CS235 Fall'22 Project Final Report: Heart Disease Classification Using Five Different Classification Techniques

Manojsai Kalaganti #1
NetID: mkala011

Archana Kallakuri #2
NetID: akall011

Sree Charan Reddy Gangireddy #3
NetID: sgang011

Nandan Paritala #4
NetID: pnand006

Vedant Chaubey #5
NetID: vchau024

ABSTRACT

One of the major medical issues for a very long time had been cardiovascular disorders. According to the World Health Association, heart conditions rank first among the top 10 major causes of mortality. A crucial first step in recovery and treatment is accurate and prompt identification. It would be necessary to build a system that could foretell the occurrence of heart problems in order to diagnose cardiac abnormalities. In the current article, our primary goal is to create an effective machine learning-based medical system that will aid in determining a patient's heart state and help a doctor determine whether or not the patient has cardiovascular disorders. We address the issue of missing data as well as the issue of imbalanced data in the publicly accessible UCI Heart Disease dataset and the Framingham dataset using a variety of data processing approaches. Additionally, we choose the best effective method for forecasting cardiovascular illnesses using machine learning. Our system was tested using a variety of measures, including accuracy, sensitivity, F-measure, and precision, which clearly show that the suggested technique performs far better than previous approaches.

KEYWORDS

data, mining, Cardiovascular diseases, Data imputation, Machine learning, Preprocessing, Normalization

ACM Reference Format:
Manojsai Kalaganti #1, Archana Kallakuri #2, Sree Charan Reddy Gangireddy #3, Nandan Paritala #4, and Vedant Chaubey #5. 2018. CS235 Fall'22 Project Final Report: Heart Disease Classification Using Five Different Classification Techniques. In *Proceedings of Data Mining Techniques (CS235 F22).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

A person can have heart disease and not feel exhausted. A few people with heart disease have symptoms. This is when there are changes or pain in the body that indicate a sickness is present. Among the symptoms of heart disease are: Pain in the chest, difficulty breathing, palpitations (the sensation that the heart is pulsing too quickly), swelling of the feet or legs Feeling weak because the body and cerebrum are not getting enough blood to supply them with oxygen, Cyanosis (skin turning a blue tone) (skin turning a blue tone). Age, sex, tobacco use, actual idleness, unreasonable liquor utilization, undesirable eating routine, weight, hereditary proclivity and family history of cardiovascular sickness, raised pulse (hypertension), raised glucose (diabetes mellitus), raised blood cholesterol (hyperlipidemia), undiscovered celiac infection, psychosocial variables, destitution and low educational status, and air contamination are all risk factors for heart disease. While the individual commitment of each risk factor varies across networks or ethnic groups, the overall commitment of these risk factors is extremely reliable. Some of these risk factors, for example, age, gender, or family ancestry/hereditary proclivity, are unchangeable; however, numerous significant cardiovascular risk factors are modifiable through lifestyle changes, social change, and drug treatment (for instance anticipation of hypertension, hyperlipidemia, and diabetes). Obese people are more likely to develop atherosclerosis of the coronary supply routes. Age is the most important risk factor for developing cardiovascular or heart diseases, with the risk increasing with age. Coronary greasy streaks can begin to frame in youth. It is estimated that 82% of people who die from heart disease are 65 or older. At the same time, the risk of stroke doubles every decade after the age of 55. Many explanations have been proposed to explain why age increases the risk of cardiovascular/heart infections. One of them corresponds to the serum cholesterol level. The serum absolute cholesterol level rises with age in many populations. This increase in men peaks between the ages of 45 and 50. In women, the increase is more pronounced until the age of 60 to 65.

As a result, preventing heart disease has become more important than ever. Great information-driven frameworks for anticipating heart diseases can improve the overall examination and prevention measure, allowing more people to live healthy lives. Al aids in the prediction of heart illnesses, and the predictions made are very accurate. It entails investigating the heart disease patient dataset and preparing relevant information. At that time, various models were developed, and predictions were made using various calculations such as KNN, SVM, Decision Tree, Nave Bayes, Random Forest.

2 PROPOSED METHODS

Our approach is to thoroughly analyze the dataset and then apply various machine learning algorithms. Firstly, we implemented a

CS235 F22, Fall 2022 quarter,, Riverside, CA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

probability-based classification method known as Naive Bayes. Further, we employed a tree-based technique such as Decision Tree Classifier. Then, we used a very popular and widely used ensemble method, Random Forest Classifier. We also used Support Vector Machine to check and handle the data's high dimensionality. Another method which we used was the KNN Classifier.

