IT414

Data Warehousing and Data Mining

Project Report

Submitted By:

Aryaman Surya(211IT011), Vaidaant Thakur(211IT075),
Verma Ayush(211IT079)

Submitted To:

Department Of Information Technology,

National Institute of Technology, Karnataka.

CERTIFICATE PAGE

This is to certify that the project report entitled “Cardiac Disease Prediction: An Effective Machine Learning Algorithm for predicting risk of Cardiac Diseases” submitted to the Department of Information Technology, National Institute of Technology Karnataka (NITK), Surathkal, in partial fulfillment for the award of the degree of Bachelor of Technology in Information Technology, is a record of bona fide work carried out by Mr. Vaidaant Thakur (Roll No. 21IT075), Mr. Aryaman Surya (Roll No. 21IT011), and Mr. Verma Ayush (Roll No. 21IT079) under my supervision and guidance.

All assistance received from various sources has been duly acknowledged. No part of this report has been submitted elsewhere for the award of any other degree.

____

Signature of Invigilator

ABSTRACT

Heart disease remains a significant global health concern, emphasizing the need for accurate predictive models to enable early diagnosis and intervention. In this study, we introduce a web-based application aimed at predicting heart disease risk utilizing machine learning algorithms. Leveraging a dataset comprising vital health metrics such as age, gender, blood pressure, and cholesterol levels, we explore the effectiveness of various machine learning models. Specifically, we employ logistic regression with L1 and L2 regularization, logistic regression with principal component analysis (PCA), support vector machine (SVM), random forest, decision tree, and k-nearest neighbours (KNN). Through comprehensive experimentation and evaluation, we discover that random forest yields the highest accuracy, surpassing other models. Our findings underscore the potential of machine learning techniques in enhancing heart disease prediction accuracy and contribute to ongoing efforts in preventive healthcare, providing a valuable tool for risk assessment and early intervention strategies.

Index Terms: UCI repository, Machine learning, Optimal Feature Subset Selection (OFSSA), ETCD, Support Vector Machine (SVM), Principal Component Analysis (PCA)

TABLE OF CONTENTS

List Of Figures

Cardiac Disease Prediction: An effective machine learning algorithm for predicting risk of cardiac diseases

Aryaman Surya
211IT011
Department of Information Technology
aryamansurya.211it011@nitk.edu.in

Verma Ayush
211IT079
Department of Information Technology
vermaayush.211it079@nitk.edu.in

Vaidaant Thakur
211IT075
Department of Information Technology
vaidaantthakur.211it075@nitk.edu.in

***Abstract*—Heart disease remains a significant global health concern, emphasizing the need for accurate predictive models to enable early diagnosis and intervention. In this study, we introduce a web-based application aimed at predicting heart disease risk utilizing machine learning algorithms. Leveraging a dataset comprising vital health metrics such as age, gender, blood pressure, and cholesterol levels, we explore the effectiveness of various machine learning models. Specifically, we employ logistic regression with L1 and L2 regularization, logistic regression with principal component analysis (PCA), support vector machine (SVM), random forest, decision tree, and k-nearest neighbors (KNN). Through comprehensive experimentation and evaluation, we discover that random forest yields the highest accuracy, surpassing other models. Our findings underscore the potential of machine learning techniques in enhancing heart disease prediction accuracy and contribute to ongoing efforts in preventive healthcare, providing a valuable tool for risk assessment and early intervention strategies.**

***Index Terms*—UCI repository Machine learning Optimal Feature Subset Selection (OFSSA) ETCD Support Vector Machine (SVM) Principal Component Analysis (PCA)**

I. Introduction

Heart disease remains a pervasive health challenge globally, contributing significantly to morbidity and mortality rates. Early detection and accurate prediction of heart disease risk are paramount for effective preventive strategies and personalized patient care. In recent years, the integration of machine learning techniques into healthcare systems has shown promising results, offering a data-driven approach to augment traditional risk assessment methods.

In this project, we undertake a comprehensive endeavor towards heart disease prediction using machine learning algorithms. Our primary objective is to develop a user-friendly web application capable of assessing an individual's risk of developing heart disease based on their health data. Through the amalgamation of advanced machine learning models and modern web technologies, we aim to provide a tool that empowers both healthcare professionals and individuals to make informed decisions regarding heart disease prevention and management.

Central to our approach is the process of feature selection, wherein we discern the most influential attributes that significantly impact heart disease prediction. Utilizing various feature selection techniques such as normalization, ANOVA, chi-square tests, correlation analysis, and information gain, we strive to identify the most pertinent predictors while mitigating the curse of dimensionality and enhancing model interpretability.

Our project encompasses multiple stages, including exploratory data analysis, model development, web application design, and performance evaluation. We employ a diverse array of machine learning algorithms, including logistic regression with regularization, support vector machine (SVM), dimensionality reduction techniques like principal component analysis (PCA), k-nearest neighbors (KNN), random forest, and decision tree. By systematically comparing the performance of these models, we seek to identify the most accurate and robust predictive model for heart disease risk assessment.

Furthermore, the user-friendly interface of the web application allows individuals to input relevant health metrics such as age, gender, blood pressure, and cholesterol levels, facilitating personalized risk assessment. Through this initiative, we aim to bridge the gap between advanced data analytics and healthcare decision-making, thereby contributing to the advancement of preventive healthcare strategies and improving patient outcomes.

II. Literature Survey

In recent years, machine learning techniques have garnered attention for their potential to enhance prediction accuracy and efficiency[1] across various domains, including healthcare. Several studies have investigated cardiac disease prediction using diverse machine-learning algorithms and feature selection methods.

For instance, Tama et al. [2] proposed a two-tier ensemble-based coronary heart disease (CHD) detection model, achieving remarkable accuracy, F1, and AUC scores. Similarly, innovative approaches such as the HDPM prediction model [3], employing outlier identification, data balancing, and classification techniques, demonstrated high accuracy rates for heart disease prediction.

Furthermore, studies have explored the effectiveness of hybrid models [4], combining multiple classifiers to improve prediction accuracy. Hybrid models integrating meta-heuristic feature selection methods with classification algorithms have shown promising results, indicating the potential for enhanced performance through model integration.[5]

Feature selection plays a crucial role in model development, with various techniques such as chi-square, information gain, and lasso being explored for identifying pertinent features. Moreover, the choice of feature selection method and classifier may vary based on dataset characteristics, necessitating robust frameworks capable of adapting to diverse datasets and classifiers.

In light of these findings, our project aims to build upon existing research by implementing progressive feature selection techniques and leveraging state-of-the-art machine learning algorithms. By incorporating advanced feature selection methods and classifiers such as SVM, kNN, DT, NB, and RF, we seek to develop an effective framework for cardiac disease prediction. Evaluation metrics such as accuracy, sensitivity, specificity, precision, and F1-score will be employed to assess the performance of our framework across diverse datasets, paving the way for more accurate and reliable cardiac disease prediction models.

III. Proposed Method

A. Exploratory Data Analysis

a) Segregating heart diseases and non-heart diseases: The provided code creates two subsets of the dataset based on the 'target' variable, where one subset represents instances with a target value of 1 (indicating the presence of heart disease) and the other represents instances with a target value of 0 (indicating absence of heart disease). Descriptive statistics for each subset are computed and transposed for better readability. Then, a matplotlib figure with two subplots is created, each representing a heatmap using seaborn.heatmap(). These heatmaps visualize the mean values of the features for both subsets, with magenta indicating instances with heart disease and cyan indicating instances without. Finally, the subplots are titled accordingly, and the layout is adjusted for better presentation using the subplots' positions and sizes to ensure they fit within the figure canvas with padding of 2 units.). Then after counting the total no of numeric features and categorial features.

b) Distribution of categorial and numeric features: It utilizes matplotlib and seaborn libraries to create subplots displaying the distribution of categorical features in a dataset. It initializes a figure with 3 rows and 2 columns, setting the overall size. Then, it iterates through each categorical feature except the generated subplots using matplotlib and seaborn libraries to visualize the distribution of numerical features in a dataset. It initializes a figure with 2 rows and 2 columns, specifying the overall size. Then, it iterates through each numerical feature except the last one, plotting a distribution plot using sns.distplot() within each subplot. Each plot displays the distribution of the respective numerical feature, customized with color and title indicating the feature's name.

c) Counting categorical and numeric features with and without diseases: In this considering categorial features In this considering a heart diseases we are plotting no of that particular features which belongs to heart diseases similarly same procedueres for non-heart diseases.Within each subplot, seaborn's countplot() function is used to display the count of each category, distinguished by the hue "target" (indicating heart disease presence or absence) and customized with a specified color palette. Additionally, text annotations are added above each bar to display the exact count.For numerical features, a scatter plot is generated using seaborn's scatterplot() function, with one feature plotted against the other. The hue parameter is set to 'target' to distinguish between instances with and without heart disease, visualized with different colors from the specified palette.

