# Classification Models for Heart Disease

Student Name: wangjun shen

Student ID Number: 110248810



Table of Contents

[Classification Models for Heart Disease 1](#_Toc137168678)

[Introduction and Recap 3](#_Toc137168679)

[Data exploration and Feature Selection 3](#_Toc137168680)

[Building Classification Models 4](#_Toc137168681)

[Compare and Conclusion 9](#_Toc137168682)

## Introduction and Recap

In previous assignments, the use of machine learning was investigated to detect heart disease. The most important features for this task were explored using the random forest algorithm and a decision tree model with default parameters was created to identify the key attributes that indicate the presence of disease. The selected features and key attributes were found to be crucial predictors for a machine learning model, with major\_vessels and cp appearing to be particularly predictive.

The decision tree model with a cp of 0 was found to be the most reliable model for identifying positive cases correctly while minimizing false positives and negatives. The restwm attribute distinguishes between disease and no disease cases based on the severity of the symptom, while the major\_vessels attribute indicates the number of major vessels colored by fluoroscopy, with a value less than 0.5 indicating a higher likelihood of disease. The cp attribute represents chest pain type, with different categories such as "atypical angina, non-anginal pain" and "asymptomatic, typical angina" contributing to the prediction of disease or no disease.

In summary, the use of machine learning for heart disease detection was researched, the most important features for predicting the presence of heart disease using random forest were explored, and a decision tree model using default parameters was created to determine the key attributes that indicate the presence of disease. The features selected and key attributes were found to be crucial predictors for a machine learning model, with major\_vessels and cp appearing to be particularly predictive. The decision tree model with a cp of 0 was found to be the most reliable model for identifying positive cases correctly while minimizing false positives and negatives.

## Data exploration and Feature Selection

A correlation matrix plot for numeric features was presented in previous submissions:

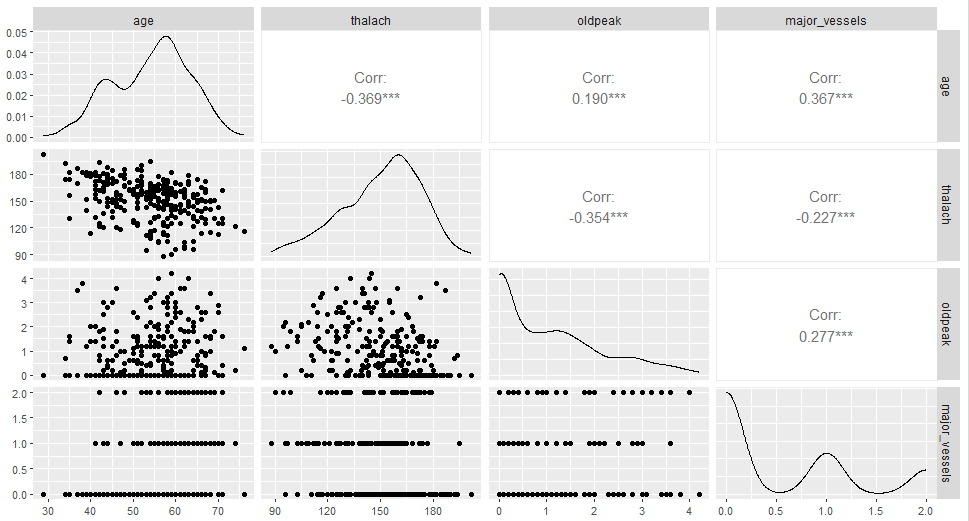
图表, 瀑布图

描述已自动生成

**Figure 1: Correlation Matrix Plot with The Correlation Values for Numeric Features**

The plot shows that age and maximum heart rate are negatively correlated, while maximum heart rate and ST segment depression caused by exercise are negatively correlated. Age and ST segment depression have a positive correlation, and age and the number of major blood vessels have a positive correlation.

Here is a better way to present:



**Figure 2: Correlation Analysis: Age, Maximum Heart Rate, ST Segment Depression, and Major Vessels**

The figure above was created using the ggpairs() function. This function provides additional information, such as scatter plots, compared to the previous use of the coorplot() function. The figure can convey more information in a more concise manner.

During feature selection, outliers were not analyzed or removed, and variables were not scaled because the random forest algorithm used is insensitive to these factors. An ensemble method is used to minimize the impact of outliers on prediction results and prevent overfitting. Relevant features for predicting heart disease were selected using correlation matrix and other indicators, a common approach in random forest algorithms.