2.1 Random Forest

2.1.1 Overview. :

The random forest algorithm is a supervised classification algorithm. As the name implies, this algorithm builds a forest out of many trees. In general, the more trees there are in the forest, the more robust it appears. Similarly, in the random forest classifier, the greater the number of trees in the forest, the higher the accuracy results. Random forest algorithms are commonly used in classification and regression applications. A decision tree is built from different samples, which are then combined and averaged for classification and regression.

```
1: procedure RANDOM FOREST
2:     for 1 to T do
3:         Draw n points D_l with replacement from D
4:         Build full decison/regression tree on D_l
               BUT: each split only consider k features, picked uniformly at random
                   new featues for every split
5:         Prune tree to minimize out-of-bag error
6:     end for
7:     Average all T trees
8: end procedure
```

Figure 1: *Random Forest Algorithm*

The method of the classes (characterization) or mean/normal expectation, along with the construction of a large number of choice trees during preparation time, are the key components of the random forest group learning strategy for arrangement, relapse, and various tasks. The simplicity and variety of this calculation make it one of the most common ones.

2.1.2 Comparing implementation with Scikit Learn. :

We experimented with various n tree and max depth values. These settings will allow us to get the most out of the RandomForestClassifier in terms of precision. While the built-in library had an accuracy of 81.4%, we were able to implement the Random Forest Procedure from scratch and achieve an accuracy of 82%. I was able to achieve marginally higher accuracy by slightly altering the pre-processing step. Instead of doing it the other way around, this time I balanced the unbalanced class before splitting the data.

2.2 Gaussian Naive Bayes

I. Overview:

Naive Bayes Classifier is a technique based on Bayes' Theorem with a 'naive' assumption of independence and equal importance amongst all the features. The main advantage of of Naive Bayes lies in it's simplicity, it is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c) as below-

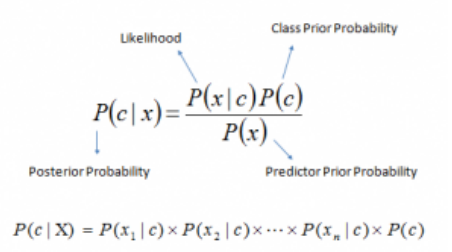

Figure 2: *Bayes rule*

In Naive Bayes unlike the other many classifiers, maximum-likelihood can be accomplished accomplished by evaluating a closed-form expression in linear time rather than through the iterative approximation which is time consuming. As described Bayes' theorem the likelihood of an event is based on previous knowledge of conditions that may be essential to the event.In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. The basic algorithm behind the Gaussian Naive Bayes is as follows.

```
1. for q = 1…w // loop for each mining models element
2.     μ[q] = 0; // initialization of mining models elements
3. end for;
4. for j = 1…m // loop for each row
5.     μ[d[j,p]]++; // increment number of row for value x_{j,p} of object x_j;
6.     for k = 1 …p-1 // loop for each column
7.         μ[φ(k-1)+(d[j, k]-1)·φ(0)+ d[j, p]]++; // increment number of rows with value x_{j,k}
                                                  // and value x_{j,p}, where φ(k)=s+∑_{q=1}^{k}(|T_q|·s)
8.     end for;
9. end for;
```

Figure 3: *Basic Naive Bayes Algorithm*

The code implementation for Guassian Naive Bayes classifier is as follows:

1. Calculation of priori probablity for each of our labels.
2. Calculating the maximum likelihood function for all of our features.
3. Calculating the posterior probablity and classifying the data into the class for which it has greater probability.

Classification using Gaussian Naive Bayes was simple but efficient and it gave us results comparable to the off-shelf implementations of the same.

2.3 Scalar Vector machines

2.3.1 Overview. :

SVM is a classis classifier algorithm used to for both Linear and Non linear datasets. It generates hyperplanes to classify dataset into two classes. After transferring nonlinear data to the appropriate higher dimension using a nonlinear mapping, it determines the ideal hyperplane. The Support vectors which are data tuples are used to draw descision boundaries, to find the hyperplane. SVM is used to calculate the biggest margin i.e. the perpendicular distance

between two classes of support vectors.

