Predicting the Risk of Heart Disease Using Key Features: A Comparative Analysis of Machine Learning Models

Lupin Cai, Leo Dai

December 9, 2023

Abstract

Heart disease is still a leading cause of death across various races in the United States, and almost half of the American population contains at least one major risk factor. With modern computational tools and techniques, there is a need to analyze and understand these risk factors better and predict heart disease earlier.

In this paper, the models we used to predict whether one will have or had heart disease are K-Nearest-Neighbor(KNN), decision tree, and neural network. Features are selected first using the Lasso coefficient. The models are then evaluated by k-fold validation, accuracy, and area under the curve of receiver operating characteristics. Lastly, the models are implemented with Python and established with an applicable website.

1 Introduction

Heart diease remains one of the leading causes of mortality worldwide, posing significant challenges to public health systems. Early and accurate prediction of heart disease is crucial for intervaention and saving the patient. In recent years, machine learning models have emerged as powerful tools for predicting whether a person will be vulnaerable to heart diseases.

In this context, we explore the efficacy of three prominent machine learning models: K-Nearest Neighbors (KNN), Decision Tree, and Neural Network, in predicting the risk of heart disease. The KNN algorithm, praised for its simplicity and effectiveness in classification tasks (Cover, Hart, 1967), is contrasted against the interpretability of Decision Tree models (Safavian, Landgrebe, 1991), and the comprehensive learning capability of Neural Networks (LeCun et al., 2015). Our study aims to perform a comparative analysis of these models on key predictive features, delineating their respective strengths and limitations in the domain of heart disease prediction.

Previous research has provided insights into various machine learning techniques for heart disease prediction, demonstrating the potential of these models in clinical settings (Chouhan et al., 2019). However, there remains a need for a comprehensive comparison that could almost immediately deliver an accurate output for anyone who can access the internet. We hypothesize that a nuanced understanding of each model's performance on key features can significantly enhance the predictive frameworks used in healthcare.

To address this, we have meticulously trained and tested the KNN, Decision Tree, and Neural Network models using a well-curated dataset from Kaggle's dataset Indicators of Heart Disease . Our findings are intended to contribute to the body of knowledge on machine learning applications in cardiology, providing a benchmark for future research and development in heart disease prediction algorithms.

2 Background

A recent study has been done using AI and Deep Learning Models for Cardiovascular Risk Prediction (Kruttanawong et al). The American Heart Association highlighted several key findings from their 2023 scientific sessions. One study demonstrated the effectiveness of AI in analyzing heart sound data from a digital stethoscope for detecting heart valve disease, outperforming traditional methods. Another study used AI and deep learning to analyze eye images of individuals with prediabetes and Type 2 diabetes to predict their risk of cardiovascular events, such as heart attacks and strokes.

Another related machine learning study in cardiovascular disease Prediction was conducted with different models: KNN, Neural Networks, Bayesian classification, Classification based on clustering, and Decision Tree (Soni et al). With the information they could gather in 2011, they provided a survey of current techniques of knowledge discovery in databases using data mining techniques that were being used at that time for medical research, particularly in Heart Disease Prediction. Whereas their results are the following the accuracy of Naive Bayes is 86.53 percent, the accuracy of Decision Tree is 89 percent, and the accuracy of KNN is 85.53 percent.

3 Methods

3.1 Model Selection

Since the output label for our dataset is binary, and the features are numerical values, we considered the following models:

1. KNN: since the target variable 'HadHeartAttack' is a binary option, we select K nearest, either Manhattan or Euclidean distance, tuples to the given tuple and find the majority votes of the k nearest tuple as a result. Its algorithm is presented below:

2. Decision tree: the tree is built by selecting the best split points through the calculation of gini index or entropy. The result is predicted by giving the features of the tuple and iterating through all branches. Its algorithm for building the tree and predicting given a tuple. The following are the formulars for gini index and entropy.
$Gini = 1 - \sum_{i=1}^{n} p_i^2$
$Entropy = - \sum_{i=1}^{n} p_i \log_2 p_i$

3. Neural Network: A neural network where most input features are binary is like a complex decision-maker that only understands ”yes” or ”no” answers. Each feature it looks at can only be in one of two states, similar to a light switch being either on or off. The network uses these simple inputs to make more complex decisions by considering how these yes/no features relate to each other and influence the final outcome. For the numerical parts, it works as combinations of perceptrons to arrive at 0 or 1 for the final prediction.

3.2 Hyperparameter Tuning

For each classifier, we use k-fold cross-validation, k = 5, to perform grid search cross-validation to acquire the performance of different parameters, and then use accuracy for evaluation and then choose the highest accuracy.

3.2.1 KNN

We tested k from 1 to 31 and for both gini index and entropy in knn.py.

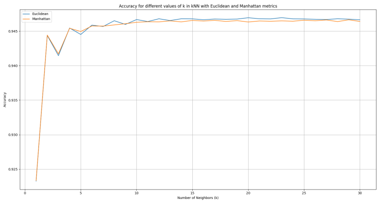

Figure 1: knn accuracy for metric and K

The best parameter we have is {metric: euclidean, n neighbors: 23}.

3.2.2 Decision Tree

We tested the minimum leaf sample from 1 to 6, maximum depth from 1 to 11, and splitting criteria of index and entropy in decisionTree.py.

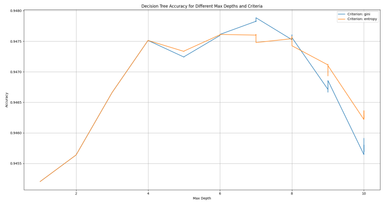

Figure 2: Decision Tree Accuracy for Max Depth

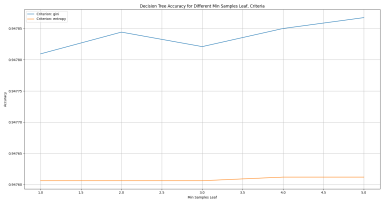

Figure 3: Decision Tree Accuracy for Min Leaf

The best parameter we have is {criterion: gini, max depth: 7, min samples leaf: 5}.

3.2.3 Neural Network

We tested the activation function ['tanh', 'relu'], alpha: [0.0001, 0.05], hidden layer sizes: [(5,), (10,), (5, 2)], solver: ['sgd', 'adam'], and learningrate': ['constant','adaptive'] in neuralNetwork.py.

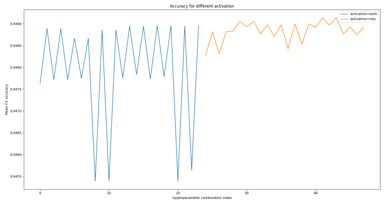

Figure 4: Neural Network Accuracy With Respect to Activation Function

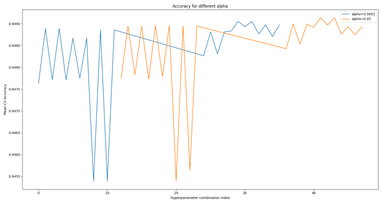

Figure 5: Neural Network Accuracy With Respect to Alpha Value

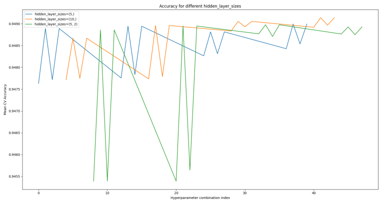

Figure 6: Neural Network Accuracy With Respect to Hidden Layer Size

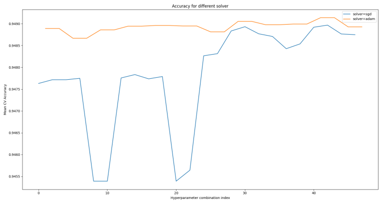

Figure 7: Neural Network Accuracy With Respect to Activation Function

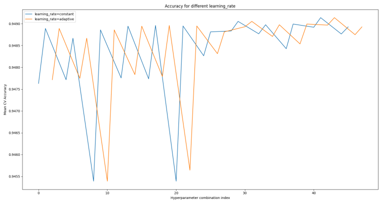

Figure 8: Neural Network Accuracy With Respect to Learning Rate

The best parameter is {activation: relu, alpha: 0.05, hidden layer sizes: (10,), learning rate: constant, solver: adam}.

4 Experiments/Results

4.1 before and after hyperparameter tuning

4.1.1 data cleaning and prepossessing

The dataset, titled "Indicators of Heart Disease (2022 UPDATE)," is a source from CDC's annual Behaviour Risk Factor Surveillance System survey, presenting over 400000 adult responses related to health status from 2022. We perform the one-hot encoding so that the categorical features within the data set will be map into new attributes with true or false or 0 or 1. The following are the attributes before converting: State, Sex, GeneralHealth, PhysicalHealthDays, MentalHealthDays, LastCheckupTime, PhysicalActivities, SleepHours, RemovedTeeth, HadHeartAttack, HadAngina, HadStroke, HadAsthma, HadSkinCancer, HadCOPD, HadDepressiveDisorder, HadKidneyDisease, HadArthritis, HadDiabetes, DeafOrHardOfHearing, BlindOrVisionDifficulty, DifficultyConcentrating, DifficultyWalking, DifficultyDressingBathing, DifficultyErrands, SmokerStatus, ECigaretteUsage, ChestScan, RaceEthnicityCategory, AgeCategory, HeightInMeters, WeightInKilograms, BMI, AlcoholDrinkers, HIVTesting, FluVaxLast12, PneumoVaxEver, TetanusLast10Tdap, HighRiskLastYear, CovidPos.

We have the following attributes after converting: PhysicalHealthDays, MentalHealthDays, PhysicalActivities, SleepHours, HadHeartAttack, HadAngina, HadStroke, HadAsthma, HadSkinCancer, HadCOPD, HadDepressiveDisorder, HadKidneyDisease, HadArthritis, DeafOrHardOfHearing, BlindOrVisionDifficulty, DifficultyConcentrating, DifficultyWalking, DifficultyDressingBathing, DifficultyErrands, ChestScan, HeightInMeters, WeightInKilograms, BMI, AlcoholDrinkers, HIVTesting, FluVaxLast12, PneumoVaxEver, HighRiskLastYear, State Alabama, ..., State Wyoming, Sex Female, Sex Male, GeneralHealth Excellent, ..., GeneralHealth Poor, LastCheckupTime 5 or more years ago, LastCheckupTime Within past 2 years (1 year but less than 2 years ago), LastCheckupTime Within past 5 years (2 years but less than 5 years ago), LastCheckupTime Within past year (anytime less than 12 months ago), RemovedTeeth 1 to 5, RemovedTeeth 6 or more, but not all, RemovedTeeth All, RemovedTeeth None of them, HadDiabetes No, pre-diabetes or borderline diabetes, HadDiabetes yes, but only during pregnancy (female), SmokerStatus Current smoker - now smokes every day, SmokerStatus Current smoker - now smokes some days, SmokerStatus Former smoker, SmokerStatus Never smoked, ECigaretteUsage Never used e-cigarettes in my entire life, ECigaretteUsage Not at all (right now), ECigaretteUsage Use them every day, ECigaretteUsage Use them some days, RaceEthnicityCategory Black only, Non-Hispanic, RaceEthnicityCategory Hispanic, RaceEthnicityCategory Multiracial, Non-Hispanic, RaceEthnicityCategory Other race only, Non-Hispanic, RaceEthnicityCategory White only, Non-Hispanic, AgeCategory Age 18 to 24, ..., AgeCategory Age 80 or older, TetanusLast10Tdap No, did not receive any tetanus shot in the past 10 years; TetanusLast10Tdap Yes, received Tdap; TetanusLast10Tdap Yes, received tetanus shot but not sure what type, TetanusLast10Tdap Yes, received tetanus shot, but not Tdap; CovidPos No, CovidPos Tested positive using home test without a health professional, CovidPos Yes.

Then we normalize the numerical values with sklearn.standardscaler. Each numerical feature in the dataset would have a mean of 0 and a standard deviation of 1. datacleaning.py

4.1.2 feature selection

We use Lasso regularization, also known as L1 regularization, for feature selection. It is a technique used in machine learning to prevent the overfitting of models. It does this by adding a penalty term to the loss function, which is proportional to the absolute value of the model coefficients. This penalty encourages the model to keep the coefficients as small as possible, leading to some coefficients becoming zero. The features we selected are physical health days, mental health days, sleep hours, weight in kilograms, had angina, had stroke, had asthma, had copd, had kidney disease, had arthritis, deaf or hard of hearing, blind or vision difficulty, difficulty walking, chest scan,alcohol drinkers, flu vax last 12 months, pneumo vax ever, sex female, sex male, general health excellent, general health fair, general health poor, general health very good, last checkup time within a year, removed teeth 1 to 5, removed teeth 6 or more, removed teeth all,removed teeth none, had diabetes no, had diabetes yes, smoker status current smoker, smoker status never smoked, e cigarette never used, race ethnicity black non hispanic, age category 18 to 24, age category 25 to 29, age category 30 to 34,age category 35 to 39, age category 40 to 44, age category 65 to 69, age category 70 to 74, age category 75 to 79, age category 80 or older, tetanus last 10 tdap, received tdap. Notice that we are solely using lasso linear regression as feature selection, not prediction. All the feature selection are being done before training the models. knn.py decisionTree.py nn.py

4.1.3 saving the model and applying the model

After we found the best hyperparameter for each model, we save them using joblib into a subfolder of the directory. Then we create a simple website using streamlit in main.py and the models we save for users to input values for all attributes for predicting whether they have the possibility to have a heart attack or not. The following is the link to the website, which was deployed on Amazon ec2.

4.2 Empirical Results

After hyperparameter tuning for each model, we tested the models using and evaluated them with accuracy and AUCROC curve (see Fig.9). The x-axis is false positive rate while the y-axis in the true positive rate. In the figureBoth accuracy and aucroc can be viewed with the table below as well. We prefer a model that has a larger area under the curve. We can observed that the aucroc for neural network classifier is larger than both KNN classifier and decision tree classifier, indicating that neural network classifier is more reliable for predicting new data entry.

	Accuracy	AUCROC
KNN	0.9470	0.84
Decision Tree	0.9479	0.86
Neural Network	0.9490	0.89

Table 1: Accuracy and AUCROC

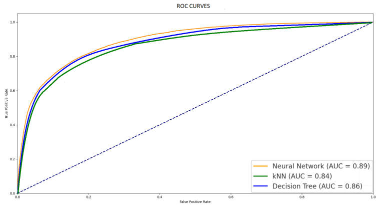

Figure 9: AUCROC

5 Discussion

In the development of our neural network model, we have achieved remarkable success, as evidenced by its impressive accuracy rate of 94.9 percent and an Area Under the Receiver Operating Characteristic (AUC-ROC) curve of 0.89. This performance marks a significant advancement over previously reported models. For instance, Soni, J., Ansari, U., Sharma, D., and Soni, S. in their 2011 study reported accuracy rates of 86.53 percent for the Naive Bayes model, 89 percent for the Decision Tree model, and 85.53 percent for the Artificial Neural Network (ANN) model. The substantial improvement in accuracy and AUC-ROC in our model demonstrates the effectiveness of the advanced techniques and methodologies employed in its development. This underscores our model's capability in delivering more reliable and precise predictions.

The results of our neural network model are quite promising when juxtaposed with the findings from the study conducted by Senthilkumar Mohan, Chandrasegar Thirumalai, and Gautam Srivastava. Our model has achieved an accuracy of 94.9 percent with an AUC-ROC of 0.89, surpassing all the models listed in the referenced study. Notably, the highest accuracy reported is by the Deep Learning model at 87.4 percent, with an HRFMLM (proposed) model closely following at 88.4 percent. Our model not only outperforms the Deep Learning model by 7.5 percent in terms of accuracy but also shows improvement over the HRFMLM model by 6.5 percent. In addition to accuracy, our neural network model also excels in the balance of classification error and AUC-ROC, indicating not only a high rate of correct predictions but also an excellent measure of the model's ability to discriminate between classes. This substantial increase in performance metrics illustrates the efficiency of our neural network model and its potential applicability in predictive tasks, setting a new standard in the domain.

In the subsequent phases of our project, after successfully launching a heart disease prediction website using our neural network model, we aim to focus on several critical areas to ensure the tool's continued relevance, accuracy. In order to increase the accuracy of the model, we have two options. The first one will be to retrain and update the model with newest data once a quarter. Another option will be to collect user input and train the model simultaneously. To maximize the utility of our prediction service, we will seek partnerships for integration with electronic health record systems. This will facilitate seamless use by healthcare professionals and ensure that our tool complements existing clinical workflows.

6 Contributions

The website is deployed at http://3.15.218.30.

The code details can be found at
github repo: https://github.com/LupinC/CS334Project.

The report details can be found at
overleaf: https://www.overleaf.com/2128328383kjqmysbdsrjkb97eaa

Lupin Cai: Responsible for coding: training, testing, saving the models, etc; deploying the website, and writing the first draft of the report.

Leo Dai: Responsible for finding the dataset, related articles, creating the power point presentation, reviewing the code, and proofreading the draft.

References

[1] T. Cover and P. Hart, ”Nearest neighbor pattern classification,” in IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21-27, January 1967, doi: 10.1109/TIT.1967.1053964.

[2] S. R. Safavian and D. Landgrebe, ”A survey of decision tree classifier methodology,” in IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, no. 3, pp. 660-674, May-June 1991, doi: 10.1109/21.97458.

[3] LeCun, Y., Bengio, Y., Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539

[4] Chouhan, V.; Singh, S.K.; Khamparia, A.; Gupta, D.; Tiwari, P.; Moreira, C.; Damaševičius, R.; de Albuquerque, V.H.C. A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images. Appl. Sci. 2020, 10, 559. https://doi.org/10.3390/app10020559

[5] Krittanawong, C., Virk, H.U.H., Bangalore, S. et al. Machine learning prediction in cardiovascular diseases: a meta-analysis. Sci Rep 10, 16057 (2020). https://doi.org/10.1038/s41598-020-72685-1

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

[7] S. Mohan, C. Thirumalai and G. Srivastava, ”Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques,” in IEEE Access, vol. 7, pp. 81542-81554, 2019, doi: 10.1109/ACCESS.2019.2923707.