A Comparison of Selected Supervised ML Classifiers on Heart Failure Prediction.

DV2578 Machine Learning Project

Lokesh Kola
Dept of Computer Science
BTH
Karlskrona, Sweden
lokl@student.bth.se

***Abstract*—Background: Nowadays heart diseases and heart attacks were very commonly occurring diseases for people. There are many causes if the treatment is delayed for heart diseases.**
Objectives: To perform prediction on Heart failure by using Decision tree, Random forest, Knearst neighbour, and support vector machine.
Method: The dataset was loaded and preprocessed. Encoding of the categorical variables was performed and the important features were extracted by using selectkbest method. By using the selected features the predictions were performed by tuning hyperparameters of every algorithm. The detailed description is in the method section.
Results: The accuracies and confusion matrix were analyzed for all algorithms. The detailed description is in the results section.
Conclusion: The highest accuracy obtained was 88% by Decision tree

***Index Terms*—Machine Learning, Heart Disease, Supervised Classifiers**

I. Introduction

The Heart stroke is the most common in the world. As per the report of 2020, 2.4M deaths were registered in WHO due to heart diseases. There are many types of Heart diseases [4]. They are Coronary Artery Disease (CAD), Heart Arrhythmias, Heart Failure, Heart Valve Disease, Pericardial Disease, Cardiomyopathy (Heart Muscle Disease), Congenital Heart Disease. The consequences are a danger if the treatment for heart diseases is delayed [5]. The proper medication is required for the patients. This project mainly focus on Heart Faliure.

Most of the papers [7], [8], [9], [10] were performed the heart disease prediction using Decision Tree(DT), Random Forest(RF), Support vector machine(SVC), KNearstNeighbour(KNN) algorithms. The dataset which was chosen is also working efficiently with these algorithms. Therefore these algorithms were selected for predicting heart disease and a comparison will be performed.

The dataset "heart.csv" is chosen for the prediction and this dataset was taken from Kaggle [6]. This dataset is the combination of some other datasets as per information from Kaggle. The total instances are 918 and 12 attributes. The dataset consists of no null values and duplicates. The 12 attributes are Age, Sex, Chest pain Type, RestingBP (Blood Pressure) in [mm Hg], Cholesterol (serum) in [mm/dl], FastingBS (Blood Sugar), RestingECG (Electro Cardio Diagram Results), MaxHR (Heart Rate), Excercise Angina, Old Peak and Heart Disease. Heart Disease is the target variable and the rest attributes are all features.

Mostly in all papers [7], [8], [9], [10], the author performed the prediction on heart failure by using a single algorithm like only with Decision Tree or Random Forest or KNN and some papers are also on comparison but not more effectively using hyperparameter tuning. Now the research gap in this project shows the comparison of the selected algorithms in prediction with feature extraction and hyperparameter tuning. The comparison of the algorithms is performed using the evaluation metrics like Accuracy, F1-score.

II. Related Work

Purushottam, Kanak, and Richa [7] performed a study on Cardiovascular disease (CVD). They developed an automated system that will predict the risk levels of patients with taking their health as a parameter. They also mentioned that the system will be working at low costs. decision tree algorithms were used to predict in this system. The prediction was performed on parts of the dataset which means the dataset was divided into two parts. The accuracy achieved is 86% and 87% for the first half and second half dataset respectively.

Archana and Rakesh [8] performed a prediction of heart disease using four Machine Learning algorithms. The algorithms chosen are SVM, DT, Linear Regression, KNN. The dataset chosen was containing more effecting attributes. The feature extraction was also performed in this prediction. The accuracies obtained were 83%, 79%, 78%, and 87% for SVM, DT, LR, and KNN respectively. In this study, tuning hyperparameters were not performed. Whereas in this project the hyperparameter tuning was performed to get efficient prediction with appropriate parameters for a particular algorithm.

