Analysis and Prediction of Heart Disease

Team 4: Alberto Cera[1], Fabio Focchi[1], Davide Fabio Loreti[1], Carlo Pegoraro[1], Flavio Yzeiri[1]

Abstract

Heart disease remains a leading cause of global mortality, necessitating early detection and intervention. This study explores the application of machine learning techniques to classify the presence of heart disease in patients using a clinical dataset sourced from Kaggle.

The dataset comprises 1025 patient entries with 14 attributes, including age, cholesterol levels, blood pressure, and other diagnostic indicators. Multiple classification models—such as Logistic Regression, Decision Trees, Random Forest, and Support Vector Machines—were trained and evaluated based on accuracy, precision, recall, and AUC-ROC metrics.

This work highlights the potential of machine learning in enhancing diagnostic workflows and supporting clinical decision-making for cardiovascular health management.

Keywords

Heart Disease — Machine Learning — Clinical Data — Classification

[1]*Universita' degli Studi di Milano Bicocca, CdLM Data Science*

Contents

Introduction

What determines the likelihood of heart disease in a patient? Clinical diagnosis often relies on a combination of demographic, physiological, and diagnostic markers, such as age, cholesterol levels, chest pain patterns, and electrocardiographic results [8]. While individual factors like elevated blood pressure or abnormal heart rate may raise suspicion, the interplay of these attributes—subjective symptoms (e.g., chest pain type) and objective measurements (e.g., resting blood pressure)—creates a complex diagnostic landscape. For instance, a patient's age or sex may influence risk stratification, yet no single metric conclusively predicts cardiovascular health [9]. Similarly, exercise-induced angina or ST segment anomalies during stress tests provide critical clues, but their interpretation depends on contextual integration with other biomarkers.

This study investigates whether machine learning models can effectively synthesize these multifaceted clinical attributes to predict heart disease presence [3]. The dataset, sourced from Kaggle, includes 1025 patient records with 14 diagnostic features, like serum cholesterol, fasting blood sugar levels, maximum heart rate achieved, and thalassemia indicators [1]. Algorithms including Logistic Regression, Decision Trees, and Random Forest were trained to classify disease status, with performance evaluated through precision, recall, and ROC-AUC metrics. Results revealed that ensemble methods, particularly Random Forest, outperformed others by leveraging feature interactions. For example, correlating exercise-induced ST depression with vessel abnormalities detected via fluoroscopy. This shows the potential of computational tools to augment traditional

diagnostic practices, offering a data-driven framework for early detection and personalized risk assessment [11].

It consists of 1025 records each with the following 14 features :

- **Age** (Numeric – ratio): Patient's age in years.
- **Sex** (Categorical – nominal): Biological sex (0 = female, 1 = male).
- **Chest pain type** (Categorical – nominal): Type of chest pain experienced (e.g., typical angina, atypical angina, non-anginal pain, asymptomatic).
- **Resting blood pressure** (Numeric – ratio): Blood pressure (mmHg) measured at rest.
- **Serum cholesterol in mg/dl** (Numeric – ratio): Serum cholesterol level (mg/dl), a blood lipid biomarker.
- **Fasting blood sugar > 120 mg/dl** (Categorical – nominal): Binary indicator of fasting blood sugar exceeding 120 mg/dl (1 = yes, 0 = no).
- **Resting electrocardiographic results** (Categorical – nominal): ECG findings at rest (e.g., normal, ST-T wave abnormality, left ventricular hypertrophy).
- **Maximum heart rate achieved** (Numeric – ratio): Highest heart rate (beats/minute) recorded during exercise.
- **Exercise induced angina** (Categorical – nominal): Presence of chest pain triggered by exercise (1 = yes, 0 = no).
- **Oldpeak** (Numeric–Continuous): ST segment depression (on ECG) during exercise compared to rest, indicating heart stress.
- **Slope of the peak exercise ST segment** (Categorical – nominal): Slope of ST segment during peak exercise (e.g., upsloping, flat, downsloping), reflecting ischemic changes.
- **Number of major vessels** (Categorical – nominal): Number of major blood vessels visible via fluoroscopy, indicating potential blockages.
- **Thal** (Categorical – nominal): Blood flow observation via thalassemia stress test (0 = normal, 1 = fixed defect, 2 = reversible defect).
- **Target** (Categorical – nominal): Whether a person has heart disease or not (0 = No, 1 = Yes).

