Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

On Design and Deployment of ML Model for Cardiac Disease Prediction

[1] Shaik Qadeer, [2]Mohammed Yousuf Khan

[1] Research Scholar, Bir Tikendrajit University
Manipur, India
[2]Research Superviser, Bir Tikendrajit University
manipur, India

E-Mail: [1] haqbei@gmail.com [2] yousuf_lect@yahoo.com

Abstract— Cardiovascular disease is among the disorders that pose the greatest hazard to life. Its high death rate is responsible for around 17 million fatalities globally. Early diagnosis helps to treat the illness at the appropriate time to prevent death. Numerous machine learning and deep learning methods can be used to analyze the presence or absence of the disease. This review paper discusses the entire process of creating an intelligent model. In this study, logistic regression techniques are utilized to classify heart illness using the UCI dataset. Pre-processing the data by decluttering the dataset, finding missing values, and selecting features by correlating each feature with the target value were done to improve the model's performance. The features that showed a strong positive correlation were selected. Next, classification is carried out by dividing the dataset in an 80:20 split-up ratio between training and testing data. By utilizing the Sklearn framework and Python scripting, the proposed model achieved an accuracy of 83.6%. Later the model is prepared with no code AI platform (Akkio) with production quality mode and obtain an efficiency of 86.9%. The effectiveness of the proposed model was thoroughly gratifying and was competent to predict evidence of having heart disease in a certain individual which exhibited quality precision 77%(87% in Akkio) and recall 89%(91% in Akkio). This prediction system for cardiac disease increases access to care while lowering costs. This research report provides us with important information that can assist us in determining whether or not the patient has a cardiac problem. The assessment of the suggested work was completed in Google Colab and afterwards in Akkio. The joblib library in Colab and an Akkio Google sheet were used for deployment.
Keywords— Cardiovascular Disease**;** Machine Learning (ML); **Google Colab; Logistic** Regression (**LR); Akkio.**

1. INTRODUCTION

The prevalence of cardiovascular problems has been steadily increasing worldwide in light of the modern culture's fascination with fast food. This paper is an extension of original conference paper presented in International Conference on Innovations in Engineering and Technology (ICIET), 2023 [1]. Despite being the primary causes of death, these illnesses are thought to be the most preventable and manageable conditions when caught early and treated appropriately [2].

Heart attacks are primarily caused by atherosclerosis. It implies that heart stroke results from the heart's inability to sufficiently pump blood throughout the body. High blood pressure is one of the main risk factors for heart disease development. A survey carried out between 2011 and 2014 found that about 35% of individuals worldwide suffer from hypertension, which is a major cause of heart disease. Similar to this, a number of additional factors, such as obesity, fat accumulation, poor diet, high cholesterol, arterial wall blockage, and inactivity, can cause heart disease. Thus, prevention is essential. It is crucial to comprehend heart diseases in order to prevent them. The fact that 47% of deaths occur outside of hospitals illustrates how frequently warning signs are disregarded.

Data from the World Health Organization show that, with 17.9 million fatalities annually, cardiovascular disorders are currently the world's largest cause of death. Heart disease encompasses a wide range of heart-related conditions. Heart diseases include conditions affecting blood vessels, such as coronary artery disease, arrhythmias, and congenital heart defects, which are abnormalities of the heart that exist from birth. Heart disease and cardiovascular disease are sometimes used synonymously and may be regarded as such. Cardiovascular disease pertains to conditions marked by constricted or clogged blood vessels, potentially resulting in a heart attack, chest pain (congestive heart failure), or stroke.

Heart conditions of today shorten a person's life. Therefore, in 2013, the World Health Organization (WHO) set goals for the prevention of non-communicable diseases (NCDs), with cardiovascular diseases contributing to 25% of the relative decline and with the expectation that by 2025, at least 50% of patients with cardiovascular diseases will have access to the right drugs and medical guidance [3]. 31% of all deaths in 2016—or 17.9 million deaths worldwide—were directly linked to cardiovascular disease. One major challenge is diagnosing cardiac issues [4].

It may be challenging to determine whether or not someone has a cardiac issue. Although there are technologies that can predict the risk of heart disease, they are either excessively costly or not very good in predicting the risk of heart disease in people. (4). According to a World Health Organization (WHO) assessment, medical practitioners can only forecast 67% of heart problems, which has prompted a great deal of research in this field [6].

