Heart Disease Prediction Report

Team Members

Mohit Monga – 102103593

Nitish Gaba – 102103310

Anraobir Singh – 102103590

Madhav Garg – 102103589

Team Name: Heart-Guard

Under the guidance of

Amardeep Singh

INDEX

1. Introduction

Heart disease is the major cause of deaths globally. More people die annually from CVDs than from any other cause, an estimated 12 million people died from heart disease every year. Heart disease kills one person every 34 seconds in the United States. Heart attacks are often a tragic event and are the result of blocking blood flow to the heart or brain. People at risk of heart disease may show elevated blood pressure, glucose and lipid levels as well as stress. All of these parameters can be easily measured at home by basic health facilities. Coronary heart disease, Cardiomyopathy and Cardiovascular disease are the categories of heart disease. The word "heart disease" includes a variety of conditions that affect the heart and blood vessels and how the fluid gets into the bloodstream and circulates there in the body. Cardiovascular disease (CVD) causes many diseases, disability and death. Diagnosis of the disease is important and complex work in medicine. Medical diagnosis is considered as crucial but difficult task to be done efficiently and effectively .The automation of this task is very helpful. Unfortunately all physicians are not experts in any subject specialists and beyond the scarcity of resources there some places. Data mining can be used to find hidden patterns and knowledge that may contribute to successful decision making. This plays a key role for healthcare professionals in making accurate decisions and providing quality services to the public. The approach provided by the health care organization to professionals who do not have more knowledge and skills is also very important. One of the main limitations of existing methods is the ability to draw accurate conclusions as needed. In our approach, we are using different data mining techniques and machine learning algorithms, “Logistic Regression"," Naïve Bayes","Support Vector Machine","K-Nearest Neighbors","Decision Tree","Random Forest" to predict the heart disease based on some health parameters.

2. Literature Survey

2.1 Features selection

The detection and prediction of heart disease an important issue. For this reason, a lot of work has been done in this area. We divide existing work into two categories: The first presents approaches that select the most relevant patient by features selection, and the second is to explore the learning algorithms that offers high accuracy. 2.1 Features selection The choice of features is one of the major challenges to train a predictive learning algorithm. Several studies have been carried out and we present the most recent works in the literature. [6] The authors made a kind of comparison of the different Machine Learning algorithms on two classes of attributes. The first one contains 10 attributes and the second one contains 14 attributes. The problem with this study is that the authors did not mention the attributes treated. The authors of [5] developed the same approach as [6] but with only six attributes (Age, Sex, Using Blood Pressure, Heart Rate, Diabetes, Hyper cholesterol). The authors of [18], in their study surveyed the most commonly used attributes in cardiac disease detection in different scientific articles, and returned three classes, mentioned in Table 1:

Table 1. The classes of the attributes used

Authors	Attributes
[14] T John Peter et al., 2012 [7] I.S Jenzi,2013 [16] S.Radhimeenakshi 2016	Number of attributes : 13 (Age, Gender, CPT, FBS, RECG, Ex-Ang, SL, Col-Ves, Thal, SC, Thalach, Old peak, RBP)
[3] Chaitrali S et al., 2012 [8] C.Kalaiselvi 2016	Number of attributes : 10 (Age, Gender, CPT, FBS, RECG, SC, Thalach, Smoking, Alchool, Obesity)
[13] Shamsher Bahadur et al., 2013 [10] Hlaudi Daniel et al., 2014	Number of attributes : 6 (CPT, Ex-Ang, Col-Ves, RBP, Num, Smoking)

2.2 Machine Learning Algorithms

An efficient Machine Learning algorithm gives more accuracy. The prediction of heart patients is very critical, because a simple mistake can lead to death of a human being. This section consists of evaluating and selecting the most frequently used algorithms with high accuracy. As in the first section, we have summarized the most recent articles. We start with [6], The authors have implemented several learning machine algorithms, Logistic Regression has given an accuracy of 93%, Random Forest 92% and Gaussian Na¨ıve Bayes 90%, we gave notice that the results are close with simple progression of Logistic Regression. The authors of [1], tested the diagnosis of heart patients by applying two techniques: genetic algorithms and the KNN algorithm. The results gave satisfaction with the KNN algorithm. The weak point of this article is that the authors did not mention their results. Same remark for [5], The authors have used artificial neural networks now unfortunately, they have not presented their results. Priyanka.N et al [15], made a comparison between two algorithms (Na¨ıve bayes and KNN), the results show that KNN gives much more accuracy than na¨ıve bayes), it reaches 85% accuracy. The authors of [17], conducted a study on the prediction of heart disease, using only dataset, the algorithms used with their accuracy of SVM (99.3%), Neural Networks (91.1%), KNN (87.2%) and decision trees (82.3%). [18] The authors conducted a survey on the most used Machine Learning techniques that give more precision, Table 2 illustrates the algorithms cited in the articles with their accuracy. At the end of the analysis of these papers, we decided to choose the important attributes of the three classes[6], and the algorithms to implement are: Neural Networks (NN), KNN and SVM, to test our dataset which contains information on Algerian patients.

