# HEART DISEASE PREDICTION USING MACHINE LEARNING

ANURADHA SAHITHI PADAVALA
DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF CENTRAL FLORIDA, an600892@ucf.edu

ABSTRACT
MOTIVATION : The objective of this project is to forecast the presence of heart disease in individuals using specific health metrics. To construct our predictive model, four distinct classification methods were employed by the researchers. Additionally, the data underwent preprocessing to fulfill the model's criteria.
RESULT : The research paper underwent implementation and enhancement through the application of the random forest and XGBoost techniques. The accuracy achieved for these methods stands at 90% for Random Forest and 86.7% for XGBoost.

**BACKGROUND**

Heart disease, or cardiovascular disease (CVD), is a general term for a variety of disorders affecting the heart, the two most common of which being ischemic heart disease and strokes. The World Health Organization lists tobacco use, excessive alcohol consumption, sedentary lifestyles, and poor diets as the main behavioral risk factors for CVD. Extended exposure to these risk factors can lead to the development of early CVD symptoms, including high blood pressure, elevated blood glucose, elevated blood lipids, and obesity. The American Heart Association has identified warning indicators for cardiovascular disease (CVD), which include symptoms including dyspnea, coughing or wheezing that does not go away, swelling in the ankles and feet, chronic fatigue, appetite loss, and cognitive decline. Furthermore, there is growing data that points to a possible connection between the start of heart disease and the Coronavirus. By enabling timely action to stop future health deterioration, early diagnosis that is both accurate and timely is essential to reducing the risk of CVD and its global impact. In light of this requirement, machine learning algorithms that can forecast the risk of CVD based on the existence of particular risk factors are desperately needed. Effective early detection can considerably lower the risks connected to CVD, reducing the burden of this common health issue worldwide. Machine learning models have shown to be useful recently in a number of medical situations. They are effective in analyzing, evaluating, and forecasting a wide range of medical disorders. This work presents a machine learning approach that uses extensive health data to predict the occurrence of cardiovascular illnesses in individuals. Taking the severity of the heart-related ailments into account, researchers are concentrating on the development of the machine learning based system for the diagnosis of the Cardiovascular diseases. In this research paper, the study is about the approach to prediction of heart disease with the help of data related to multiple patients. The researchers explore four classification methods – Multilayer Perceptron, Support Vector Machines, Random Forest and Naïve Bayes.

**IMPLEMENTATION METHOD**

1. DATA COLLECTION

The dataset, sourced from Kaggle, consists of 297 instances, each characterized by 13 attributes. This collection of data serves as a valuable resource for analysis and exploration, offering insights into various aspects encapsulated by the specified attributes.

2. DATA PREPROCESSING

The effectiveness of a machine learning model hinges significantly on the caliber of the data employed in its construction, underscoring the crucial role of data pre-processing. This essential phase encompasses tasks such as purging corrupted or missing data points, addressing outliers, and engaging in processes like data transformation, resampling, and feature selection.

3. DATA CLEANING & VISUALIZATION

Initially, a scrutiny for missing values revealed none. Subsequently, an examination for outliers was conducted, exposing some anomalies. Given that the mild outliers contributed substantively to the ultimate diagnosis, only the extreme outliers were expunged. Identification of extreme outliers was accomplished using expressions (1) and (2), leveraging the interquartile range (IQR) as a gauge for data dispersion. Here, Q1 and Q3 denote the lower and upper quartiles, respectively.

$(75\% \times Q3) + 3 \times IQR$ (1)
$(25\% \times Q1) - 3 \times IQR$ (2)

Data points surpassing the threshold established by the first expression and falling below that of the second were eliminated. Consequently, two instances out of the original 297 were excluded from the dataset.

Following this, a correlation coefficient matrix was generated to scrutinize the relationships between various attributes and the output. In the correlation matrix, wherein the coefficient not only indicates the strength of the association between variables but also denotes the direction of the correlation—whether it is positive or negative.

4. CHECKING FOR IMBALANCES

Distortions in prediction accuracy can arise from imbalances in the output. To assess and address this concern, the balance of the output variable "target" was scrutinized. Following a thorough inspection, it was determined that the data exhibited a balanced distribution, maintaining a 9:11 ratio between the two categories. Consequently, there was no necessity to perform any resampling of the data.

5. DATA TRANSFORMATION

Transformation becomes necessary when dealing with datasets that incorporate diverse data formats or when amalgamating different datasets. In the context at hand, the transformation process was applied to convert nominal features into factors, rendering them compatible for utilization in Rstudio.

6. DIMENSIONALITY REDUCTION

Within the realm of machine learning, dimensionality reduction is a pivotal process aimed at diminishing the number of features. This reduction serves to mitigate complexity and forestall overfitting, accomplished through either feature selection or extraction.

Feature selection entails cherry-picking a subset of features from the initial set and employs methods such as CFS (Correlation-based Feature Selection), the Chi-squared test, and ridge regression. In the context of this paper, the chosen feature selection method was CfsSubsetEval, which assesses the value of a subset by considering both the individual predictive capacity of each feature and the extent of redundancy between them. The Weka software was employed for feature selection due to its array of attributes evaluators offering various testing and utilization options.

Diverging slightly from feature selection, feature extraction involves generating a new set of features from the original set. Principal Component Analysis (PCA) stands out as a widely utilized method in this regard. PCA calculates the projection of the original data into a more compact dimension space, facilitating the reduction of complexity while preserving essential information.

7. TRAIN TEST SPLIT

In the domain of machine learning, it is customary to partition the data into training and testing sets. The training set is employed to train the model, while the testing set is reserved for assessing the model's performance and predicting output. In this study, a hold-out method was implemented, allocating 90% of the data for training purposes and reserving the remaining 10% for testing the model's predictions.

8. APPLYING ALGORITHM

In the research paper, the authors experimented with Multilayer Perceptron, Support Vector Machines, Random Forest and Naïve Bayes. But as part of implementation and improving this research paper, Random Forest and XGBoost algorithms were used to improve the existing accuracy.

MULTILAYER PERCEPTRON - A Multilayer Perceptron (MLP) is a type of artificial neural network extensively utilized in machine learning applications. It falls under the category of feed forward neural networks, wherein data progresses unidirectionally—from the input layer through hidden layers to the output layer. The architecture involves multiple layers of nodes, including an input layer, one or more hidden layers, and an output

layer, with weighted connections between neurons. Activation functions introduce non-linearity, and the training process, often employing back-propagation, adjusts weights to minimize the disparity between predicted and actual outputs. Key considerations in MLPs include the choice of activation functions, loss functions, and hyperparameters such as the number of layers and neurons. Despite their versatility, MLPs may face challenges like overfitting, prompting the use of techniques like regularization and dropout. These networks serve as fundamental components in deep learning, contributing to diverse applications like image recognition, natural language processing, and regression tasks.

SUPPORT VECTOR MACHINES - Support Vector Machines (SVM) stand out as a robust category of supervised learning algorithms extensively utilized for both classification and regression tasks within the field of machine learning. The SVM methodology involves identifying an optimal hyperplane that effectively separates data points of distinct classes within a high-dimensional space. This algorithm is designed to maximize the margin, representing the distance between the hyperplane and the nearest data points of each class. SVMs demonstrate remarkable efficacy in addressing scenarios characterized by intricate decision boundaries and in spaces with a high number of dimensions. Furthermore, their adaptability extends to managing non-linear relationships through the incorporation of kernel functions, facilitating the transformation of input features into higher-dimensional spaces. The broad applicability of SVMs spans diverse domains, including image recognition, text classification, and bioinformatics, attributed to their versatile nature and capacity to generalize effectively across various data types.

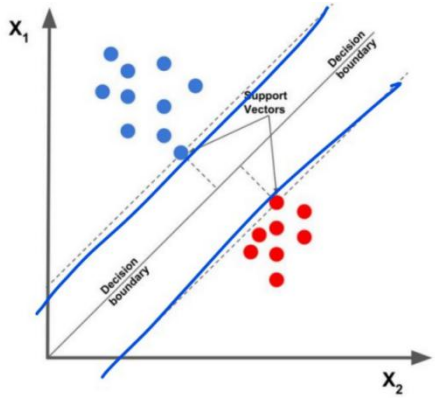



FIG.1 REPRESENTATION OF SUPPORT VECTOR MACHINE

NAIVE BAYES - Naïve Bayes stands out as a widely employed probabilistic classification algorithm in the realm of machine learning due to its efficiency and simplicity. Operating on the principles of Bayes' theorem, this algorithm assumes feature independence, hence the term "naïve." It computes the probability of an instance belonging to a specific class by assessing the likelihood of observed features. Despite its straightforward nature, Naïve Bayes consistently delivers impressive performance, particularly in tasks such as text classification, spam filtering, and sentiment analysis. Notably effective in scenarios with limited training data, the algorithm retains its popularity across various applications, emphasizing its efficiency and straightforward implementation in situations demanding swift and reliable classification.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

FIG.2 REPRESENTATION OF BAYES THEOREM

RANDOM FOREST - Random Forest stands as a highly effective ensemble learning method extensively utilized in machine learning for tasks encompassing both classification and regression. This algorithm functions by creating numerous decision trees during the training phase and derives predictions through the amalgamation of these individual trees. Each tree undergoes training on a random subset of the dataset, and the ultimate prediction is determined through a majority vote (in classification) or averaging (in regression) across all trees. Renowned for its proficiency in mitigating overfitting, adeptness in handling sizable datasets, and suitability for high-dimensional feature spaces, Random Forest exhibits inherent diversity among its individual trees, contributing to the model's resilience and adaptability. This characteristic renders it less vulnerable to outliers or noise. Acknowledged for its versatility and robust performance, Random Forest finds applications across diverse domains such as finance, healthcare, and image analysis, where precise and steadfast predictions hold paramount importance.

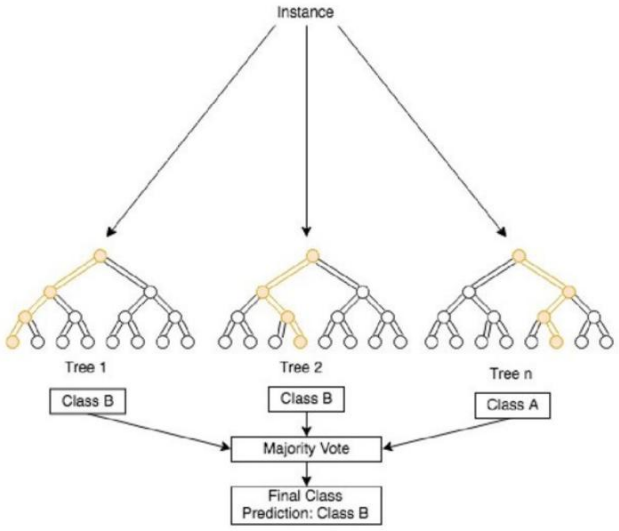



FIG.3 REPRESENTATION OF RANDOM FOREST

XGBOOST - XGBoost, an abbreviation for eXtreme Gradient Boosting, is a sophisticated and widely embraced machine learning algorithm recognized for its outstanding predictive capabilities. Operating within the gradient boosting framework, XGBoost systematically constructs an ensemble of weak learners, typically in the form of decision trees, and optimizes their contributions to elevate predictive accuracy. Proficient in tasks spanning regression, classification, and ranking, it showcases resilience against overfitting through the implementation of regularization techniques. XGBoost incorporates advanced features like parallel processing, tree pruning, and handling missing data, enhancing both efficiency and scalability. Renowned for its remarkable performance in machine learning competitions, XGBoost has become a preferred choice for data scientists and practitioners seeking superior results across a diverse array of applications, ranging from finance and healthcare to online advertising.

## 9. EVALUATION METRICS

In the world of machine learning, think of evaluation metrics as our trusty tools for measuring how well a model is doing its job across different tasks. These metrics give us clear numbers to understand if the model can handle new data effectively, helping us make smart choices when picking or improving a model. Depending on what the model is doing, we use different metrics:

Classification Metrics: These metrics, like accuracy, precision, recall, and the F1 score, tell us how good the model is at correctly figuring out the category of something.

Regression Metrics: For tasks where we're predicting values, metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) let us know how close our predictions are to the real values.

Clustering Metrics: When dealing with groups of data, metrics such as the Silhouette Score and Davies-Bouldin Index help us understand how tight and separate those groups are.

Ranking Metrics: In tasks that involve ranking items, metrics like Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (NDCG) show us how well the model can create a ranked list.

Anomaly Detection Metrics: For spotting unusual things in our data, metrics like Precision-Recall and Area Under the Receiver Operating Characteristic curve (AUROC) evaluate how good the model is at catching anomalies.

Choosing the right metric depends on what we're trying to achieve, and we often use a mix of these metrics to get a complete picture of how well our model is doing in different aspects.

In the research paper, the authors chose the following evaluation metrics for the experiment -

### CONFUSION MATRIX

Imagine you're training a machine learning model to distinguish between patients who have a certain medical condition and those who don't. You want your model to be accurate and reliable, but how do you measure its

performance? This is where the confusion matrix steps in as a valuable tool, breaking down predictions into different categories to give you a detailed picture of how well your model is doing.

Let's delve into the confusion matrix, a cornerstone in evaluating classification tasks. It's like a report card for your model, providing a breakdown of its predictions.

In the medical scenario, we have four key categories:

True Positives (TP): These are the moments your model shines. It correctly predicts that patients have the medical condition, and you can trust these positive identifications.

True Negatives (TN): On the flip side, true negatives represent the instances where the model gets it right by correctly identifying patients without the medical condition. It's a reassuring nod to the model's ability to discern the absence of the condition.

False Positives (FP): Oops, here's where the model trips up. False positives occur when the model predicts that a patient has the medical condition, but they actually don't. It's a bit like a false alarm, and it can be concerning if it happens too frequently.

False Negatives (FN): This is another stumble. False negatives happen when the model fails to identify patients who actually have the medical condition. It's like missing a critical piece of the puzzle and could lead to undetected health issues.

| | | Predicted class | |
|---|---|---|---|
| | | P | N |
| Actual class | P | True Positive (TP) | False Negative (FN) |
| | N | False Positive (FP) | True Negative (TN) |

FIG.4 REPRESENTATION OF CONFUSION MATRIX

Now, let's use these components to derive some key metrics that tell us more about the model's performance.

Accuracy: Think of accuracy as an overall grade for your model. It's calculated by adding up the correct predictions (TP + TN) and dividing by the total number of predictions (TP + TN + FP + FN). Essentially, accuracy tells you how often your model gets it right.

Precision (Positive Predictive Value): Precision is like the model's ability to make positive predictions accurately. It's calculated as TP / (TP + FP). In our medical example, precision helps us understand how trustworthy the model is when it says someone has the condition.

Recall (Sensitivity or True Positive Rate): Recall is all about capturing all the positive instances. It's calculated as TP / (TP + FN). In the medical context, recall helps us gauge the model's ability to identify all patients with the condition, ensuring that none are missed.

Specificity (True Negative Rate): Specificity is the counterpart to recall. It measures the model's ability to correctly identify negative instances and is calculated as TN / (TN + FP). This is particularly important in scenarios where identifying those without the condition is crucial.

F1 Score: The F1 score is like a balanced report card that considers both precision and recall. It's the harmonic mean of the two and provides a more nuanced understanding of the model's overall performance.

The beauty of the confusion matrix lies in its ability to offer insights beyond just accuracy. It allows you to see where your model excels and where it may need improvement. If false positives are causing concern, you might want to fine-tune the model's sensitivity. On the other hand, if false negatives are a worry, adjusting specificity could be the key.

In conclusion, the confusion matrix is your model's way of revealing its strengths and weaknesses. It's a personalized breakdown that empowers you to make informed decisions about its performance. Whether you're in the realm of health care or any other field, understanding the nuances of the confusion matrix is like having a compass to navigate the landscape of machine learning evaluation.

