Predicting heart disease using machine learning models

Student’s Name

Institutional affiliation

**Introduction**

According to statistics, it is approximated that about 26 million people have become victims of heart diseases. It is reported that acute heart failure is the most significant cause of adult hospitalization in the United States and Europe, where 1% - 2% of the total hospital admission in these regions is associated with heart disease. Despite the improvement in mortality rate associated with heart failures, researches have reported that heart failure is still one of the leading deaths mostly in adult people worldwide. The advancement in technology and artificial intelligence has contributed significantly to the effort applied in the field of medicine in detecting and treatment of heart diseases. Various machine learning algorithms have demonstrated high performance of clinical outcomes in predicting the heart diseases that have helped in selecting therapies for diagnosed patients. In this study, we utilized the heart disease dataset from the Kaggle repository to build a prediction model for the disease using four classification machine learning algorithms; logistic regression, KNearestNeighbor; Decision Tree, and the Random Forest classifier.

**The dataset**

As pointed out, the dataset for this project was obtained from the Kaggle machine learning repository. The data contains 303 cases (rows) and 14 features (columns). The 14th feature is the target variable, which presents a correct diagnose of the disease in binary (1= the person has heart disease, 0 = the person has no heart disease). The table below shows the features and their description.

|  |  |  |
| --- | --- | --- |
| **#** | **Attribute** | **Description** |
| **1** | Age | Age |
| **2** | Sex | Sex |
| **3** | Cp | Chest Pain Type (4 values) |
| **4** | Trestbps | Rest Blood Pressure in mm Hg at admission to the hospital |
| **5** | Chol | Serum Cholestoral in mg/dl |
| **6** | FBS | Fasting Blood Sugar if > 120 mg/dl (1 = true; 0 = false) |
| **7** | Restecg | Resting Electrocardiographic Result (values 0,1,2) |
| **8** | Thalach | Max. Heart Rate |
| **9** | Exang | Exercise Induced Angina (1 = yes; 0 = no) |
| **10** | Oldpeak | St Depression Induced by Exercise Relative to Rest |
| **11** | Slope | The Slope of The Peak Exercise St Segment |
| **12** | Ca | Number of major vessels (0-3) colored by flourosopy |
| **13** | Thal | 3 = Normal; 6 = Fixed defect; 7 = Reversable defect |
| **14** | Num(Target) | Diagnosis of Heart Disease (angiographic disease status) |

**Previous literature**

A significant amount of studies has been carried out to determine whether heart diseases can be predicted using various machine learning and deep learning models with higher precisions that can be used in hospital settings. Greater accuracies give confidence to clinicians when diagnosing and administering medical prescriptions to patients with heart infections. Kwon et al. (2019) conducted prediction models for predicting acute heart failure and in-hospital mortality using deep learning models and machine learning models. Random forest and logistic regression models are some machine learning algorithms they used that we also employed in detecting the presence or absence of the disease. In their study, they used a dataset with 12654 patients as the training set for the models and another dataset with 4759 patients as the testing test for the model performance. Using a receiver operating characteristic curve (ROC curve), the deep learning model performed better than the machine learning models. However, the random forest outperformed logistic regression by scoring AUC=0.756 against logistic regression AUC = 0.720.

Ali et al. (2019) also employed machine learning in the detection of heart disease. The authors applied four machine learning models; decision tree, naïve Bayes, support vector machine, and random forest for classification of patients. After the training and testing of the models, the Gaussian naïve Bayes demonstrated the highest accuracy (91.21%), random forest (89.01%), support vector machine (86.81%), and lowest decision tree (70.33%). According to the authors, feature engineering and preprocessing are essentials for improving model performance.

**Data acquisition.**

After downloading the dataset from Kaggle, it was loaded into the Jupyter Notebook for analysis via the pandas library in python. The library is essential for loading and performing critical manipulation on the data-frame such as sub-setting and slicing.

**Data exploration**

Data exploration is critical in machine learning, as it gives a quick understanding of the properties and pattern of the data set. Data exploration is also essential in understanding the dimension of the data. The descriptive statistics were produced to understand the distribution of continuous variables. The table below shows the mean and standard deviation for the variables. The mean is the average value from the 303 cases of the dataset. The standard deviations are more significant indicating greater variations of data points in the variables.

