HEART DISEASE PREDICTION USING DATA MINING CLASSIFICATION ALGORITHMS

S M RAHID HAQUE[1], MD. ATIK FOYSAL[1], ARUPKUMAR DAS[1], MD. SHAHIDUL ISLAM LEAON[1], DR. MD. ABDULLAH - AL - JUBAIR[2].

[1]Student of Computer Science Department, [2]Assistant Professor of Computer Science Department, American International University-Bangladesh, Dhaka.

Abstract

A range of conditions which affect heart is called heart diseases or “cardiovascular diseases”. This disease can bring out heart attack, chest pain, stroke etc. By reviewing some research paper related to heart disease prediction it was identified that most of the paper using singular algorithm to predict the disease using machine learning algorithm. Some of them indicates that they can’t use optimization techniques to improve their model performance. For these results, they have faced some problem to predict heart disease in an efficient manner by using their proposed system. To overcome these problems and for getting more accurate results in this medical study is very crucial that’s why four different classification algorithms were implemented to predict heart disease and find out the effectiveness of these algorithms. In this study Principal Component Analysis (PCA) dimensionality reduction technique was applied which helped to get better results with the aim of better accuracy by using these algorithms since medical diagnosis is sensitive. For this approach data was collected from UCI repository which was found in Kaggle and it is named as “Heart Disease UCI”. It was observed that without Principal Component Analysis (PCA) Logistic Regression performed best to predict heart disease with Principal Component Analysis (PCA) k-Nearest Neighbors (KNN) achieved greater accuracy compared to other classification algorithms. By applying PCA it was identified that accuracy for other algorithms like Decision Tree and Naive Bayes also increased compared to originally.

Keywords: Cardiovascular Diseases, Machine Learning, PCA, Heart Disease Prediction, Classification Algorithm

1. Introduction

Proper health is the key to a country’s well-being. A range of conditions that affect the heart are called heart diseases. It is also called “cardiovascular diseases”. In this disease, blood vessels can be prevented from moving and it can bring about heart attack, chest pain, stroke etc. Some other conditions also considered as heart diseases such as the heart state which attack the heart’s muscle valves or pattern.

The concept of ‘data mining’ describes the process of extracting patterns from vast quantities of data. Data mining includes variety of algorithms like classification for supervised learning and clustering for unsupervised learning.

1.1. Problem Statement

In previous clinical trials, machine learning approaches were used to forecast heart disease. However, these researches concentrated on the singular results of specific machine learning techniques but not using any optimization of techniques for improvement [1] [2]. In addition, these researchers tried to use hybrid optimization methods for an optimized classification of machine learning. For heart disease prediction using data mining techniques some of the researches don’t provide valid accuracy results [3] as they haven’t provided any optimization techniques in improving the classification model and haven’t checked the performance of the model in the obtained result. So, there is a need for study to find out effectiveness and how classification algorithms work in predicting heart disease.

1.2. Objectives

The overarching goal of this study is to see whether machine learning classification algorithms will accurately predict heart disease. Based on the reviews of previous research the objectives of this research are to predict heart diseases using data mining classification techniques such as Naive bias, K-Nearest neighbor, Logistic Regression Decision Tree. Then online open-source platform is used which is https://jupyter.org/ where it can develop the proposed machine learning algorithms. Finally, it must figure out how to improve the model, compare the accuracy and performance of different classification models.

2. Background Study

According to World Health Organization, heart disease has been one of the top killers, with millions of individuals dying annually. This is a type of disease that attacks both the heart and vascular system of the body and causes a range of heart related infections like coronary heart disease, hypertension,cerebrovascular disease etc. [4]. According to earlier findings, the chances of developing coronary heart disease was 34.9 percentage for males and 24.2 percentage for females at the age of 70 and 42.4 at age 40 respectively [5]. Heart disease shows early signs of symptoms like breathlessness, orthopnea, fatigue, tiredness, ankle inflammation etc. and diagnosis includes Chest Radiograph, Electrocardiogram (ECG), Echocardiography etc. which is really pricey [6].

In this modern age there is data in nearly all regards of our life like it can be in hospitals, business stores, social media where large volumes of data are collected like medical records, sales records etc. and data mining can help turn those data into a knowledge [7]. Data mining is seen as an adaptation in information technology from early database systems in 1970 where data are tabulated in relational database analysis systems [8]. Then in 1980 it was researched and progressed to advanced database storage techniques where there was an abundance of storage systems and equipment to store large amounts of data and finally in 1990 web-based databases like XML was introduced where data was plenty and information gathering using data mining now far easier. Data mining classification algorithms can be used to predict the labeled class of any data set and to solve real-world problems including heart disease [7] [8].