Singh, Sinha, and Singh [9] predicted heart disease using RF. The prediction was performed by using a different number of splits and trees. And the dataset instances were 303. The

final accuracy obtained by applying 3-fold, 5-fold, and 10-fold cross-validation are 85.81%.

Nida and Muhammad [10] developed an Efficient Heart Disease Prediction System using KNN. This study mainly focused on cardiovascular disease. The dataset used for this study has 14 attributes. The literature review of different related papers was performed and made a table for readability. The predictions were made for three cases namely before Re-sampling, After Re-sampling, and Using SMOTE. The accuracy achieved for KNN is 79.20%, which is the highest among all cases.

The motivation behind choosing the algorithms is the above papers. In this project, Heart Failure Prediction was performed with a dataset containing 918 instances and 12 attributes. The feature extraction and hyperparameter tuning were performed for efficient prediction.

III. Method

The method in the following sections

A. Importing Required Modules

Initially, the required modules are pandas, NumPy, LabelEncoder, SelectKBest, f_regression, Decision tree Classifier, metrics, SVC, KNNclasifier, RFclassifier, traintestsplit, GridSearchCV, StratifiedKFold, evaluations metrics accuracy score, f1-score, confusion matrix. These modules are imported in the first step.

B. Loading the Dataset

The dataset was loaded to a data frame by using the pandas. The dataset consists of 918 instances and 12 attributes.

C. Pre-processing Data

The dataset was checked for null values by using df.isnull().sum() . No null values were found in the dataset. Similarly checked for duplicates and none was found. The dataset was very clean.

D. Encoding

The categorical variables in the dataset are Sex, ChestPainType, RestingECG, ExerciseAngina, ST_Slope. The labels or categories for sex are male or female, for chestpaintype is asymptomatic(ASY) or Non-Anginal pain(NAP) or Atypical Angina (ATA) or Typical Angina (TA), for RestingECG is Normal or LVH or ST, for ExcericeAngina is Yes or No, for ST_Slope is Up or Down or Flat. The categories for each categorical variable are encoded by using Label Encoder. Then correlation check for every variable by using df.corr().

E. Univariate Analysis

The Univariate analysis was performed. Every attribute was analyzed by using the boxplot diagram. The Box plot for every attribute is as follows.

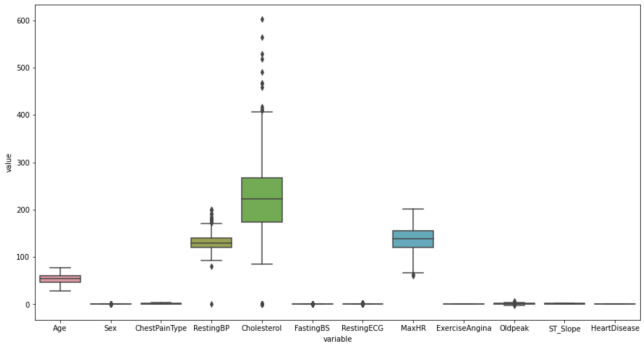

Fig. 1. Boxplot of all the Features

F. Feature Extraction

Then the feature extraction was performed by using SelectKBest algorithm. If the prediction was performed with all attributes then it might obtain less accuracy because some attributes might not correlate with the target variable. So the features which were more affecting the target variable were extracted and the prediction was performed by using the selected features.

The SelectKBest algorithm is based on f_regression metrics as a test because the some of the values in the oldpeak feature are negative in the dataset.The f_regression is used to identify potentially predictive feature for a downstream classifier, irrespective of the sign of the association with the target variable [11]. The values lies in between [-1,1].

The test was performed on all features and provided the scores. The top 6 features were selected for the prediction. There is no specific reason behind choosing top 6 and just more than half features among all features might predict efficiently. Those were mentioned in a table as follows.