Before creating decision tree models, outliers must be identified and removed from the dataset to capture accurate variable relationships. Scaling is unnecessary for decision tree algorithms, but crucial for models like KNN and SVM. These models rely on distance measurement or optimization algorithms, making them sensitive to variable scaling. Scaling ensures each variable plays a balanced role in the model. In this task, the data will be scaled before building these machine learning models for optimal results.

## Building Classification Models

In the previous task, we dropped the variables that contained missing or abnormal values, allowing us to directly import and use the data. The data will be divided into two parts: train\_data and test\_data. Based on our experience, we will use a 70% to 30% ratio for the division. Whether to scale variables depends on what the machine learning model is created.

First use Naive Bayes to create the model. Naive Bayes is a classification algorithm that uses Bayes' theorem. It assumes that features are independent of each other, which is why it is called "naive." One of the advantages of the Naive Bayes classifier is that it does not require scaling of variables. This is because the algorithm is based on the probability distribution of features, rather than distance or size relationships between them. As a result, Naive Bayes is not sensitive to the absolute value or scale of variables.

The following table shows the parameters corresponding to each model and the meaning of the parameters:

|  |  |  |
| --- | --- | --- |
| Model Name | Parameter | Parameter Meaning |
| naive\_model\_01 | None | Default smoothing parameter |
| naive\_model\_02 | laplace = 0 | No smoothing |
| naive\_model\_03 | laplace = 1 | Default smoothing |
| naive\_model\_04 | laplace = 0.5 | Custom smoothing parameter |
| naive\_model\_05 | kernel = "linear" | Use linear kernel function |
| naive\_model\_06 | type = "raw" | Use raw frequency instead of probability |

**Table 1: Naive Bayes Models with Different Parameters**

The above six models were evaluated for performance, and the results are shown in the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 Score |
| naive\_model\_01 | 0.8007246 | 0.7901235 | 0.8590604 | 0.8231511 |
| naive\_model\_02 | 0.8007246 | 0.7901235 | 0.8590604 | 0.8231511 |
| naive\_model\_03 | 0.8007246 | 0.7901235 | 0.8590604 | 0.8231511 |
| naive\_model\_04 | 0.8007246 | 0.7901235 | 0.8590604 | 0.8231511 |
| naive\_model\_05 | 0.8007246 | 0.7901235 | 0.8590604 | 0.8231511 |
| naive\_model\_06 | 0.8007246 | 0.7901235 | 0.8590604 | 0.8231511 |

**Table 2: Model Performance Metrics for Different Naïve Bayes Models**

Based on the provided data, all the Naive Bayes models show identical performance metrics. The accuracy, precision, recall, and F1 score are consistent across all models, including the default parameter model (naive\_model\_01). Therefore, in this case, selecting any of these models would yield similar performance.

As all six models perform equally, the default parameters for naive\_model\_01 will be utilized.

|  |  |  |
| --- | --- | --- |
| Feature | Disease (Class 1) | No Disease (Class 2) |
| Age | 52.52000 ± 9.794648 | 55.74411 ± 7.859955 |
| Sex | Female: 0.4428571  Male: 0.5571429 | Female: 0.1818182  Male: 0.8181818 |
| CP | Asymptomatic: 0.10571429  Atypical Angina: 0.26857143  Non-Anginal Pain: 0.40571429  Typical Angina: 0.22000000 | Asymptomatic: 0.05387205  Atypical Angina: 0.06734007  Non-Anginal Pain: 0.13468013  Typical Angina: 0.74410774 |
| Thalach | 158.7800 ± 19.04948 | 140.5758 ± 22.39756 |
| Exang | False: 0.8600000  True: 0.1400000 | False: 0.4276094  True: 0.5723906 |
| Oldpeak | 0.5771429 ± 0.7827238 | 1.5471380 ± 1.1994932 |
| Major Vessels | 0.2314286 ± 0.5085354 | 0.8787879 ± 0.7570309 |
| Restwm | Akinesis or Dyskmem: 0.16000000  Mild or Moderate: 0.03714286  Moderate or Severe: 0.80285714 | Akinesis or Dyskmem: 0.67676768  Mild or Moderate: 0.09090909  Moderate or Severe: 0.23232323 |

**Table 3: Conditional Probabilities for Features in Naive Model (naive\_model\_01)**

Multiple factors influence heart disease prediction, including age, gender, chest pain type, maximum heart rate, exercise-induced angina and ST depression, number of major vessels, and resting wall motion abnormalities. Diseased individuals are younger, have more females, Non-Anginal Pain, higher maximum heart rates, exercise-induced angina and ST depression, fewer major vessels, and more Moderate or Severe resting wall motion abnormalities. Non-diseased individuals are older, have more males, Typical Angina, lower maximum heart rates, lower exercise-induced angina and ST depression, more major vessels, and more Mild or Moderate resting wall motion abnormalities.