3. Methodology

In the study, heart diseases prediction framework is introduced and this approach aims to analyze performance of different classification algorithms and find out effectiveness of the classification algorithms.

Datasets are encrypted and loaded within this proposed system. After that the data is divided into two sections, with one half of the sample being used for training and another half being used for evaluating our model. Then different classification techniques are used such as Decision tree, K-nearest neighbor, Logistic Regression and Naive Bayes analyze performance by checking accuracy, precision, recall, f1 score, confusion matrix diagram of the algorithms to find the most effective classification algorithm. At last, PCA feature selection technique is applied to improve performance of the algorithm.

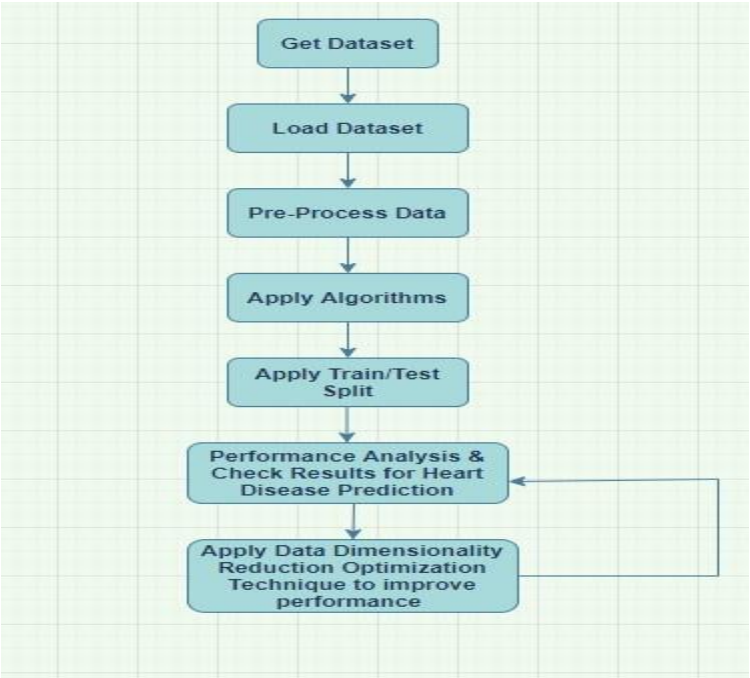

Get Dataset
Load Dataset
Pre-Process Data
Apply Algorithms
Apply Train/Test Split
Performance Analysis & Check Results for Heart Disease Prediction
Apply Data Dimensionality Reduction Optimization Technique to improve performance

Figure 1: Proposed Framework of Heart Disease Prediction

3.1. Dataset Collection

For this research, collection of data is the first phase of the proposed technique to predict the heart disease. In this process, the data will be collected from UCI repository which was found in Kaggle and it is named as “Heart Disease UCI” [9]. This public data is widely used by ML researchers around the world for heart disease research in data mining as stated in the website. The “target” column holds if a patient has heart disease or not and it denoted by 1 or 0 which will be used for prediction.

This data has 13 attributes (age, gender, cp, trest bps, cholesterol, fbs, Rest ECG, Thalach, exang, old peak, slope, CA, Thal) 1 class label attributes (Diagnosis of heart disease) and 303 instances of data originally.

3.2. Loading and Pre-Processing Data

After collecting the data second phase is about loading the data by reading the .csv file. For reading the .csv file the command used for it was the pd.read and pd was imported as import pandas as pd. Then the data was ultimately loaded inside the notebook for pre-processing.

After loading the data next step is about pre-processing it. The author in [11] said that in actuality datasets are extremely vulnerable because of noise, absent and conflicting data due to natural error or large datasets. Data pre-processing is the way to solve these issues. In this phase there are many preprocessing approaches that can used in the dataset. It can be data cleaning, data reduction, data integration etc. In visualization we have to check if the target data is balanced or not. In this research we are tracking the dataset to detect any null values. In order to do that isna() function will be used to detect null and in addition the sum of null values with .sum() will be calculated for each columns. Moreover, we will also apply the feature scaling method. This method is mainly used to normalize the heart disease dataset. In order to that library must transferred from sklearn which is StandardScaler and this function will be loaded inside a variable so that the object can be created to apply this method.