Table 2. The most frequently cited algorithms in the papers

Algorithms	Authors	Accuracy
Neural networks	[14] T John Peter et al., 2012 [3] Chaitrali S et al., 2012	78% 100%
KNN	[14] T John Peter et al., 2012 [8] C.Kalaiselvi 2016	75% 87%
SVM	[14] T John Peter et al., 2012 [3] Chaitrali S et al., 2012 [13] Shamsher Bahadur et al., 2013 [19] B.Venkata-lakshmi et al., 2014	76% 99% 99% 84%

3. Methodology

In this section we shall learn about the various classifiers used in machine learning to predict heart disease. We shall also explain our proposed methodology to improve the accuracy. The different methods used are defined below. The output is the accuracy metrics of the machine learning models. Then, the model can be used in prediction.

Dataset Description

The heart data set originated from Kaggle website, which contains medical information of patients, including their

1. age
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholestoral in mg/dl
6. fasting blood sugar > 120 mg/dl
7. resting electrocardiographic results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by flourosopy
13. thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

The names and social security numbers of the patients were recently removed from the database, replaced with dummy values.

We performed exploratory data analysis (EDA) on the dataset to gain insights into the data and visualize the distribution of the features. We then split the data into training and testing sets. Heart dataset containing 303 cases. The objective is to predict based on the measures to predict if the patient is diabetic or not.

Shape of dataset

```
In [4]: dataset.shape
Out[4]: (303, 14)
```

Printing out a few columns

```
In [5]: dataset.head(5)
Out[5]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

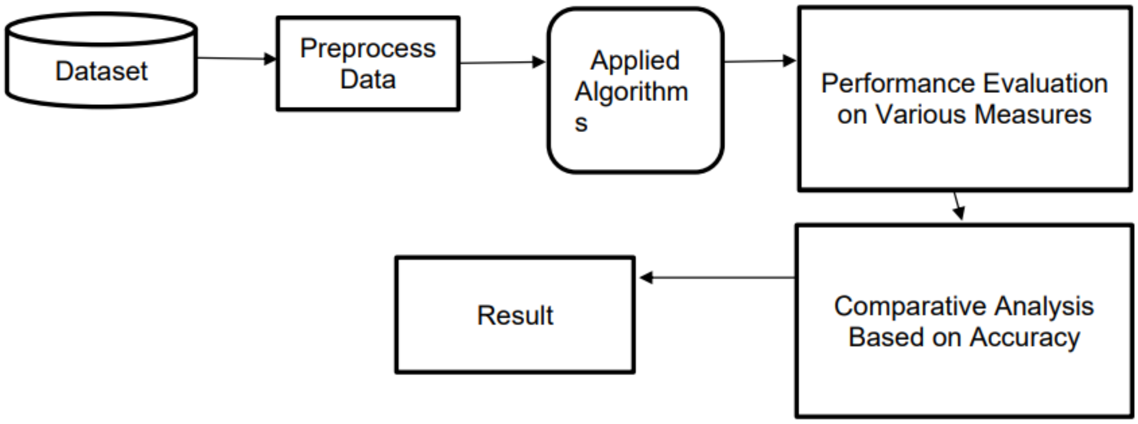
➔ The heart data set consists of 303 data points, with 14 features each.

➔ “target” is the feature we are going to predict, 0 means No disease, 1 means heart disease.

```
In [8]: dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

➔ There is no null values in dataset.

Flowchart:



