Heart Failure Prediction using KNN, Naïve Bayes, Logistic Regression and Decision Trees

Pradipkumar Rajasekaran 5033

Sonaxy Mohanty 5033

Abstract

In this report, we highlight the hypotheses, experiments, and results of experiments designed to create a machine learning model that will help in early detection and prediction of Heart Failure using various algorithms - K nearest neighbors, Naïve Bayes, Logistic Regression and Decision Tree.

1. Project Domain

We are aware of the fact that cardiovascular diseases (CVDs) are the leading cause of death worldwide, killing an estimated 17.9 million people each year, accounting for 31% of all deaths. Heart attacks and strokes account for four out of every five CVD deaths, and one-third of these deaths occur in adults under the age of 70. CVDs are a common cause of heart failure and we intend to predict heart disease based on the 12 attributes that we have in our dataset. We're going to use a dataset of 36kB from Kaggle which has 12 attributes – Age, Sex, ChestPainType (like Typical Angina, Atypical Angina, Non-Anginal Pain, Asymptomatic), RestingBP, Cholesterol, FastingBS (1 if FastingBs > 120 mg/dl, 0 otherwise), RestingECG (Normal: normal electrocardiogram, ST: having ST wave abnormality or LVH: definite left ventricular hypertrophy), MaxHR (maximum heart rate achieved between 60 and 202), ExerciseAngina (Y person has exercise induced angina, N otherwise), Oldpeak (numeric value), ST_Slope (Up, Flat, Down), HeartDisease (1 if heart disease present, 0 otherwise).

2. Learning Methods

We chose KNN, Naïve Bayes, Logistic Regression and Decision Tree to create our prediction models. We started with implementing K- Nearest Neighbors algorithm, where we considered K neighbors who had a plurality vote in determining if a person will suffer heart failure or not. Secondly, we implemented Naïve Bayes algorithm. Here we tried to establish connections between medical condition attributes in the dataset and heart disease classification to predict the heart failure. Thirdly, we implemented Decision Tree where sequential splits are performed that separated the data into targeted groups. Finally, we implemented Logistic regression method. Here, we trained the logistic regression model which mapped the input attributes and the expected output based on a threshold value.

3. Literature Review

Our first reference (Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm) discusses using a combination of KNN and Genetic Algorithm (GA) for better classification. Here the proposed model employed genetic search as a goodness metric to eliminate redundant and unnecessary features, and to prioritize the attributes that are most important for classification. Then the classifier's accuracy is calculated which reflects the classifier's ability to accurately categorize unknown samples. Genetic algorithms are good algorithms for search and optimization problems. Each solution in a genetic algorithm is represented as a chromosome. Chromosomes are made up of genes which are elements that represent the problem. A collection of chromosomes is called a population. The proposed model (KNN + GA) in the paper gives an accuracy of 100% whereas our KNN model gives an accuracy of 57.6%. Using a combination of KNN and genetic algorithm is a great step in the future for our project.

Our second reference (The Prediction of Heart Disease Using Naive Bayes Classifier and Particle Swarm Optimization (PSO)) where the proposed model applies PSO to select optimal features and then apply Naïve Bayes on these relevant features. Particle swarm consists of N particles. The position of each particle represents the potential solution. The particles change state based on the tendency to keep inertia, tendency to switch to the most optimal position and to switch state according to the swarm's most optimal position. The proposed model (Naïve Bayes + PSO) gives an accuracy of 87.91% whereas our Naive Bayes model gives an accuracy of 89.1%.

Our third reference (Using Decision Tree for Diagnosing Heart Disease Patients) discusses that Decision Tree is one of the successfully used data mining techniques used to help healthcare professionals in the diagnosis of heart disease. In order to improve health outcomes accuracy, this paper systematically suggests to first perform data discretization (unsupervised or supervised), then perform multiple classifier voting, i.e., dividing the data into smaller, equal subsets and developing a decision tree classifier on each subset. The third step the paper suggests is to decide on which Decision Tree type – Information Gain, Gini Index and Gain Ratio, and then performing reduced error pruning. The proposed model in the paper - Nine Voting Equal Frequency Discretization Gain Ratio Decision Tree gave an accuracy of 84.1% while our Information Gain Decision Tree model gave an accuracy of 81%.