3.3. Performing Train-Test Split on Algorithms

After pre-processing train-test split will be applied in the dataset. As stated by [10], it is necessary to train the data based on training data and make a prediction on the test data. So, the test data is basically used to assess the performance of the model.

In this research, dataset is divided into 70 percentage for training and remaining 30 percentage for testing purpose. Therefore, among the 303 instances roughly 91 samples will be used for testing purposes and other samples for training. Random state is also specified which is 40 for data splitting so that model perform with repeated data every time we run. To accomplish all of this, we must first install the Scikit-Learn library and then access the train test split() function from the Python library.

In order to do perform the train-test split next step is to find the suitable algorithms. We have to apply train-test split on wide variety of supervised data mining classification algorithms. Classification algorithms that will be used are Decision Tree, KNN, Logistic Regression and Naive Bayes. For all the algorithms again python machine learning library will be used and transfer the suitable algorithms.

3.4. Checking Performance and Results of the Algorithms

In the second last phase, after applying algorithms next step will be is to check the performance of each of them. In this phase, accuracy score must be determined first. The number of correctly estimated instances in comparison to the overall number of instances will be determined by the accuracy ranking. Then confusion matrix will be applied to check the performance of each algorithm in more detail. Precision, recall, f1 score, and accuracy will be created by the confusion matrix, and the results will be mentioned in the results section of the confusion matrix article.

3.5. Checking Performance and Results of the Algorithms

In the final phase, PCA optimization techniques will be used and compare the result again with applying it. PCA is used to increase the algorithm’s efficiency by making it work quicker by dramatically reducing training time, and it does so by reducing measurements [12]. It also helps reducing overfitting so algorithms like decision tree with low number of features in dataset can work properly with improved accuracy and performance.

4. Implementation and Evaluation

4.1. Algorithms

4.1.1. Decision Tree

In order to implement the decision tree certain steps are followed in our study. At first the Decision tree is loaded using scikit-learn libraries and other required libraries. Then the feature and target variables are separated and train-test split is also applied. In order to build decision tree performance is improved by setting parameters with pre-pruning by selecting the max depth and attribute selection measurement like entropy [13]. Then the performance is evaluated using confusion matrix.

4.1.2. Logistic Regression

The primary goal of this technique is to estimate the probabilities of events. When the address variable is categorical and only has a few distinct variations, this method is used. In order to apply in this following research at first the logistic regression libraries are from scikit-learn from python. Then the data is loaded and split using train-test method. Then finally confusion matrix will be generated to determine the performance of the model.

4.1.3. K-nearest Neighbor

In order to implement KNN in our study python scikit-learn technique was used. At first the data was loaded then train/test spilt was applied. KNN was imported from the library and fitted into the data. One of the crucial steps in applying KNN is to determine the value of ‘K’ [14]. In order to find the value of ‘K’ for loop was applied to test the accuracy in range of neighbors from 1 to 100 in our research. K was determined by plotting the graph with the respect to value of K and testing accuracy. The value of ‘K’ was chosen where accuracy was highest with small value of K to minimize overfitting.

4.1.4. Naive Bayes

In this study dataset composed of binary labels which is “1” or “0” so it is best to use Gaussian Naïve bayes technique to apply in the dataset to get the heart disease prediction. It also only measures the mean and standard deviation from the training dataset and it is easier to apply. So, in order to do that we used sklearn techniques to import Gaussian Naïve bayes from python library. Train/test method was applied and algorithm is fit inside the dataset. Finally, the prediction was made and the accuracy value was assessed.

4.2. Performance Measurement using Confusion Matrix

In order to get more information about the performance of the algorithms on accuracy is confusion matrix and it summarizes overall performance of the classification algorithms that is used in this research.

1) True Positive (TP): True Positive occurs when actual value was positive and predicted label value was also positive. In this case it means a patient has heart disease and algorithm also predicted that he/she actually has it.

2) True Negative (TN): True Negative represents when actual value was negative and predicted label was also negative. For instance, in this case, it means patient does not have any heart disease and also predicted to not have any.