Proposed Model Diagram

IV. RESULT & DISCUSSION

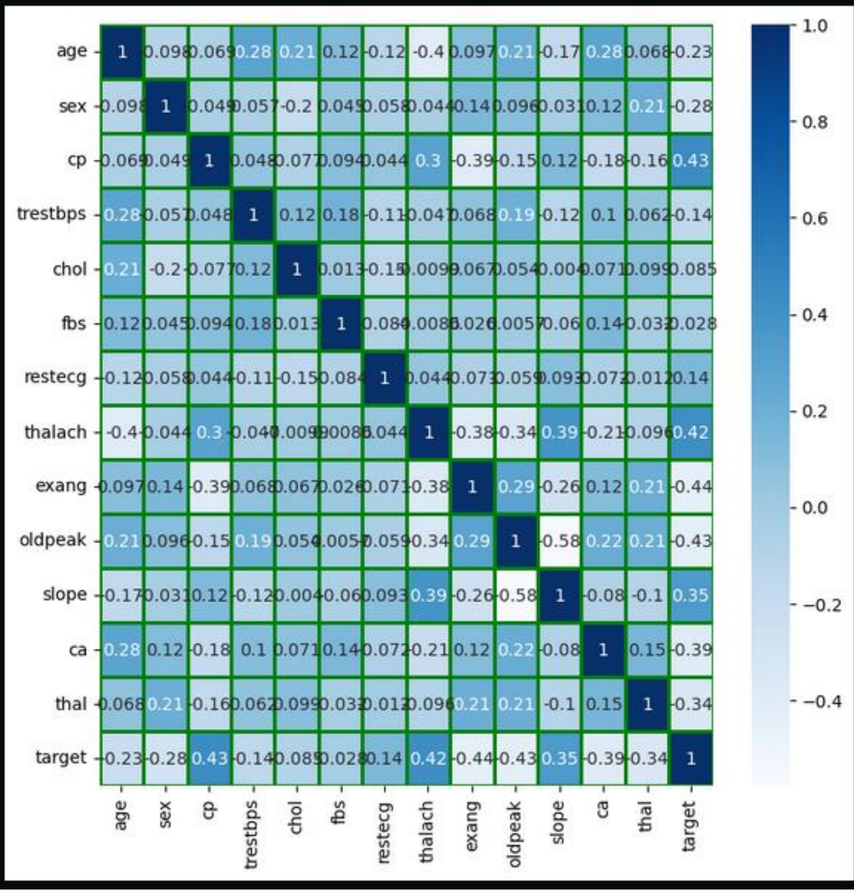
Correlation Matrix:

Checking correlation between columns

```
In [13]: print(dataset.corr()["target"].abs().sort_values(ascending=False))

target      1.000000
exang       0.436757
cp          0.433798
oldpeak     0.430696
thalach     0.421741
ca          0.391724
slope       0.345877
thal        0.344029
sex         0.280937
age         0.225439
trestbps    0.144931
restecg     0.137230
chol        0.085239
fbs         0.028046
Name: target, dtype: float64
```

Heatmap:

```
In [67]: plt.figure(figsize=[8,8])
         sns.heatmap(dataset.corr(),annot=True, cmap='Blues', linecolor='Green', linewidths=2.0)
         plt.show()
```

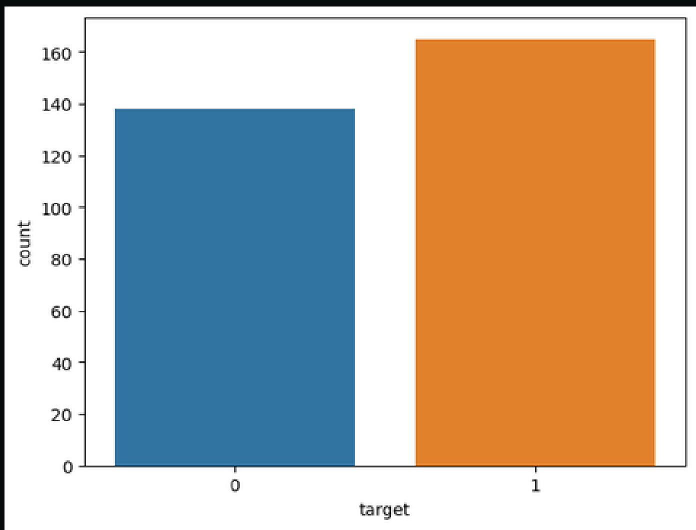


EDA for TARGET class:

Exploratory Data Analysis (EDA)

First, analysing the target variable:

```
In [15]: y = dataset["target"]

         sns.countplot(y)


         target_temp = dataset.target.value_counts()

         print(target_temp)

1    165
0    138
Name: target, dtype: int64
```




```
In [16]: print("Percentage of patience without heart problems: "+str(round(target_temp[0]*100/303,2)))
         print("Percentage of patience with heart problems: "+str(round(target_temp[1]*100/303,2)))

Percentage of patience without heart problems: 45.54
Percentage of patience with heart problems: 54.46
```

The above graph shows that the data is biased towards datapoints having target value as 0 where it means that heart disease was not present actually. The number of heart patients is almost same as the number of diabetic patients.