The goal of our analysis is to predict the presence of heart disease in patients using machine learning techniques and evaluate the predictive performance of the models. This report is organized as follows:

1. **Initial data exploration:** We analyze clinical attributes from the dataset, focusing on heart disease presence.
2. **Preprocessing:** We clean the dataset by removing duplicates, encoding categorical features (e.g., chest pain type), addressing outliers and adjusting skewness.
3. **Post Data Visualization**: We visualize feature distributions after getting data cleaned to uncover real insights.
4. **Models:** We implement classification algorithms (Logistic Regression, SVM, Decision Trees, Random Forest) to predict heart disease.
5. **Valuation**: We compare model performance using metrics like accuracy, recall and AUC-ROC to identify the most reliable classifier.

1. Initial Data Exploration

This data provides information about heart disease since 1988. This data exploration is performed before applying any pre-processing steps. For target class, the ratio of healthy people and heart patients is almost same. We performed uni-variate exploratory analysis to observe the distribution of various attributes. Bi-variate exploratory analysis was also performed to see how target class is affected by each feature.

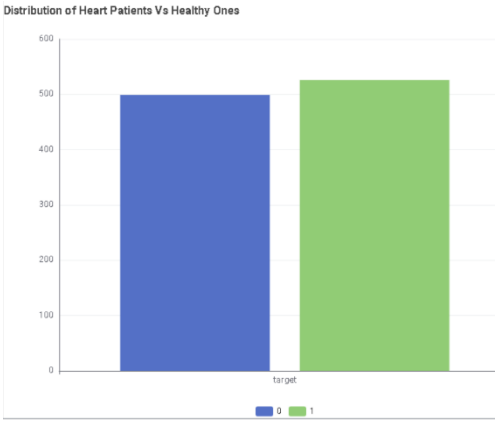

Figure 1. Distribution of Target Class

For example, we come to know that those patients having on-anginal pain have a very low chance of getting heart disease as we know that this pain occurs due to skeletal muscles or stomach issues. We also find outliers when we separately draw the distribution of target class versus age but no outliers in age attributes if box plot is drawn overall.

The distribution of this feature is shown in Figure 1.

We used bar charts to visualize categorial variables and box plots to visualize numerical variables. We also came to know that people with exercise induced angina had a very high risk of getting heart disease. Box plot visualization of distribution of age according to target class is shown below. Various important insights were also gained as a result of this data exploration process.

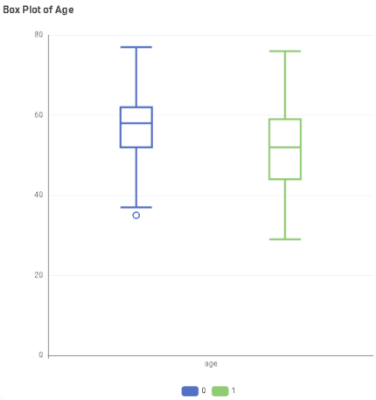

Figure 2. Age Distribution According to Target Class

2. Preprocessing

To enhance dataset's readiness for modeling, preprocessing steps were applied to address key data limitations.

2.1 Duplicates Removal and Categorical Pre-processing

First, we removed the duplicate rows from the data as there was a chance of getting biased results. There was a total of 723 duplicates. So, our new dataset had 302 rows left.

