**DSA1101 Individual Assignment 2**

**Introduction**

The task given is to explore the dataset on the topic of heart disease. The dataset given consists of 300 patients with various patient attributes. The purpose of the task is to determine a classifier best suited to predict the heart disease status of an individual, given the types of classifiers we have learned thus far.

**Exploring the data**

The dimensions of the given data are 300 rows and 13 columns. The 13 columns include a response variable: *<disease>* as well as 12 input variables: *<age, sex, chest.pain, bp, chol, fbs, resting.ecg, heart.rate, angina, st.depression, vessels, blood.disorder>*

*<blood.disorder>* had 2 missing values, which was replaced with its modal category to maintain data integrity. A summary of the variables was recorded in *figure 1* below, which includes: Quantitative variable’s *range, mean, median and Interquartile range (IQR)* as well as Categorical variable’s *frequency table*.

Among these 13 variables, the following variables were treated as **Categorical**: *<disease, sex, chest.pain, fbs, resting.ecg, angina, blood.disorder>* and the rest was treated as **Quantitative**: *<age, bp, chol, heart.rate, st.depression, vessels>*.

|  |
| --- |
| *Figure 1: Summary of variables* |

**Fitting appropriate variables into classifiers**

|  |  |
| --- | --- |
| For the **quantitative** variables, the association with *<disease>* was observed visually through side-by-side boxplot comparisons, while the strength and direction of the relationship were quantified using the point-biserial correlation coefficient. From *figure 3*, only *<chol>* was statistically insignificant (p\_value>0.05). *Plot C* in *figure 2* also shows a significant overlap in the boxplots, further indicating a weak association. Therefore, the quantitative variable *<chol>* was removed, while the rest was kept.  *Figure 3* also shows that *<heart.rate>* has a negative coefficient (-0.418), which indicates a negative association between higher heart rate and the likelihood of having the disease.  According to *figure 2 and 3*, *<heart.rate, st.depression, vessels>* has a moderate association with *<disease>* while *<age, bp>* has a weaker association in terms of strength. | **A group of boxes with numbers and letters  AI-generated content may be incorrect.**  Figure 2: Boxplots of Quantitative variable against disease  A screenshot of a computer code  AI-generated content may be incorrect.  Figure 3: Point-biserial correlation of Quantitative variables |

|  |  |
| --- | --- |
| For the **categorical** variables, the significance of their association with the response variable was assessed using a proportion table, and the odds of success was calculated and illustrated in *figure 4*. For example, the odds of a male having the disease was 0.343, while the odds of a female having the disease was 1.24. From *figure 4*, the difference in odds in *plot C* was insignificant. On further inspection, the odds values were 0.835 and 0.957 respectively. Given that both odds values were also close to 1, there is no significant trend, therefore the input does not have a strong association with *<disease>* and was dropped. | **A group of gray bars with black text  AI-generated content may be incorrect.**  Figure 4: Barplots of Categorical variable with success odds |

**Proposing the classifiers**

To find the most suitable classifier, each model was trained and tested on the dataset, leveraging all available data to evaluate each model’s performance.

I trained and tested on the entire dataset, leveraging all available data to evaluate model performance. Each classifier underwent 5-fold cross-validation to ensure robust and reliable evaluation, reducing the risk of overfitting. Quantitative predictors were standardized using the dataset’s mean and standard deviation, ensuring algorithms like K-Nearest Neighbors (KNN) weren’t skewed by differing feature scales.

Linear regression was unsuitable due to the binary, categorical nature of the response variable, which violates the normality assumption required for linear regression. Instead, supervised classifiers—Decision Tree (DT), Logistic Regression (LR), K-Nearest Neighbors (KNN), and Naive Bayes (NB)—were employed. Unsupervised methods like K-means and association rules were excluded, as the dataset includes a labeled response variable.

Performance was assessed using True Positive Rate (TPR), Precision, Receiver Operating Characteristic (ROC) curves, and Area Under the Curve (AUC) for each classifier. TPR was prioritized because missing a heart disease case (false negative) has severe consequences, potentially delaying critical treatment. An optimal classifier maximizes TPR (close to 1) and achieves high overall accuracy.

**Decision Tree (DT)**

|  |  |
| --- | --- |
| The DT classifier used the "class" method to produce categorical outputs. For each of the 5 folds, a confusion matrix was generated to compute the TPR which was then averaged across folds. Hyperparameters (maxdepth and minsplit) were tuned by testing various combinations, evaluated via TPR, and visualized in a heatmap, illustrated in *figure 5*, where lighter shades indicate higher TPR. | Figure 5: Heatmap of TPR with regards to minsplit and maxdepth |
| The optimal parameters, maxdepth = 4 and minsplit = 8, yielded the highest average TPR (0.739) during cross-validation. Applying this model to the dataset produced a TPR of 0.797, Precision of 0.859, and AUC of 0.883. | A diagram of a number of blood disorder  AI-generated content may be incorrect.  Figure 6: Decision Tree Plot |

**Naïve Bayes Model (NB)**

The NB classifier employed the naiveBayes function to predict categorical outcomes ("yes" or "no" for heart disease) based on conditional probabilities. The model was trained and tested on the whole dataset. It achieved a TPR of 0.804, precision of 0.841, and AUC of 0.912.