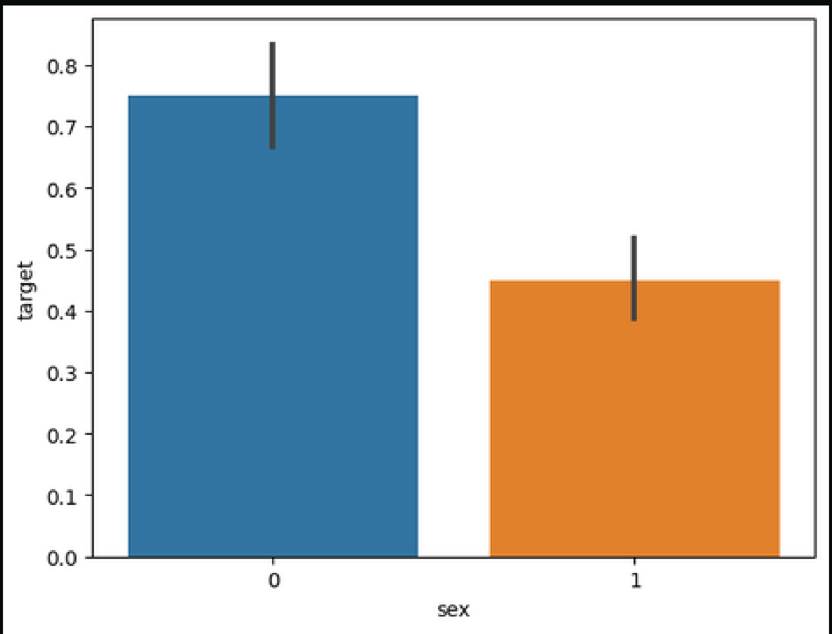
Analyzing the 'sex' feature:

Analysing the 'Sex' feature

```
In [17]: dataset["sex"].unique()
Out[17]: array([1, 0], dtype=int64)
```

We notice, that as expected, the 'sex' feature has 2 unique features

```
In [18]: sns.barplot(dataset["sex"],y)
Out[18]: <AxesSubplot:xlabel='sex', ylabel='target'>
```



We notice, that females are more likely to have heart problems than males

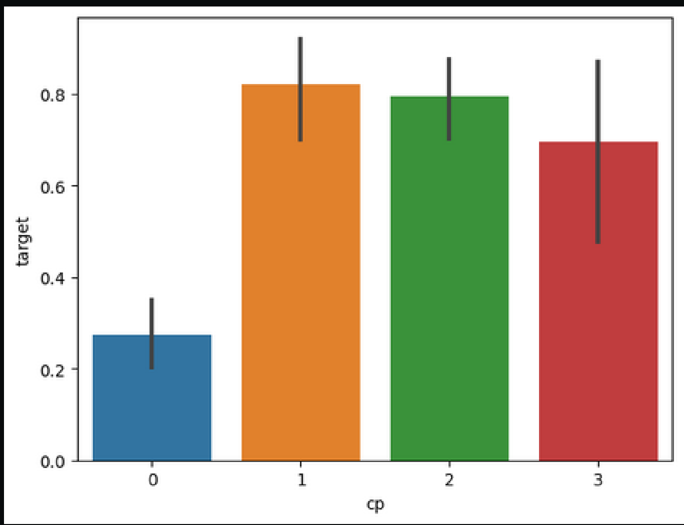
Analyzing the ‘chest pain type’ feature:

Analysing the 'Chest Pain Type' feature

```
In [19]: dataset["cp"].unique()
Out[19]: array([3, 2, 1, 0], dtype=int64)
```

As expected, the CP feature has values from 0 to 3

```
In [20]: sns.barplot(dataset["cp"],y)
Out[20]: <AxesSubplot:xlabel='cp', ylabel='target'>
```



We notice, that chest pain of '0', i.e. the ones with typical angina are much less likely to have heart problems

Analyzing the 'fbs' feature:

Analysing the FBS feature

```
In [21]: dataset["fbs"].describe()
```

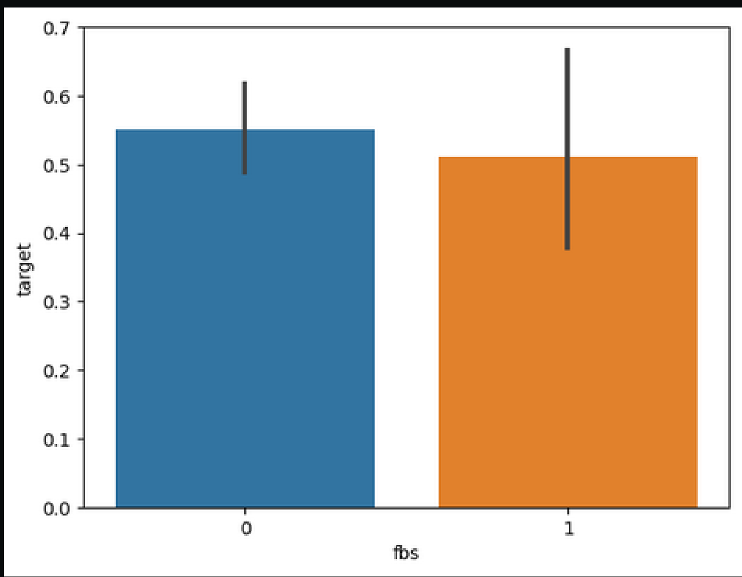
```
Out[21]: count    303.000000
         mean       0.148515
         std        0.356198
         min        0.000000
         25%        0.000000
         50%        0.000000
         75%        0.000000
         max        1.000000
         Name: fbs, dtype: float64
```

```
In [22]: dataset["fbs"].unique()
```

```
Out[22]: array([1, 0], dtype=int64)
```

```
In [23]: sns.barplot(dataset["fbs"],y)
```

