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| Decision Tree | 74.59 | 0.85 | 0.816 | 0.645 | 0.721 | 0.505 |
| Random Forest | 75.41 | 0.817 | 0.796 | 0.694 | 0.741 | 0.514 |
| Naive Bayes | 72.13 | 0.767 | 0.75 | 0.677 | 0.712 | 0.446 |
| Logistic Regression | 81.97 | 0.85 | 0.845 | 0.79 | 0.817 | 0.641 |
| Support Vector | 81.15 | 0.867 | 0.855 | 0.758 | 0.803 | 0.628 |
| Gradient Boosting | 77.87 | 0.817 | 0.807 | 0.742 | 0.773 | 0.56 |
| XGBoost | 76.23 | 0.85 | 0.824 | 0.677 | 0.743 | 0.535 |
| Ensemble | 73.77 | 0.783 | 0.768 | 0.694 | 0.729 | 0.478 |

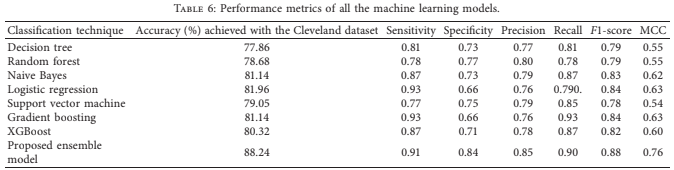


Figure : Cleveland Dataset Study Results (Doppala, Bhattacharyya, Janarthanan, & Baik, A Reliable Machine Intelligence Model for Accurate Identification of Cardiovascular Diseases Using Ensemble Techniques, 2022)

As seen in tables above, the accuracy achieved by the logistic and support vector classifiers most closely approximates the study results. The specificity of the precision and recall closely mirror that of the results and the remaining metrics closely approximate that of the study authors. The remaining classifiers do not closely align with the study results with the worst performing being the Naïve Bayes model followed by the ensemble model. The logistic model also provides the greatest recall and F1 score.

In an attempt to improve the classification of the stage 1 results and more closely align with the study results, additional hyperparameter tuning was performed on the worst performing models. Tuning of the hyperparameters using GridSearchCV did not positively impact the performance metrics except in the case of the decision tree and the ensemble model.

Table : Cleveland Dataset Study Replication Hyperparameter tuning

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| Decision Tree | 76.23 | 0.833 | 0.811 | 0.694 | 0.748 | 0.531 |
| Random Forest | 75.41 | 0.817 | 0.796 | 0.694 | 0.741 | 0.514 |
| Gradient Boosting | 75.41 | 0.783 | 0.776 | 0.726 | 0.75 | 0.51 |
| XGBoost | 76.23 | 0.85 | 0.824 | 0.677 | 0.743 | 0.535 |
| Ensemble | 75.41 | 0.783 | 0.776 | 0.726 | 0.75 | 0.51 |

The parameters tuned for the decision tree model are listed below along with the best criteria

**Decision Tree Best Parameters:**

'criterion': 'gini', 'max\_depth': None, 'min\_samples\_leaf': 4, 'min\_samples\_split': 15

**Random Forest Best Parameters:**

min\_samples\_split= 15, min\_samples\_leaf= 4, max\_depth= 10, criterion = 'entropy'

**Support Vector Machine Best Parameters:**

probability=True, random\_state = 42, kernel= 'sigmoid',gamma= 0.1, degree= 2, C= 10

**Gradient Boosting Best Parameters:**

random\_state = 42, learning\_rate=0.1, max\_depth= 4, min\_samples\_leaf=4, min\_samples\_split= 2, n\_estimators= 200

## IEEE Dataset Stage 1 Results

The table presented below details the results of the stage 1 modeling replicating the procedure applied by the study authors for the IEEE dataset. For reference the study results are presented in the figure below. There were a total 544 of duplicate records found within the IEEE. This is primarily due to the Statlog Heart Disease dataset duplicating the instances of the Cleveland dataset, a case well documented on the IEEE dataport and in literature such as the study, *Enhancing Heart Disease Prediction Accuracy through Machine Learning Techniques and Optimization* (Peddakrishna & Chandrasekhar, 2023).

Table : IEEE Dataset Study Replication Performance Statistics

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| Decision Tree | 79.89 | 0.75 | 0.819 | 0.835 | 0.827 | 0.587 |
| Random Forest | 84.78 | 0.795 | 0.855 | 0.887 | 0.87 | 0.687 |
| Naive Bayes | 85.6 | 0.801 | 0.86 | 0.896 | 0.878 | 0.704 |
| Logistic Regression | 84.78 | 0.782 | 0.848 | 0.896 | 0.872 | 0.687 |
| Support Vector | 83.7 | 0.75 | 0.83 | 0.901 | 0.864 | 0.664 |
| Gradient Boosting | 84.51 | 0.795 | 0.854 | 0.882 | 0.868 | 0.682 |
| XGBoost | 86.14 | 0.814 | 0.868 | 0.896 | 0.882 | 0.715 |
| Ensemble | 85.05 | 0.795 | 0.855 | 0.892 | 0.873 | 0.693 |

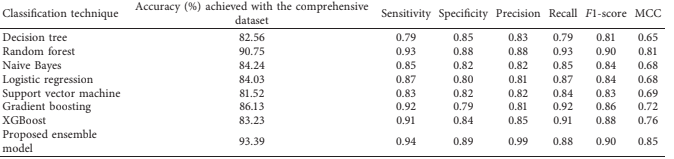


Figure : IEEE Dataset Study Results (Doppala, Bhattacharyya, Janarthanan, & Baik, A Reliable Machine Intelligence Model for Accurate Identification of Cardiovascular Diseases Using Ensemble Techniques, 2022)

Naïve Bayes, Logistic Regression, Support Vector, Gradient Boosting and XGBoost all have accuracies that approximate or exceed that of the study authors. The MCC which measures the strength of the binary classification is relatively strong in the study and somewhat weaker in the replication with the XGBoost classifier indicating the greatest strength between the predicted and true correlations. The performance metrics of the Ensemble model are lower than the those calculated by the study authors with the recall and F1 score approximating the study. The recall for all classifiers exceeds 0.80 and the models perform well at identifying true positives.