There was a total of 8 categorical columns excluding the target class. Out of those 8, two of them had more categories mentioned in the data as compared to the information provided by the owner of dataset. Those two attributes were 'ca' and 'thal'. So, there was a need for merging categories that were non-existent in the real data collection process. So, we decided to combine the least occurring category of those attributes with the most occurring category. In this way, we got rid of untrue observations that were a part of our data.

There were no missing values in the data. So, we had no trouble dealing with them [2].

By the end of this phase, we had got our data cleaned from duplicates and all the issues of categorical variables were resolved.

2.2 Feature Filtration and Outlier Detection

We performed correlation analysis to observe how closely every variable was correlated to the target class. As we know, the target class is categorical variable. So, we did not expect to get very high values of correlation for any variable. The correlation range was very low to medium. The highest value of positive correlation was observed for 'cp' variable with a correlation value of 0.432. The highest negative correlation was observed for 'exang' variable with a correlation value of -0.436. The lowest correlation value was observed for 'fbs' variable and had a value of -0.027.

We removed columns with very low values of correlations. In this regard, three variables were removed from the data namely 'chol', 'fbs' and 'restecg'. Our new dataset had only 10 features left.

Now, we have moved on to dealing with numerical variables i-e detecting and removing outliers and adjusting the skewness of variables to make them normally distributed which will help our models to perform better.

Although 'age' variable was showing some outliers when plotted separately for the target class categories. But effectively, this variable had no outliers and followed an almost normal distribution curve.

The numerical attribute 'trestbps' followed an almost normal distribution despite its outliers. To deal with normally distributed data, we use the technique of standard deviations to remove outliers. We filtered out that 15 values of this attribute were falling either above or below 2 standard deviations of this attribute. So, we filtered out those rows and removed them from the data. Now, our data has 287 rows left.

The two remaining numerical variables i-e 'thalach' and 'oldpeak' had skewed distributions. So, we cannot use the previous technique for these two variables. To detect outliers in such cases, we used the technique of interquartile range multiplied by some factor (in our case, it was 1.5) to determine outliers. 5 outliers were detected as a result of this process and were removed from the data. Now, our data has 282 rows remaining.

Another important check on numerical variables is to see whether they fall under the defined range of skewness i-e between -0.5 to 0.5. If any attribute has skewness beyond this range, then we need to apply some technique or algorithm to make the data normally distributed as a high value skewness will result in variability of the data which will affect our model outcome.

Out of all the 4 numerical variables, the 'oldpeak' variable had a skewness value of 0.9279 which is well above the defined range and make this distribution moderately skewed. So, we applied the technique of using square root of the all the values of this attribute instead of actual values. In this way, our skewness was reduced to 0.0671. In this way we got our attribute with a normal distribution.

2.3 Encoding Categorical Variables

Before moving further, we have got a completely cleaned dataset with all the necessary changes made to both types of variables. Now comes the phase where we have to encode

the categorical variables as they can't be directly used for modeling. They should be handled carefully in order to avoid problems in the development of predictive models.

So, we used the technique of one-hot encoding for those categorical variables which had more than 2 categories and the binary categorical variables that were two in number were kept as they were because binary categorical variables were already in correct form. Later on, before inputting the data into machine learning models, the target class was converted to string format so it could be accessible by the models.

2.4 Feature Scaling and Class Imbalance Check

As we know, categorical variables contain either 0 or 1. So they need not to be scaled before applying predictive models.

But as far as numerical variables are concerned, they may have values ranging between large limits. So, it may cause difficulty for ML models to comprehend them. So, we need to adopt some methodology to scale those numerical attributes so they should fall under the same range as other variables are.

For this purpose, we need to apply the method of normalization to downscale all the numeric variables within the range of 0-1. So, we got our entire data in the range between 0 and 1.

Then, another important issue is checking whether there is an imbalance in the target class or not. This is crucial as it may result in untrue predictions and biased results.

For this purpose, we divided the count of both categories of class and obtained a value of 1.27. For a class to be considered as balanced, this value should be less than or equal to 3. As we can clearly see that our values is less than 3. So, we have no class imbalance issues. To deal with such imbalance, the most used technique is SMOTE.