3) False Positive (FP): False Positive happens when actual value was negative but the algorithm prediction came positive. For example, in this research it can be said that patient do not have any heart disease but predicted to have the disease

4) False Negative (FN): False Negative happens when actual value was positive but the algorithm prediction came negative. In this research, it can be said that patient have heart disease in their body but predicted to not have any.

5) Accuracy: One of the common performance detection techniques used by confusion matrix is the accuracy measurement. It determines how frequently the algorithm can yield actual output. It can be measured as the ratio of number of correction prediction to the total number of predictions that is obtained by the classifier.

$$\text{Accuracy: } (TP + TN)/(TP + TN + FP + FN) \quad (1)$$

6) Precision: Precision measures how much percentage of patients that actually predicted to have heart disease among the patients who are classified to have heart disease. It can be calculated as:

$$\text{Precision: } TP/(TP + FP) \quad (2)$$

7) Recall: Recall calculates how much patients have predicted to get heart disease among the patients that actually have this disease. It can be calculated as:

$$\text{Recall: } TP/(TP + FN) \quad (3)$$

8) F1-Score: The f1-score is taken as the harmonical mean of accuracy as well as recall. The maximum value is given when the same accuracy and reminder value are the same.

$$\text{F1−score: } (\text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall}) * 2 \qquad (4)$$

4.3. Train-Test Split

In order to implement train/test split in the research we used scikit-learn technique from python library by passing the train test split() function and applied it to solve our classification problems. In this study, we also used 70 percentage for training the data and other 30 percentage for testing purpose and also 42 selected as random state so that same data used every time instead of using different

4.4. Result Evaluation

1) Performance of Decision Tree before and after applying PCA:

The figure below demonstrates the original results of Decision tree(left) and result after applying Decision tree with PCA (right). It can be interpreted that 35 patients have heart disease positive which is true positive with 14 patients false negative originally without PCA whereas 34 predicted to have heart disease correctly and only 8 patients with false negative after PCA which is an improvement. Therefore, the achieved recall value is increased which is 0.714 to 0.809. Decision tree also performed relatively well compared to original as the accuracy is increased from 0.791 to 0.846 which is quite significant. The total result of recall and f1-score was also shown to have a beneficial influence with P

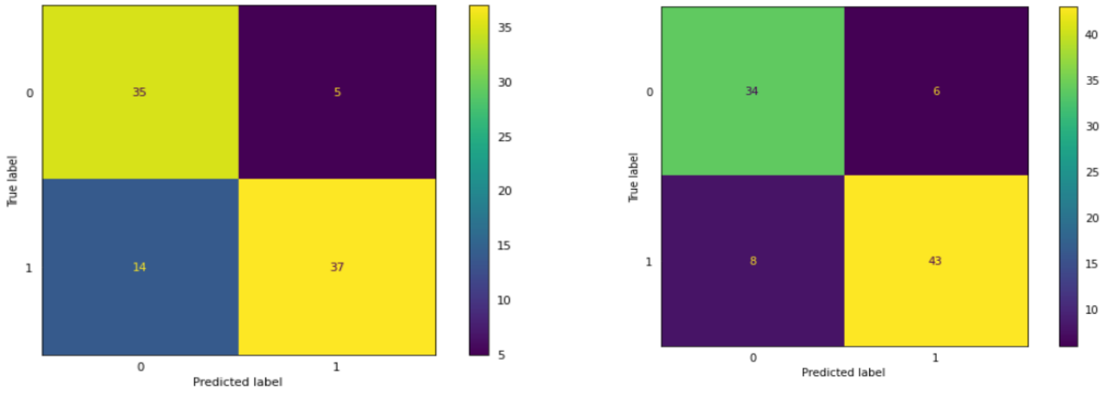
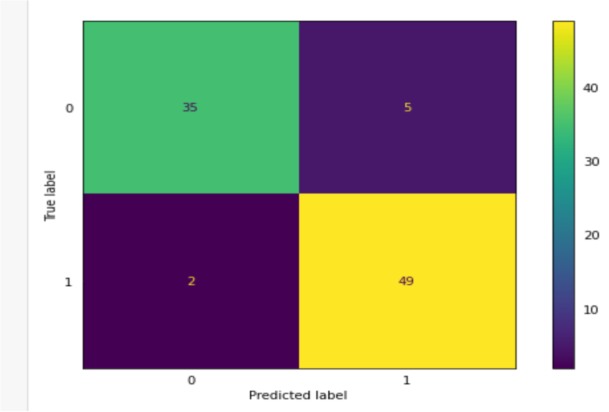