Feature	Score
ST_Slope	415.830346
ExerciseAngina	296.144771
Oldpeak	178.615120
MaxHR	174.913585
ChestPainType	161.185346
Sex	94.253184

TABLE I
TABLE OF SELECTED FEATURES.

The other 4 attributes(”RestingBP”, ”Cholesterol”, ”FastingBS”, ”RestingECG”) were dropped from the data frame. Then the train-test split was performed by using the train-test algorithm. 80% of data were given for training and 20% of data was used for testing.

G. Prediction With Hyper-Parameter Tuning

The hyperparameter tuning was performed for effective prediction. The accuracy might not be obtained if the prediction was performed without tuning hyperparameters. The hyperparameter tuning was performed by using GridSearchCV. The parameters are the object of selected algorithms and their

different parameters. The code of all the algorithms was pasted in the appendix section

After tuning the hyperparameters the gridseaechcv will give the perfect parameters for every algorithm by performing Stratified 10-fold cross-validation. The obtained parameters were used for prediction. similarly for the selected algorithms tuning was performed and predictions were made. The results were discussed in the Results section.

IV. Results & Analysis

A. Results

After the prediction are made, some evaluation metrics was used to know the efficiency of predictions. The accuracy and confusion matrix were used as a evaluation metrics.

The results obtained were shown in table as follows.

Algorithms	Accuracy
DT	**88.77**%
RFC	87.75%
SVC	85.71%
KNN	86.73%

TABLE II
TABLE OF ACCURACIES

The DT classifier had achieved 88.77% accuracy whereas RFC got 87%, SVC got 85% and KNN got 87% accuracies. The bar graph was drawn for the accuracies of the algorithms is as follows. The confusion metrics was also calculated for every algorithms. All confusion matrix was mentioned below.

1) The confusion matrix for DT is [[65 21] [11 87]].
2) The confusion matrix for RF is [[69 17] [12 86]].
3) The confusion matrix for SVC is [[64 22] [14 84]].
4) The confusion matrix for KNN is [[66 20] [13 85]].

B. Analysis / Discussion

The predictions were performed by tuning all algorithm's hyperparameters by providing the training data. Hence all the four selected algorithms performed very closely in the terms of accuracy. The papers which were discussed in the related work sections meant that some of the selected algorithms was not efficiently performed whereas in this project with the tuning hyperparameters obtained best performance.

The figure 2 shows that all the selected algorithms were performed a bit closer accuracy. Whereas the accuracy obtained by DT is 88.77%, RFC is 87.75%, SVC is 85.71%, and KNN with 86.73%. All the obtained accuracies were close enough because of the hyperparameter tuning. Amoung all selected algorithms DT was performed well and achieved a slightly higher accuracy than other algorithms.

The SVC algorithm was obtained with 85.71% accuracy, which is lowest accuracy but not worse. The second highest accuracy obtained by RFC with 87.75%. Similarly then KNN with 86.73%. Hence by the results obtained every selected algorithm is performing efficiently. The DT algorithm is obtained highest performance among other selected algorithms. The DT algorithm is recommended for the prediction of heart failure.