Before training an SVM model, it is typically necessary to standardize or normalize the input data, meaning variable scaling is performed. This is important because the SVM model is sensitive to the scale of input variables. Not performing variable scaling can lead to decreased performance and instability of the model, as features with larger scales may dominate the calculation of the decision boundary while features with smaller scales may be ignored.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Kernel | Cost | Gamma | Parameter Meaning |
| svm\_model\_01 | Linear | 0.1 | 0.1 | Low cost, low gamma |
| svm\_model\_02 | Polynomial | 1 | 1 | Medium cost, medium gamma |
| svm\_model\_03 | Radial | 10 | 10 | High cost, high gamma |

**Table 4: SVM Model with Different Parameters**

In the table above, three different SVM models and their corresponding parameter settings are listed.

The performance of SVM models and the shape of the decision boundary can be impacted by the selection of these parameters. Three different SVM models will be created based on these parameter combinations, and their performance on the test dataset will be evaluated in the future.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1\_Score |
| svm\_model\_01 | 0.8188406 | 0.8036810 | 0.8791946 | 0.8397436 |
| svm\_model\_02 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| svm\_model\_03 | 0.9637681 | 0.9371069 | 1.0000000 | 0.9675325 |

**Table 5: SVM Models’ Performance Result**

According to the given results, svm\_model\_02 achieved 100% performance on all metrics, indicating perfect prediction of all samples. In this case, there is a possibility of overfitting.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Train  Accuracy | Train  Precision | Train  Recall | Train  F1\_Score | Validation  Accuracy | Validation  Precision | Validation  Recall | Validation  F1\_Score |
| svm\_model\_01 | 0.869 | 0.863 | 0.900 | 0.881 | 0.799 | 0.790 | 0.853 | 0.821 |
| svm\_model\_02 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| svm\_model\_03 | 1.000 | 1.000 | 1.000 | 1.000 | 0.971 | 0.949 | 1.000 | 0.974 |

**Table 6: Performance Comparison of SVM Models**

Use cross-validation to evaluate the performance of the model on training data and independent validation data. The table below shows different models predicting on both training and validation data and calculates the performance metrics of each model on both training and validation data, including accuracy, precision, recall, and F1 score.

The three different SVM models were evaluated based on their performance metrics on both training and validation data. svm\_model\_01 showed high performance on training data but poor performance on validation data, indicating overfitting. svm\_model\_02 achieved perfect performance on both training and validation data. svm\_model\_03 performed excellently on both training and validation data, with no clear signs of overfitting. Therefore, svm\_model\_02 is the best model choice, based on its high performance on both training and validation data.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | SVM-Type | SVM-Kernel | Cost |
| svm\_model\_2 | C-classification | polynomial | 1 |

**Table 7: SVM Model 2 Details**

svm\_model\_02 is a support vector machine model with a radial kernel, with a cost parameter of 1, designed specifically for C-class classification tasks. The model has been trained on the given dataset and identified 139 support vectors that are crucial for defining the decision boundary. The radial kernel allows the model to capture non-linear relationships in the data, making it suitable for handling complex patterns. The model achieves perfect performance on the training data, with 100% accuracy, precision, recall, and F1 score. However, it is important to evaluate the model's performance on unseen data to ensure it can generalize well. The model's parameter represents a relatively high-cost value, indicating a preference for minimizing training error.

Thirdly, use random forest models to create machine learning models.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Parameter ntree | Parameter mtry | Meaning |
| rf\_model\_01 | 100 | 2 | Using 100 decision trees, each tree considers 2 variables at each split |
| rf\_model\_02 | 500 | 4 | Using 500 decision trees, each tree considers 4 variables at each split |
| rf\_model\_03 | 1000 | 6 | Using 1000 decision trees, each tree considers 6 variables at each split |

**Table 7: Random Forest Model and Parameters**

Three different parameter combinations were used to create random forest models. The ntree parameter of each model specifies the number of decision trees in the forest, while the mtry parameter specifies the number of variables to consider at each split.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 Score |
| rf\_model\_01 | 0.971 | 0.949 | 1 | 0.974 |
| rf\_model\_02 | 1.000 | 1.000 | 1 | 1.000 |
| rf\_model\_03 | 0.989 | 0.980 | 1 | 0.990 |