Finally, we had to split our data into training and test sets. We used a ratio of 0.8 (train) and 0.2 (test). The method of sampling was set to 'linear sampling'.

Now, our data is fully ready to be used as an input into our machine learning models.

3. Post Data Exploration

After getting everything preprocessed, we wanted to observe our cleaned data.

We visualized our categorical attributes to see how they were distributed according the target class (bivariate) as well as their overall distribution was also visualized (univariate).

The next things to be sure of was whether the numerical columns have gotten rid of their outliers or not.

Through visualizations, we have observed that the behaviour of categorical variables with respect to the target class is not affected and the only effect that took place was that the categories for two attributes were reduced but their distribution was almost same.

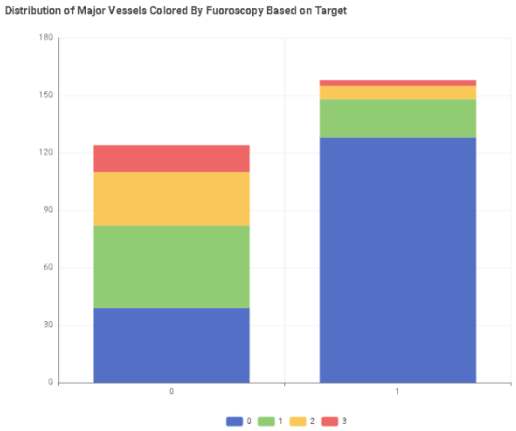

Figure 3. Vessel Distribution According to Target Class

We also observed that there were no outliers left in any numerical variable. For example, below mentioned is the distribution of resting blood pressure based on target class. We can clearly see that no outliers are left in this variable after preprocessing.

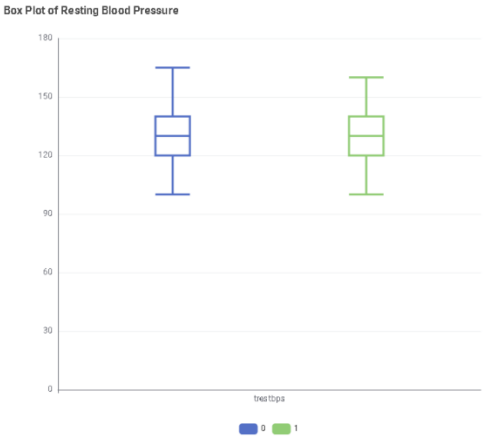

Figure 4. Vessel Distribution According to Target Class

4. Models

Now we have the task to finally think of using the most suitable models for our study. We have decided to use 4 different classification algorithms, each of which is mentioned below:

- ***Logistic Regression (LR),*** employing a sigmoid function, models binary outcomes using linear combinations of input features;
- ***Support Vector Machine (SVM),*** leveraging kernel-based transforms, by maximizing margin hyperplanes;

- ***Decision Tree (DT),*** following CART methodology, splits data using entropy/Gini criteria to form classification rules;

- ***Random Forest (RF),*** an ensemble of decision trees, aggregates predictions via majority voting to reduce overfitting.

Each of these algorithms was trained using three different approaches: Default Parameter Selection, Cross Validation and Optimized Parameter Selection.

4.1 Default Parameter Selection

In this first approach, we trained all four models (Logistic Regression, SVM, Decision Tree, Random Forest) using their default hyperparameters to establish baseline performance. The dataset, split via an 80-30 train-test ratio, retained all 14 clinical attributes—including age, cholesterol, and ST depression metrics—to reflect real-world diagnostic scenarios. Linear sampling ensured unbiased representation of heart disease cases. This phase aims to assess inherent model behavior without optimization, prioritizing generalizability and identifying initial trends in feature importance [4].