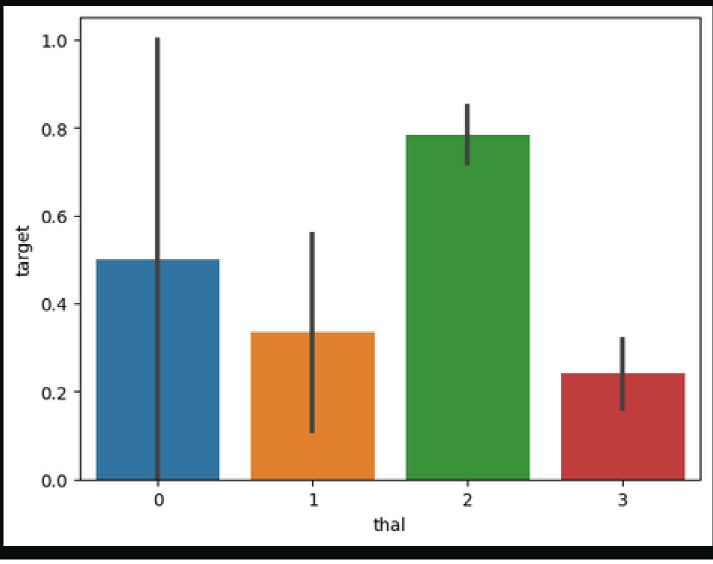
```
Out[23]: <AxesSubplot:xlabel='fbs', ylabel='target'>
```



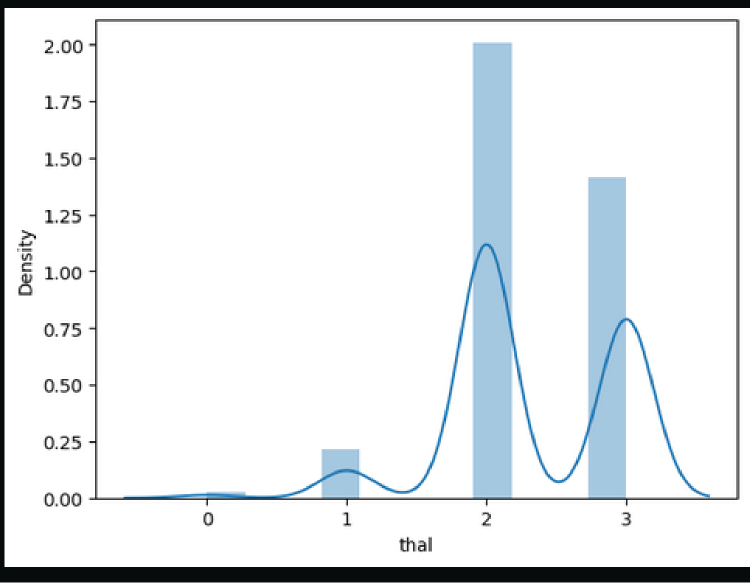
Nothing extraordinary here

Analyzing the ‘thal’ feature:

```
In [34]: ### Analysing the 'thal' feature

In [35]: dataset["thal"].unique()
Out[35]: array([1, 2, 3, 0], dtype=int64)

In [36]: sns.barplot(dataset["thal"],y)
Out[36]: <AxesSubplot:xlabel='thal', ylabel='target'>
```




```
In [37]: sns.distplot(dataset["thal"])
Out[37]: <AxesSubplot:xlabel='thal', ylabel='Density'>
```



Train Test Split (Training And Validation)

IV. Train Test split

```
In [38]: from sklearn.model_selection import train_test_split

         predictors = dataset.drop("target",axis=1)
         target = dataset["target"]

         X_train,X_test,Y_train,Y_test = train_test_split(predictors,target,test_size=0.20,random_state=0)

In [39]: X_train.shape
Out[39]: (242, 13)

In [40]: X_test.shape
Out[40]: (61, 13)

In [41]: Y_train.shape
Out[41]: (242,)

In [42]: Y_test.shape
Out[42]: (61,)
```

Model Fitting:

V. Model Fitting

```
In [43]: from sklearn.metrics import accuracy_score
```

Algorithms Used:

k-Nearest Neighbors:

The k-NN algorithm is arguably the simplest machine learning algorithm. Building the model consists only of storing the training data set. To make a prediction for a new data point, the algorithm finds the closest data points in the training data set, its "nearest neighbors."