B. Features Selection Process

a) Normalization: Normalization and standardization are preprocessing techniques in machine learning to adjust numerical data to a similar scale for improved algorithm performance. Normalization, done with MinMaxScaler, rescales features to a range between 0 and 1, preventing dominance by larger-magnitude features. Standardization, implemented with StandardScaler, transforms features to have a mean of 0 and a standard deviation of 1, making data more Gaussian-like and less sensitive to outliers.

b) Correlation: Correlation measures the strength and direction of a linear relationship between two variables. In a nutshell, positive correlation indicates that as one variable increases, the other tends to increase as well, while negative correlation suggests that as one variable increases, the other tends to decrease. A correlation value close to 1 or -1 indicates a strong relationship, while values close to 0 indicate a weak relationship. Analyzing correlations between features helps identify redundant or highly influential features, which can guide feature selection and model building processes, enhancing the overall effectiveness of machine learning algorithms.

c) Chi-Square: Chi square applied to categoricaldata is also telling us to remove fbs, restecg, thal and if we observe we can see that in the heatmap these features had very less correlation to the target feature. Chi-squared statistical tests for feature selection, aiming to identify the most relevant categorical features for predicting the target variable. It computes chi-squared scores and p-values for each categorical feature concerning the target variable. The chi-squared scores indicate the strength of association between the categorical feature and the target, while the p-values signify the statistical significance of these associations. Features with lower p-values and higher chi-squared scores are considered more informative.

d) Analysis of variance (ANOVA): ANOVA (Analysis of Variance) is a statistical technique used to determine whether there are statistically significant differences between the means of three or more independent groups. In the context of feature

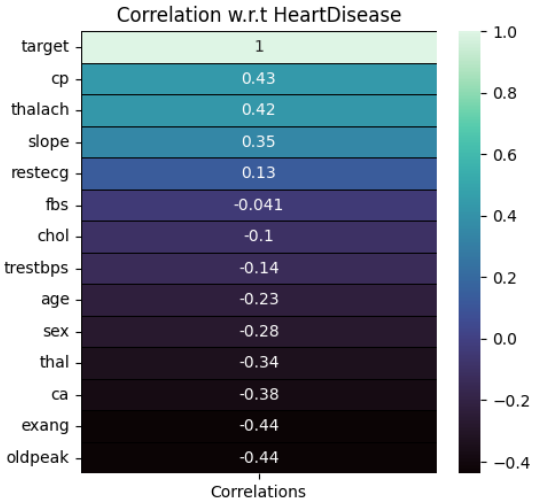


Fig. 1. Correlation Heatmap

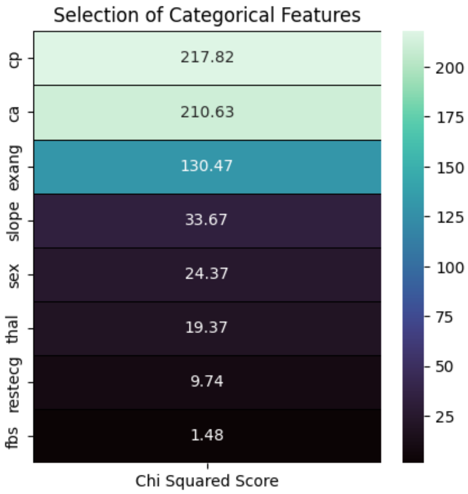


Fig. 2. Chi-Square Heatmap

selection, ANOVA calculates the F-statistic for each numerical feature by comparing the variance between multiple groups (defined by different categories of the target variable) with the variance within each group. A higher F-statistic suggests that the means of the numerical feature vary significantly across the categories of the target variable, indicating the feature's relevance for predicting the target.

e) InformationGain: the information gain (mutual information) for each categorical feature in the dataset using scikit-learn's mutualinfoclassifier. For every categorical feature, it computes the reduction in uncertainty about the target variable ('target') given the feature's value. The resulting information gain values are then visualized in a bar plot, sorted in descending order to highlight the most informative features, aiding in feature selection for predictive modeling.

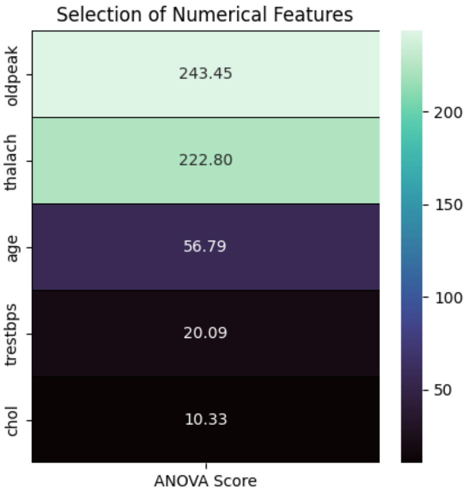


Fig. 3. ANOVA Heatmap

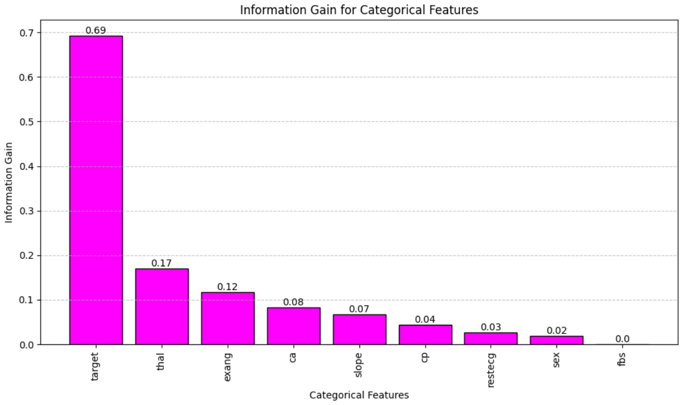


Fig. 4. Information Gain

C. Features Extraction Process

a) PCA: Principal Component Analysis (PCA) is utilized for dimensionality reduction and feature extraction. Categorical features in the DataFrame 'dfpca' are encoded using LabelEncoder to convert them into numerical values. PCA is then initialized with the desired number of components. The PCA model is fitted to the preprocessed data frame. Feature importances are obtained by calculating the absolute values of the PCA components. The index of the most important feature for each component is determined, and the corresponding feature names are extracted from the DataFrame.

b) Modelling Process: Classification of models, the process involves several key steps. First, the data is split into training and testing sets. Then, various classifiers are trained on the training set and evaluated on the testing set using performance metrics such as accuracy, cross-validation score, ROC-AUC score, confusion matrix, and classification report. These metrics provide insights into the model's predictive ability, generalization performance, and ability to distinguish between classes. Additionally, visualizations like ROC-AUC plots and confusion matrices aid in interpreting model performance and identifying areas for improvement.

c) Modeling Process Using Logistic Regressions: Logistic Regression (LR) stands as a widely recognized linear classification algorithm for its efficacy in binary classification endeavors. Within the provided code, LR is instantiated through the Logistic Regression class sourced from the scikit-learn library. Herein, the algorithm is configured with a designated regularization parameter and a specified random state to ensure reproducibility across iterations. The inclusion of a confusion matrix and classification report within the code furnishes comprehensive insights into the LR model's performance. These evaluation metrics delineate precision, recall, and F1-score metrics for each class, furnishing a detailed depiction of the model's discriminative capabilities and classification proficiency.

d) Modeling Process Using SVM: Support Vector Machine (SVM) is another widely used classification algorithm known for its effectiveness in handling complex datasets. In the provided code, SVM is implemented using Scikit-learn's SVC class with a specified regularization parameter and a random state. The confusion matrix and classification report provide detailed insights into the model's performance, including precision, recall, and F1-score for each class Using Logistic Regression.

e) Modeling process using Random Forest: RandomForestClassifier to construct a Random Forest classification model. It imports the RandomForestClassifier class from the sklearn.ensemble module and instantiates it with 100 decision trees (nestimators=100) and a fixed random state for reproducibility (random-state=0). However, the code lacks the fitting step to train the model on actual data. Typically, this would involve using the fit() method of the model instance with training data and corresponding labels. Thus, while the code sets up the Random Forest classifier configuration, it requires additional steps to be trained and used for making predictions.

f) Modeling process using Decision Tree: DecisionTreeClassifier to instantiate a Decision Tree classification model. It imports the DecisionTreeClassifier class from the sklearn.tree module and creates an instance with a fixed random state for reproducibility (random-state=0). However, the code lacks the fitting step, which is crucial for training the model on actual data. Typically, this would involve using the fit() method of the model instance with training data and corresponding labels. Therefore, while the code sets up the Decision Tree classifier configuration, it requires additional steps to be trained and utilized for making predictions.

g) Modeling process using KNN: The KNeighborsClassifier class from the sklearn.neighbors module. Following this, an instance of the KNeighborsClassifier class is created with the parameter neighbors set to 5, indicating that the model will consider the 5 nearest neighbors when making predictions. However, the code lacks the fitting step where the model is trained on actual data. Typically, this would involve using the fit() method of the model instance with training data (Xtrain) and corresponding labels (ytrain). This step is crucial for the model to learn patterns from the data and make accurate predictions. Therefore, the provided code snippet sets up the KNN classifier configuration but requires additional steps to train the model on data.