In rural India, there is a significant shortage of hospitals and high-quality medical care providers. Only 58% of doctors in urban areas and 19% in rural areas have medical degrees, according to a 2016 WHO report. According to statistics, one person in the USA dies from a heart attack every 40 seconds. Up until 2012, Turkmenistan had the highest death rate. (712 out of 100,000 persons). Kazakhstan, on the other hand, has the second-highest death rate from heart disease. India is currently rated 56th in this series [7]. The study found that 0.4 million (28.0%) of the 1.3 million cardiovascular fatalities among individuals aged 30-69 were attributable to stroke, and 0.9 million (68.4%) to coronary heart disease. Machine learning might be a useful tool for predicting any heart disease in humans [8].

Medical science has a significant challenge in the form of heart ailments. Neural networks, decision trees, KNNs, and other collaborative ensemble approaches can be used for heart condition predicting. This article also discusses some noteworthy aspects of using SVM to assess the precision of heart disease prediction. Additionally, it demonstrates how ML will support future efforts to combat cardiac disease.

The associations between the predictor components and the result variable are tested and analyzed using SVM [7]. SVM is more effective at making predictions than discriminant analysis since it takes fewer data[8]. SVM is a classification algorithm that forecasts a discrete result based on a range of variables. Group size, for instance, can be mixed, discrete, continuous, or dichotomous [9] at order to predict the antidepressant usage trend at a tertiary care center, researchers employed a binary SVM technique. They found that outlier patterns might be used to predict people with depression [10]. One of the most widely used methods for minimizing vertical discrepancies and assisting in the creation of the best possible regression line is least squares. To assess an estimator's robustness, robust plane regression seeks to fit a straight line across a set of two-dimensional points without being affected by outliers [11].

Regression analysis can be done using a variety of techniques, including logistic, multiple, and linear, depending on the dependencies and kind of input data. The percentage of measurements that can be manipulated most readily without causing the estimator to yield an incorrect result is known as the breakdown point. Since the mean of all observations has an infinite value if one observation has an infinite value, the mean breakdown point is 0.5. On the other hand, the median value does not change. Only when more than half of the possible outcomes are infinite is the mean irrational. The median absolute deviation is the "single most accurate ancillary scale computation". There are numerous mathematical techniques for replacing missing values. For example, the mean and median are used to replace missing values that are either numeric or character-based. Finding the critical factors that affect the presence or absence of heart illness is the aim of this research. It has also been highlighted that heart disease rarely has any obvious symptoms before to death, making it a silent killer. Both data mining and deep learning sets are used by the previous system [7] [8][15]. An important but challenging activity that necessitates clinical diagnosis effectiveness and precision. The cost of clinical testing should be reduced by employing suitable computer-based data and decision support systems. Data mining is the process of using computer techniques to identify patterns and consistency in data sets. Moreover, computers are more likely to create and classify the numerous attributes or categories directly as a result of the development of data mining over the past 20 years. By knowing the risk factors associated with the disease, medical services providers can identify people who are at high risk for heart disease. Risk factors for heart disease were determined using mathematical analysis using data such as age, blood pressure, total cholesterol, diabetes, hypertension, obesity, inactivity, fasting blood sugar, and so on.

Numerous literary works identify cardiac issues with data mining and machine learning. This is a rudimentary examination of it. In September 2018, M. Marimuthu, M. Abinaya, K. S. Hariesh, K. Mandhan Kumar, and V. Pavithra published a study on cardiac ailment using machine learning and analytics. The findings demonstrate that the researchers' conclusion—drawn from a survey of the literature—was that combinational and more sophisticated models are required to improve the prediction accuracy of heart disease.

Because of its accurate and effective outcomes, these machine learning algorithms have been used in several recent research articles to detect the disease. The decision tree classifier was implemented as part of an efficient decision tree-based cardiac disease prediction approach reported by Sharma Purshottam et al. in 2015. They were able to attain an accuracy rate of 83.3 percent by applying this technique. Likewise, heart disease prediction using basic Bayes and modified K-means, Sairabi H. Mujawar et al. 2015 saw the publication of this essay. Heart disease afflicted 83% of the population with an accuracy rate of detection of 83%, while 85% of cases went unreported [14]. This illustrates how the technique being used affects the accuracy rate.

Detrano et al.'s work, which employed the SVM technique to obtain 77 percent accuracy on a Cleveland dataset, provides another illustration. Saw et al. put the enhanced SVM classification model for the heart disease dataset into practice. For HD prediction, C4.5 trees and rapid decision trees have both been used. Trees and features from the first iteration of the proposed model have been employed. When utilizing SVM to predict cardiac sickness, a hybrid model that combines a fitness function, appropriate genetic algorithms, and a rule encoding technique has been provided to quickly generate the rules. Gudadhe et al. reviewed the process and found that the accuracy was 80.41 percent. Kahramanli and Allahverdi were able to obtain accuracy rates of 84.24 percent for the Pima Indian diabetes dataset and 84.8 percent for the Cleveland HD dataset by merging fuzzy and crisp values in health data. Various machine learning classification models could be used to boost intelligence. For the Pima Indian diabetes dataset and the Cleveland HD dataset, the artificial and fuzzy-based models created by Kahramanli and Allahverdi [13] had accuracy rates of 84.24 percent and 85.8 percent, respectively. The paper "Heart disease prediction utilizing machine learning and data mining approaches" [15] by Dr. Sameer Patel, Prof. Tejpal Upadhyay, and Jaymin Patel from Nirma University in Gujarat is another example.

This review paper gives an insight into the entire of development of such software using a code and a node method of AI development. The purpose of the article is to cover up the approaches of data processing required to model correct deployment.

2. PRPOSED SYSTEM

The suggested model is offered to cover every step of putting a conventional machine learning model creation into practice. To illustrate the complete process of developing a health care model, each phase is explored in detail. Preparing the data, developing the model, and deploying the model (for inference operation) were the procedures that were covered. In order to make an accurate prediction, the right dataset must be chosen, and before using any preprocessing techniques, the features must be appropriately extracted and picked. The goals of preprocessing are to eliminate noise, eliminate redundant data, and make the data consistent. In this case, LR is used on the test platform to predict the stroke of a dataset's user. The LR algorithm's coefficients (Beta values b) must be estimated using the training data. The optimal outcomes are calculated using the maximum-likelihood estimate. The most popular machine learning learning strategy is a maximum-likelihood estimation for logistic regression and a least square method for linear simple regression, despite the fact that it does make assumptions about the distribution of normalized data. For categorization, the incredibly effective Decision Tree method can be applied. The three phases of the proposed system's implementation—feature engineering, model development, and model deployment—are covered in this part and are covered in subsections A, B, and C, respectively. This expansion of the work also adds a no code AI platform, a novel technique for AI development.

A. Feature Engineering

This subsection discusses the resources that were used to gather the data. The machine learning repository at the University of California, Irvine, or UCI. There are 271 samples of heart disease data stored on the Machine Learning Repository website. Table I illustrates the encoding procedure used in feature engineering. where "0" and "1" represent the outcomes of having or not having four heart diseases, respectively, according to the response variable (Y). Age, gender, type of chest pain, resting blood pressure in mmHg, serum cholesterol in mg/dl, fasting blood sugar in the range of 120 mg/dl, resting electrocardiogram, maximum heart rate, exercise-induced angina, old peak (thalassemia), x11, x12, and x13 are the independent variables. Predictions about the independent variable (X's) lead to the dependent variable, Y. The table, which can be utilized in the model development process, covers the encoding details.

TABLE I. DESCRIPTION OF DATA

Variable	**Description**	
	General	***Variable Type***
Y	It's a Label(1-Presence,0-Absence)	Dependent(Qualitative)
X1	Patients age	Ind.(Numerical)
X2	Gender(0-F, 1-M)	Ind.(Categorical)
X3	Chest Pain Type(0-Angina, 2-Non-Angina, 3-Asymtomatic)	Ind.(Categorical)
X4	Blood Pressure(Resting)-in mmHg	Ind.(Categorical)
X5	Serum Chlolestral-in mg/dl	Ind.(Numerical)
X6	Fasting Blood sugar>120mg/dl (0-F,1-True)	Ind.(Categorical)
X7	ECG(Resting)(0-N,1-STtoT wave abnormal,2-Left ventricl hypertrophy)	Ind.(Categorical)
X8	Maximum Heart rate	Ind.(Categorical)
X9	Inducted Angina(exercise)- 0-No,1-YES	Ind.(Categorical)
X10	OldPeak(Due to ST depression induced during exerecise)	Ind.(Numerical)
X11	Old Peak slope 0-Upsloping, 1-Flat, 2-downsloping)	Ind.(Categorical)
X12	Major vessels no.-0-3	Ind.(Categorical)
X13	Thalassemia: 0-N,1-Fixed defect, 2-Reversible defect	Ind.(Numerical)

In Akkio platform, after uploading the data the feature engineering can be done automatically by selecting a proper transform from the drop down menu. First we need to pick a data source to start, for data upload. The options are: Table, Salesforce, Google analytics etc. In our case as we have data available in CSV form, so Table option is selected. Then we need to add outcome of work, which will be available in report. Then for data engineering, three options are available (data cleaning, data merging and chat data preparation). In our case data cleaning is used as pre-processing

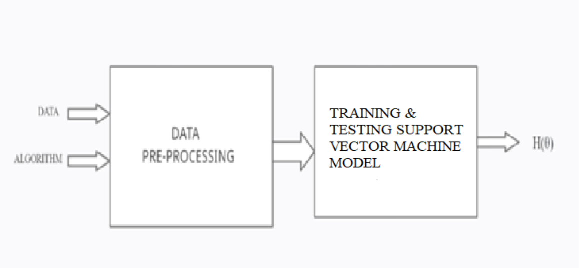

Fig. 1. Block diagram of proposed model

B. Development of ML Model

Logistic regression analysis is used to forecast the issue at hand; in this process, categorical data is transformed into numerical data. Let's examine how the machine learning model is being applied. The block design of this model for importing the libraries is displayed in Figure 1, while the block diagram of the system architect for the suggested heart failure prediction system is displayed in Figure 2. Since raw data cannot be used by our machine learning system, we must process the data set after we acquire it. We accomplish this by preparing the data for learning. Data selection is the process of selecting the information to be used in attack identification. The dataset contains information about an individual's age, gender, heart disease, hypertension, type of work, place of residence, weight, smoking status, and history of stroke. Run the following cell in a Pandas data-frame to import patient data from a CSV file:

Line Number	Code
L1	import pandas as pd1
L2	!wget https://Website.csv
L3	Heart_DataS = pd1.read_csv('/heart .csv')

Our data file is located using line number 2 (L2), and data is read from the located file using line number 3 (L3). The information is contained in the columns listed in Table 1. In this case, Y is the label and x is the input (feature vector). The following can be used to carry out the pre-processing of the data:

Line Number	Code
L1	Heart_DataS .head() Heart_DataS .tail()
L2	Heart_DataS .shape Heart_DataS .info()
L3	Heart_DataS .isnull().sum()
L4	Heart_DataS ['target'].value_counts()

The first five and last five rows of data are obtained using line number one (L1). The dimension of the data can be obtained by using code at L2. L3 is utilized to identify missing data, and the following outcome was obtained:

"age 0 sex 0 cp 0 trestbps 0 chol 0 fbs 0 restecg 0 thalach 0 exang 0 oldpeak 0 slope 0 ca 0 thal 0 target 0".

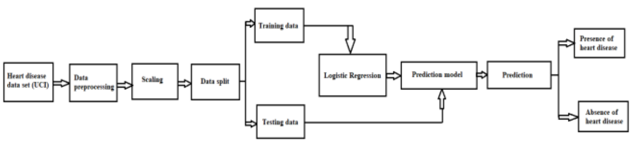

Fig. 2. Block diagram of architecture of proposed model

Since no data is missing, we can move forward. We obtained the following result: "1-> 165 0-> 138," which may be used to verify the output label L4 distribution. In other words, 165 data have the label "defective heart," while 138 have the label "healthy heart."

We can train a classifier to find a statistical association between the features and the label value by using the known label values in our dataset. But how can we determine whether our model is working? When we apply it to new data that it wasn't trained on, how can we be sure that it will make the right predictions? We may leverage the fact that we have a large dataset with known label values to compare the predicted labels with the

---

known labels in the test set. Only a portion of this dataset may be used to train the model; the remainder can be kept aside for model testing. The train test split function, which ensures a statistically random distribution of training and test data, is one of several functions in the scikit-learn package of the Python programming language that may be used to build a machine learning model. We will use it to split the data into two categories: training (80%) and testing (20%). This is accomplished using the code below:

LN	Code
L1	X1 = Heart_DataS .drop(columns ='target', axis=1) Y1 = Heart_DataS ['target']
L2	from sklearn.model_selection import train_test_ split
L3	X_train1, X_test1, Y_train1, Y_test1 = train_tes t_split(X1,Y1, test_size=0.2, random_state=0)
L4	print(X.shape, X_train.shape, X_test.shape)

To distinguish between the dependent and independent variables, utilize line L1. The required library is imported via the code in L2 in order to carry out the data splitting process. Splitting is done by codes at L3, and L4 prints the dimensions of the whole data, train, and test data as (303, 13), (212, 13), and (91, 13). The Binary Classification Model under consideration may now be trained and assessed. Alright, so we can start training our model now that we have the training features (X train) and training labels (Y train) set up. There are several algorithms that we may apply to train the model. Despite its name, logistic regression is a well-known classification technique, therefore we'll utilize it in this case. A regularization parameter, also known as a hyper parameter, has to be set in addition to the labels and training features. This is used to offset sample bias and support the model's capacity to generalize by preventing the model from being overfit to the training set. This is accomplished using the code below:

LN	Code
L1	from sklearn.linear_model import LogisticRegressi on
L2	reg = 0.01
L3	model = LogisticRegression(C=1/reg, solver="libli near").fit(X_train, Y_train)
L4	print (model)

To accomplish the data splitting procedure, the required library is imported using the code in L1. L2 set the hyperparameter to zero. Logistic regression is used in code at L3 to train the model, and L4 prints the model type as "LogisticRegression (C=100.0, solver='liblinear')".

We can now utilize the test data we withheld to evaluate the model's prediction accuracy after utilizing the training set to train it. Once more, Scikit-Learn can help us do this. First, let's use the model to predict the labels for our test set. Next, we'll compare the predicted labels with the known labels. This is accomplished using the code below:

LN	Code
L1	predictions = model.predict(X_test1)
L2	print('Predicted labels: ', predictions)
L3	print('Actual labels: ' ,Y_test1)

Using test data, the L1 algorithm performs a prediction operation. L2 generates forecast labels. The actual label printing is done by code at L3. The arrays of labels are too big to fit on the notebook output, so we can only compare a limited set of values. Even if we printed out every possible combination of projected and

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

actual label, it would still be difficult to evaluate the model in this manner due to the sheer number of labels. Fortunately, scikit-learn has certain metrics and other tricks under its sleeve that we may use to assess the model's performance. The most apparent thing you may want to do is check the accuracy of the forecasts. Stated differently, what is the number of labels that the model accurately predicted? This is accomplished using the code below:

LN	Code
L1	from sklearn.metrics import accuracy_score
L2	print('Accuracy: ', accuracy_score(Y_test1, predictions))

To determine accuracy, a library from Sklearn is imported using the code in L1. The prediction accuracy is printed by L2. The observed value is 83.6%. While accuracy seems like a sensible statistic to use for assessment (and it is, in some ways), you should be cautious about drawing too strong conclusions about a classifier's performance based just on its accuracy. Just keep in mind that it's only an evaluation of the percentage of situations that were accurately predicted. Fortunately, additional measures exist that provide light on our model's effectiveness. With Scikit-Learn, you can produce a classification report that offers more information than simply accuracy. There are further assessment techniques. This is accomplished using the code below:

LN	Code
L1	from sklearn.metrics import precision_score1, recall_score1
L2	print("Overall Precision:",precision_score1(Y_test, predictions)) print("Overall Recall:",recall_score1(Y_test, predictions))

To determine accuracy and recall, a library from Sklearn is imported using the code in L1. The accuracy and recall of the forecasts are printed by L2. Overall Precision is 77.8%, and Overall Recall is 89.3%, according to the observations.

The ROC curve and confusion matrix are the other assessments.

Confusion Matrix: The accuracy and recall ratings are built from four possible prediction outcomes.

True Positives: Both the anticipated and actual labels are 1.

False Positives: When a label is expected to be 1, but it is really zero.

False Negatives: When a projected label of 0 is given while the actual label is 1.

True Negatives: Both the anticipated and actual labels are 0.

Observe that the diagonal line from top left to bottom right represents the accurate (true) predictions; if the model is performing well, these figures ought to be far higher than the incorrect projections. This is accomplished using the code below:

LN	Code
L1	from sklearn.metrics import confusion_matrix1
L2	Cm1 = confusion_matrix1(Y_test1, predictions) print (cm1)

The Sklearn library is imported using the code in L1 in order to determine the confusion matrix. L2 computes and outputs the predictions' confusion matrix. It has been noted to be:

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

TN(32)	FP(12)
FN(5)	TP(42)

Whether a forecast obtains a 1 or a 0 depends on the threshold to which the anticipated probability are compared. If we modified the threshold, it would affect the predictions, which would change the metrics in the confusion matrix. Examining the true positive rate, often referred to as recall, and the false positive rate for a range of possible thresholds is a popular method for classifier assessment.

Aggregation from composite sources, data sampling, lowering the linear or non-linear dimensions, feature generation and selection, discretization and binarization followed by attribute modification are a few of the preprocessing techniques that are used. The majority of the most recent implementations of discretization use an entropy-based approach.

ROC curve: Whether to assign a prediction a score of 1 or 0 depends on the threshold at which the projected probability are evaluated. If we modified the threshold, it would affect the predictions, which would change the metrics in the confusion matrix. Analyzing the true positive rate, often referred to as recall, and the false positive rate for a variety of possible thresholds is a standard method for classifier assessment. A graph known as a receiver operator characteristic (ROC) chart is created by plotting these rates against all possible thresholds.

The received operator characteristic (ROC) chart is created by plotting these rates against each possible threshold. This is accomplished using the code below:

LN	Code
L1	from sklearn.metrics import roc_curve1 from sklearn.metrics import confusion_matrix1 import matplotlib import matplotlib.pyplot as plt1 %matplotlib inline
L2	y_scores = model.predict_proba(X_test) fpr, tpr, thresholds = roc_curve1(Y_test, y_scores[:, 1]) fig = plt1.figure(figsize=(6, 6)) plt1.plot([0, 1], [0, 1], 'k--') plt1.plot(fpr, tpr) plt1.xlabel('FPR- →(False Positive Rate)') plt1.ylabel('TPR→(True Positive Rate)') plt1.title(' Curve- ROC ') plt1.show()

To import a library from Sklearn for ROC curve calculations, use the code in L1. The true positive rate (TPR) and false positive rate (FPR) are printed by L2. Figure 3 illustrates the ROC curve to demonstrate how the analysis should be interpreted.

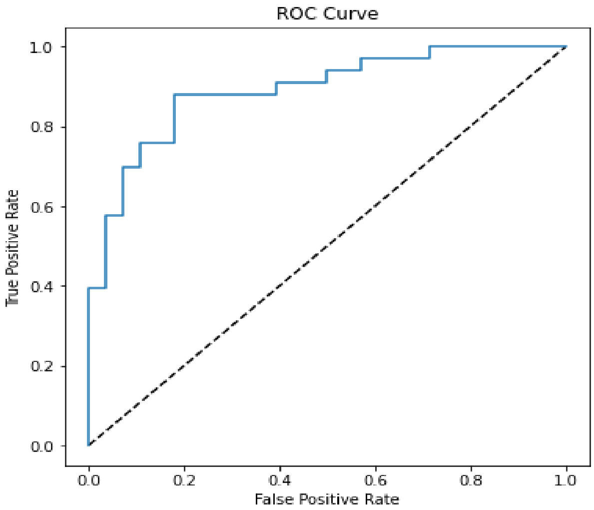

Fig. 3.ROC curve of proposed model

The model's overall performance is measured by the area under the curve (AUC), which has a range of 0 to 1. The model performs better the closer this value is to 1. This is accomplished using the code below:

LN	Code
L1	from sklearn.metrics import roc_auc_score
L2	auc = roc_auc_score(Y_test,y_scores[:,1]) print('AUC: ' + str(auc))

To import a library from Sklearn for AUC calculations, utilize the code in L1. L2 computes and outputs the prediction's AUC. The AUC that was noted is AUC: 0.877.

While in Akkio, the model development is pretty simple, we need a select from the main option of model development as "predict" or "forecast". The predict option is used for numerical or categorical output, whereas forecast is used for generative AI applications. In our case the predict option is used, as our application is of first form. The next step is the selection of independent variable, our independent variable is 'Y'. Finally we need to select the option of model creation (Fast, High quality or production quality) based upon our requirement. Then we need to select create ML model option. The system takes some time to perform both training and testing operation. It is important to note that, the cloud service (Akkio) make use of GPU for compute engine, which is very fast then the CPU.

C. Model Deployment

We may now draw conclusions using the model. Now that our trained model is pretty helpful, we may save it and utilize it in the future to predict labels for fresh data. The deployment procedure is depicted in Figure 3. The code that follows is utilized to deploy our model and input the crucial parameters or data.

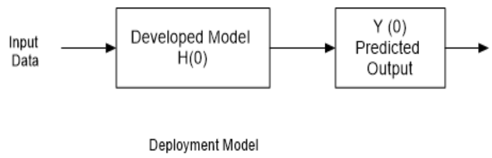

Fig. 4. Model for deployment

LN	Code
L1	import joblib
L2	filename1 = './heart_model.pkl' joblib.dump(model, filename1)

The "joblib" library is imported for deployment using the code in L1. L2 turns the model into a pickle file. The model may be loaded and used to predict values for newly observed data for which the label is not known. This is accomplished using the code below:

LN	Code
L1	model = joblib.load(filename1)
L2	import numpy as np1
L3	X_new1 = np1.array([[63,1,3,145,233,1,0,150,0,2.3,0 ,0,1]]) print ('New FedSample: {}'.format(list(X_new1[0]))) pred1 = model.predict(X_new1) print('Predicted Result1 is {}'.format(pred1[0]))

The developed model is imported using the code included in L1. In order to generate an array and provide a sample for prediction, L2 grabbed a library from numpy. L3 code has the ability to feed samples in order to anticipate outcomes using the intended model. The outcome attained is:

New FedSample: [63.0, 1.0, 3.0, 145.0, 233.0, 1.0, 0.0, 150.0, 0.0, 2.3, 0.0, 0.0, 1.0]

Predicted Result1 is 1.

The model deployment with Akkio method is professional. Here we need to select a place of deployment from “Pick an endpoint to deploy”. The options available are API (Configure and deploy a production ready API), Web app(create a web page to run our model), or the clouds(Salesforce, Zapier, Google sheet etc.). In our case we have deployed on Google sheet, whose address is: https://docs.google.com/spreadsheets/d/1DrF34S0uNwTjMKu_2ZAh3lNKe-7effSkaK3BbumYWEY/edit#gid=0. If we feed the input, the model is doing prediction automatically and giving the result of classification a few is shown:

Fed Sample													Akkio_ Result	Probability target is 1	Probability target is 0
63	1	1	3	145	233	1	0	150	0	2.3	0	0	1	0.87	0.12
56	1	2	130	256	1	0	142	1	0.6	1	1	1	0	0.30	0.69
43	1	2	130	315	0	1	162	0	1.9	2	1	2	1	0.75	0.24
60	1	0	117	230	1	1	160	1	1.4	2	2	3	0	0.07	0.92

D. Results and Discussion

This section discusses the results obtained in the entire process. Let us start with accuracy. It is observed to be 83.6 %(86.9% with Akkio). Although accuracy appears to be a reasonable metric to use for evaluation (and to some extent, it is), you should be careful not to infer too much about a classifier's performance from its accuracy. The next is precision and recall. It is observed to be: Overall Precision= 77.78 %(87% with Akkio) and Overall Recall= 89.6%(91% with Akkio). The other metric can be calculated with confusion matrix where we obtained TN=32, TP=42 and FP and FN are observed to be very less (12 and 5 respectively). The AUC value is observed to be 87.7%.

The bar graph of heart disease v/s sex shows that (shown in Figure 5.), the more men are likely to be effected than women.