Hyperparameter tuning produced the following results. The performance metrics appear to be equal or lower than those previously calculated.

Table : IEEE Dataset performance metrics after tuning hyperparameters

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| Decision Tree | 77.45 | 0.718 | 0.797 | 0.816 | 0.807 | 0.536 |
| Random Forest | 84.78 | 0.795 | 0.855 | 0.887 | 0.87 | 0.687 |
| Gradient Boosting | 85.05 | 0.801 | 0.858 | 0.887 | 0.872 | 0.693 |
| XGBoost | 86.14 | 0.814 | 0.868 | 0.896 | 0.882 | 0.715 |

## Mendeley Dataset Stage 1 Results

The table presented below details the results of the stage 1 modeling replicating the procedure applied by the study authors for the Mendeley dataset. For reference the study results are presented in the figure below. The performance metrics closely approximates that of the study authors. The recall remains high indicating strong identification of the true positives and the MCC demonstrates a strong correlation between predicted and true classifications. As the metrics for the Mendeley dataset closely approximate the study, further hyperparameter tuning is not presented.

Table : Mendeley Dataset Study Replication Performance Statistics

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| Decision Tree | 92.75 | 0.933 | 0.944 | 0.923 | 0.934 | 0.854 |
| Random Forest | 96.00 | 0.944 | 0.956 | 0.973 | 0.964 | 0.919 |
| Naive Bayes | 94.00 | 0.911 | 0.93 | 0.964 | 0.947 | 0.879 |
| Logistic Regression | 96.00 | 0.939 | 0.952 | 0.977 | 0.964 | 0.919 |
| Support Vector | 96.75 | 0.955 | 0.964 | 0.977 | 0.971 | 0.934 |
| Gradient Boosting | 97.50 | 0.95 | 0.961 | 0.995 | 0.978 | 0.95 |
| XGBoost | 97.50 | 0.966 | 0.973 | 0.982 | 0.977 | 0.949 |
| Ensemble Model | 97.50 | 0.966 | 0.973 | 0.982 | 0.977 | 0.949 |

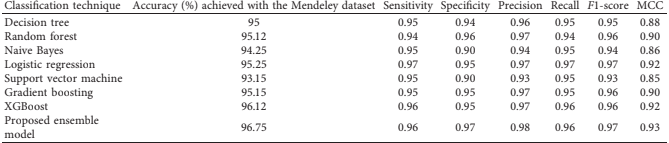


Figure : Mendeley Dataset Study Results (Doppala, Bhattacharyya, Janarthanan, & Baik, A Reliable Machine Intelligence Model for Accurate Identification of Cardiovascular Diseases Using Ensemble Techniques, 2022)

## Cleveland Dataset Stage 2 Results

### Feature Selection: Simple Filters

Following simple filter analysis, the feature trestbps, chol, fbs, restecg and features was found to have an insignificant relationship with the target variables were dropped from the dataset.

After dropping the necessary features, the classification was performed again using a 60:40 test train split. The results are presented in the table below. The computational speed was measured in seconds and the memory usage was calculated in megabytes. The logistic regression model and support vector models result in the greatest accuracy and recall. The computational speed is also lower in the logistic model compared to others, however the memory consumption is lowest for the random forest model. The MCC indicates a low to moderate binary classification with the decision tree performing the worst. On the basis accuracy, recall and efficiency logistic regression is a better performing classifier.

The accuracy of the classifiers falls short of the study paper. There are moderate improvements to sensitivity and specificity. The MCC score that defines the performance of the binary classification is reduced as are the remaining performance metrics. The performance is also poorer compared to the full dataset that was deployed in the replication stage. Note that ensemble model consists of only a gradient and random forest model.

Table : Cleveland Dataset performance metrics following simple filter feature selection

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** | **Computational Speed (s)** | **Memory Usage (MB)** |
| Decision Tree | 71.31 | 0.767 | 0.745 | 0.661 | 0.701 | 0.43 | 0.010111 | 263.77 |
| Random Forest | 72.95 | 0.817 | 0.784 | 0.645 | 0.708 | 0.468 | 0.407271 | 201.67 |
| Naive Bayes | 75.41 | 0.817 | 0.796 | 0.694 | 0.741 | 0.514 | 0.046504 | 202.7 |
| Logistic Regression | 79.51 | 0.817 | 0.814 | 0.774 | 0.793 | 0.591 | 0.022038 | 203.27 |
| Support Vector | 79.51 | 0.867 | 0.849 | 0.726 | 0.783 | 0.598 | 0.033995 | 244.57 |
| Gradient | 77.87 | 0.817 | 0.807 | 0.742 | 0.773 | 0.56 | 0.108624 | 203.93 |
| XGBoost | 77.05 | 0.833 | 0.815 | 0.71 | 0.759 | 0.547 | 0.760564 | 222.71 |
| Ensemble | 73.77 | 0.8 | 0.778 | 0.677 | 0.724 | 0.481 | 0.882053 | 261.97 |

Hyperparameters were tuned for the best performing model in terms of accuracy and recall, that being the logistic regression model. The Tuning parameters are listed below. There is an improvement to the performance metrics using the tuned parameters. Specificity has declined and recall and MCC remain high indicating a binary classification and identification of true positives.

Table : Cleveland dataset classifiers with tuned hyperparameters

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| Logistic Regression | 85.6 | 0.801 | 0.86 | 0.896 | 0.878 | 0.704 |

**Logistic Regression Best Parameters:**

{'C': 1.0, 'max\_iter': 100}

A stratified k-fold cross validation was performed for the top model and the results presented below. The number of folds used was 10, the number of repetitions was 3. K-folds cross validation provides a better estimate of the accuracy of the data and was applied to the consistent dataset. The performance metrics appear improved compared to the hold-out method. The mean of metrics is presented below. The standard deviation is presented below as a percent and appears to show low spread with the exception of the MCC metric being somewhat higher. The low standard deviations indicate good model stability.