**Logistic Regression (LR)**

|  |  |
| --- | --- |
| The LR classifier used the "response" type to generate probabilities of heart disease (where 1 indicates presence). An initial model with all predictors was fitted to assess variable importance. Significance of the variables was the key criteria (p-value >= 0.05). Among the statically significant variables in *figure 7*, *<Sex, heart.rate, angina>* were dropped as their values were close to 1 (0.80-1.20 range). | A screenshot of a computer  AI-generated content may be incorrect.  Figure 7: Variable Coefficients of initial LR model |
| A graph of True Positive Rate (TPR) and False Positive Rate (FPR) against threshold was then plotted to identify the optimal threshold. We prioritize TPR (>=0.90) when we choose the ideal threshold. *Figure 8* shows that the threshold of 0.1806 maximizes TPR while minimizing FPR, as such we chose the threshold value of 0.1806. Probabilities were then converted to binary outcomes. The model achieved a TPR of 0.978, precision of 0.621, and AUC of 0.872. | Figure 8: Plot of TPR & FPR against threshold |

**KNN Regression (KNN)**

|  |  |
| --- | --- |
| The KNN classifier predicts class labels based on the majority vote of the k nearest neighbors, using Euclidean distance. To ensure fair distance calculations, training predictors were scaled using the training set’s mean and standard deviation, with test data scaled accordingly. Odd k values (3 to 21) were tested across 5-fold cross-validation, with mean TPR as the key metric. From *figure* 9, K = 3 was selected for its best TPR, reflecting the priority of detecting heart disease cases. This model yielded a TPR of 0.826, precision of 0.905, and AUC of 0.953. | Figure 9: Plot of TPR against K values. |

**ROC AND AUC**

|  |  |
| --- | --- |
| A Receiver Operating Characteristic (ROC) curve was employed as a diagnostic tool to evaluate classifier performance, given the binary nature of the response variable (heart disease: "yes" or "no"). The Area Under the ROC Curve (AUC) for Naive Bayes (NB), Logistic Regression (LR), Decision Tree (DT), and K-Nearest Neighbors (KNN) was 0.912, 0.872, 0.893, and 0.953. A higher AUC indicates superior ability to discriminate between positive (disease) and negative (no disease) classes, with KNN and NB outperforming the others, followed by DT and LR. | Figure 10: AUC plots for all models |

**Comparing between classifiers**

|  |  |  |  |
| --- | --- | --- | --- |
| Decision Tree  TPR: 0.797  Precision: 0.859  AUC: 0.893 | Naïve Bayes  TPR: 0.804  Precision: 0.841  AUC: 0.9122 | Logistic Regression  TPR: 0.928  Precision: 0.621  AUC: 0.872 | K-Nearest Neighbors  TPR: 0.826  Precision: 0.905  AUC: 0.953 |

Figure 11: Table of Goodness of Fit between different classifiers

Overall, the goodness of fit for all four classifiers is well-suited for predicting heart disease. From *figure 11*, We can see that LR has the highest TPR, followed by KNN, NB and DT. KNN has the highest Precision, followed by DT, NB and LR. Finally, KNN has the highest AUC value, followed by NB, LR and DT. The criterion for choosing the best classifier is maximizing TPR while effectively balancing sensitivity and precision. Although LR has the highest TPR, it has a relatively low precision. Hence, KNN’s high TPR, precision and AUC makes it the preferred classifier for this dataset.

**Pros and Cons of the proposed classifier**

K-Nearest Neighbors (KNN) offers a straightforward way to predict heart disease by classifying patients based on the majority vote of their three nearest neighbors in a scaled feature space, using predictors like *<chest.pain, vessels>*. For instance, a patient with high *<chest.pain>* is often near others with heart disease, making risk easy to understand through closeness. This simplicity lets KNN adapt to data patterns without needing complex rules or assumptions about how predictors like *<st.depression>* behave, which is great for messy medical data. It achieved a test AUC of 0.953 with k = 3, showing strong performance.

However, KNN depends heavily on distance, so noisy or unimportant features like *<age, rest.ecg>* can confuse it if not carefully scaled. It also struggles with big datasets, as it calculates distances to all 300 training points for every prediction, slowing things down if the dataset grows. Unlike Logistic Regression, KNN doesn’t naturally highlight key predictors, so all features count equally, which might weaken the focus on critical ones like *<vessels>*.

**--- End ---**