Figure 2: Decision Tree Confusion Matrix before (left) & after applying PCA (right)*2)* Performance of Logistic Regression before and after PCA:The figure below illustrates the initial result of Logistic Regression (left) and result of this algorithm with PCA (right). It can be seen that 35 patients predicted to have heart disease but only 2 patients are incorrectly predicted to not have the disease without PCA but 36 patients predicted whereas again 5 patients wrongly predicted in terms of true positive and false positive with PCA. It can be seen that there is a change in accuracy as it is decreased from 0.923 to 0.901. So, overall, there is a decrease in progress after applying PCA therefore it is not recommended to use PCA with Logistic regression as it performs relatively worse.



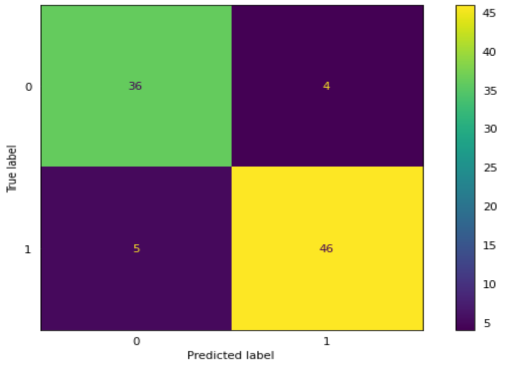


Figure 3: Logistic Regression Confusion Matrix before (left) & after applying PCA (right)

3) Performance of KNN before and after PCA:

In the results below, it depicts the results of confusion matrix of KNN algorithm before(left) and applying (right) PCA technique. Initially, it can be seen that without PCA 34 patients have predicted to have heart disease but only 2 incorrectly identified to don't have heart disease and 7 patients incorrectly predicted as heart disease patient. But in case of PCA, 6 patients wrongly predicted to have this disease, so PCA performed slightly better compared to initially. The initial result also shows that KNN performed with the accuracy of 0.901 but with applying PCA it is increased to 0.912. Recall, precision and f1-score also increased slightly with PCA so it is better to use PCA with KNN algorithm as it performs relatively better.

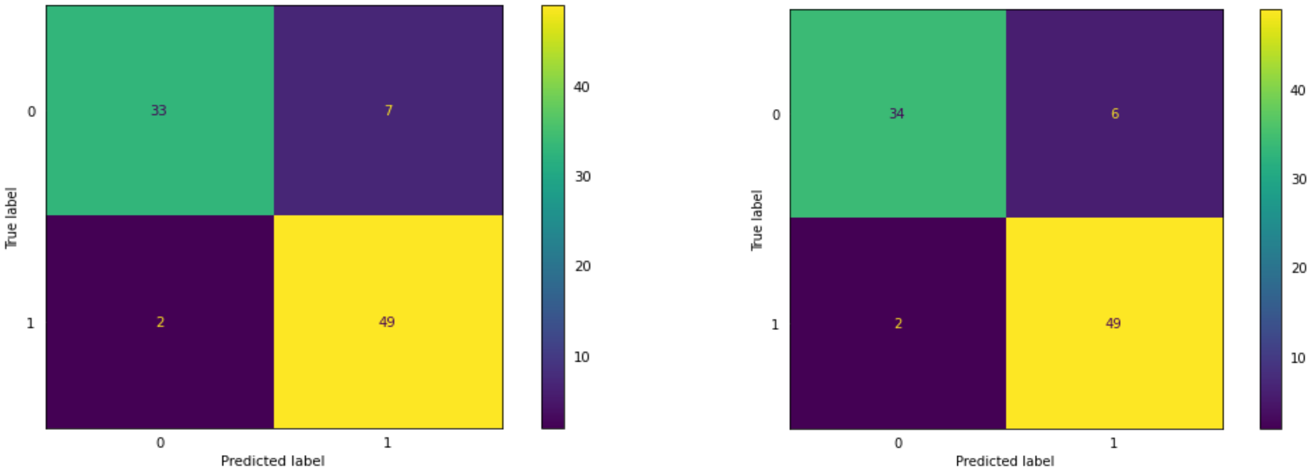


Figure 4: KNN Confusion Matrix before (left) & after applying PCA (right)

4) Performance of Naïve Bayes before and after PCA:

The figure below represents the results of confusion matrix before and after applying PCA. It can be seen that in both cases 5 patients wrongly predicted to have heart disease. However, without PCA, six patients were incorrectly predicted to be safe of heart disease, while only five patients were incorrectly predicted to be free of heart disease with PCA. Accuracy of the result in predicting heart disease also slightly increased after applying PCA from 0.879 to 0.890. Recall, precision, f1 score all increased drastically. So, overall, it can be seen that there is an improvement in the result after applying PCA and no result seen any decrease in value after applying.