4.2 Cross Validation

To rigorously assess model robustness, each classifier (Logistic Regression, SVM, Decision Tree, Random Forest) was evaluated using *10-fold* cross-validation. The dataset was divided into 10 stratified subsets, with each iteration training on 9 subsets and testing on the remaining 1. This approach preserved the class distribution of heart disease cases (positive/negative) across folds, mitigating sampling bias. Linear sampling ensured consistency with the prior 80-30 split methodology.

By averaging performance metrics—including accuracy, precision, recall, and F1-score—across all folds, we gained insights into model stability and generalization. Unlike single train-test splits, this method reduced variance in results, particularly critical for clinical data where false negatives (missed diagnoses) carry high risks. All 14 features, such as cholesterol levels and exercise-induced angina, were retained to mirror real-world diagnostic complexity. The process highlighted how algorithms differently prioritized variables like ST depression or thalassemia results, informing subsequent hyperparameter tuning.

4.3 Optimized Parameter Selection

The last approach involved tuning the parameters of each classification model and then using those optimized parameters based on achieving the maximum accuracy of features. For this purpose, we ran parameter optimization loop for each of the models. For logistic regression model, we tuned *sigma* and *stepsize* to maximize the model accuracy. For decision tree model, we tuned *number of records per node* and *number of threads* to get the optimized performance.

For support vector machine algorithm, we tuned *overlapping penalty* to get the maximum performance of it. For random forest model, we tuned *number of models* parameter to get our optimized evaluation scores.

5. Evaluation

The evaluation techniques used are ROC curve evaluation and evaluation of accuracy, precision and recall.

5.1 Default Parameter Selection

When logistic regression model was applied using default parameters, we obtained an accuracy of 84.21 % with recall of 0.89 and precision of 0.8. Although these metrics are good enough there is still a chance of getting a better performance with other two techniques. The AUC value is 0.912 which labels this model as an excellent one. The ROC curve is shown below for logistic regression model applied with default parameters.

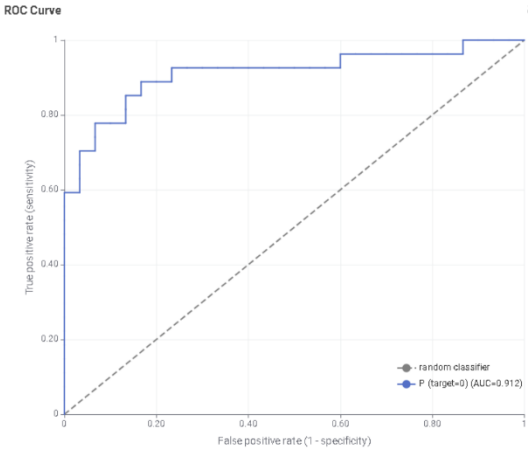

Figure 5. ROC Curve of *Logistic Regression* Model

When decision tree model was applied using default parameters, we obtained an accuracy of 77.19 % with recall of 0.741 and precision of 0.761. These metrics are not good enough in this case as it may put life of a person at risk if they are not declared as heart patient and treated at the right time. The AUC value is 0.752 which labels this model as a satisfactory one.

When support vector machine model was applied using default parameters, we obtained an accuracy of 84.21 % with recall of 0.89 and precision of 0.8. Although these metrics are good enough there is still a chance of getting a better performance with other two techniques. The AUC value is 0.912 which labels this model as an excellent one. The ROC

curve is shown below for both decision tree and SVM models applied with default parameters.