Our fourth reference (Estimation of Prediction for Getting Heart Disease Using Logistic Regression Model of Machine Learning) suggests that the proposed system uses the data available to create a model that tries to predict if a patient has heart disease by reading data and also by data exploration. The proposed model also uses Random Search Algorithm on Logistic Regression Model where

random combinations of hyperparameters are used to improve the accuracy of predicting data. The proposed model gives an accuracy of 87% whereas our Logistic Regression model which makes use of ROC and Youden Index to get a better threshold value for hyperparameter gives an accuracy of 91.8%.

Our fifth reference (Deep Neural Network as an alternative to Boosted Decision Trees for PID) discusses why Neural Networks are better than boosted Decision Trees by using binary classification of particles (tau neutrinos and electronic neutrinos) as an example. For the Boosted Decision Tree AdaBoost used, it yields an accuracy of 94.5%. With neural network the accuracy is 95.2% which is slightly better than the boosted decision tree.

Our sixth reference (Heart Failure Prediction Using Machine Learning Techniques) compares SVM, Naive Bayes, Logistic Regression, Decision Tree and KNN with accuracies 85.2%, 75%, 83%, 68% and 81% respectively. The SVM performs the best for the classification problem discussed in the paper.

4. Hypothesis

We hypothesized that we can create an efficient model by implementing various commonly used algorithms – K nearest neighbors, Naïve Bayes, Logistic Regression and Decision Tree.

5. Experiments

First, we started by splitting the data into test (20% of total data), train (60% of total data) and validation (20% of total data) (3-way split) which was used for KNN, Decision Tree and Logistic Regression, whereas for Naïve Bayes we divided the data into only two sets – train (80% of total data) and test (20% of total data) (2-way split). Not all the features in our dataset were numerical, so we had to change them from categorical to numerical using the pandas function pd.get_dummies(). Then we dropped the columns with categorical values and inserted the numerical values columns created. But these dummy values were generated for KNN, Naïve Bayes and Logistic Regression models only. Decision Tree model used both categorical and numerical data in its algorithm. After that, we divided the dataset into features(X) and labels(Y). The dataset was then ready to be used.

Experiment1: We implemented K- Nearest Neighbors, for which we created three functions - one to calculate the Euclidean distance, second function to train the model and third is our main function for KNN where we invoke all the related functions – train, validate and test our model. Using the training and validation data set, we looped through different k values and stored that the K value for which we got the best F1-score. Then we merged the training and validation set and trained our model with K = 15 (best hyperparameter). Using the merged data, we made predictions for the test data set with Accuracy: 57.6% F1 score: 62.1%. We compared our model's test score with the prebuilt sklearn model for K = 15 for which Accuracy: 75.5% and F1 score: 78.5%. Our model performed poorly as compared to the sklearn model (Figure1).

Experiment2: We implemented Naïve Bayes, for which we created three functions - one to calculate the Gaussian probability density, second function to predict the target labels and third is our main function for Naïve Bayes where we invoke all the related functions – train and test our model. Using the training data, we made predictions for the test data set with Accuracy: 89.1% F1 score: 90.7%. We compared our model's test score with the prebuilt sklearn model for which Accuracy: 89.6% and F1 score: 91.1%. Our model performed marginally close to the sklearn model (Figure2).

Experiment3: We implemented Decision Tree, for which we observed that the first split it makes is based on Sex attribute. Then the split happens for ST Slope, followed by Resting ECG and Resting BP. To select the best hyperparameter on our validation data set, we calculated both Gini Index and Entropy. But based on our validation results, we saw that Entropy performed better. Also we checked for multiple maximum depth of the tree and minimum number of samples splits, and four our model performing better for depth of 6 and samples split for 20. Then based on our merged data (training and validation), we made predictions for the test data set. Accuracy: 81% F1 score: 84.6%. We compared our model's test score with the prebuilt sklearn model for maximum depth of 6 and minimum samples split for 20 for which Accuracy: 82.6% and F1 score: 85.2%. Our model performed marginally close to the sklearn model (Figure3).

Experiment4: We implemented our last hypothesized algorithm Logistic Regression for maximum iterations of 1500, for which we first looped through various learning rates using the validation data set and chose the best learning rate to be 0.01 based on our F1-score. Then we implemented ROC Curve and using Youden Index heuristic we calculated the best threshold value to be 0.2871. After finding the best hyper parameters for our merged data (training and validation), we made predictions for the test data set. Accuracy: 91.8% F1 score: 93.4%. We compared our model's test score with the prebuilt sklearn model for maximum iterations of 1500 and no penalty for which Accuracy: 92.4% and F1 score: 93.6%. Our model performed marginally close to the sklearn model (Figure4).

6. Metrics

We used various metrics – Confusion Matrix, Accuracy, Precision, Recall and F1- Score to evaluate our KNN, Naïve Bayes, Logistic Regression and Decision Tree models.

7. Results

The hypothesis that that we can create an efficient model by implementing various commonly used algorithms – K nearest neighbors, Naïve Bayes, Logistic Regression and Decision Tree holds true for every algorithm except KNN. We observed that our models predicted with an accuracy of 91.8% using Logistic Regression, 81% using Decision Tree, 89.1% using Naïve Bayes and 57.6% using KNN.

8. Summary and Future Work

We have presented our implementation of the KNN, Naïve Bayes, Logistic Regression and Decision Tree algorithms on the prediction of Heart Failure. For all the implemented algorithms we saw good results for all the models except KNN.

In order to enhance the accuracy of predictions based on KNN model we can implement GA to eliminate redundant and unnecessary features. To improve our Naïve Bayes model, we can implement m-estimate as we have few data points, some predicted values for the

examples maybe zero even though true values are not. We can also enhance the Naïve Bayes model by implementing PSO to select optimal features. To improve our Decision Tree model, we can do data discretization as part of preprocessing for our data set. We can also enhance the performance of our Logistic Regression model by using regularization. We can implement Ensemble methods to improve our accuracy for predictions using Decision Tree.

9. Contributions

Pradip implemented the KNN and Decision Tree algorithms. Sonaxy implemented the Naïve Bayes and Logistic Regression algorithms. Each members formulated hypotheses, ran experiments, and analyzed their respective algorithms and came up with the performance metrics. Pradip reviewed the M. Akhil jabbar 2013 and Mai Shouman 2011 and Denis Stanev Riccardo 2021 papers. Sonaxy reviewed the Uma N Dulhare 2018 and Montu Saw 2019 and Prasanta Kumar Sahoo 2021 papers.

10. Novelty

We had to use pandas function pd.get_dummies() to convert some categorical attributes to numerical to use in KNN, Naïve Bayes and Logistic Regression. For choosing the best threshold value using Logistic Regression, we used ROC curve. And then based on the ROC curve heuristic of Youden Index we got the desired threshold value.

11. Figures

Metrics	Our Model	Sklearn Model
Confusion Matrix	[[64 31] [47 42]]	[[82 16] [29 57]]
Accuracy	57.6%	75.5%
Precision	67.4%	83.7%
Recall	57.7%	73.9%
F1-Score	62.1%	78.5%

Figure 1. KNN Model Evaluation Results.

Metrics	Our Model	Sklearn Model
Confusion Matrix	[[97 6] [14 67]]	[[98 6] [13 67]]
Accuracy	89.1%	89.7%
Precision	94.2%	94.2%
Recall	87.4%	88.3%
F1-Score	90.7%	91.2%

Figure 2. Naïve Bayes Model Evaluation Results.

Metrics	Our Model	Sklearn Model
Confusion Matrix	[[96 20] [15 53]]	[[92 13] [19 60]]
Accuracy	81.0%	82.6%
Precision	82.8%	87.6%
Recall	86.5%	82.9%
F1-Score	84.6%	85.2%

Figure 3. Decision Tree Model Evaluation Results.

Metrics	Our Model	Sklearn Model
Confusion Matrix	[[106 10] [5 63]]	[[103 6] [8 67]]
Accuracy	91.8%	92.4%
Precision	91.4%	94.5%
Recall	95.5%	92.8%
F1-Score	93.4%	93.6%

Figure 4. Logistic Regression Model Evaluation Results.

12. References

[1] M. Akhil jabbar, B.L. Deekshatulu, Priti Chandra 2013. Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm.

[2] Uma N Dulhare 2018. Prediction system for heart disease using Naive Bayes and particle swarm optimization.

[3] Mai Shouman, Tim Turner, Rob Stocker 2011. Using Decision Tree for Diagnosing Heart Disease Patients.

[4] Montu Saw, Tarun Saxena, Sanjana Kaithwas, Rahul Yadav, Nidhi Lal 2019. Estimation of Prediction for Getting Heart Disease Using Logistic Regression Model of Machine Learning.

[5] Denis Stanev Riccardo, Riva, Michele Umassi 2021. Deep Neural Network as an alternative to Boosted Decision Trees for PID

[6] Prasanta Kumar Sahoo, Pravalika Jeripothula 2021. Heart Failure Prediction Using Machine Learning Techniques

[7]https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction