Article Title: An Empirical Evaluation of Regression, AdaBoost and TensorFlow Models for Heart Disease Prediction

Ms. Hiralkumari Babulal Chauhan

Abstract:

The Heart disease is a leading cause of morbidity and death worldwide, so, requiring accurate and reliable prediction models. The study draws upon a comprehensive dataset containing clinical and demographic data collected from a various population on earth. We assessed performance in terms of prediction accuracy, sensitivity, specificity and Area under the receiver operating characteristic curve through a rigorous evaluation. It explores the impact of various factors like feature selection, data pre-processing and model optimization, etc., on the performance of each technique. The results of comparative analysis provide valuable insights of the strengths Deep learning using TansorFlow platform for heart disease prediction. The research also, guides researchers and practitioners in selecting the most applicable technique. It have citations selected from recent and reputable sources to support the analysis and findings. Furthermore, the study highlights the importance of continuously improving prediction models to enhance heart disease diagnosis and intervention strategies. The study divided into sections, namely: literature, Methodology of algorithms, database exploration and processing, results comparison, conclusion and limitation with future work.

Keywords - Regression, Adaboost, Tensorflow, Precision, Recall, F1-Score, Sensitivity, Specificity, Accuracy

1. Introduction:

Now a days, Heart disease is become one of the leading causes of passing away worldwide, therefore prediction and early detection are essential steps for effective prevention and treatment. With progression of machine learning techniques, the predictive models have become increasingly important in the health field. This study conducts a comparative analysis of three popular machine learning algorithms named, regression, Adaboost and Tensorflow to determine effectiveness in the area heart disease prediction. Regression is a traditional statistical analysis technique that pursues to establish a connection between a dependent and one or more independent variables. Adaboost is an abbreviation for Adaptive Boosting, which ensembles a learning technique that combines weak learners to construct a robust and precise predictive model. All the smart devices used in current era are generating a lot amount of data, which is difficult to manage and store for further operation without cluster [1]. By leveraging Tensorflow's proficiencies, this study has develop sophisticated deep learning models capable of analysing complex patterns and extracting high-level features from large-scale heart disease datasets.

The objective of this study is to evaluate the predictive performance of three different techniques, Regression, Adaboost and Tensorflow in the context of heart disease prediction. It will utilize a comprehensive dataset consisting of various patient characteristics and medical attributes to train and test this models. This study shall not only contribute in the growing body of knowledge in cardiovascular research but also offer valuable insights for healthcare professionals in developing personalized treatment plans with preventive strategies. Addition to this, this analysis will shed light on the strengths and limitations of each algorithm included, guiding future research and describes advancements in the field of machine learning for heart disease prediction.

2. Methodology:

The enlisted 3 algorithms basic functionality is described here in the following section.

2.1 Regression Algorithm

Regression algorithm are falls under supervised learning category they are used to predict outcome variable based on predictor variables. There are many types of regression algorithm,

1. Linear regression has two types, single regression simple one where one dependent variable and one independent variable, while multiple regression means more than one independent variables are used to find one dependent variable

2. Polynomial regression, there is relationship of independent and dependent variable as an Nth degree polynomial. It is also known as extended linear regression

3. Ridge Regression. It is known as L2 regularization, also. It introduces penalty term for large coefficients to prevent overfitting problem

4. lasso regression, it is also known as L1 regularization it uses absolute value Coefficient as penalty. It often results in sparser model

5. Elastic Net regression, where it combines Ridge and lasso regularization to provide balance this two regression.

6. Decision tree regression, it uses model to create relationship between Independent and dependent variables

7. Random forest regression, an assembled method to build decision tree and average the prediction for better accuracy and resolve the overfitting problem

8. Support vector regression, known as extended vector machine for regression problem it helps to find hyperplane that best fit the data

9. K nearest neighbour (KNN), this predicts dependent variable by averaging values of K nearest neighbours

10. Gradient boost regression, builds a series of weak learners sequentially with the tree correcting errors of previous ones

11. XGboost regression, it optimise scalable version of gradient boosting

12. LightGBM regression, this is gradient boosting framework which uses tree based learning algorithm, it is designed for distributed and effective training

13. Catboost regression, a gradient boosting library to categorise features support and training efficiently

14. Neural network regression, utilise artificial neural network for Complex relationship of variables

LogR (Logistic regression) stands as highly utilised regression alcohol Rhythm for task involving binary classification. This algorithm estimates3probability of occurring input features using logistic Function. LogR has been applied in heart disease prediction with of smart devices demonstrates effectiveness in identifying patients risk of heart disease [2].

One more Regression method that is used in this paper is Logistic regression, a statistical way used for binary classification or multi class problems, here outcome variable is categorical (like, True -False, Yes - No) having two possible classes (0 or 1).

Logistic regression uses Sigmoid function, a logistic function given below to model probability of a class

$$F(Z) = \frac{1}{1 + e^{-z}}$$

Here, Z is a linear combination of the input features. The Linear Combination Z is calculated as follows,

$$Z = b_o + b_1X_1 + b_2X_2.......b_nX_n$$

Here, bo = bias value.

b_1 , b_2.......b_n = Coefficients

X_1 , X_2.......X_n = Features

The value F(Z) is applied to input feature z to find probability Class-1.

$$P(Y - 1) = \frac{1}{1 + e^{-(bo + \, b1X1 \, + \, b2X2 ......+ bnXn)}}$$

Probability of Class-0 is complementary to Class-1, found by equation,

$$P(Y - 0) = 1\, P(Y - 1)$$

A decision boundary can chosen based on threshold probability generated. The model is trained by adjustment of coefficients (b1 , b2.......bn) to minimize difference among the predicted and actual class labels. A cost function- cross-entropy loss is used in logistic regression, that measures difference between predicted probabilities and true labels.

Logistic regression which one is used in this study, models the probability of an instance for a particular class using a sigmoid function of logistic regression. IT is trained by adjusting coefficients to minimize the difference between predicted probabilities and actual labelled values. This regression is widely used for binary or multi classification tasks in several fields, like medicine, finance, etc. The selection process of best suitable regression algorithm is usually depends on characteristics of data set interpreter ability required for application.

2.2 **Theory Of AdaBoost**:

AdaBoost (Adaptive Boosting) is an ensemble learning algorithm that integrates numerous classifier to form a robust and more powerful classifier. It iteratively trains sequence of weak classifier on weighted version of training datasets. In each iteration algorithm adjust the weight of Misclassified samples to prioritize correct classification in subsequent iteration.

The working of AdaBoost algorithm is explained as follows,

1. Weights initialisation: It assigns equal weights to all training samples in initial iteration.

2. Classify train week classifier: A weak classifier exhibits performance marginally superior to random guessing.

3. Evaluate weak classifiers: It calculate the error rate of each weak classifier on the weighted training data. The error rate is determined by comparing the predicted and actual labels of the training samples.

4. Updating weights: it increases weights of misclassified samples and decreases weights of correctly classified samples. This step gives higher significance to misclassified samples in the consequent iterations.

5. Combination of weak classifiers: It assign weights to the weak classifiers based on lower error rates. The weights reflect in impact of each weak classifier into the final prediction [3].

6. Repetition of steps 2-5: By Repeating process, for a predefined number of iterations, it reaches at a specified performance threshold value.

7. Predictions making: the predictions of the weak classifiers are aggregate to derive the final prediction. In final prediction the contribution of weak classifiers is influenced by their weights, where classifiers with higher weights exert a more significant impact prediction process [4].

AdaBoost leverages collective wisdom of multiple weak classifiers to create a strong classifier. The weights of the training samples are adaptively adjusting to focus on the most challenging samples and its improvement. It selects a weak classifier that performs better than random guessing and adjusts weights of misclassified samples, it accentuates importance of these samples in succeeding iterations. The iterative process continues until a specified number of weak classifiers is amalgamated in a form a strong classifier.

AdaBoost has improved accuracy and robustness of heart disease prediction models and widely employed in the classification of heart disease based on database collected from smart devices, such as electrocardiograms (ECG). Finally, AdaBoost has proven as a powerful technique for disease prediction using smart devices. It has ability to handle complex relationships variables, capacity to improve prediction accuracy and these make it a valuable tool in healthcare domain.

2.3 **Theory Of TensorFlow**:

A TensorFlow is open-source framework established for a variety of ML (machine learning) and deep learning applications building and training various types of NN (neural networks).

The main concept of TensorFlow is rooted in several fundamental concepts like, linear algebra and techniques in ML.

The key concepts underlying TensorFlow is illustration of computations as computational graphs. The Computational graphs are directed acyclic graphs where nodes represent mathematical operations while edges represent flow of data between operations. TensorFlow uses a system of tensors to denote and manipulate data, therefore it is named "TensorFlow". The core idea of TensorFlow is to define a computation graph that represents mathematical operations involved in ML model. The graph is then executed by running required computations on available hardware resources, such as GPUs. It allows efficient parallel execution and automatic differentiation, which is crucial for training NNs using techniques, such as backpropagation. TensorFlow is applied for Heart disease prediction [5].

The math's operations in TensorFlow are stated using tensors, which are multidimensional arrays. Tensors may scalars 0-dimensional, 1-dimensional vector, 2-dimensional matrices or higher-dimensional arrays. TensorFlow provides a rich set of operations for transforming and manipulating tensors. It includes element wise operations, reductions, matrix multiplications, etc. ML models in TensorFlow are implementing using Keras API, is a library to provide high level of abstractions. The Keras is easy-to-use interface for users to quick building and training NNs, with marginal code. TensorFlow supports lower-level operations and customizations for more fine-grained control over user's models. TensorFlow is an open-source library that developed and maintained by Google and ML community, there isn't any single publication associated with its theory [6]. Infect, TensorFlow is built upon and inspired by several research papers and conceptions in the field of ML and deep learning.

2.4 Dataset Exploration:

The heart database available on Kaggle, is a curated dataset collected for training and evaluating ML models for heart related predictions. The database contains information of various features related to cardiac health and corresponding labels representing specific heart related conditions. The database undergoes two key stages of a pre-processing phase and application of ML algorithms to identify the most accurate algorithm. The data process is divided in into three important phases are explained in detail here.

This data set is containing of four databases: Cleveland, Hungary, Switzerland, and Long Beach V dates from 1988. In this study, we utilize a dataset comprising individuals who have undergone analyses and tests aimed at identifying the presence of heart disease. The dataset is structured as a matrix, with each row corresponding to individual patients and each column representing the specific factors or attributes (features) under consideration for testing. The dataset has 14 attributes, including predicted attribute are used here out of 76 attributes. The categorical data Sex is converted in binary form, in terms of 1's and 0's to replace male and Female, Respectively. The attributes are listed below in Table-1.

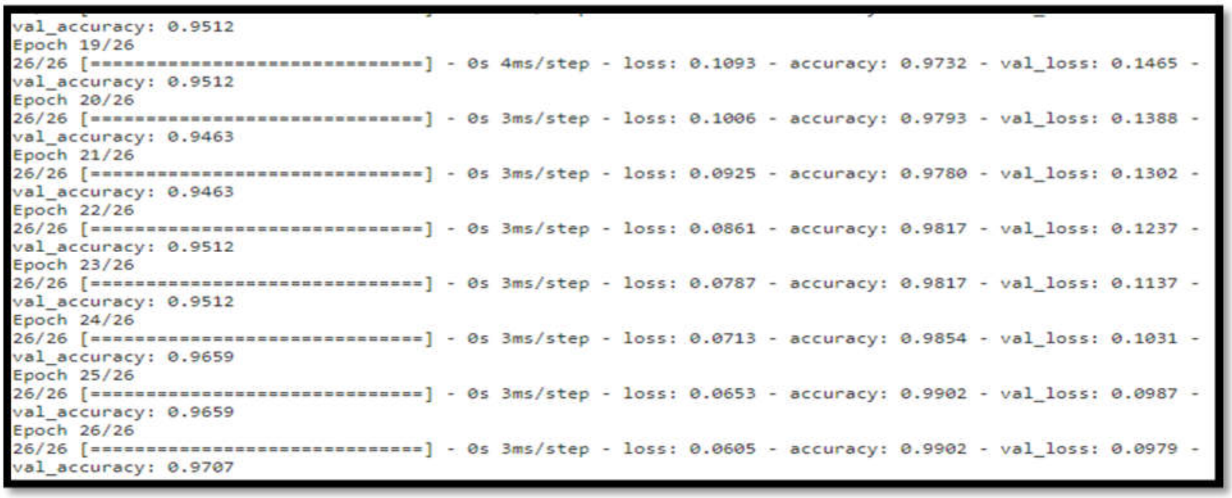
Table 1 [Pre-processed Database]

Age	Sex	CP	Trest bps	Chol	Fbs	Reste cg	Thal ach	Exang	Oldpeak	Slope	CA	Thal	Target
68	1	2	118	277	0	1	151	0	1	2	1	3	1
55	1	0	140	217	0	1	111	1	5.6	0	0	3	0
42	1	0	136	315	0	1	125	1	1.8	1	0	1	0
49	1	2	118	149	0	0	126	0	0.8	2	3	2	0
53	0	0	138	234	0	0	160	0	0	2	0	2	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
51	1	3	125	213	0	0	125	1	1.4	2	1	2	1
51	1	0	140	261	0	0	186	1	0	2	0	2	1
70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
35	0	0	138	183	0	1	182	0	1.4	2	0	2	1
58	1	2	112	230	0	0	165	0	2.5	1	1	3	0

This crucial step involves dataset analysis. The info function is essential, providing comprehensive details about the database. It plays a vital role in understanding and interpreting dataset effectively. These algorithms are applied to train dataset. The “target” column in the dataset is predicted outcome that refers to the presence of heart disease in the patient. If, it is integer valued 0, it indicates no disease and if it is integer valued 1, it indicates that patient is having disease.

3 Results Analysis:

While training of a model epochs refer to one complete pass through whole training dataset during the. The parameters of Training a model are updating iteratively in order to minimize error. By exploring entire dataset multiple times, multiple epochs improve model's performance. Below Figure-1 shows epoch values for TanseFlow.



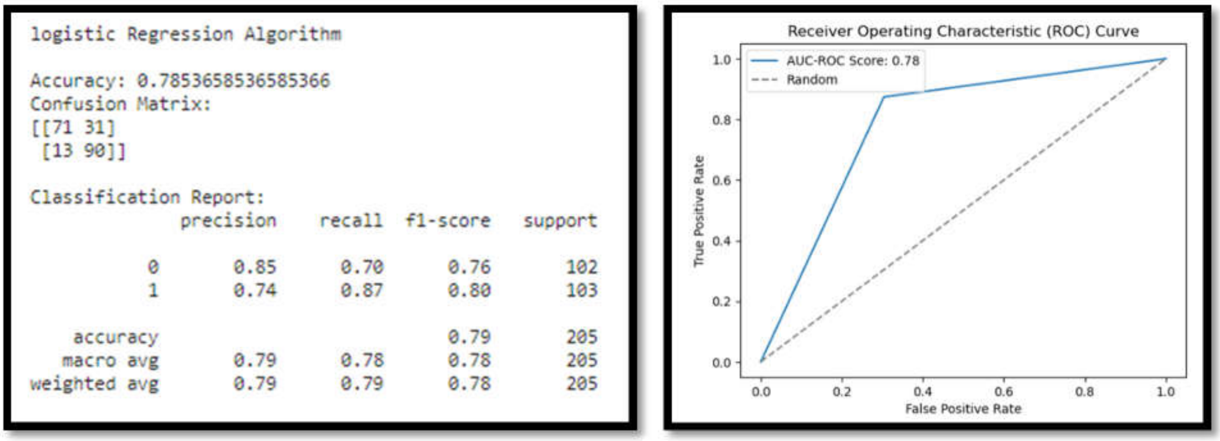
```
val_accuracy: 0.9512
Epoch 19/26
26/26 [==============================] - 0s 4ms/step - loss: 0.1093 - accuracy: 0.9732 - val_loss: 0.1465 -
val_accuracy: 0.9512
Epoch 20/26
26/26 [==============================] - 0s 3ms/step - loss: 0.1006 - accuracy: 0.9793 - val_loss: 0.1388 -
val_accuracy: 0.9463
Epoch 21/26
26/26 [==============================] - 0s 3ms/step - loss: 0.0925 - accuracy: 0.9780 - val_loss: 0.1302 -
val_accuracy: 0.9463
Epoch 22/26
26/26 [==============================] - 0s 3ms/step - loss: 0.0861 - accuracy: 0.9817 - val_loss: 0.1237 -
val_accuracy: 0.9512
Epoch 23/26
26/26 [==============================] - 0s 3ms/step - loss: 0.0787 - accuracy: 0.9817 - val_loss: 0.1137 -
val_accuracy: 0.9512
Epoch 24/26
26/26 [==============================] - 0s 3ms/step - loss: 0.0713 - accuracy: 0.9854 - val_loss: 0.1031 -
val_accuracy: 0.9659
Epoch 25/26
26/26 [==============================] - 0s 3ms/step - loss: 0.0653 - accuracy: 0.9902 - val_loss: 0.0987 -
val_accuracy: 0.9659
Epoch 26/26
26/26 [==============================] - 0s 3ms/step - loss: 0.0605 - accuracy: 0.9902 - val_loss: 0.0979 -
val_accuracy: 0.9707
```

Figure 1 [Results in each Epoch]

Confusion Matrix is generated for three different algorithms using 1000 data records of database. Performance parameters confusion matrix is a tool used to evaluate effectiveness of models. This matrix displays TP (true positives), TN (true negatives), FP (false positives) and FN (false negatives). Given figures – shows Confusion Matrix for Logistic Regression, Adaboost and TansorFlow respectively

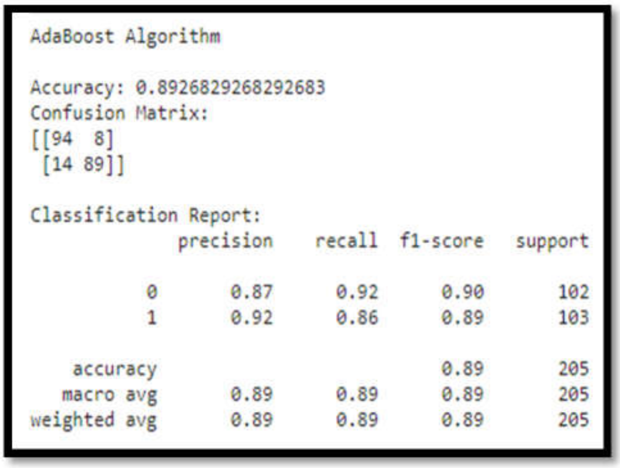
Confusion matrix values can utilised to find precision, recall, and F1 score for aiding in the understanding a model's performance. Here, Precision measures the accuracy of the positive predictions, generated by number of TP divided by the sum of TP & FP. Precision value is used when required to minimize the number of FP, in case where cost of FP is high. The Recall is ability of model to predict all positive instances, calculated by number of TP divided by the sum of TP & FN. It is effectively used in minimizing the number of instances those are actually positive but predicted as negative, so it can reduce cost of FN. The next parameter is F1-score a harmonic mean of precision and recall values, especially used in imbalance between the classes or no clear preference is given. The last value given in classification report is Support, a number of actual occurrences of true instances of the class in specified dataset. Support is not part of calculations but just gives key idea of sharing of classes in dataset. These metrics helps to optimize and refine ML models for more efficient outcomes by clear interpretation.

The classification reports of three algorithms, summarizes outcome of ML model in terms of precision, recall, F1 score, and support for each class with offering insights into predictive capabilities. AUC-ROC (Area Under the Receiver Operating Characteristic) is a performance metric for binary models which measuring ability to separate outs the classes. The higher value of AUC-ROC nearer to 1, indicates that model performances superior. The figures – 2, 3 and 4 shows Classification Reports and AUC-ROC for Logistic Regression, Adaboost and TansorFlow, respectively.



```
logistic Regression Algorithm

Accuracy: 0.7853658536585366
Confusion Matrix:
[[71 31]
 [13 90]]

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.70      0.76       102
           1       0.74      0.87      0.80       103

    accuracy                           0.79       205
   macro avg       0.79      0.78      0.78       205
weighted avg       0.79      0.79      0.78       205
```

Figure 1 [Classification Reports and AUC-ROC for Logistic Regression]



```
AdaBoost Algorithm

Accuracy: 0.8926829268292683
Confusion Matrix:
[[94  8]
 [14 89]]

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.92      0.90       102
           1       0.92      0.86      0.89       103

    accuracy                           0.89       205
   macro avg       0.89      0.89      0.89       205
weighted avg       0.89      0.89      0.89       205
```

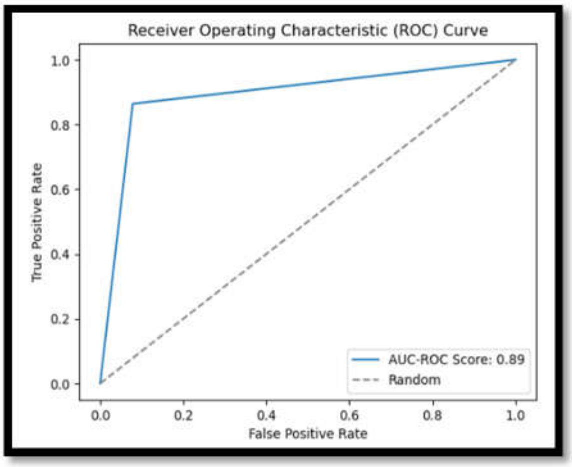
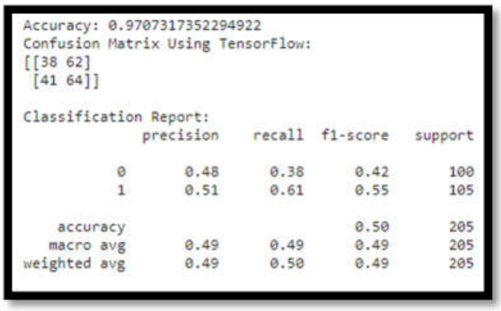
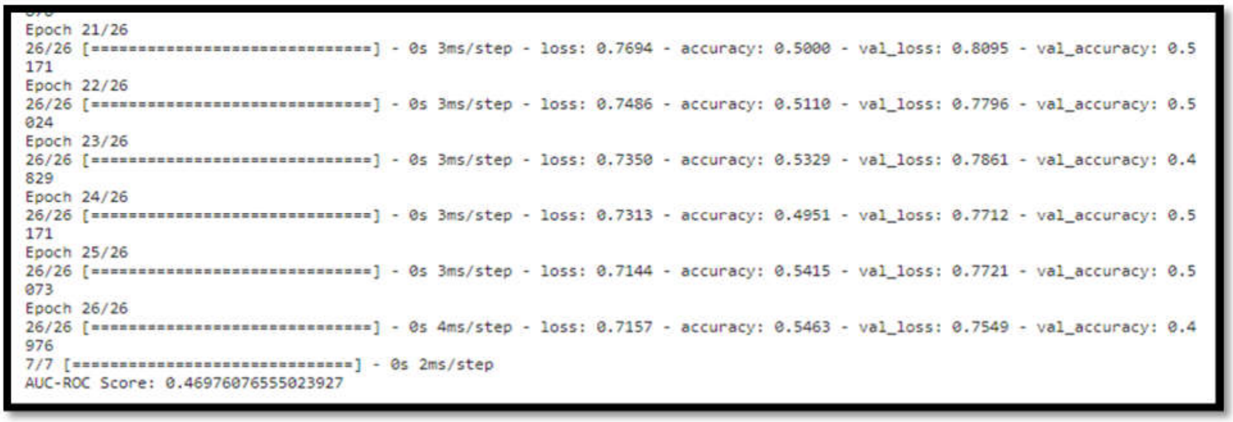


Figure 3 [Classification Reports and AUC-ROC for AdaBoost]



```
Accuracy: 0.9707317352294922
Confusion Matrix Using TensorFlow:
[[38 62]
 [41 64]]

Classification Report:
              precision    recall  f1-score   support

           0       0.48      0.38      0.42       100
           1       0.51      0.61      0.55       105

    accuracy                           0.50       205
   macro avg       0.49      0.49      0.49       205
weighted avg       0.49      0.50      0.49       205
```



```
Epoch 21/26
26/26 [==============================] - 0s 3ms/step - loss: 0.7694 - accuracy: 0.5000 - val_loss: 0.8095 - val_accuracy: 0.5
171
Epoch 22/26
26/26 [==============================] - 0s 3ms/step - loss: 0.7486 - accuracy: 0.5110 - val_loss: 0.7796 - val_accuracy: 0.5
024
Epoch 23/26
26/26 [==============================] - 0s 3ms/step - loss: 0.7350 - accuracy: 0.5329 - val_loss: 0.7861 - val_accuracy: 0.4
829
Epoch 24/26
26/26 [==============================] - 0s 3ms/step - loss: 0.7313 - accuracy: 0.4951 - val_loss: 0.7712 - val_accuracy: 0.5
171
Epoch 25/26
26/26 [==============================] - 0s 3ms/step - loss: 0.7144 - accuracy: 0.5415 - val_loss: 0.7721 - val_accuracy: 0.5
073
Epoch 26/26
26/26 [==============================] - 0s 4ms/step - loss: 0.7157 - accuracy: 0.5463 - val_loss: 0.7549 - val_accuracy: 0.4
976
7/7 [==============================] - 0s 2ms/step
AUC-ROC Score: 0.46976076555023927
```

Figure 2 [Classification Reports and AUC-ROC value for TansorFlow]

The results clearly shows the accuracy for Logistic Regression, Adaboost and TansorFlow is 0.7853, 0.8927 and 0.9707, respectively, while AUC-ROC value for Logistic regression is 0.78, for AdaBoost is 0.89 and for TansorFlow is 0.98 . So, this comparative analysis demonstrates that deep learning model implemented using TensorFlow consistently yield superior accuracy in heart disease prediction compared to linear regression and Adaboost. As shown in Figure

Yet, carefully consideration of factors like dataset characteristics, model's complexity, hyper-parameter tuning and evaluation metrics would have strong influence on accuracy achievement.

The ability of deep learning models to automatically learn intricate patterns from large datas ets   contributes to their superior performance. However, it is important to note that regress ion based methods and Adaboost can also provide reliable predictions, especially when interpretability and simplicity are essential.

4. Conclusion:

ML and deep learning techniques such as Regression, Adaboost, and TensorFlow have proven their effectiveness in heart disease prediction by having more than 75 percent accuracy. Regression model provides a straightforward approach, whereas Logistic Adaboost and TensorFlow offer ensemble and deep learning capabilities, respectively. This comparative study has shown that deep learning model is implemented using TensorFlow has achieved above 98 percent accuracy. This model can give faster and better result for heart patient’s data. It can be useful to doctor to predict heart attack probability with accuracy. However, choice of suitable algorithm depends on specific dataset, available numbers of resources and required level of interpretability. The performance of the algorithms may vary grounded on the specific dataset and feature selection process.

References:

1. Yu, Z., Wang, K., Wan, Z. et al. Popular deep learning algorithms for disease prediction: a review. Cluster Comput 26, 1231–1251 (2023). https://doi.org/10.1007/s10586-022-03707-y
2. Ambrish G, Bharathi Ganesh, Anitha Ganesh, Chetana Srinivas, Dhanraj, Kiran Mensinkal, Logistic regression technique for prediction of cardiovascular disease, Global Transitions Proceedings, Volume 3, Issue 1, 2022, Pages 127-130, ISSN 2666-285X, https://doi.org/10.1016/j.gltp.2022.04.008.
3. Mahesh TR, Dhilip Kumar V, Vinoth Kumar V, Asghar J, Geman O, Arulkumaran G, Arun N. AdaBoost Ensemble Methods Using K-Fold Cross Validation for Survivability with the Early Detection of Heart Disease. Comput Intell Neurosci. 2022 Apr 18; 2022:9005278. doi: 10.1155/2022/9005278. PMID: 35479597; PMCID: PMC9038394.
4. Wang K, Yan LZ, Li WZ, Jiang C, Wang NN, Zheng Q, Dong NG, Shi JW. Comparison of Four Machine Learning Techniques for Prediction of Intensive Care Unit Length of Stay in Heart Transplantation Patients. Front Cardiovasc Med. 2022 Jun 21;9:863642. doi: 10.3389/fcvm.2022.863642. PMID: 35800164; PMCID: PMC9253610.
5. Ajay Sharma, Tarun Pal, Varun Jaiswal, Chapter 12 - Heart disease prediction using convolutional neural network, Editor(s): Ayman S. El-Baz, Jasjit S. Suri, Cardiovascular and Coronary Artery Imaging, Academic Press, 2022, Pages 245-272, ISBN 9780128227060, https://doi.org/10.1016/B978-0-12-822706-0.00012-3.
6. M. Abadi et *al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." Software available from tensorflow.org, 2015.*