K Nearest Neighbors

```
In [53]: from sklearn.neighbors import KNeighborsClassifier

         knn = KNeighborsClassifier(n_neighbors=7)
         knn.fit(X_train,Y_train)
         Y_pred_knn=knn.predict(X_test)

In [54]: Y_pred_knn.shape
Out[54]: (61,)

In [55]: score_knn = round(accuracy_score(Y_pred_knn,Y_test)*100,2)

         print("The accuracy score achieved using KNN is: "+str(score_knn)+" %")

         The accuracy score achieved using KNN is: 67.21 %
```

```
The accuracy score achieved using KNN is: 67.21 %
```

Logistic regression:

Logistic Regression is one of the most common classification algorithms.

Logistic Regression

```
In [44]: from sklearn.linear_model import LogisticRegression

         lr = LogisticRegression()

         lr.fit(X_train,Y_train)

         Y_pred_lr = lr.predict(X_test)

In [45]: Y_pred_lr.shape
Out[45]: (61,)

In [46]: score_lr = round(accuracy_score(Y_pred_lr,Y_test)*100,2)

         print("The accuracy score achieved using Logistic Regression is: "+str(score_lr)+" %")

         The accuracy score achieved using Logistic Regression is: 85.25 %
```

```
The accuracy score achieved using Logistic Regression is: 85.25 %
```

Decision Tree:

This classifier creates a decision tree based on which, it assigns the class values to each data point. Here, we can vary the maximum number of features to be considered while creating the model.

Decision Tree

```
In [56]: from sklearn.tree import DecisionTreeClassifier

max_accuracy = 0


for x in range(200):
    dt = DecisionTreeClassifier(random_state=x)
    dt.fit(X_train,Y_train)
    Y_pred_dt = dt.predict(X_test)
    current_accuracy = round(accuracy_score(Y_pred_dt,Y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x

#print(max_accuracy)
#print(best_x)


dt = DecisionTreeClassifier(random_state=best_x)
dt.fit(X_train,Y_train)
Y_pred_dt = dt.predict(X_test)
```

```
In [57]: print(Y_pred_dt.shape)

(61,)
```

```
In [58]: score_dt = round(accuracy_score(Y_pred_dt,Y_test)*100,2)

print("The accuracy score achieved using Decision Tree is: "+str(score_dt)+" %")

The accuracy score achieved using Decision Tree is: 81.97 %
```

`The accuracy score achieved using Decision Tree is: 81.97 %`

SVM:

Support Vector Machine is a supervised learning algorithm used for classification and regression analysis. It works by finding the hyperplane that maximally separates the different classes of data. The hyperplane is found by maximizing the margin between the support vectors, which are the data points closest to the decision boundary.

```
SVM

In [50]: from sklearn import svm

         sv = svm.SVC(kernel='linear')

         sv.fit(X_train, Y_train)

         Y_pred_svm = sv.predict(X_test)

In [51]: Y_pred_svm.shape

Out[51]: (61,)

In [52]: score_svm = round(accuracy_score(Y_pred_svm,Y_test)*100,2)

         print("The accuracy score achieved using Linear SVM is: "+str(score_svm)+" %")

         The accuracy score achieved using Linear SVM is: 81.97 %
```

```
The accuracy score achieved using Linear SVM is: 81.97 %
```

Naïve Bayes:

Naive Bayes is a probabilistic machine learning algorithm based on Bayes' theorem. It is used for classification and is popularly used for text classification and spam filtering.

```
Naive Bayes

In [47]: from sklearn.naive_bayes import GaussianNB

         nb = GaussianNB()

         nb.fit(X_train,Y_train)

         Y_pred_nb = nb.predict(X_test)

In [48]: Y_pred_nb.shape

Out[48]: (61,)

In [49]: score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)

         print("The accuracy score achieved using Naive Bayes is: "+str(score_nb)+" %")

         The accuracy score achieved using Naive Bayes is: 85.25 %
```

```
The accuracy score achieved using Naive Bayes is: 85.25 %
```

Random Forest:

This classifier takes the concept of decision trees to the next level. It creates a forest of trees where each tree is formed by a random selection of features from the total features.

Random Forest

```
In [61]: from sklearn.ensemble import RandomForestClassifier

max_accuracy = 0


for x in range(2000):
    rf = RandomForestClassifier(random_state=x)
    rf.fit(X_train,Y_train)
    Y_pred_rf = rf.predict(X_test)
    current_accuracy = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x

#print(max_accuracy)
#print(best_x)

rf = RandomForestClassifier(random_state=best_x)
rf.fit(X_train,Y_train)
Y_pred_rf = rf.predict(X_test)
```

```
In [62]: Y_pred_rf.shape

Out[62]: (61,)
```

```
In [63]: score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2)

print("The accuracy score achieved using Decision Tree is: "+str(score_rf)+" %")

The accuracy score achieved using Decision Tree is: 90.16 %
```