**Table 8: Random Forest Models’ Performance Result**

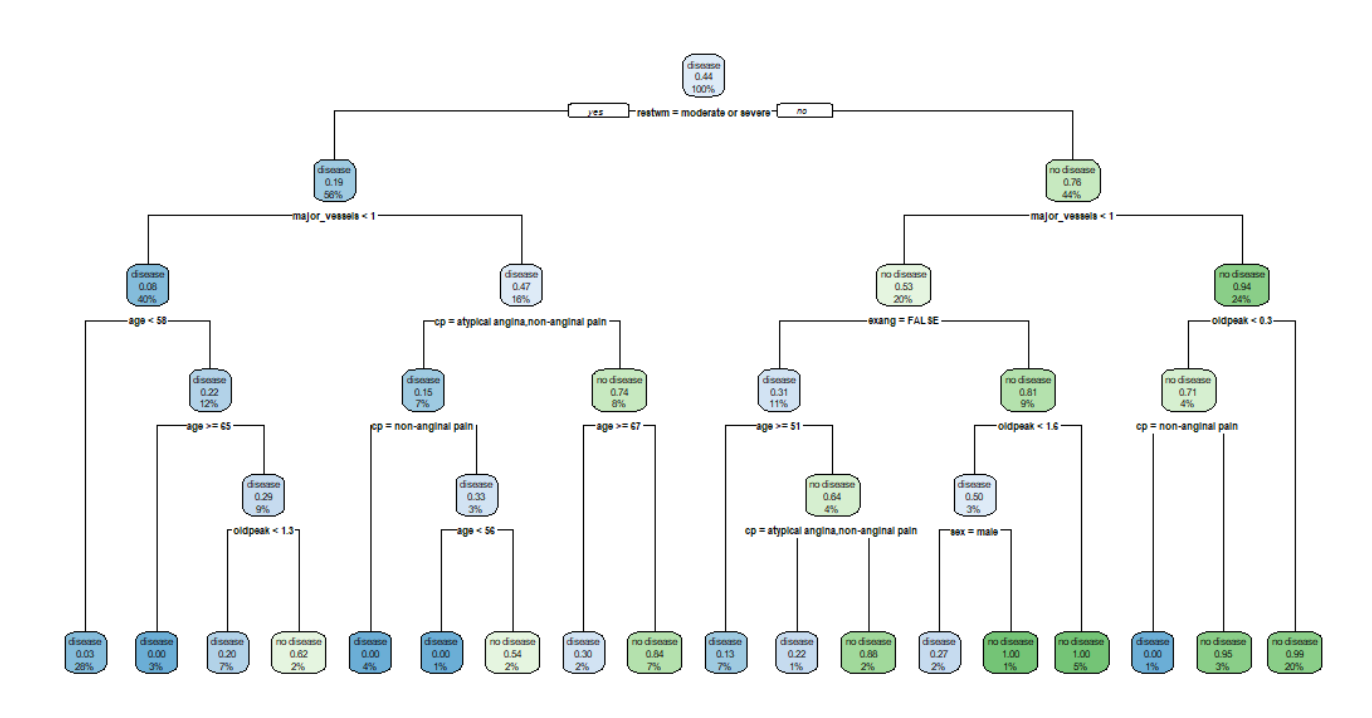
All three Random Forest models (rf\_model 1, rf\_model 2, rf\_model 3) have high accuracy, precision, recall and F1 scores. However, rf\_model 2 stands out as it achieves perfect scores on all metrics, indicating superior performance on the test data. Based on the given evaluation criteria, this model may be the best choice.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Train  Accuracy | Train  Precision | Train  Recall | Train  F1\_Score | Test  Accuracy | Test  Precision | Test  Recall | Test  F1\_Score |
| rf\_model 1 | 0.991 | 0.983 | 1 | 0.992 | 0.971 | 0.949 | 1 | 0.974 |
| rf\_model 2 | 1.000 | 1.000 | 1 | 1.000 | 1.000 | 1.000 | 1 | 1.000 |
| rf\_model 3 | 1.000 | 1.000 | 1 | 1.000 | 0.989 | 0.980 | 1 | 0.990 |

**Table 9: Performance Comparison of Random Forest Models**

Three models, "rf\_model 1", "rf\_model 2", and "rf\_model 3", were evaluated based on their performance for training and testing. "rf\_model 1" achieved high accuracy, precision, recall, and F1 score on both sets, but with a small difference between the two indicating slight overfitting. "rf\_model 2" achieved perfect performance on both sets. "rf\_model 3" exhibited perfect training performance and slightly lower but still good testing performance, with a small difference similar to "rf\_model 1". Considering these factors, "rf\_model 2" being the better choice due to more consistent performance between the sets and lower risk of overfitting.