SVM could also be used for generating hyperplanes for multiclasses. However for our study we are restricting ourselves to two class classifier. We are trying to achieve maximum margin hyperplane for our data set which helps in easily classifying the data. Following is the pseudo code and the basic equation of the algorithm:

Algorithm 1 Pseudo-code of SVM algorithm

Inputs:Determine the various training and test data.
Outputs:Determine the calculated accuracy.
Select the optimal value of cost and gamma for SVM.
while (stopping condition is not met) **do**
    Implement SVM train step for each data point.
    Implement SVM classify for testing data points.
end while
Return *accuracy*

Figure 4: *Scalar vector machines*

$$y_n\left[w^T\phi(x) + b\right] = \begin{cases} \geq 0 \; if \; correct \\ < 0 \; if \; incorrect \end{cases}$$

Figure 5: *SVM Equation*

Where y and x are related to the datasets and w is the weight.

In addition to using the SVM algorithm we have also considered the cost function with hinge loss and implemented reguralization in order to optimise the weights.

$$Hypothesis: \quad h_\theta(x) = \begin{cases} 1 & \text{if } \theta^T f >= 0 \\ 0 & \text{otherwise} \end{cases}$$

$$Cost\ Function: \quad J(\theta) = C[\sum_{i=1}^{m} y^{(i)} Cost_1(\theta^T(f^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T(f^{(i)})]$$

Figure 6: *Cost funciton*

2.3.2 Comparing implementation with Scikit Learn. : Using the Sklearn libraries SVM algorithm we were able to achieve an accuracy of 84.6% on testing data and 84.8% on traning data. On the other side the algorithm we implemented achieved an accuracy of 84.56% on test and 84.82% on train dataset. This signifies that the algorithm we implemented is well optimised for out dataset.

2.4 Decision Tree

2.4.1 . I. Overview:
Decision trees are used both for classification and Regression. Each leaf node contains a class label, each internal node represents a test on an attribute, and each branch represents the result of a test. The root node of a tree is the node at the top. Classification rules are represented by the routes from root to leaf nodes.By using information gain as a criterion, we try to estimate the information contained by each attribute.

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

Figure 7: *Expected information needed to classify a tuple in dataset D is*

$$Info_A(D) = \sum_{j=1}^{v} |D_j| / |D| \times Info(D_j)$$

Figure 8: *Information needed after using attribute A to split D into v partitions*

$$SplitInfo_A(D) = -\sum_{j=1}^{v} |D_j| / |D| \times \log_2 |D_j| / |D|$$

Figure 9: *Potential information generated after using attribute A to split D into v partitions*

$$GainRatio\ (A) = Gain\ (A) / SplitInfo_A(D)$$

Figure 10: *Gain Ratio for attribute A*

2.4.2 Comparing implementation with Scikit Learn. :
The accuracy has slightly improved from the one we casted during the midterm report. While we worked on Decision Tree using library, we could get a total modal accuracy of 73%. After changing some parameters in the algorithm we ultimately were able to reach a total modal accuracy of 77%

2.5 K Nearest Neighbors Classifier

2.5.1 . I. Overview:
K Nearest Neighbors Classifiers of KNN Classifier is a non parametric supervised learning classifier which uses the distance metric to make predictions on how and where to group the data point.

2.5.2 Comparing implementation with Scikit Learn. :
The accuracy has slightly improved from the one we casted during the midterm report. While we worked on KNN Classifier using the library, we could get a total accuracy of 0.73. After changing some parameters in the algorithm we ultimately were able to reach a total accuracy of 0.82

Then we compared the accuracy with different distances and also with different dimensionality reductions.

(1) Distance Metrics considered
   a) Euclidean
   b) Manhattan
   c) Hamming
   d) Minkowski
   e) Chebyshev

(2) Dimensionality Reduction considered
   a) PCA%
   b) LDA%
   c) NCA%
   d) Sparse PCA%
   e) Fast ICA%

We were able to get the same results on our KNN implementation which we implemented from scratch. and achieved an accuracy of 83.2, greater than the sci-kit implimentation of KNN.

3 EXPERIMENTAL EVALUATION

Specific metrics like f1 score, precision, recall, and accuracy are used as the basis for the evaluation criteria used to gauge the algorithm's performance. Additionally, we track how long it takes for each algorithm to train. The confusion matrix displays the classification algorithm's results, which serve as the foundation for calculating various parameters. As a result, the confusion metrics help determine whether a classification algorithm frequently labels objects incorrectly by comparing predicted values with actual values.