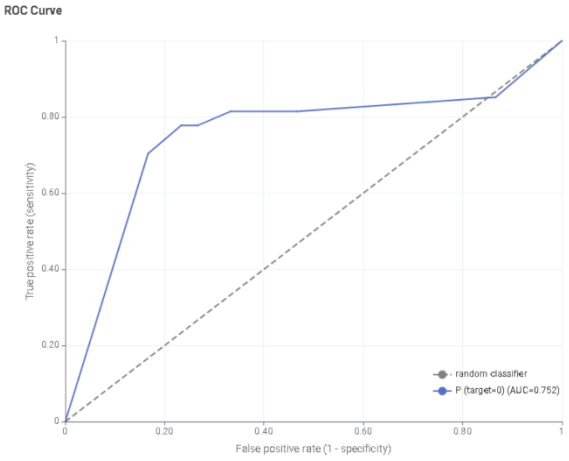

Figure 6. ROC Curve of *Decision Tree* Model

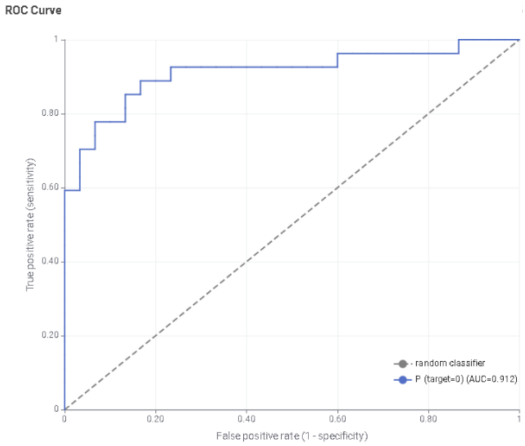

Figure 7. ROC Curve of *SVM* Model

Random Forest model applied with default parameters, gave the best accuracy of 85.97 % with recall of 0.89, precision of 0.83 and AUC value of 0.928 which labels this model as an excellent one. Its ROC curve is shown below.

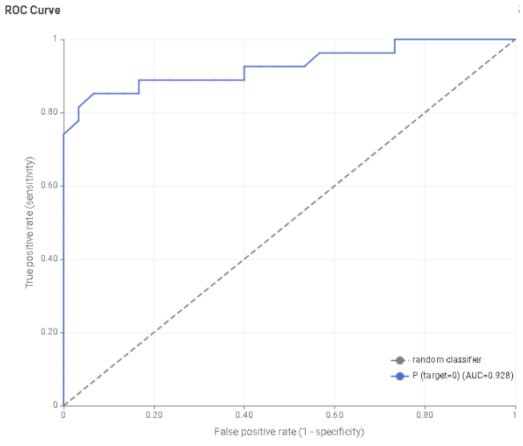

Figure 8. ROC Curve of *Random Forest* Model

5.2 Optimized Parameter Selection

When logistic regression model was applied using tuned parameter values, we obtained an accuracy of 87.72 % with recall of 0.89 and precision of 0.86. This showed that our model has improved after optimizing the parameters. The AUC value is 0.926, which labels this model as an excellent one and is better than the previous model. The ROC curve is shown below for logistic regression model applied with tuned parameters.

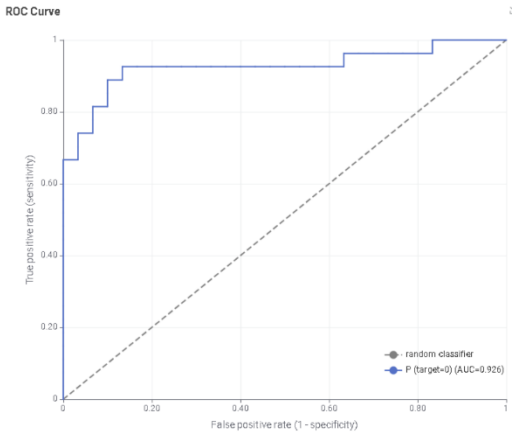

Figure 9. ROC Curve of Tuned *LR* Model

When decision tree model was applied using tuned parameter values, we obtained an accuracy of 84.21 % with recall of 0.89 and precision of 0.8. This showed that our model has improved immensely from 77 to 84 precent after optimizing the parameters. The AUC value is 0.910, which labels this model as an excellent one and is a lot better than the previous model. The ROC curve is shown below for decision tree model applied with tuned parameters.