The accuracy score achieved using Decision Tree is: 90.16 %

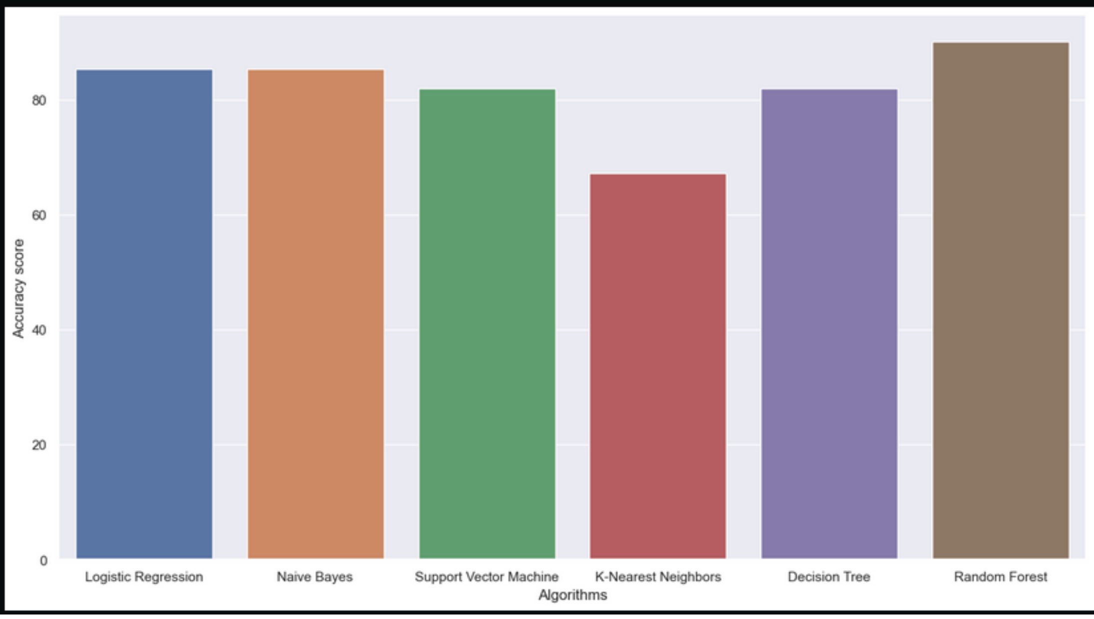
Accuracy Comparison:

VI. Output final score

```
scores = [score_lr,score_nb,score_svm,score_knn,score_dt,score_rf]
algorithms = ["Logistic Regression","Naive Bayes","Support Vector Machine","K-Nearest Neighbors","Decision Tree","Rand

for i in range(len(algorithms)):
    print("The accuracy score achieved using "+algorithms[i]+" is: "+str(scores[i])+" %")

The accuracy score achieved using Logistic Regression is: 85.25 %
The accuracy score achieved using Naive Bayes is: 85.25 %
The accuracy score achieved using Support Vector Machine is: 81.97 %
The accuracy score achieved using K-Nearest Neighbors is: 67.21 %
The accuracy score achieved using Decision Tree is: 81.97 %
The accuracy score achieved using Random Forest is: 90.16 %
```

Graph Comparing Different Algorithms:

```
In [62]: sns.set(rc={'figure.figsize':(15,8)})
         plt.xlabel("Algorithms")
         plt.ylabel("Accuracy score")

         sns.barplot(algorithms,scores)

Out[62]: <AxesSubplot:xlabel='Algorithms', ylabel='Accuracy score'>
```



This shows the accuracy values for all six machine learning algorithms.
It shows that Random forest algorithm gives the best accuracy with 90.16% accuracy.

4. Results and future scope

One of the important real-world medical problems is the detection of Heart Disease at its early stage. In this study, systematic efforts are made in designing a system which results in the prediction of heart disease. Based on the results, it seems that the Random Forest algorithm is the best performing model with an accuracy of 90.16%. Logistic Regression and Naive Bayes have similar performance with an accuracy of 85.25%, followed by Support Vector Machine and Decision Tree with 81.97%, and finally K-Nearest Neighbors with 67.21%.

To further improve the model, feature selection and engineering techniques can be applied to identify the most important features for predicting heart disease. This can lead to more accurate models with better generalization. Additionally, hyperparameter tuning can be performed to optimize the model performance. Furthermore, exploring other machine learning algorithms or ensemble methods like Gradient Boosting or Neural Networks can also improve the accuracy of the model. Finally, the model can be deployed as a web application or mobile application for easy access and use by healthcare professionals or individuals.

The future scope of a heart disease prediction system using machine learning is vast and promising. Some potential areas of development and advancement include:

1. Integration with wearable technology: With the increasing popularity of wearable devices such as smartwatches and fitness trackers, integrating a heart disease prediction system with these devices can provide real-time monitoring and early detection of heart disease. This can allow for timely intervention and preventive care, improving patient outcomes.