## SIGNIFICANT RESULTS FROM THE IMPLEMENTATION

The heart disease prediction model was constructed using four chosen machine learning techniques, and the outcomes were acquired through a three-stage process to achieve the optimal final model. Initially, predictions were made without data cleaning in the first stage, followed by predictions after data cleaning in the second stage. The final stage involved improving performance by conducting predictions after applying feature selection.

| Metric \ Model | MLP | SVM | RF | NB |
|---|---|---|---|---|
| **Accuracy** | 52.46 % | 75.41% | 77.05% | 70.49% |

FIG.5 Accuracy before removing the outliers

| Metric \ Model | MLP | SVM | RF | NB |
|---|---|---|---|---|
| **Accuracy** | 81.67% | 88.33% | 86.67% | 86.67% |

FIG.6 Accuracy after removing the outliers

Above is the method chosen by the researchers to build a heart disease prediction model.

## COMPARISION BETWEEN THE IMPLEMENTATION AND ORIGINAL PAPER'S RESULTS

The paper was implemented and improved by using the random forest and XGBoost methods. Where the accuracy for the above mentioned methods are 90% for Random Forest and 86.7 for XGBoost.

```
#Create a Gaussian Classifier
clf=RandomForestClassifier(n_estimators=100)

#Train the model using the training sets y_pred=clf.predict(X_test)
clf.fit(X_train,y_train)

y_pred=clf.predict(X_test)
```

```
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9
```

FIG.7 Accuracy for Random Forest

```
# Calculate accuracy
accuracy = accuracy_score(y_test, predictions)
print(f"XGBoost Model Accuracy: {accuracy * 100.0}%")
```

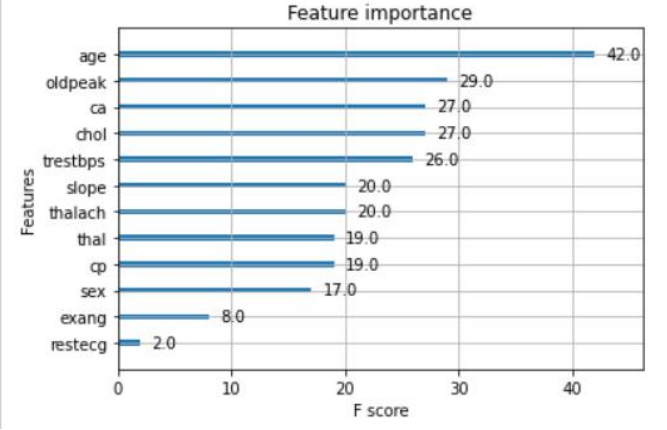
```
XGBoost Model Accuracy: 86.66666666666667%
```

FIG.8 Accuracy for XGBoost

## FEATURE IMPORTANCE

XGBoost provides a built-in function to plot feature importance. After training an XGBoost model, you can use the plot_importance() function to visualize the importance of each feature. This plot will display a bar chart showing the relative importance of each feature based on the chosen metric. You can customize the metric and other parameters according to your specific needs.

```
xgb.plot_importance(bst)
plt.show()
```



The results in the existing research paper were as follows -
Accuracy for MLP - 81.67%
Accuracy for SVM - 88.33%
Accuracy for RF - 86.67%
Accuracy for NB - 86.67%
Whereas, after implementation and improvement of the research paper, the results were increased to **90% accuracy for the Random Forest Method** and 86.67% for XGBoost Method.
Thus, there was an improvement of 2% when compared with the existing results.

## CONCLUSION AND INSIGHTS

In this project, the goal is to predict whether individuals have heart disease based on certain health measurements. To build our prediction model, researchers used four different classification methods. Before diving into the modeling process, we carefully collected and cleaned the data, making sure to handle any missing values or outliers. We also preprocessed the data to ensure it meets the requirements of our model. This involved visually exploring imbalances in the data and examining the correlation matrix. Also, after implementation and improving the existing research paper with few more techniques in my own python code, there was an increase in the accuracy for the Random Forest Method; XGBoost was also implemented.

## REFERENCES


[1] S. Rehman, E. Rehman, M. Ikram, and Z. Jianglin, “Cardiovascular disease (CVD): assessment, prediction and policy implications,” BMC Public Health, vol. 21, no. 1, p. 1299, 2021, doi: 10.1186/s12889-021-11334-2.
[2] O. Atef, A. B. Nassif, M. A. Talib, and Q. Nassir, “Death/Recovery Prediction for Covid-19 Patients using Machine Learning,” 2020.
[3] A. B. Nassif, I. Shahin, M. Bader, A. Hassan, and N. Werghi, “COVID-19 Detection Systems Using Deep-Learning Algorithms Based on Speech and Image Data,” Mathematics, 2022.
[4] H. Hijazi, M. Abu Talib, A. Hasasneh, A. Bou Nassif, N. Ahmed, and Q. Nasir, “Wearable Devices, Smartphones, and Interpretable Artificial Intelligence in Combating COVID-19,” Sensors, vol. 21, no. 24, 2021, doi: 10.3390/s21248424.
[5] O. T. Ali, A. B. Nassif, and L. F. Capretz, “Business intelligence solutions in healthcare a case study: Transforming OLTP system to BI solution,” in 2013 3rd International Conference on Communications and Information Technology, ICCIT 2013, 2013, pp. 209–214, doi: 10.1109/ICCITechnology.2013.6579551.
[6] A. Nassif, O. Mahdi, Q. Nasir, M. Abu Talib, and M. Azzeh, “Machine Learning Classifications of Coronary Artery Disease.” Jan. 2018.
[7] A. F. Otoom, E. E. Abdallah, Y. Kilani, A. Kefaye, and M. Ashour, “Effective diagnosis and monitoring of heart disease,” Int. J. Softw. Eng. its Appl., vol. 9, no. 1, pp. 143–156, 2015, doi: 10.14257/IJSEIA.2015.9.1.12.
[8] K. Vembandasamyp, R. R. Sasipriyap, and E. Deepap, “Heart Diseases Detection Using Naive Bayes Algorithm,” IJISETInternational J. Innov. Sci. Eng. Technol., vol. 2, no. 9, 2015, Accessed: Dec. 11, 2021. [Online]. Available: www.ijiset.com.
[9] A. U. Haq, J. P. Li, M. H. Memon, S. Nazir, R. Sun, and I. GarciáMagarinõ, “A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms,” Mob. Inf. Syst., vol. 2018, 2018, doi: 10.1155/2018/3860146.
[10] D. Shah, S. Patel, · Santosh, and K. Bharti, “Heart Disease Prediction using Machine Learning Techniques,” vol. 1, p. 345, 2020, doi: 10.1007/s42979-020-00365-y.
[11] K. Pahwa and R. Kumar, “Prediction of heart disease using hybrid technique for selecting features,” 2017 4th IEEE Uttar Pradesh Sect. Int. Conf. Electr. Comput. Electron. UPCON 2017, vol. 2018-January, pp. 500–504, Jun. 2017, doi: 10.1109/UPCON.2017.8251100.
[12] H. Jindal, S. Agrawal, R. Khera, R. Jain, and P. Nagrath, “Heart disease prediction using machine learning algorithms,” doi: 10.1088/1757-899X/1022/1/012072.
[13] D. Murphy, “Using Random Forest Machine Learning Methods to Identify Spatiotemporal Patterns of Cheatgrass Invasion through Landsat Land Cover Classification in the Great Basin from 1984 - 2011,” 2019.
[14] S. Liu, Z. Fang, and L. Zhang, “Research on Urban Short-term Traffic Flow Forecasting Model,” J. Phys. Conf. Ser., vol. 1237, no. 5, Jul. 2019, doi: 10.1088/1742-6596/1237/5/052026.
[15] “Support Vector Machines (SVM) | LearnOpenCV #.” https://learnopencv.com/support-vector-machines-svm/ (accessed