Table : Cleveland dataset classifiers with tuned hyperparameters mean performance metrics

|  |  |
| --- | --- |
| **Accuracy** | 85.36 |
| **Precision** | 0.86 |
| **Recall** | 0.82 |
| **F1 Score** | 0.83 |
| **MCC** | 0.71 |
| **Specificity** | 0.89 |

Table : Cleveland dataset classifiers with tuned hyperparameters standard deviation performance metrics

|  |  |
| --- | --- |
| **Accuracy** | 5.55 |
| **Precision** | 0.07 |
| **Recall** | 0.09 |
| **F1 Score** | 0.07 |
| **MCC** | 0.11 |
| **Specificity** | 0.07 |

### Feature Selection: Embedded Methods

Embedded methods were tested on the random forest, gradient boosting and xgboost classifiers and the results are presented below.

|  |  |
| --- | --- |
|  |  |

Figure : Feature Importance Results for Selected Classifiers

With regards to permutation-based measures magnitude is the factor that determines the importance. The following variables were dropped for each classifier.

Table : Features removed following feature importance analysis

|  |  |
| --- | --- |
| **Classifier** | **Variables Dropped** |
| Random Forest | ['restecg\_2', 'chol','cp\_2', 'fbs', 'ca\_2', 'slope\_2', 'thal\_2', 'restecg\_1', 'cp\_1', 'sex', 'oldpeak'] |
| Gradient Boosting | ['thal\_2','restecg\_1','cp\_1','age','chol','fbs','ca\_3','restecg\_2','cp\_2','slope\_1'] |
| XGBoost | ['restecg\_2','ca\_3', 'cp\_1','thal\_2' 'slope\_2','restecg\_1','cp\_1','exang','ca\_1'] |

The model was trained and tested using a 60:40 test train split and the results are presented below. The calculated accuracy is lower than the study authors. The accuracy has improved compared to the full dataset for the random forest model. Since recall is of greatest importance the gradient boosting model was subject to hyperparameter tuning and k-fold cross validation to obtain a better measure of accuracy and stability. The results are presented in the table below. The accuracy obtained after k-folds is lower than that of a single iteration in addition, the standard deviation of the accuracy is quite high but less so for the remaining metrics. The model does appears to have a somewhat volatile stability.

Table : Cleveland dataset performance metrics for random forest classifier following embedded feature selection

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| Random Forest | 76.23 | 0.8 | 0.789 | 0.726 | 0.756 | 0.527 |
| Gradient Boosting | 77.87 | 0.883 | 0.857 | 0.677 | 0.757 | 0.572 |
| XGBoost | 72.95 | 0.717 | 0.73 | 0.742 | 0.736 | 0.459 |

Table : Cleveland dataset classifiers with mean performance metrics for Gradient Boosting Classifiers

|  |  |
| --- | --- |
| **Accuracy** | 76.57 |
| **Precision** | 0.77 |
| **Recall** | 0.71 |
| **F1 Score** | 0.73 |
| **MCC** | 0.53 |
| **Specificity** | 0.82 |

Table : Cleveland dataset classifiers standard deviation performance metrics for Gradient Boosting Classifiers

|  |  |
| --- | --- |
| **Accuracy** | 7.31 |
| **Precision** | 0.1 |
| **Recall** | 0.12 |
| **F1 Score** | 0.09 |
| **MCC** | 0.15 |
| **Specificity** | 0.1 |

### Feature Selection: Backward Elimination

The features remaining for backward elimination are presented in the table below for each classifier. An ensemble model cannot be subjected to feature selection. The feature selection was performed using mlxtend.feature\_selection in Python.

Table : Features removed during backward elimination

|  |  |
| --- | --- |
| **Classifier** | **Features Remaining** |
| Decision Tree | ('age', 'fbs', 'exang', 'oldpeak', 'cp\_3', 'slope\_2', 'ca\_3', 'thal\_2', 'thal\_3') |
| Random Forest | ('age', 'trestbps', 'chol', 'exang', 'oldpeak', 'cp\_2', 'slope\_1', 'ca\_3') |
| Naïve Bayes | ('sex', 'exang', 'cp\_1', 'cp\_2', 'cp\_3', 'slope\_1', 'slope\_2', 'ca\_1', 'ca\_2', 'ca\_3', 'thal\_2', 'thal\_3', 'trestbps', 'thalach', 'oldpeak') |
| Logistic Regression | ('age', 'sex', 'chol', 'exang', 'oldpeak', 'cp\_1', 'cp\_2', 'cp\_3', 'restecg\_2', 'slope\_1', 'slope\_2', 'ca\_1', 'ca\_2', 'thal\_2', 'thal\_3') |
| Support Vector | ('age', 'sex', 'chol', 'fbs', 'thalach', 'exang', 'cp\_1', 'cp\_3', 'restecg\_1', 'slope\_1', 'slope\_2', 'ca\_1', 'ca\_2', 'ca\_3', 'thal\_2', 'thal\_3') |
| Gradient Boosting | ('age', 'sex', 'trestbps', 'chol', 'fbs', 'thalach', 'exang', 'cp\_2', 'cp\_3', 'slope\_1', 'slope\_2', 'ca\_1', 'ca\_2', 'thal\_3') |
| XGBoost | ('age', 'trestbps', 'exang', 'oldpeak', 'cp\_2', 'cp\_3', 'slope\_1', 'ca\_1', 'ca\_3', 'thal\_3') |

The performance of all classifiers is documented below. The best performing classifier is the logistic model. It has the highest accuracy and a strong recall value. The computational speed is towards the minimum of the range and memory consumption is minimal.