Table 2: summary statistics of continuous variable distribution

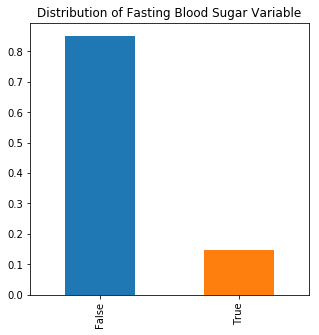
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **age** | **trestbps** | **Chol** | **thalach** | **Oldpeak** |
| **Mean** | 54.37 | 131.62 | 246.26 | 149.65 | 1.04 |
| **Standard deviation** | 9.08 | 17.54 | 51.83 | 22.91 | 1.16 |

**Distribution of categorical variables**

Upon exploring the target feature, 54% of sampled cases were had heart disease, while the remaining 46% had no heart disease. Based on gender, 68% of the sample size were male, while the remaining 32% of the cases were female. Based on the fasting blood sugar (FBS) variable, 85% of the cases had an FBS level of less than 120, while 15% had FBS higher level than 120. Also, 67% of the cases recorded no angina induced by exercise, while 33% of the cases recorded angina caused by exercise. The bar charts below show the distributions.



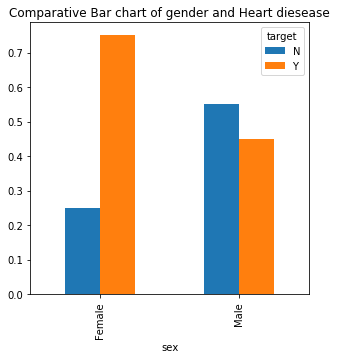




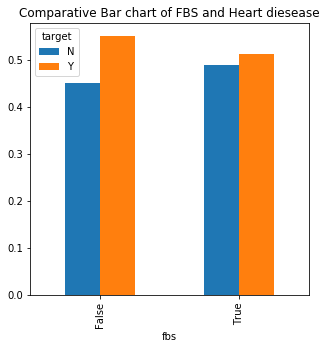


**Data visualization and insight generation**

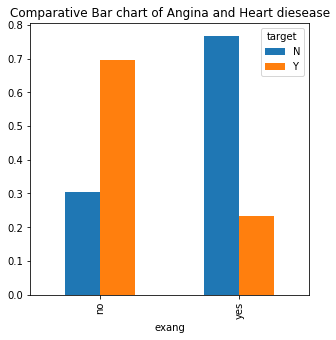
We developed crosstabulations of other categorical features with the target variable to get some insight into the characteristics of people who are affected by the disease. The comparative bar chart of sex and target shows that more females than males have been diagnosed with heart disease. This suggests that females may be practicing a lifestyle that could increase their vulnerability to contracting heart infection.



The comparative bar graph below indicates that cases that had an FBS level of less than 120 were diagnosed with heart disease compared to cases that had an FBS level higher than 120.

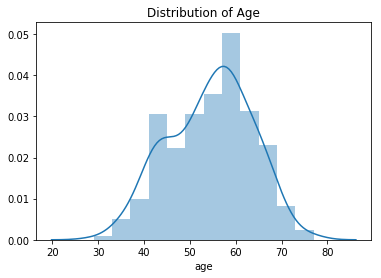


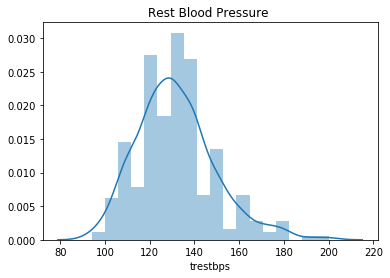
The comparative bar graph below shows examples that recorded no angina were diagnosed with heart disease compared to those cases that recorded angina induced by exercise. This may suggest that people who carry out physical exercise are less vulnerable to heart diseases compared to their counterparts.