- True positive (TP) – Are the instances where the actual class of a datapoint is 1 and the predicted class is also 1
- False positive (FP) – Are the instances where the actual class of datapoint is 0 and predicted is 0.
- False negative (FN) – Are the instances where the actual class of datapoint is 0 and predicted is 1.
- True negative (TN) – Are the instances where the actual class of datapoint is 1 and predicted is 0.

(a) F1_score:
When a f1-score reaches its best value at 1 and its worst score at 0, it is interpreted as a weight of recall average and precision. The formula for f1-score is:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

(b) Recall:
Recall measures how much relevant data is retrieved from any machine learning algorithm. It is concerned with the ability to locate all related occurrences in the data. A recall is represented by the following equation:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

(c) Precision: Precision is the quality of being precise and correct. Precision conveys the concept of correctly predicted events. It quantifies positive predictions by measuring the proportion of true positives among all positives, and this is calculated as:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

(d) Accuracy: Accuracy is a critical metric for describing an algorithm's performance. It defines the stage at which an algorithm can correctly predict positive and negative cases and is measured using the formula:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP} + \text{FP} + \text{FN+TN}}$$

(1) Random Forest
   a) Accuracy: 82%
   b) F1_Score: 82%
   c) Recall: 82%
   d) Precision: 85%

(2) Naive Bayes
   a) Accuracy: 81.72%
   b) F1_Score: 80.83%
   c) Recall: 81.7%
   d) Precision: 80.17%

(3) Decision Tree
   a) Accuracy: 77%
   b) F1_Score: 76.89%
   c) Recall: 76.24%
   d) Precision: 77.12%

(4) SVM
   a) Accuracy: 84.6%
   b) F1_Score: 77%
   c) Recall: 85%
   d) Precision: 72%

(5) KNN
   a) Accuracy: 83.72%
   b) F1 Score: 15.8%
   c) Recall: 31.0%
   d) Precision: 11.1%

4 RELATED WORK

1. Framingham heart Study: The dataset which we used was from the Framingham Heart study(FHS). FHS was launched in 1948 to identify common factors or characteristics that contribute to

cardiovascular disease. FHS has gathered multi featured data for almost two generations to predict the risk of heart attack for given factors such as family patterns, diabetes, smoking etc. More information of FHS could be found in the given link:
Source: https://www.framinghamheartstudy.org/fhs-risk-functions/cardiovascular-disease-10-year-risk/

2.Soni, J., Ansari, U., Sharma, D. and Soni, S., 2011. Predictive data mining for medical diagnosis: An overview of heart disease prediction. International Journal of Computer Applications, 17(8), pp.43-48.

The problem definition considered in this paper is similar. They are using Decision Tree, Bayesian classifier, Decision Trees, KNN, Neural Networks etc for Heart disease prediction albeit on a different set.They have observed Decision trees and Bayesian classifiers to perform well.

3.Apurb Rajdhan, Avi Agarwal,Milan Sai,Poonam Ghuli,Dundigalla Ravi: Heart Disease Prediction using Machine Learning

This paper makes use of a heart disease dataset. The proposed work predicts the chances of Heart Disease and classifies patient's risk level by implementing different data mining techniques such as Naive Bayes, Decision Tree and Random Forest. Thus, this paper presents a comparative study by analysing the performance of different machine learning algorithms which is similar to our methodolgy.

4. V.V. Ramalingam,Ayantan Dandapath,M Karthik Raja: Heart disease prediction using machine learning techniques: a survey

This paper surveys numerous models built using these methods and algorithms and evaluates their effectiveness. Models are built using supervised learning techniques including Support Vector Machines (SVM), K-Nearest Neighbour (KNN), Naive Bayes, Decision Trees (DT), and Random Forest (RF).

5.T. J. Peter and K. Somasundaram, "An empirical study on prediction of heart disease using classification data mining techniques," IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012), 2012, pp. 514-518.

This paper measures the performance of different classifiers for prediction Cardio Vascular disease. Along with classification they have used different preprocessing techniques to improve the predictions. They found that Naive Bayes with CFS attribute selection showed better results amongst Decision tree classifier, Naive Bayes, K-NN and Neural Networks.

5 DISCUSSION & CONCLUSIONS

For the Heart Disease classification we have taken 5 classifiers, Gaussian Naive Bayes, Decision Trees, Random Forest, SVM and K-NN. We preprocessed our dataset first and then built our implementation from scratch and compared their performance to the off-the shelf implementations. Upon comparing the results of each classifier we can see that the accuracy for all the classifiers ranged from 0.7 to 0.8.