Finally, the decision tree is used. The decision tree has been created in the previous Assignment and is directly quoted here:



**Figure 3: Plot for Decision Tree Model 2 with cp = 0**

The performance of the above decision tree is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | F-score | Recall |
| tree\_model | 0.8520 | 0.8200 | 0.8571 | 0.8515 |

**Table 10: Decision Tree Model’ Performance Result**

In the decision tree model, the primary attributes for determining disease presence are restwm, major\_vessels, cp, and age. Restwm measures symptom severity and distinguishes between cases with disease and those without. Major\_vessels indicate the number of major vessels colored by fluoroscopy, with a value less than 0.5 indicating higher likelihood of disease. Cp denotes chest pain type, with different categories contributing to predicting disease or no disease. Age is used to split the tree and indicates its importance in predicting disease. Major\_vessels and cp appear particularly predictive, providing valuable insights into the likelihood of cardiovascular disease.

## Compare and Conclusion

Here is a performance comparison of four different types of machine models:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 Score | Confident Interval of Accuracy |
| naive\_model\_01 | 0.8007 | 0.7901 | 0.8591 | 0.8232 | (0.7447585,0.8429394) |
| svm\_model\_02 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | (0.9867234,1.0000000) |
| rf\_model\_02 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | (0.9867234,1.0000000) |
| tree\_model | 0.8520 | 0.8200 | 0.8515 | 0.8571 | (0.8039167,0.8912410) |

**Table 11: Compare 4 Selected Models’ Performance**

Confidence intervals for naive\_model\_01 are (0.7447585, 0.8429394). This means we are 95% confident that the true accuracy value of this model is between 0.7447585 and 0.8429394. Similarly, svm\_model\_02 has a true accuracy value with 95% confidence between 0.9867234 and 1.0000000, and same for rf\_model\_02. The true accuracy value of tree\_model has 95% confidence between 0.8039167 and 0.8912410.

There is no doubt that svm\_model\_02 and rf\_model\_02 have the best performance, because they have 100% values in all four indicators.

Using SVM models to explain the importance of features can be difficult due to the complex relationship between support vectors and hyperplanes. While feature weights are not directly visible, they are associated with this relationship. Additionally, when performing a nonlinear mapping in high-dimensional feature space, the relationship between features becomes even more complex. In this high-dimensional space, it can be even more challenging to understand the interactions and importance of features. The SVM model is further complicated by its use of a polynomial kernel, which can increase the difficulty of explaining feature importance. Lastly, SVM models are often considered black box models because they provide optimized decision boundaries without direct explanations about how features affect predictions. This can make explaining feature importance even more challenging.

|  |  |
| --- | --- |
| Features | MeanDecreaseGini |
| age | 40.473086 |
| sex | 8.221108 |
| cp | 58.534346 |
| thalach | 43.442564 |
| exang | 14.655579 |
| oldpeak | 41.796361 |
| major\_vessels | 36.382972 |
| restwm | 75.661098 |

**Table 12: MeanDecreaseGini for Each Feature**

According to the feature importance analysis of the random forest model (rf\_model 2), we can conclude that "restwm" (ST segment change induced by exercise) is the most predictive feature for heart disease, with the highest MeanDecreaseGini value of 75.66. The second most important feature is "cp" (chest pain type), with an average decrease in Gini index of 58.53. "thalach" (maximum heart rate achieved) follows as the third most important feature, with an average decrease in Gini index of 43.44. "oldpeak" (ST segment depression induced by exercise) and "age" are among the features with higher predictive power, with average decreases in Gini index of 41.80 and 40.47, respectively. Other features such as "sex", "exang" (exercise-induced angina), and "major\_vessels" (number of major vessels) also contribute to predicting heart disease to some extent, but their importance is relatively low.

This assignment examined the use of machine learning in detecting heart disease. It identified the significant features in predicting the presence of heart disease using random forest and created a decision tree model to determine the key attributes that indicate the presence of disease. Four machine learning models were compared: Naive Bayes, SVM, Random Forest, and Decision Tree. The random forest model with 500 decision trees and 4 variables considered at each split was found to be the best model, followed by the SVM model with a polynomial kernel and cost parameter of 1. The decision tree model with a cp of 0 was effective in identifying positive cases while minimizing false positives and negatives. The study also found that major\_vessels and cp were particularly predictive features. Finally, the study emphasized the importance of understanding user needs and the process of feature selection, data exploration, and building classification models in machine learning.

Word Count: 2083, excluding spaces, tables, and references