Cardiovascular Disease Prediction using Classification Models Report

INTRODUCTION: -

The World Health Organization estimates that cardiovascular disease causes millions of deaths worldwide each year. One of the leading causes of morbidity and mortality among the global population is this disease. One of the most crucial topics in the data analysis area is Cardiovascular prediction. Numerous studies have been carried out to identify the most important risk factors for cardiovascular disease and to precisely estimate the overall risk. This disease is also referred to as a silent killer because it causes a person to pass away without any evident signs. In high-risk individuals, an early diagnosis of disease is crucial for helping them decide whether to change their lifestyle, which lowers consequences.

There is not many research employing machine learning (ML) to predict Cardiovascular at the professional level, even though ML models have been found to perform better than clinical risk estimates when compared to statistical methods. This study employed machine learning techniques on a dataset gathered during a health assessment survey of patients from various age groups. We are predicting that weather a person is having cardiovascular disease or not using different features like age, exercise routine, smoking etc.

PROBLEM STATEMENT: -

In this project, we are predicting that patient has blood pressure disease or not using several factors such as age, gender, height etc. This project will help medical and health professionals to evaluate the patient medical condition more precisely and accurately.

DATASET: -

Dataset we are using in this case study is from Kaggle named as “cardiovascular disease prediction” and contains medical information of patients. Dataset contains 70000 patients records and 13 features. We are using “cardio” as our target variable. It is a Binary class classification problem, where value of class is 0 and 1 (0 means disease not detected and 1 means disease

detected). Dataset is balanced and consist of 14 attributes other than target attribute. Dataset is cleaned and does not contain any null values or string column, so we don't need much data prepossessing.

SPLITTING DATA: -

Now we are splitting our dataset into train test datasets using sklearn train test function. We are splitting dataset with the ration of 70-30%. So, we will train our model on 70% data and test it on 30% test data.

Model Evaluation: -

We are evaluating performance of our model's using accuracy, F1 Score, AUC graph, ROC_AUC score, sensitivity, specificity, miss rate, precision, and recall values. Before we move forward to model implementation, it is necessary to know following terms.

1) Precision: -

The proportion of True Positives to All Positives is known as precision. For our problem statement, that would be the measure of cases that we correctly identify having a disease out of all the cases having it.

2) Recall: -

The recall is the number of our model rightly identifying True Positives values. Thus, for all the cases who have disease, recall tells us how many we correctly identified as having a disease.

3) Sensitivity: -

The sensitivity of a machine learning model indicates how effectively it can recognize successful examples. Other names for it include the true positive rate (TPR) or recall.

Sensitivity is used to evaluate model performance since it allows us to ascertain how many examples the model was able to correctly identify.

4) Specificity: -

Specificity is the number of true negatives that are rightly predicted by the model.

5) Miss Rate: -

The miss rate is commonly known as false positive rate – is the likelihood that the test will fail to detect a true positive. FN/FN+TP, where FN is the number of false negatives and TP is the number of true positives (FN+TP being the total number of positives), is the formula used to compute it.

6) False Positive Rate: -

The ratio of false positives to true negatives, or false positives to true negatives plus true negatives (FP+TN), is used to compute the false positive rate. It's the likelihood that an error will occur, meaning that a positive result will be reported when a negative value exists.

7) Classification Report: -

The accuracy of predictions made by a classification algorithm is evaluated using a classification report. How many of the forecasts came true, and how many were wrong? More specifically, the metrics of a categorization report are predicted using True Positives, False Positives, True Negatives, and False Negatives.

```
from sklearn.metrics import classification_report
print(classification_report(testy,y_pred))
```

[13] ✓ 0.1s

```
              precision    recall  f1-score   support

           0       0.91      0.94      0.92        63
           1       0.93      0.90      0.92        60

    accuracy                           0.92       123
   macro avg       0.92      0.92      0.92       123
weighted avg       0.92      0.92      0.92       123
```

Models Implementation: -

For prediction of cardiovascular disease, we are using 3 different models which are Gradient boosting, Naïve bayes, and K nearest neigbor classifier models.

1- K- Nearest Neighbor Classifier: -

The supervised machine learning approach known as the k-nearest neighbors (KNN) model is straightforward and simple to apply. It can be used to tackle classification and regression issues. KNN searches for nearby nodes and returns results based on them. It determines its closest neighbors by calculating the distance to various neighbors. According to similarity in a particular group of nearby data points, KNN classifies data points.

We are also using KNN-model in this project; we implement model with different number of neighbors and its performance is 68%.

Features	Results
Model	KNN
Accuracy	68.25%
AUC Score	0.682
Specificity	0.699
Sensitivity	0.665
Recall	0.665
Precision	0.688
Miss Rate (FNR)	0.300
Miss Rate (FPR)	0.334

2- Naïve Bayes Classifier: -

A group of supervised learning algorithms known as "naive" Bayes methods utilize Bayes' theorem with the "naive" assumption that every pair of features is conditionally independent given the value of the class variable. On our dataset, Naïve bayes is performing average as compared to the other models.

Features	Results
Model	Naïve Bayes
Accuracy	58.50%
AUC Score	0.685
Specificity	0.699
Sensitivity	0.665
Recall	0.665
Precision	0.688
Miss Rate (FNR)	0.334
Miss Rate (FPR)	0.300

3- Gradient Boosting Classifier: -

The supervised machine learning algorithm gradient boosting is used to solve classification and regression problems. By transforming weak learners into strong learners, we can enhance the model prediction of any given algorithm using this ensemble technique. The weak student is successively corrected by predecessors, becoming a strong learner as a result. When compared to other algorithms, gradient boosting classifiers are frequently quick and require less storage.

In this project, we also use Gradient boosting classifier and it's giving us optimum results as compared to other algorithms. Gradient boosting converts and enhance weak learner by reducing its error, but if the data is noisy than accuracy of gradient may affect sometime. We implement gradient boosting and it's fitting quite well.

Features	Results
Model	Gradient Boosting
Accuracy	73.36%
AUC Score	0.73
Specificity	0.787
Sensitivity	0.683
Recall	0.683
Precision	0.759
Miss Rate (FNR)	0.316
Miss Rate (FPR)	0.216

RESULTS: -

From results, we concluded that Gradient boosting performing better than other two algorithm, both in term of accuracy and ROC Score. We can further use some enhancement techniques to increase accuracy of our model like by implementing parameters after hyperparameter tuning or doing cross validation.