Table : Cleveland Dataset performance metrics following backward elimination feature selection

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** | **Computational Speed (s)** | **Memory Usage (MB)** |
| Decision Tree | 66.39 | 0.75 | 0.706 | 0.581 | 0.637 | 0.335 | 0.014288 | 261.71 |
| Random Forest | 68.03 | 0.733 | 0.709 | 0.629 | 0.667 | 0.364 | 0.304591 | 260.94 |
| Naive Bayes | 76.23 | 0.85 | 0.824 | 0.677 | 0.743 | 0.535 | 0.07381 | 261.71 |
| Logistic Regression | 81.15 | 0.8 | 0.81 | 0.823 | 0.816 | 0.623 | 0.018614 | 260.94 |
| Support Vector | 77.87 | 0.833 | 0.818 | 0.726 | 0.769 | 0.562 | 0.022847 | 260.94 |
| Gradient | 78.69 | 0.817 | 0.81 | 0.758 | 0.783 | 0.575 | 0.069642 | 274.05 |
| XGBoost | 72.95 | 0.783 | 0.764 | 0.677 | 0.718 | 0.463 | 0.065644 | 261.71 |

K-fold cross validation was performed on the logistic model to obtain a better understanding of the model stability results are presented in the tables below. The standard deviation of the accuracy is quite not unlike the previous model, however the deviations for the remaining performance metrics are moderate.

Table : Cleveland dataset classifiers with mean performance metrics for Logistic Model

|  |  |
| --- | --- |
| **Accuracy** | 82.97 |
| **Precision** | 0.84 |
| **Recall** | 0.79 |
| **F1 Score** | 0.81 |
| **MCC** | 0.66 |
| **Specificity** | 0.86 |

Table : Cleveland dataset classifiers standard deviation performance metrics for Logistic Model

|  |  |
| --- | --- |
| **Accuracy** | 6.85 |
| **Precision** | 0.09 |
| **Recall** | 0.12 |
| **F1 Score** | 0.08 |
| **MCC** | 0.14 |
| **Specificity** | 0.09 |

A simple filter applied to the Cleveland dataset appears to give the best accuracy and strong recall using the logistic regression model and 10-fold cross-validation.

## IEEE Dataset Stage 2 Results

### Feature Selection: Simple Filters

Following simple filter analysis, the feature ‘resting ecg’ was found to have an insignificant chi-squared value and was dropped from the dataset. The table below documents the performance metrics following simple filter feature selection. There are improvements to the performance metrics compared to the full dataset. Accuracy improved for Naïve Bayes, Logistic Regression, Support Vector and Gradient Boosting classifiers. There are marked improvements to specificity as well, exempting random forest, XGBoost and the ensemble model. Recall or the true positive is of most interest during disease classification and unfortunately has declined with the simple filter method.

The filtered model performance metrics computed using a 60:40 test train split demonstrate a higher accuracy compared to the study results in the case of Naïve Bayes, Logistic Regression, Support Vector Machine and XGBoost. There are improvements to recall observed for the Decision tree, Naïve Bayes and Gradient Boosting.

The efficiency of the model can be evaluated using the computation speed and memory. The most efficient model following simple filters was logistic regression. Naïve Bayes demonstrates a comparable computational speed and the greatest accuracy and recall.

Table : IEEE Dataset Simple Filter Performance Metrics

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** | **Computational Speed (s)** | **Memory Usage (MB)** |
| Decision Tree | 79.62 | 0.763 | 0.825 | 0.821 | 0.823 | 0.583 | 0.007592 | 278.57 |
| Random Forest | 83.15 | 0.776 | 0.841 | 0.873 | 0.856 | 0.653 | 0.218987 | 278.57 |
| Naïve Bayes | 86.14 | 0.808 | 0.864 | 0.901 | 0.882 | 0.715 | 0.015864 | 278.57 |
| Logistic Regression | 85.6 | 0.801 | 0.86 | 0.896 | 0.878 | 0.704 | 0.010808 | 278.57 |
| Support Vector | 84.78 | 0.795 | 0.855 | 0.887 | 0.87 | 0.687 | 0.058467 | 278.57 |
| Gradient Boosting | 84.78 | 0.814 | 0.864 | 0.873 | 0.869 | 0.688 | 0.098315 | 278.57 |
| XGBoost | 84.51 | 0.788 | 0.851 | 0.887 | 0.868 | 0.681 | 0.054945 | 279.09 |
| Ensemble | 83.15 | 0.763 | 0.835 | 0.882 | 0.858 | 0.653 | 0.293246 | 279.09 |

As there are no hyperparameters to tune for the Naïve Bayes classifier, the Logistic Regression model was tuned. Results are presented below. Despite hyperparameter tuning, the performance metrics have not improved. K-fold cross-validation was performed to obtain a measure of model stability. The standard deviation is of the accuracy and remaining metrics appears to be reasonable. The low variance between folds would indicate that the model is stable.

**Logistic Regression Best Parameters:** {'C': 1.0, 'max\_iter': 100}

Table : IEEE dataset classifiers with tuned hyperparameters

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| Logistic Regression | 85.6 | 0.801 | 0.86 | 0.896 | 0.878 | 0.704 |

Table : IEEE dataset classifiers with mean performance metrics for Logistic Model

|  |  |
| --- | --- |
| **Accuracy** | 86.64 |
| **Precision** | 0.87 |
| **Recall** | 0.89 |
| **F1 Score** | 0.88 |
| **MCC** | 0.73 |
| **Specificity** | 0.83 |

Table : IEEE dataset classifiers standard deviation performance metrics for Logistic Model

|  |  |
| --- | --- |
| **Accuracy** | 3.37 |
| **Precision** | 0.03 |
| **Recall** | 0.04 |
| **F1 Score** | 0.03 |
| **MCC** | 0.07 |
| **Specificity** | 0.05 |

### Feature Selection: Embedded Methods

Embedded methods were tested on the random forest, gradient boosting and xgboost classifiers and the results are presented below.

|  |  |
| --- | --- |
|  |  |
|  |  |

Figure : Feature Importance Results for Selected Classifiers

With regards to permutation-based measures magnitude is the factor that determines the importance. The following variables were dropped for each classifier.