D. User Interface Designing

In this project, we are utilizing five different models for the prediction of heart and non-heart diseases: decision tree, random forest, support vector machine (SVM), logistic regression, and k-nearest neighbors (KNN) algorithm. The user interface is designed using HTML and CSS for the frontend, and Flask for the backend. The Flask backend is responsible for handling user requests and executing predictions using the five trained machine learning models. Upon receiving user input through an HTML form, the backend loads the respective models using the pickle library and passes the input data to each model for prediction. The predictions are then returned to the frontend, where they are displayed to the user. This seamless interaction between the Flask backend and the HTML/CSS frontend creates an intuitive user experience, allowing users to input their data and receive accurate predictions from multiple machine learning models with ease. The design details are explained further in the results and analysis section.

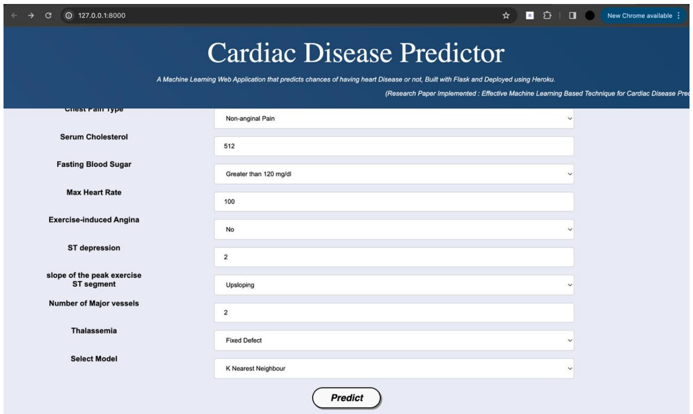
IV. Result And Analysis

In this the segment is integral to a web application focused on predicting heart disease. Users input 11 health parameters including age, sex, chest pain type, cholesterol level, fasting blood sugar, maximum heart rate achieved, exercise-induced angina, ST depression induced by exercise, slope of the peak exercise ST segment, number of major vessels colored by fluoroscopy, and thalassemia type. Upon submission, the server-side code processes this data, converts it into a numpy array, and selects one of five machine learning models: Decision Tree, Support Vector Machine (SVM), Random Forest, Logistic Regression, or K-Nearest Neighbors (KNN), based on the user's choice. After selecting the appropriate model, it's loaded from a pre-trained file using pickle. Finally, the loaded model predicts the likelihood of heart disease from the input data, assigning a binary result (0 for no heart disease, 1 for heart disease), stored in the variable myprediction.

V. Conclusion And Future Work

In conclusion, the heart disease prediction project has successfully demonstrated the feasibility of utilizing machine learning algorithms for accurate risk assessment. Through the implementation of Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree, k-nearest Neighbors (KNN), Random Forest, and ensemble methods, alongside rigorous feature selection techniques, the project has yielded promising results in predicting heart disease risk within a user-friendly web application framework.

The integration of various machine learning models has provided valuable insights into the predictive capabilities and performance metrics of each approach. Logistic Regression



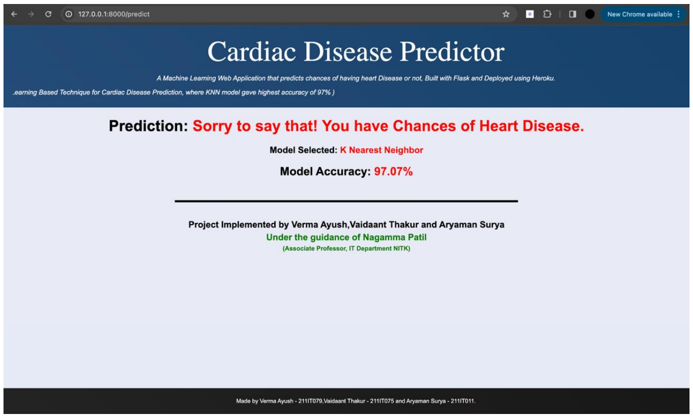
Cardiac Disease Predictor

A Machine Learning Web Application that predicts chances of having heart Disease or not, Built with Flask and Deployed using Heroku.

(Research Paper Implemented : Effective Machine Learning Based Technique for Cardiac Disease Pre

Chest Pain Type	Non-anginal Pain
Serum Cholesterol	512
Fasting Blood Sugar	Greater than 120 mg/dl
Max Heart Rate	100
Exercise-induced Angina	No
ST depression	2
slope of the peak exercise ST segment	Upsloping
Number of Major vessels	2
Thalassemia	Fixed Defect
Select Model	K Nearest Neighbour

Fig. 5. User Interface (Home Page)



Cardiac Disease Predictor

A Machine Learning Web Application that predicts chances of having heart Disease or not, Built with Flask and Deployed using Heroku.

.earning Based Technique for Cardiac Disease Prediction, where KNN model gave highest accuracy of 97%)

Prediction: Sorry to say that! You have Chances of Heart Disease.

Model Selected: K Nearest Neighbor

Model Accuracy: 97.07%

Project Implemented by Verma Ayush,Vaidaant Thakur and Aryaman Surya

Under the guidance of Nagamma Patil

(Associate Professor, IT Department NITK)

Made by Verma Ayush - 211IT079,Vaidaant Thakur - 211IT075 and Aryaman Surya - 211IT011.

Fig. 6. User Interface (Result Page)

showcased robust performance, while SVM exhibited high accuracy rates. Decision Tree, KNN, and Random Forest algorithms contributed significantly to the predictive power, each offering unique advantages in terms of interpretability, scalability, and accuracy.

The inclusion of advanced feature selection methods, including normalization, ANOVA, chi-square tests, correlation analysis, and information gain, has enhanced the models' ability to discern relevant predictors. Furthermore, the incorporation of evaluation metrics such as confusion matrices and classification reports has provided comprehensive assessments of model performance, enabling informed decision-making in healthcare contexts.

Looking ahead, future work will focus on refining the predictive models to achieve even higher accuracy rates. This may involve exploring alternative algorithms, fine-tuning hyperparameters, and incorporating additional features for improved prediction outcomes. Additionally, efforts to enhance the user interface with more sophisticated styling and interactive features will be prioritized to enhance usability and engagement.

Furthermore, considerations for scalability, interpretability, and deployment optimization will be paramount for wider adoption in clinical and healthcare settings. By addressing these areas of improvement, the heart disease prediction web application will continue to evolve as a valuable tool for healthcare professionals and individuals in managing and preventing heart disease effectively.

In summary, while the current iteration of the project has yielded promising results, ongoing research and development efforts will be essential to further enhance predictive accuracy, usability, and deployment readiness. Through continuous refinement and optimization, the project aims to make meaningful contributions to the field of preventive healthcare and improve patient outcomes in the fight against heart disease.

References

[1] J. Mishra, S. Tarar, Chronic Disease Prediction using Deep Learning BT - Advances in Computing and Data Sciences, 2020, pp. 201–211.

[2] B.A. Tama, S. Im, S. Lee, Improving an intelligent detection system for coronary heart disease using a two-tier classifier ensemble, Biomed. Res. Int. 2020 (2020) http://dx.doi.org/10.1155/2020/9816142.

[3] N.L. Fitriyani, M. Syafrudin, G. Alfian, J. Rhee, HDPM: An effective heart disease prediction model for a clinical decision support system, IEEE Access 8 (2020) 133034–133050, http://dx.doi.org/10.1109/ACCESS.2020.3010511.

[4] M. Tarawneh, O. Embarak, Hybrid Approach for Heart Disease Prediction using Data Mining Techniques BT - Advances in Internet, Data and Web Technologies, 2019, pp. 447–454.

[5] M. Tarawneh, O. Embarak, Hybrid Approach for Heart Disease Prediction using Data Mining Techniques BT - Advances in Internet, Data and Web Technologies, 2019, pp. 447–454.

[6] A.L. Bui, T.B. Horwich, G.C. Fonarow, Epidemiology and risk profile of heart failure, Nat. Rev. Cardiol. 8 (1) (2011) 30–41, http://dx.doi.org/10.1038/ nrcardio.2010.165.

[7] TY - JOUR AU - Wadhawan, Savita AU - Maini, Raman PY - 2022/08/01 SP - 109709 T1 - ETCD: An effective machine learning based technique for cardiac disease prediction with optimal feature subset selection VL - 255 DO - 10.1016/j.knosys.2022.109709 JO - Knowledge-Based Systems ER -