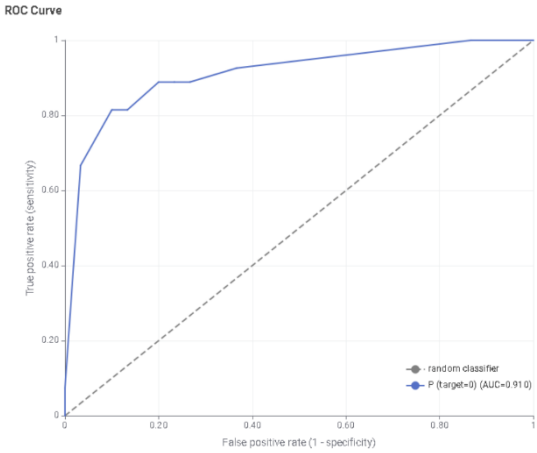

Figure 10. ROC Curve of Tuned *DT* Model

When support vector machine model was applied using tuned parameter values, we obtained an accuracy of 87.72 % with recall of 0.89 and precision of 0.86. This showed that

our model has improved after optimizing the parameters. The AUC value is 0.922, which labels this model as an excellent one and is better than the previous model. The ROC curve is shown below for SVM model applied with tuned parameters.

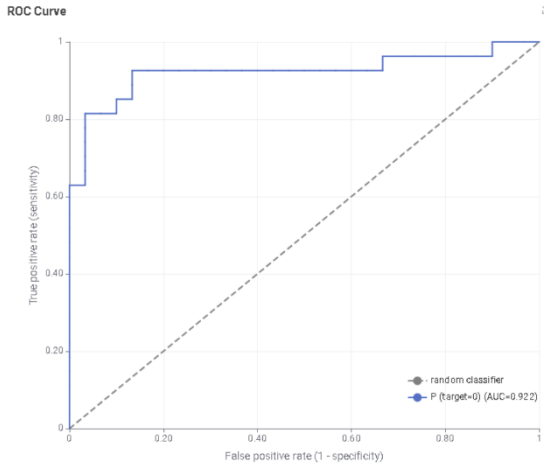

Figure 11. ROC Curve of Tuned *SVM* Model

When random forest model was applied using tuned parameter values, we obtained an accuracy of 87.72 % with recall of 0.89 and precision of 0.86. This demonstrated how this model was already very accurate before.

The AUC value is 0.929, which labels this model as an excellent one and it's even better than the previous model. The ROC curve is shown below for random forest model applied with tuned parameters.

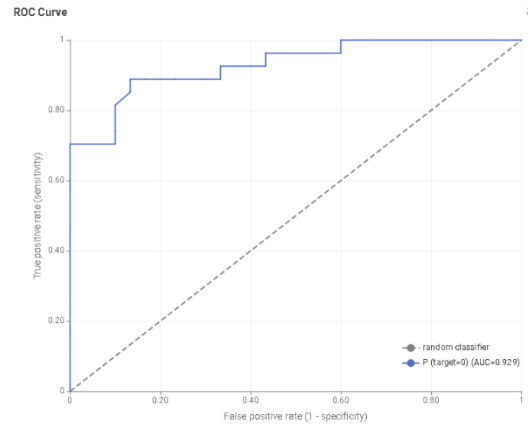

Figure 12. ROC Curve of Tuned *RF* Model

5.3 Cross Validation

When logistic regression model was applied using 10-fold cross-validation split, we obtained an accuracy of 89.65 % with recall of 0.89 and precision of 0.94. These metrics have shown that our model can perform greatly on unseen or new data because the tuned model and cross-validated model have almost similar values of evaluation metrices. There is no chance of overfitting and underfitting which proves that our model is performing at its best.

When decision tree model was applied using 10-fold cross-validation split, we obtained an accuracy of 79.31 % with recall of 0.72 and precision of 0.93. These metrics have shown that our model is not performing well on unseen or new data because the tuned model and cross-validated model have almost very different values of evaluation metrices. The is a high chance of overfitting which proves that our model is not performing as desired.