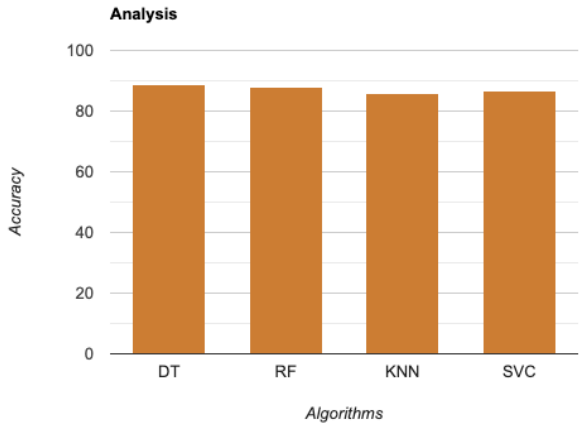

Fig. 2. The analysis of obtained accuracies

V. Conclusions

Heart Failure disease was not a neglectable disease. The symptoms of heart failure must be taken seriously and proper medication must be required. For this purpose prediction of heart failure by symptoms is very essential. In this study, the selected four machine learning algorithms were used to predict the heart failure. In this process, features selection and hyperparameter tuning was performed for better results. This study concludes that the Decision Tree (DT) Classifier obtains the slighter highest accuracy of 88% among RF, SVC, and KNN. The RF, SVC, and KNN algorithms were also performed good but DT has achieved a bit slighter high accuracy. Therefore DT is recommended for Heart disease prediction.

References

[1] Flach, P. (2012). Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press.

[2] Shaheen, H. , Mohamed, M. , Basset, F. , Rashed, M. , Theruvan, N. and Mosbah, S. (2022) Heart Failure Prediction in Athletic Heart Remodeling among Long Distance Runners. World Journal of Cardiovascular Diseases, 12, 1-10. doi: 10.4236/wjcd.2022.121001.

[3] fedesoriano. (September 2021). Heart Failure Prediction Dataset. Retrieved [Date Retrieved] from https://www.kaggle.com/fedesoriano/heart-failure-prediction.

[4] Types of Heart Diseases from https://www.webmd.com/heart-disease/heart-disease-types-causes-symptoms

[5] Kathleen Dracup, Debra K.Moser Mickey, Eisenberg, Hendrika Meischke, Angelo A.Alonzo, Allan Braslow, Causes of delay in seeking treatment for heart attack symptoms.

[6] Heart failure Prediction Dataset https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

[7] Purushottam, K. Saxena and R. Sharma, "Efficient heart disease prediction system using decision tree," International Conference on Computing, Communication & Automation, 2015, pp. 72-77, doi: 10.1109/CCAA.2015.7148346.

[8] A. Singh and R. Kumar, "Heart Disease Prediction Using Machine Learning Algorithms," 2020 International Conference on Electrical and Electronics Engineering (ICE3), 2020, pp. 452-457, doi: 10.1109/ICE348803.2020.9122958.

[9] Singh Y.K., Sinha N., Singh S.K. (2017) Heart Disease Prediction System Using Random Forest. In: Singh M., Gupta P., Tyagi V., Sharma A., Ören T., Grosky W. (eds) Advances in Computing and Data Sciences. ICACDS 2016. Communications in Computer and Information Science, vol 721. Springer, Singapore. https://doi-org.miman.bib.bth.se/10.1007/978-981-10-5427-3_63

[10] Nida Khateeb and Muhammad Usman. 2017. Efficient Heart Disease Prediction System using K-Nearest Neighbor Classification Technique. In ¡i¿Proceedings of the International Conference on Big Data and Internet of Thing¡/i¿ (¡i¿BDIOT2017¡/i¿). Association for Computing Machinery, New York, NY, USA, 21–26. DOI:https://doi-org.miman.bib.bth.se/10.1145/3175684.3175703

[11] sklearn.feature_selection.f_regression from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_regression.html

VI. Appendix

The hyperparameter tuning code are as follows:

parameters = "criterion":["gini","entropy"]
, "max_depth":[1,2,3,4,5,6,7,None]
grid_search = GridSearchCV(estimator =DTC or RFC or KNN or SVC,
param_grid = parameters,
scoring = 'accuracy',
cv = StratifiedKFold(10),
n_jobs = -1)
grid_search = grid_search.fit(X_train, y_train)

The best parameters for prediction are

1) for DT is 'criterion': 'entropy', 'max_depth': 3
2) for RF is 'bootstrap': False, 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 48
3) for SVC is 'C': 1000, 'kernel': 'linear'
4) for KNN is 'algorithm': 'auto', 'leaf_size': 30, 'metric': 'minkowski', 'metric_params': None, 'n_jobs': 1, 'n_neighbors': 5, 'p': 2, 'weights': 'distance'