Table : Features removed following feature importance analysis

|  |  |
| --- | --- |
| **Classifier** | **Variables Dropped** |
| Random Forest | ['resting bp s', 'chest pain type\_3','max heart rate','resting ecg\_1','ST slope\_3','fasting blood sugar','chest pain type\_2'] |
| Gradient Boosting | ['age','resting ecg\_1','ST slope\_2','ST slope\_3','resting bp s','chest pain type\_3','sex','resting ecg\_2','chest pain type\_2'] |
| XGBoost | ['resting ecg\_2','resting bp s', 'resting ecg\_1','age','chest pain type\_3','ST slope\_3','exercise angina','max heart rate'] |

Table : IEEE dataset performance metrics for random forest classifier following embedded feature selection

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| Random Forest | 80.98 | 0.731 | 0.814 | 0.868 | 0.84 | 0.608 |
| Gradient Boosting | 83.42 | 0.776 | 0.842 | 0.877 | 0.859 | 0.659 |
| XGBoost | 85.33 | 0.795 | 0.856 | 0.896 | 0.876 | 0.698 |

XGBoost appears to give the highest accuracy and has performed the best across all metrics. Performing k-folds cross-validation does not result in an accuracy significantly greater than that of the original test train split. The XGBoost classifier performs better than the proposed model in the study. The standard deviation remains relatively moderate indicating good model stability.

Table : IEEE dataset classifiers with mean performance metrics for XGBoost

|  |  |
| --- | --- |
| **Accuracy** | 85.84 |
| **Precision** | 0.87 |
| **Recall** | 0.88 |
| **F1 Score** | 0.87 |
| **MCC** | 0.71 |
| **Specificity** | 0.83 |

Table : IEEE dataset classifiers standard deviation performance metrics for XGBoost

|  |  |
| --- | --- |
| **Accuracy** | 4.45 |
| **Precision** | 0.04 |
| **Recall** | 0.05 |
| **F1 Score** | 0.04 |
| **MCC** | 0.09 |
| **Specificity** | 0.06 |

### Feature Selection: Backwards Elimination

The features remaining for backward elimination are presented in the table below for each classifier. An ensemble model cannot be subjected to feature selection. The feature selection was performed using mlxtend.feature\_selection in Python.

Table : Features removed during backward elimination

|  |  |
| --- | --- |
| **Classifier** | **Features Remaining** |
| Decision Tree | ('sex', 'fasting blood sugar', 'chest pain type\_2', 'chest pain type\_3', 'chest pain type\_4', 'resting ecg\_2', 'ST slope\_2') |
| Random Forest | ('sex', 'resting bp s', 'cholesterol', 'max heart rate', 'exercise angina', 'oldpeak', 'chest pain type\_2', 'chest pain type\_3', 'chest pain type\_4', 'resting ecg\_1', 'ST slope\_1', 'ST slope\_2') |
| Naïve Bayes | ('sex', 'fasting blood sugar', 'exercise angina', 'chest pain type\_2', 'chest pain type\_3', 'chest pain type\_4', 'resting ecg\_1', 'resting ecg\_2', 'ST slope\_1', 'ST slope\_2', 'ST slope\_3', 'age', 'resting bp s', 'cholesterol', 'oldpeak') |
| Logistic Regression | ('sex', 'fasting blood sugar', 'exercise angina', 'oldpeak', 'chest pain type\_2', 'chest pain type\_4', 'resting ecg\_2', 'ST slope\_1', 'ST slope\_2', 'ST slope\_3') |
| Support Vector | ('sex', 'cholesterol', 'fasting blood sugar', 'exercise angina', 'oldpeak', 'chest pain type\_4', 'ST slope\_2') |
| Gradient Boosting | ('sex', 'fasting blood sugar', 'exercise angina', 'oldpeak', 'chest pain type\_2', 'chest pain type\_4', 'ST slope\_1', 'ST slope\_2') |
| XGBoost | ('age', 'sex', 'resting bp s', 'cholesterol', 'fasting blood sugar', 'exercise angina', 'oldpeak', 'chest pain type\_2', 'chest pain type\_4', 'ST slope\_1', 'ST slope\_2', 'ST slope\_3') |

The performance of all classifiers is documented below. The best performing classifier in terms of accuracy is the support vector machine. It has the highest accuracy and a strong recall value. The computational speed and memory usage is towards the minimum of the range.

Table : IEEE Dataset performance metrics following backward elimination feature selection

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** | **Computational Speed (s)** | **Memory Usage (MB)** |
| Decision Tree | 82.07 | 0.801 | 0.851 | 0.835 | 0.843 | 0.634 | 0.016455 | 261.59 |
| Random Forest | 83.7 | 0.776 | 0.842 | 0.882 | 0.862 | 0.664 | 0.831384 | 262.1 |
| Naive Bayes | 85.6 | 0.788 | 0.853 | 0.906 | 0.879 | 0.704 | 0.019461 | 263.91 |
| Logistic Regression | 83.15 | 0.769 | 0.838 | 0.877 | 0.857 | 0.653 | 0.011889 | 261.59 |
| Support Vector | 86.14 | 0.814 | 0.868 | 0.896 | 0.882 | 0.715 | 0.01823 | 261.84 |
| Gradient Boosting | 82.88 | 0.763 | 0.834 | 0.877 | 0.855 | 0.647 | 0.112811 | 261.84 |
| XGBoost | 84.78 | 0.814 | 0.864 | 0.873 | 0.869 | 0.688 | 0.661045 | 263.65 |

K-fold cross validation was performed on the support vector machine model to obtain a better understanding of the model stability results are presented in the tables below. The standard deviation of the accuracy is moderate and the deviations for the remaining performance metrics are moderate. The standard deviations across the folds indicate model that is stable.

Table : IEEE dataset classifiers with mean performance metrics for support vector machine

|  |  |
| --- | --- |
| **Accuracy** | 86.89 |
| **Precision** | 0.87 |
| **Recall** | 0.89 |
| **F1 Score** | 0.88 |
| **MCC** | 0.74 |
| **Specificity** | 0.84 |

Table : IEEE dataset classifiers standard deviation performance metrics for support vector machine

|  |  |
| --- | --- |
| **Accuracy** | 3.63 |
| **Precision** | 0.04 |
| **Recall** | 0.04 |
| **F1 Score** | 0.03 |
| **MCC** | 0.07 |
| **Specificity** | 0.05 |

A backward elimination applied to the IEEE dataset appears to give the best accuracy and strong recall using the support vector machine model and 10-fold cross-validation.

## Mendeley Dataset Stage 2 Results

### Feature Selection: Simple Filters

Following simple filter analysis, the features age, gender, exerciseangia were found to have insignificant relationships with the target variables and were dropped from the dataset. The table below documents the performance metrics following simple filter feature selection. The accuracy of the simple filters method closely approximates that the study results as well as the recall and remaining performance metrics. Note that the accuracy was already large.

The filtered model performance metrics computed using a 60:40 test train split demonstrate an accuracy similar to the full dataset. In the case of Logistic Regression and Support Vector Machine there is a decline in accuracy. Recall remains roughly same compared to the full dataset.

The efficiency of the model can be evaluated using the computation speed and memory. The most efficient model following simple filters was the decision tree.

Table : Mendeley Dataset Simple Filter Performance Metrics

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** | **Computational Speed (s)** | **Memory Usage (MB)** |
| Decision Tree | 93.0 | 0.927 | 0.941 | 0.932 | 0.936 | 0.859 | 0.008738 | 197.6 |
| Random Forest | 96.25 | 0.944 | 0.956 | 0.977 | 0.966 | 0.924 | 0.336036 | 202.68 |
| Naïve Bayes | 94.0 | 0.911 | 0.93 | 0.964 | 0.947 | 0.879 | 0.024183 | 203.19 |
| Logistic Regression | 94.75 | 0.922 | 0.939 | 0.968 | 0.953 | 0.894 | 0.033456 | 203.54 |
| Support Vector | 95.25 | 0.939 | 0.951 | 0.964 | 0.957 | 0.904 | 0.021854 | 204.37 |
| Gradient Boosting | 97 | 0.944 | 0.956 | 0.991 | 0.973 | 0.94 | 0.101133 | 204.37 |
| XGBoost | 96.75 | 0.95 | 0.96 | 0.982 | 0.971 | 0.934 | 0.090567 | 222.9 |
| Ensemble | 97.0 | 0.961 | 0.969 | 0.977 | 0.973 | 0.939 | 0.29127 | 223.5 |

Hyperparameters were tuned for the XGBoost model. Results are presented below. Despite hyperparameter tuning, the performance metrics have not improved. K-fold cross-validation was performed to obtain a measure of model stability. The standard deviation of the accuracy and remaining metrics appears to be reasonable. The low variance between folds would indicate that the model is stable.

**XGBoost Best Parameters:** {'colsample\_bytree': 1.0, 'learning\_rate': 0.1, 'max\_depth': 5, 'n\_estimators': 50, 'subsample': 0.9} Table : Mendeley dataset classifiers with tuned hyperparameters

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| XGBoost | 96.75 | 0.95 | 0.96 | 0.982 | 0.971 | 0.934 |

Table : Mendeley dataset classifiers with mean performance metrics for XGBoost

|  |  |
| --- | --- |
| **Accuracy** | 97.17 |
| **Precision** | 0.98 |
| **Recall** | 0.97 |
| **F1 Score** | 0.97 |
| **MCC** | 0.94 |
| **Specificity** | **0.97** |

Table : Mendeley dataset classifiers standard deviation performance metrics for XGBoost

|  |  |
| --- | --- |
| **Accuracy** | 1.58 |
| **Precision** | 0.02 |
| **Recall** | 0.02 |
| **F1 Score** | 0.01 |
| **MCC** | 0.03 |
| **Specificity** | **0.03** |

### Feature Selection: Embedded Methods

Embedded methods were tested on the random forest, gradient boosting and xgboost classifiers and the results are presented below.

|  |  |
| --- | --- |
|  |  |
|  |  |

Figure : Feature Importance Results for Selected Classifiers

With regards to permutation-based measures magnitude is the factor that determines the importance. The following variables were dropped for each classifier.

Table : Features removed following feature importance analysis

|  |  |
| --- | --- |
| **Classifier** | **Variables Dropped** |
| Random Forest | ['exerciseangia', 'age', 'chestpain\_1', 'noofmajorvessels\_1', 'chestpain\_3', 'slope\_1', 'noofmajorvessels\_3', 'fastingbloodsugar', 'restingrelectro\_1', 'gender', 'oldpeak', 'maxheartrate', 'serumcholestrol', 'noofmajorvessels\_2', 'restingrelectro\_2'] |
| Gradient Boosting | ['noofmajorvessels\_3', 'exerciseangia', 'chestpain\_3', 'restingrelectro\_1', 'age', 'noofmajorvessels\_1', 'chestpain\_1', 'noofmajorvessels\_2', 'fastingbloodsugar', 'maxheartrate', 'gender', 'serumcholestrol'] |
| XGBoost | ['exerciseangia', 'age', 'chestpain\_1', 'noofmajorvessels\_1', 'chestpain\_3', 'slope\_3', 'noofmajorvessels\_3', 'fastingbloodsugar', 'restingrelectro\_1', 'gender', 'oldpeak', 'maxheartrate', 'serumcholestrol', 'noofmajorvessels\_2', 'restingrelectro\_2'] |

Table : Mendeley dataset performance metrics for random forest classifier following embedded feature selection

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| Random Forest | 86.75 | 0.844 | 0.875 | 0.887 | 0.881 | 0.732 |
| Gradient Boosting | 94.75 | 0.922 | 0.939 | 0.968 | 0.953 | 0.894 |
| XGBoost | 95.75 | 0.944 | 0.955 | 0.968 | 0.962 | 0.914 |

XGBoost appears to give the highest accuracy and has performed the best across all metrics. Performing k-folds cross-validation does not result in an accuracy significantly greater than that of the original test train split. The XGBoost classifier performs as well as the model proposed by the study authors. The standard deviation remains low indicating good model stability.

Table : Mendeley dataset classifiers with mean performance metrics for XGBoost

|  |  |
| --- | --- |
| **Accuracy** | 95.2 |
| **Precision** | 0.96 |
| **Recall** | 0.96 |
| **F1 Score** | 0.96 |
| **MCC** | 0.9 |
| **Specificity** | 0.94 |

Table : IEEE dataset classifiers standard deviation performance metrics for XGBoost

|  |  |
| --- | --- |
| **Accuracy** | 2.09 |
| **Precision** | 0.02 |
| **Recall** | 0.02 |
| **F1 Score** | 0.02 |
| **MCC** | 0.04 |
| **Specificity** | 0.04 |

### Feature Selection: Backwards Elimination

The features remaining for backward elimination are presented in the table below for each classifier. An ensemble model cannot be subjected to feature selection. The feature selection was performed using mlxtend.feature\_selection in Python.

Table : Features removed during backward elimination

|  |  |
| --- | --- |
| **Classifier** | **Features Remaining** |
| Decision Tree | ('gender', 'restingBP', 'serumcholestrol', 'maxheartrate', 'exerciseangia', 'oldpeak', 'chestpain\_1', 'chestpain\_2', 'chestpain\_3', 'restingrelectro\_1', 'restingrelectro\_2', 'slope\_2', 'slope\_3', 'noofmajorvessels\_1', 'noofmajorvessels\_2', 'noofmajorvessels\_3') |
| Random Forest | ('gender', 'restingBP', 'serumcholestrol', 'fastingbloodsugar', 'chestpain\_1', 'chestpain\_2', 'restingrelectro\_2', 'slope\_2', 'slope\_3') |
| Naïve Bayes | ('fastingbloodsugar', 'chestpain\_1', 'chestpain\_2', 'chestpain\_3', 'restingrelectro\_1', 'restingrelectro\_2', 'slope\_2', 'slope\_3', 'noofmajorvessels\_1', 'noofmajorvessels\_2','restingBP', 'serumcholestrol', 'maxheartrate', 'oldpeak') |
| Logistic Regression | ('age', 'gender', 'restingBP', 'serumcholestrol', 'fastingbloodsugar', 'maxheartrate', 'oldpeak', 'chestpain\_1', 'chestpain\_2', 'chestpain\_3', 'restingrelectro\_1', 'restingrelectro\_2', 'slope\_1', 'slope\_2', 'slope\_3', 'noofmajorvessels\_1', 'noofmajorvessels\_2', 'noofmajorvessels\_3') |
| Support Vector | ('gender', 'restingBP', 'serumcholestrol', 'exerciseangia', 'oldpeak', 'chestpain\_1', 'chestpain\_2', 'chestpain\_3', 'restingrelectro\_2', 'slope\_1', 'slope\_2', 'slope\_3', 'noofmajorvessels\_2') |
| Gradient Boosting | ('gender', 'restingBP', 'serumcholestrol', 'fastingbloodsugar', 'maxheartrate', 'exerciseangia', 'oldpeak', 'chestpain\_2', 'chestpain\_3', 'restingrelectro\_1', 'slope\_1', 'slope\_2', 'slope\_3') |
| XGBoost | ('gender', 'restingBP', 'serumcholestrol', 'fastingbloodsugar', 'maxheartrate', 'exerciseangia', 'oldpeak', 'chestpain\_1', 'chestpain\_2', 'chestpain\_3', 'slope\_1', 'slope\_2', 'slope\_3', 'noofmajorvessels\_1', 'noofmajorvessels\_3') |

The performance of all classifiers is documented below. The best performing classifier in terms of accuracy is the XGBoost. It has the highest accuracy and recall value. The computational speed and memory usage is towards the maximum of the range. The linear methods, logistic regression and support vector machine are most efficient in terms of memory consumption, decision tree was most efficient from the perspective of computational speed.

Table : Mendeley Dataset performance metrics following backward elimination feature selection

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Specificity** | **Precision** | **Recall** | **F1 Score** | **MCC** | **Computational Speed (s)** | **Memory Usage (MB)** |
| Decision Tree | 94.75 | 0.955 | 0.963 | 0.941 | 0.952 | 0.894 | 0.012678 | 282.39 |
| Random Forest | 96.5 | 0.944 | 0.956 | 0.982 | 0.969 | 0.929 | 0.998488 | 266.73 |
| Naive Bayes | 93.75 | 0.872 | 0.905 | 0.991 | 0.946 | 0.877 | 0.043865 | 266.99 |
| Logistic Regression | 95.25 | 0.922 | 0.939 | 0.977 | 0.958 | 0.904 | 0.102162 | 266.73 |
| Support Vector | 96.75 | 0.95 | 0.96 | 0.982 | 0.971 | 0.934 | 0.05587 | 266.73 |
| Gradient Boosting | 96.25 | 0.933 | 0.948 | 0.986 | 0.967 | 0.925 | 0.107876 | 266.99 |
| XGBoost | 97.75 | 0.955 | 0.965 | 0.995 | 0.98 | 0.955 | 0.801514 | 266.99 |

K-fold cross validation was performed on the XGBoost model to obtain a better understanding of the model stability results are presented in the tables below. The standard deviation of the accuracy is low as are the deviations for the remaining performance metrics. The standard deviations across the folds indicate model that is stable.

Table : Mendeley dataset classifiers with mean performance metrics for XGBoost

|  |  |
| --- | --- |
| **Accuracy** | 97.67 |
| **Precision** | 0.98 |
| **Recall** | 0.98 |
| **F1 Score** | 0.98 |
| **MCC** | 0.95 |
| **Specificity** | 0.97 |

Table : Mendeley dataset classifiers standard deviation performance metrics for XGBoost

|  |  |
| --- | --- |
| **Accuracy** | 1.3 |
| **Precision** | 0.02 |
| **Recall** | 0.01 |
| **F1 Score** | 0.01 |
| **MCC** | 0.03 |
| **Specificity** | 0.02 |

A backward elimination applied to the Mendeley dataset appears to give the best accuracy and strong recall using the support vector machine model and 10-fold cross-validation.

# Conclusions & Recommendations

While the performance metrics of the Cleveland dataset replication phase did not match those of study authors even after hyperparameter tuning, there were instances where the results closely aligned. The logistic regression classifier and support vector classifier closely approximated the accuracy of the study metrics. In case of the IEEE dataset, the best performing classifiers in terms of accuracy and overall metrics were Naïve Bayes and Logistic Regression. High accuracy and recall were achieved in stage 1 of modeling across the Mendeley dataset comparable to study results.

The tree-based methods tended to be more computationally expensive and inefficient while in some cases underperforming linear methods. The Cleveland dataset was optimized in terms of accuracy using simple filter methods applied prior to logistic regression while recall was also strong. The accuracy of the IEEE dataset was optimized following backward elimination and application of support vector machine. The Mendeley dataset demonstrated close alignment to study results but also consistently high performance metrics. Again, backward elimination prior to support vector machine achieved the greatest accuracy with good recall. In the above cases, the model accuracy improved after the application of k-fold cross validation. The cross-validation was performed multiple times and the standard deviation across the folds indicates a fairly stable model.

The datasets evaluated within the project have a low number of instances making generalizations to larger populations difficult. The datasets have been anonymized and were collected multiple decades in past. Since the data is of a personal health nature future research would require anonymization.

Further studies could include larger datasets or the application of classifiers such as k-nearest neighbours or artificial neural networks. In addition, parametric tests such as the Mann-Whitney U test metric could be calculated for the distributions resulting from the individual classifiers used to construct the ensemble model. If the predictions result in similar distributions, a different classifier should be used in the ensemble approach. The scalability of the classifiers could be tested by manually introducing a biased record to the dataset and observing the impact. Further work could include analyzing the degree of overfitting by comparing the accuracy of the trained data to that of the test data. Using other hyperparameter tuning methods such as population-based training may aid in improving alignment of performance metrics to study results (Hertel, Sadowski, & Collado , 2023).

# GitHub Repository Link:

Relevant Codes may be found at https://github.com/CIND820/CIND820CapstoneProject

# Bibliography

scikit-learn. (2023, 11 30). *Feature importances with a forest of trees*. Retrieved from scikit-learn: https://scikit-learn.org/stable/auto\_examples/ensemble/plot\_forest\_importances.html

2023 Python Software Foundation. (2023, 10 20). *psutil 5.9.6*. Retrieved from PyPi: https://pypi.org/project/psutil/

Brownlee, J. (2020, August 28). *How to Scale Data With Outliers for Machine Learning*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/robust-scaler-transforms-for-machine-learning/

Brownlee, Jason. (2020, August 27). *Feature Importance and Feature Selection With XGBoost in Python*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/

Brownlee, Jason. (2021, February 17). *A Gentle Introduction to XGBoost for Applied Machine Learning*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

Detrano, R., Janosi, A., Steinbrunn , W., Pfisterer, M., Schmid, J.-J., Sandhu, S., . . . Froelicher, V. (1989, August). International Application of a New Probability Algorithm for the Diagnosis of Coronary Artery Disease. *Cardiology, 65*(5).

Doppala, B. P., Bhattacharyya, D., Janarthanan, M., & Baik, N. (2022, March 8). A Reliable Machine Intelligence Model for Accurate. *Hindawi Journal of Healthcare Engineering*. doi:https://doi.org/10.1155/2022/2585235

Doppala, B. P., Bhattacharyya, D., Janarthanan, M., & Baik, N. (2022, March). A Reliable Machine Intelligence Model for Accurate Identification of Cardiovascular Diseases Using Ensemble Techniques. *Hindawi Journal of Healthcare Engineering, 2022*, 13. doi:https://doi.org/10.1155/2022/2585235

Government of Canada. (2017, July 18). *Heart disease in Canada: Highlights from the Canadian Chronic Disease Surveillance System*. Retrieved from Government of Canada: https://www.canada.ca/en/public-health/services/publications/diseases-conditions/heart-disease-canada-fact-sheet.html

Hertel, L., Sadowski, P., & Collado , J. (2023, 12 01). *Population Based Training*. Retrieved from SHERPA: https://parameter-sherpa.readthedocs.io/en/latest/algorithms/keras\_mnist\_mlp\_population\_based\_training.html

Jason Brownlee. (2021, April 27). *A Gentle Introduction to Ensemble Learning*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/what-is-ensemble-learning/

Johnson, K., & Kuhn, M. (2019). *Models, Feature Engineering and Selection: A Practical Approach for Predictive.* Boca Raton: CRC Press.

Kuhn, M., & Johnson, K. (2019). *Feature Engineering and Selection : A Practical Approach for Predictive Models.* Boca Raton: CRC Press LLC.

Peddakrishna, S., & Chandrasekhar, N. (2023, March). Enhancing Heart Disease Prediction Accuracy through Machine Learning Techniques and Optimization. *processes, 11*(4), 1210. doi:https://doi.org/10.3390/pr11041210

sci-kit learn. (2023, 11 30). *sklearn.ensemble.GradientBoostingClassifier*. Retrieved from sci-kit learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

scikit-learn developers. (2023, 9 22). *1. Supervised learning*. Retrieved from scikit learn: https://scikit-learn.org/stable/supervised\_learning.html#supervised-learning

scikit-learn developers. (2023, 10 20). *Feature importances with a forest of trees*. Retrieved from scikit-learn: https://scikit-learn.org/stable/auto\_examples/ensemble/plot\_forest\_importances.html

Siddhartha, M. (2023, 10 19). *HEART DISEASE DATASET (COMPREHENSIVE)*. Retrieved from IEEE DataPort: https://ieee-dataport.org/open-access/heart-disease-dataset-comprehensive

Statology. (2023, 11 30). *How to Calculate Matthews Correlation Coefficient in Python*. Retrieved from Statology: https://www.statology.org/matthews-correlation-coefficient-python/

The imbalanced-learn developers. (2023, 10 20). *User Guides*. Retrieved from imbalanced-learn: https://imbalanced-learn.org/stable/user\_guide.html#user-guide

1. The Cleveland Heart Disease dataset is available through the UCI Machine Learning Repository at the link https://archive.ics.uci.edu/dataset/45/heart+disease [↑](#footnote-ref-1)
2. The Cardiovascular Disease dataset is publicly available through the Mendeley Data Center at the link https://data.mendeley.com/datasets/dzz48mvjht/1 [↑](#footnote-ref-2)
3. The Comprehensive Heart Disease dataset is publicly available through the IEEEDataPort at the link

   <https://ieee-dataport.org/open-access/heart-disease-dataset-comprehensive> [↑](#footnote-ref-3)