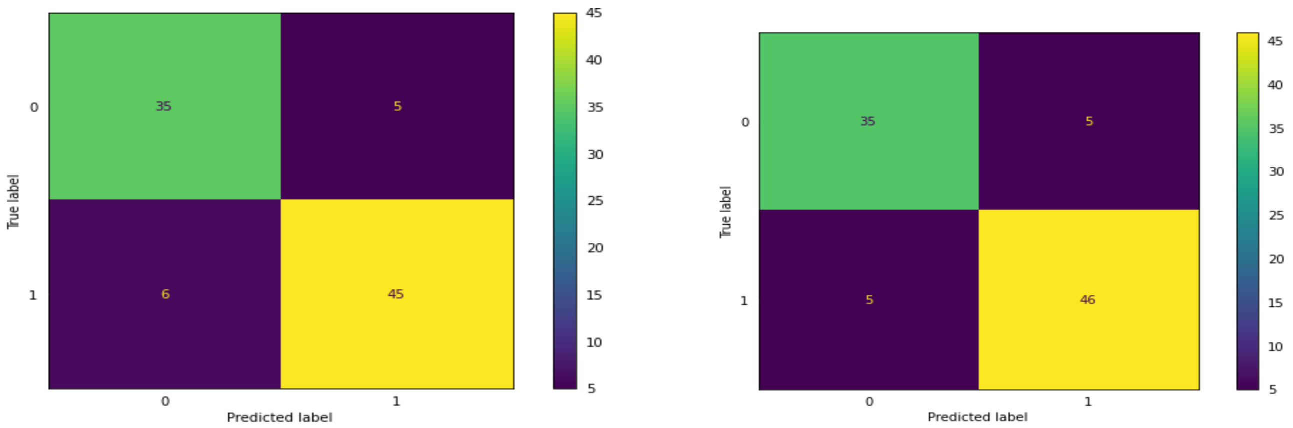


Figure 5: Naïve Bayes Confusion Matrix before (left) & after applying PCA (right)

Findings

After applying dimensionality reduction technique with PCA there are 9 attributes remaining in the dataset based on the explained variance. In the dataset those 9 attributes are trest bps, cholesterol, fbs, Rest ECG, Thalach, exang, old peak, CA, Thal with target attributes. From the above result evaluation proof is provided that Logistic Regression outperformed other models in heart disease diagnosis, with an accuracy of 0.923. Other algorithms like KNN and Naïve Bayes also performed well and results are close to Logistic Regression with accuracy 0.901 and 0.879 respectively. But Decision Tree performed worse out of all the four algorithms with only 0.791 accuracy which is very low compared to others. After the development of PCA with these algorithms it can be seen that KNN performed well as accuracy increased to 0.912 compared to other algorithms. One of the highlights of using PCA is that Decision Tree seen huge improvement in terms of accuracy as it increased from 0.791 to 0.846 so it can be concluded that reducing dimension of the dataset with PCA can increase performance of Decision Tree significantly. Naïve Bayes also seen slight improvement while using PCA with accuracy of 0.890. In introducing PCA performance metrics for the logistic regression decreased all over the board. But there is an improvement for all the other three algorithms which is Decision Tree, KNN and Naïve Bayes by applying PCA. As dimensionality of the dataset reduced with PCA so the accuracy of the decision tree increased greatly since number of leaf reduced. But with introduction of PCA logistic regression performance reduced, since minimizing the number of features in the dataset it resulted in reducing our accuracy slightly. By applying PCA for KNN and Naïve Bayes we can see that there are slight changes in accuracy. The table below shows the results of the accuracy before and after applying PCA:

Table 1: Result of Accuracy, Precision, Recall, F1-Score without PCA**Algorithms**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Decision Tree	0.791	0.875	0.714	0.786
Logistic Regression	0.923	0.875	0.945	0.908
K-Nearest Neighbor	0.901	0.825	0.943	0.880
Naïve Bayes	0.879	0.875	0.853	0.864