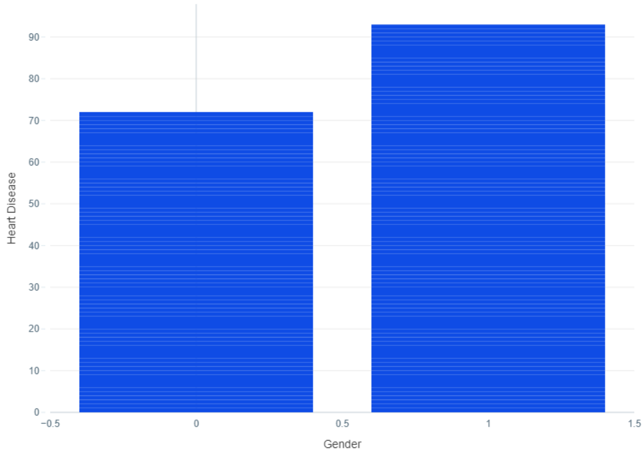

Fig. 5. Bar chart of Heart disease v/s Gender

3. CONCLUSION

One day, the prevalence of heart disease could surpass the current level of occurrence. Heart problems can be deadly and result in many deaths annually. Manually estimating the probability of developing heart disease using the above indicated risk factors is challenging. Prior detection uses medical traits or features to determine whether a certain patient has early-stage cardiac disease. The end-to-end architecture of our model tackles the primary problem confronting the medical community: how to predict cardiovascular illness in low-income or developing countries more efficiently and correctly in order to prevent the disease's compounding effect. Early diagnosis reduces risk and improves quality of life. Twenty test instances were then given to it in order to confirm the forecast it had made. This program will save doctors' time by predicting cardiac issues in non-medical workers. It is still an unexplored area that needs to be used in order to improve cardiac illness prediction. It is possible to improve the present methods by combining the concepts of aggregation and privacy.
This review article provides an overview of the whole software development process for both coding and non-coding platforms. This article covers everything from data processing to the deployment of models.

DATA AVAILABILITY

The dataset for the proposed work is available at: https://www.kaggle.com

REFERENCES

[1] S. Qadeer, A. G. M, M. Y. Khan and F. Taranum, "Design and Deployment of ML Model for Cardiac Disease Prediction-A review," 2023 International Conference on Innovations in Engineering and Technology (ICIET), Muvattupuzha, India, 2023, pp. 1-7, doi: 10.1109/ICIET57285.2023.10220910

[2] Avinash Golande, Pavan Kumar T. Heart disease prediction using effective machine learning techniques. In International Journal of Recent Technology and Engineering (IJRTE), 2019.

[3] The Lancet Global Health. The changing patterns of cardiovascular diseases and their risk factors in the states of India: The global burden of disease study 1990-2016, DOI:https://doi.org/10.1016/S2214-109X(18)30407-8.

[4] Himanshu Sharma, M A Rizvi. Prediction of heart disease using machine learning algorithms: A survey. , https://doi.org/10.17762/ijritcc.v5i8.1175.

[5] A. Gavhane, “Prediction of Heart Disease using Machine Learning”, Proceedings of International Conference on Electronics, Communication and Aerospace Technology, pp.1275-1278, 2018.

[6] P.S. Kumar, “A Computational Intelligence Method for Effective Diagnosis of Heart Disease using Genetic Algorithm”, International Journal of Bio-Science and BioTechnology, Vol. 8, No. 2, pp. 363-372, 2016.

[7] J. Nahar and T. Imam, “Computational Intelligence for Heart Disease Diagnosis: A Medical Knowledge Driven Approach”, Expert Systems with Applications, Vol. 40, No. 1, pp. 96-104, 2013.

[8] J. Nahar and T. Imam, "Association Rule Mining to Detect Factors which Contribute to Heart Disease in Males and Females", Expert Systems with Applications, Vol. 40, No. 4, pp. 1086-1093, 2013.

[9] Tulay Karaylan and Ozkan Kilic, "Prediction of Heart Disease using Neural Network", Proceedings of International Conference on Computer Science and Engineering, pp. 719-723, 2017.

[10] T.P. Thomas, "Human Heart Disease Prediction System using Data Mining Techniques", Proceedings of International Conference on Circuit, Power and Computing Technologies, pp. 12-18, 2016.

[11] D.K. Ravish and N.R. Shenoy, "Heart Function Monitoring, Prediction, and Prevention of Heart Attacks: using Artificial Neural Networks", Proceedings of International Conference on Contemporary Computing and Informatics, pp. 1-6, 2014.

[12] Kazaam, E., Ghurbana, B., Obineche, E. et al. Hypertension — still an important cause of heart failure? J Hum Hypertens 19, 267–275 (2005) doi:10.1038/sj.jhh.1001820

[13] M. Marimuthu, M. abinaya, K S Hariesh, K Madhankumar, V Pavithra. A review on heart disease prediction using machine learning and data analytics approach.

[14] Jaymin Patel, Prof. Tejpal Upadhyay, and Dr. Samir Patel. Heart disease prediction using machine learning and data mining techniques.

[15] Mohammad Zakwan, Shaik A. Qadeer, Mohammed Yousuf Khan Application of deep learning algorithm in hydrometry, Under print in International Journal of Hydrology Science and Technology, 2023.