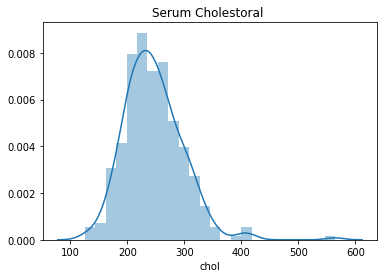


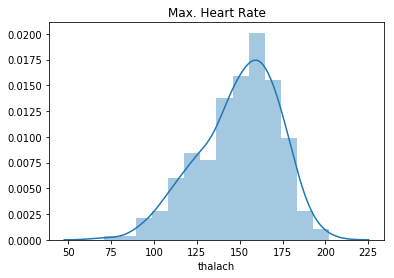
**Univariate exploration of continuous variables**

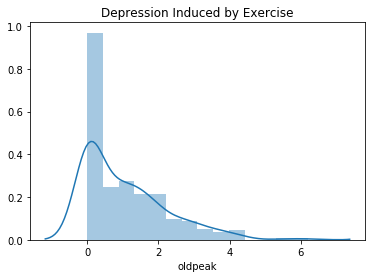
Most Machine learning algorithms produce high performance when the distribution of numerical features follows a normal distribution. Further, most algorithms are affected by the presence of a high amount of outliers in the dataset. Therefore, visualizing the distribution of the variables helps in detecting the presence of outliers that may affect performance. Using the seaborn library in python, we visualized the distribution of Age, trestbps, chol, thalach, and oldpeak. From the distributions plots of the respective variables, the data points followed a normal distribution with minimal outliers which may not adversely affect the performance of the models.







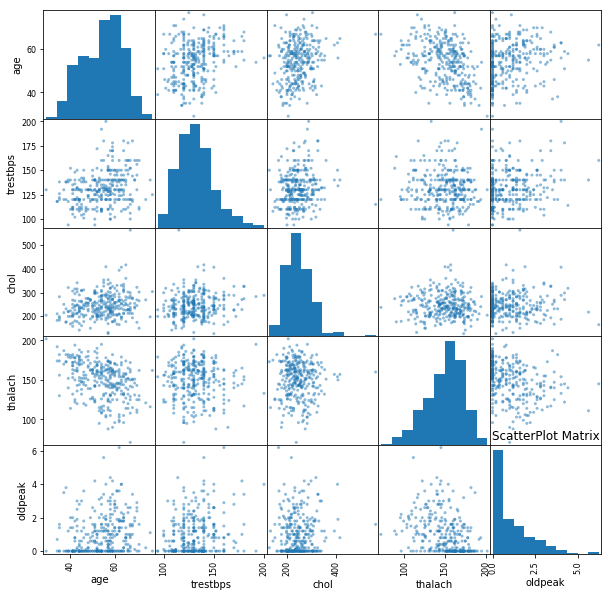




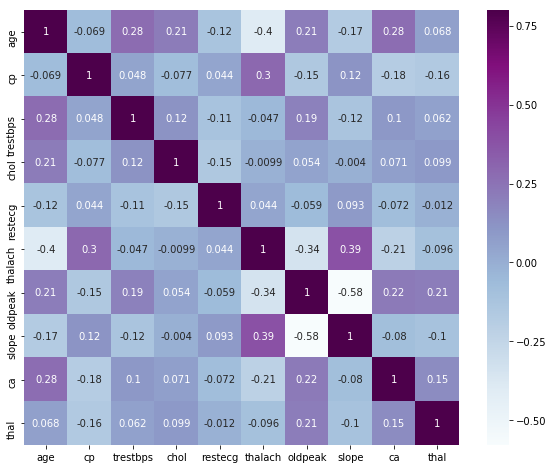
**Bivariate visualization of continuous data**

Multicollinearity which refers to the high correlation between the predictors is a critical factor that usually adversely affects the performance of many machine learning algorithms in both regression and classification models. Bivariate visualization is thus is essential to detect highly correlated features and consequently apply the appropriate feature selection method to eliminate features with multicollinearity.

Using the panda’s library, we constructed the scatterplot matrix shown in the figure below. The leading diagonal shows a histogram of the features, which indicates normally distributed data points. The scatterplot of the variables with each other shows the features are less correlated with each other, and they may not introduce multicollinearity in the machine learning models.



Using the seaborn library in python, we produced a heatmap to show the strength and to visualize the correlation coefficient of each feature with other features. As evident in the heatmap below, the features are less correlated, and hence they are suitable for predicting heart disease with high-performance accuracy.



**Machine learning models**

**Logistic Regression**

Logistic regression is a supervised machine learning model that takes a binary variable as a dependent variable and try to predict its outcome using categorical and numerical predictors. This algorithm predicts the binary outcome by calculating probabilities from the predictors and assigns a case a positive class (binary =1) or negative class (binary = 0) depending on the specified threshold, default is 0.5 (Uddin et al. 2019). A case whose probability is below the specified or the default threshold is classified as a negative class, while the variables whose probabilities are equal to or greater than the threshold are classified in the negative class. Regarding our research cases with probabilities, less than 0.5 are classified as not having heart disease, while cases with probabilities greater than or equal to 0.5 are classified to have heart disease.

**Decision Tree (DT)**

DT is also a supervised machine learning algorithm for both regression and regression found in the sci-kit learn library in python. This model solves classification problems through tree representation. As Uddin et al. (2019) report, each node of the tree represents a specific feature, while each leaf represents a class (target). The cases are assigned to positive or negative class depending on the conditions satisfied which are generated automatically by the algorithm

**KNearestNeighbor (KNN)**

KNN is another supervised machine learning algorithm for both classification and regression models found in python scikit learn library. This model predicts a class of an instance using a known nearest neighbor. The K in the algorithm specifies the number of neighbors the model should use. Instances closer to a negative class neighbor are classified as negative, while those close to a positive class neighbor are classified as positive (Uddin et al. 2019).

**Random Forest (RF)**

Like the algorithms above, RF is also a supervised algorithm in sci-kit learn library for both classification and regression problems. RF is often classified as an ensemble algorithm since it trains many DT algorithms and then selects the decision tree with the highest performance from the decision tree forest (Uddin et al. 2019). Therefore, the classification approach is that of a decision tree but performed on many trees, to find an excellent performing DT, hence the name random forest.

**Data slicing**

The new data frame (X) was created containing the first 13 columns as features and another data frame (y) constating of the binary target only. This was achieved using the pandas library in python.

**Training and testing datasets**

The training dataset is the proportion of the dataset with labels that the algorithms are trained to be able to master the patterns with the data that resulted in the class labels. On the other hand, the testing set is the proportion that is previously unseen by the models, which is used to measure the generalization performance of the models. The newly created X and y data frames were split randomly into 75% (227 instances) training and 25% (76 instances) testing sets. This was achieved using train\_test\_split function in scikit learn.

**Machine learning pipelines for preprocessing and model fitting**

Since there were some outliers in the data, we had to conduct data preprocessing to reduce their effect. The training and testing datasets were normalized using the MinMax preprocessor function found in scikit learn. The MinMax function transforms the data points such that they fall in the range of 0-1. We utilized the pipeline function also found in the scikit learn module wrap each model and preprocessing function and then fitted the models to the training set. The fitted model was then used to predict the test labels from testing features.

**Overfitting and underfitting**

Overfitting refers to when the model demonstrates an excellent generalization on the training set by scoring higher accuracy but generalizes poorly on the testing set (Vabalas et al. 2019). Underfitting, on the other hand, is when the model performs poorly on the training set but generalizes well on the testing set (set (Vabalas et al. 2019)). Using a larger dataset and hyperparameter tuning are considered a solution for the problems. We computed the accuracies of both training and testing sets for the four models to determine which of them generalizes best.

**Analysis results of the models**

After fitting the four pipelines to the training set, we introduced them to the testing set to determine how well they could generalize on new examples. Using a score metric in sci-kit learn, we produced both the training and testing accuracy to identify the model with great generalization capability. As evident in the table, the LR scored 86.34% and 76.32% on training and testing set respectively, KNN scored 85.90% and 76.32% on training and testing set respectively, DT scored 100% and 78.95% on training and testing set respectively. RF scored 99.56% and 75.00% on training and testing sets, respectively. From the results, the DT and RF are overfitting highly. The LR and KNN are also overfitting but can be preferred over DT and RF.