When support vector machine model was applied using 10-fold cross-validation split, we obtained an accuracy of 89.65 % with recall of 0.89 and precision of 0.94. These metrics have shown that our model can perform greatly on unseen or new data because the tuned model and cross-validated model have almost similar values of evaluation metrices. There is no chance of overfitting and underfitting which proves that our model is performing at its best.

When random forest model was applied using 10-fold cross-validation split, we obtained an accuracy of 89.65 % with recall of 0.83 and precision of 1. These metrics have shown that our model can perform greatly on unseen or new data because the tuned model and cross-validated model have almost similar values of evaluation metrices. The is no chance of overfitting and underfitting which proves that our model is performing at its best.

After evaluating all the four models using respective metrices, we have concluded that the best performing model is the ***Random Forest Model*** as it gave the highest values of accuracy as well as precision and a good value for recall [5]. The AUC value of this model is also the highest among all others. So, for this problem of detecting heart disease, our winning model is the Random Forest Model

The worst performing model is the ***Decision Tree Model*** based on the poor values of evaluation metrices.

Conclusion

As highlighted in the introduction, predicting the presence of heart disease using measurable clinical attributes is a complex challenge, given the interplay of subjective symptoms (e.g., chest pain types) and objective biomarkers (e.g., cholesterol levels) [6]. This complexity is mirrored in the performance of the models we evaluated. While no algorithm achieved flawless accuracy, the results underscore the potential of machine learning to augment clinical judgment.

The **Random Forest model** emerged as the most effective classifier, delivering the highest accuracy (87.72%), precision (86%), recall (89%), and AUC-ROC (0.929). Its ensemble

design allowed it to capture intricate interactions between features—such as how age and ST depression jointly influence risk—outperforming simpler models. Conversely, the **Decision Tree** lagged significantly, with lower metrics (accuracy: 84.21%, AUC: 0.91), likely due to overfitting and its inability to generalize beyond training patterns.

While these results are promising, further improvements could involve integrating advanced techniques like **neural networks** or **gradient-boosted trees** to handle nonlinear relationships [7]. Additionally, refining preprocessing—such as adopting robust imputation for missing values or incorporating domain-specific feature engineering (e.g., combining blood pressure and age into a risk score)—could enhance predictive power [10]. Expanding the dataset with variables like family history or lifestyle factors might also better capture the multifaceted nature of cardiovascular health. Ultimately, this work highlights the viability of machine learning as a supplementary tool for early diagnosis, provided models are transparent and validated against diverse clinical populations.

References

[1] Kaggle (2019), Heart Disease Dataset. Retrieved from: *https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset/data*

[2] Allison P. D. (2024). Missing Data in Clinical Datasets: Challenges and Solutions

[3] Folorunso, S. O., Awotunde, J. B., & Adeniyi, E. A. (2022). Heart Disease Classification Using Machine Learning Models. In Informatics and Intelligent Applications (pp. 63–74). Springer

[4] Malibari, A. A., et al. (2024). Enhancing Heart Disease Classification with M2MASC and CNN-BiLSTM

[5] Khan, M. A., et al. (2024). Audio Signal-Based Heart Disease Detection Using Feature Ensemblers

[6] Ingole, B. S., et al. (2024). Advancements in Heart Disease Prediction: A Machine Learning Approach for Early Detection

[7] Shah, D., Patel, S., & Bharti, S. K. (2020). Heart Disease Prediction Using Machine Learning Techniques. SN Computer Science

[8] IEEE Xplore. (2024). Heart Disease Prediction Using Machine Learning, Conference Publication

[9] IEEE Xplore. (2024). Cardio Prognosis: Machine Learning Approaches to Heart Disease Prediction, Conference Publication

[10] Springer. (2024). A Comprehensive Review of Deep Learning-Based Models for Heart Disease Prediction

[11] IEEE Xplore. (2024). Heart Disease Prediction Using Machine Learning: A Data-Driven Approach, Conference Publication