Table 2: Result of Accuracy, Precision, Recall, F1-Score with PCA

Algorithms	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Decision Tree	0.846	0.850	0.809	0.828
Logistic Regression	0.901	0.900	0.878	0.889
K-Nearest Neighbor	0.912	0.850	0.944	0.895
Naïve Bayes	0.890	0.875	0.875	0.875

Limitation and Future Direction

In this research we are limited to use aged dataset from Kaggle and the dataset size was small too. It contains only 303 features of data. We are separated from each other due to pandemic so it is harder to collect any recent data together for us to apply. It is better to use large data large data for prediction as it is difficult to know how quick the algorithm runs with small data and compare with PCA and each algorithm.

In future we want to collect data from cardiac hospitals from Bangladesh when the pandemic ends and apply the algorithms to see the results for prediction. It will help to analyze the effectiveness of the results in our country's perspective. We also want to demonstrate the algorithms' success in a ROC curve mode. It will help to analyze the results in graphical forms which will help to easily identify results from the curves. We are also keen on developing a hybrid model by using the mixture of some of the algorithms that we used in this research to get better and accurate heart disease prediction. There are also more feature selection optimization techniques that we can apply despite only using PCA in future. We proposed to apply mutual information gain feature selection, chi-square method, Correlation heat map etc. in future to find more precise results of the algorithms that we applied. Also, we can apply more machine learning algorithms like SVM, Random Forest and gradient boosting algorithms for more results for comparison in future works.

Conclusion

In this study there were different classification algorithms which were used to predict heart disease and analyzed performance of these algorithms. PCA was also optimized as an optimization technique to get better result for this strategy. By analyzing all the classification algorithms, it can be interpreted that Logistic Regression performed very well compared to other classification techniques. But with PCA all results of accuracy like Decision Tree, KNN and Naïve bayes seen better results compared to originally.

Reference

1. Khourdifi, Y., Bahaj, M. (2019). Heart Disease Prediction and Classification Using Machine Learning Algorithms Optimized by Particle Swarm Optimization and Ant Colony Optimization. International Journal of Intelligent Engineering and Systems, 12(1), 242–252. https://doi.org/10.22266/ijies2019.0228.24 Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.
2. Patel, J., Tejalupadhyay, S., Patel, S. B. (2016). Heart Disease prediction using Machine learning and Data Mining Technique. Journal - IJCSCVolume: 7.
3. Rajesh, N., T, M., Hafeez, S., Krishna, H. (2018). Prediction of Heart Disease Using Machine Learning Algorithms. International Journal of Engineering Technology, 7(2.32), 363.
4. South Eastern Health and Social Care Trust. (2013). Cardiovascular Disease. Communications Department
5. Lloyd-Jones, D. M., Larson, M. G., Beiser, A., Levy, D. (1999). Lifetime risk of developing coronary heart disease. The Lancet, 353(9147), 89–92.
6. Ministry of Health Kenya. (2015). Kenya National Guideline for Cardiovascular Diseases Management. DIVISION OF NONCOMMUNICABLE DISEASES.
7. Bramer, M. (2020). Principles of Data Mining (Undergraduate Topics in Computer Science) (4th ed. 2020 ed.). Springer.
8. Han, J., Kamber, M., Pei, J. (2011). Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems) (3rd ed.). Morgan Kaufmann.
9. Heart Disease UCI. (2018, June 25). Kaggle. https://www.kaggle.com/ronitf/heart-disease-uci
10. Bronshtein, A. (2020, March 24). Train/Test Split and Cross Validation in Python - Towards Data Science. Medium. https://towardsdatascience.com/train-test-split-and-cross-validationin-python-80b61beca4b6
11. Acuna, E. (2011). Preprocessing in Data Mining. International Encyclo- ˜ pedia of Statistical Sci-ence, 1083–1085.
12. Kumar, N., Kumar, N., Profile, V. M. C. (2019). Advantages and Disadvantages of Princi-pal Component Analysis in Machine Learning. The Professionals Point.
13. Navlani, A. (2018, December 28). Decision Tree Classification in Python. Datacamp
14. Sanjay.M (2018, November 2). Machine Learning — KNN using scikit-learn - Towards Data Science. Medium. https://towardsdatascience.com/knn-using-scikit-learn-c6bed765be75