Table 3: Training and testing sets accuracies

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **LR accuracy (%)** | **KNN accuracy (%)** | **DT accuracy (%)** | **RF accuracy (%)** |
| **Training set** | 86.34 | 85.90 | 100.00 | 99.56 |
| **Testing set** | 76.32 | 76.32 | 78.95 | 75.00 |

**Other performance measures**

Apart from the accuracy metric, we evaluated the performance of the models using the following parameters. First is the Precision score. According to Cokelaer et al. (2015), this metric measure identifies only individuals predicted to have heart disease, dividing true positive (have heart disease) by the sum of true positive and false positive (predicted to have but they do not have heart disease). Second is the recall score. This performance metric measures the models’ capability to identify people with heart disease, by dividing the true positive by sum of true positive and false negative (predicted not having but truly they have heart diseases) (Cokelaer et al. 2015). The last one is the Receiver Operating Characteristic Area Under Curve (ROC\_AUC) score. Ideally, models with ROC\_AUC scores of 0.7 and above are accepted to have good prediction capabilities and can be adopted. In medical settings, prediction with high recalls is preferred to those with high precisions. As evident in the table below, the models have higher precisions than recalls. The scores of ROC\_AUC are more significant than 0.7, which means the models have capabilities to predict heart disease.

Table 4: Precision, recall, and ROC\_AUC scores

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Precision score**  **(%)** | **Recall score**  **(%)** | **ROC\_AUC score (%)** |
| **LR** | 81.08 | 73.17 | 76.59 |
| **KNN** | 81.08 | 73.17 | 76.59 |
| **DT** | 83.78 | 75.61 | 79.23 |
| **RF** | 75.68 | 73.68 | 75.00 |

**Results of hyperparameter tuned Logistic and Random Forests**

The Logistic regression and random forest models were tuned using Gridsearch and k=5 cross-validation folds to see if the performance can be improved. The mean accuracy and standard deviation are shown in the table below, where the performances have improved through hyperparameter tuning.

Table 5: Accuracy of 5-folds tuned Models

|  |  |  |
| --- | --- | --- |
| Model | Mean accuracy (%) | Standard deviation |
| Logistic regression | 82.16 | 0.058 |
| Random Forest | 83.17 | 0.016 |

**Conclusion**

The project was conducted to determine whether we can predict heart disease in people using machine learning models and achieve greater accuracy in predicting. From the four models, we achieved good performances for predictions (over 70% in the test set) using metrics such as accuracy, precision, recall, and ROC\_AUC scores. However, the models are overfitting in the testing sets since the scores in the training set are over 80%. This could be due to a small number of examples in the training set. A larger dataset can provide a solution to overfitting. Hyperparameter tuning also improves model performance, as evident in the accuracies of logistic regression and random forest hyperparameter tuned models. To sum up, we can predict heart disease using the models. However, their prediction can be improved by adding more examples and tuning their respective hyperparameters.

References

Ali, M., Khan, M. D., Imran, M. A., & Siddiki, M. (2019). *Heart disease prediction using machine learning algorithms* (Doctoral dissertation, BRAC University).

Cokelaer, T., Bansal, M., Bare, C., Bilal, E., Bot, B. M., Neto, E. C., ... & Hoff, B. (2015). DREAMTools: a Python package for scoring collaborative challenges. *F1000Research*, *4*.

Kwon, J. M., Kim, K. H., Jeon, K. H., Lee, S. E., Lee, H. Y., Cho, H. J., ... & Hwang, K. K. (2019). Artificial intelligence algorithm for predicting mortality of patients with acute heart failure. *PloS one*, *14*(7).

Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, *19*(1), 1-16.

Vabalas, A., Gowen, E., Poliakoff, E., & Casson, A. J. (2019). Machine learning algorithm validation with limited sample size. *PloS one*, *14*(11).