2. Use of advanced machine learning algorithms: While the current models used in heart disease prediction are effective, there is scope for further improvement using more advanced machine learning algorithms. For example, deep learning models such as neural networks can be used to capture more complex relationships between the features and the target variable, leading to more accurate predictions.
3. Personalized risk assessment: Machine learning algorithms can be used to develop personalized risk assessment models based on the individual's medical history, lifestyle, and other relevant information. This can provide more accurate risk estimates and tailored treatment recommendations.
4. Integration with electronic health records (EHRs): Integrating the heart disease prediction system with EHRs can provide a comprehensive view of the patient's medical history, allowing for more accurate risk assessment and treatment planning.
5. Mobile application development: Developing a mobile application for heart disease prediction can improve access to healthcare services, especially in rural or remote areas where healthcare facilities are limited. This can provide early detection and intervention for heart disease, reducing healthcare costs and improving patient outcomes.

Overall, the future of heart disease prediction using machine learning is promising, and there is ample scope for development and advancement. With further research and development, these systems can become an essential tool in preventive care and personalized treatment planning.

5. *References*

1. Babu, S., Vivek, E., Famina, K., Fida, K., Aswathi, P., Shanid, M., Hena, M.: Heart disease diagnosis using data mining technique. In: 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA). vol. 1, pp. 750–753. IEEE (2017)

2. Cai, J., Luo, J., Wang, S., Yang, S.: Feature selection in machine learning: A new perspective. Neurocomputing 300, 70–79 (2018)

3. Dangare, C.S., Apte, S.S.: Improved study of heart disease prediction system using data mining classification techniques. International Journal of Computer Applications 47(10), 44–48 (2012)

4. Fang, X., Hodge, B.M., Du, E., Zhang, N., Li, F.: Modelling wind power spatialtemporal correlation in multi-interval optimal power flow: A sparse correlation matrix approach. Applied energy 230, 531–539 (2018)

5. Gavhane, A., Kokkula, G., Pandya, I., Devadkar, K.: Prediction of heart disease using machine learning. In: 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA). pp. 1275–1278. IEEE (2018)

6. Hasan, S., Mamun, M., Uddin, M., Hossain, M.: Comparative analysis of classification approaches for heart disease prediction. In: 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2). pp. 1–4. IEEE (2018)

7. Jenzi, I., Priyanka, P., Alli, P.: A reliable classifier model using data mining approach for heart disease prediction. International Journal of Advanced Research in Computer Science and Software Engineering 3(3) (2013)

8. Kalaiselvi, C.: Diagnosing of heart diseases using average k-nearest neighbor algorithm of data mining. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). pp. 3099–3103. IEEE (2016)

9. Lopez, A.D., Mathers, C.D., Ezzati, M., Jamison, D.T., Murray, C.J.: Global and regional burden of disease and risk factors, 2001: systematic analysis of population health data. The Lancet 367(9524), 1747–1757 (2016)

10. Masethe, H.D., Masethe, M.A.: Prediction of heart disease using classification algorithms. In: Proceedings of the world Congress on Engineering and computer Science. vol. 2, pp. 22–24 (2014)

11. Organization, W.H.: The world health report 2002: reducing risks, promoting healthy life. World Health Organization (2016)

12. Organization, W.H., of Canada, P.H.A., of Canada, C.P.H.A.: Preventing chronic diseases: a vital investment. World Health Organization (2015)

13. Patel, S.B., Yadav, P.K., Shukla, D.: Predict the diagnosis of heart disease patients using classification mining techniques. IOSR Journal of Agriculture and Veterinary Science (IOSR-JAVS) 4(2), 61–64 (2013)

14. Peter, T.J., Somasundaram, K.: An empirical study on prediction of heart disease using classification data mining techniques. In: IEEE-International conference on advances in engineering, science and management (ICAESM-2012). pp. 514–518. IEEE (2012)

15. Priyanka, N., RaviKumar, P.: Usage of data mining techniques in predicting the heart diseases—na¨ıve bayes & decision tree. In: 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT). pp. 1–7. IEEE (2017)

16. Radhimeenakshi, S.: Classification and prediction of heart disease risk using data mining techniques of support vector machine and artificial neural network. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). pp. 3107–3111. IEEE (2016)

17. Raju, C., Philipsy, E., Chacko, S., Suresh, L.P., Rajan, S.D.: A survey on predicting heart disease using data mining techniques. In: 2018 Conference on Emerging Devices and Smart Systems (ICEDSS). pp. 253–255. IEEE (2018)

18. Shanmugasundaram, G., Selvam, V.M., Saravanan, R., Balaji, S.: An investigation of heart disease prediction techniques. In: 2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA). pp. 1–6. IEEE (2018)

19. Venkatalakshmi, B., Shivsankar, M.: Heart disease diagnosis using predictive data mining. International Journal of Innovative Research in Science, Engineering and Technology 3(3), 1873–7 (2014)

20. https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset