Prediction of Heart diseases using Machine Learning Classifiers

CSE 575 Statistical Machine Learning (Fall 2019)

Abhishek Muralikrishnan
1217200439
Arizona State University
Tempe Arizona USA
amuralik@asu.edu

Sai Teja Padakandla
1217167588
Arizona State University
Tempe Arizona USA
spadaka1@asu.edu

Ashmi Chheda
1217018972
Arizona State University
Tempe Arizona USA
achheda1@asu.edu

Abstract

Prediction of diseases in healthcare field is possible with the current trends and technology available. It can help thousands of doctors and patients with detecting and mitigating the effects of diseases. In this paper, we focus on heart disease prediction and have built models based on 5 classifier algorithms and compare the performance of the algorithms based on their accuracy of prediction of the disease. Furthermore, we built models for Epileptic seizure dataset also and compared the results to find out which algorithm is best suited for disease prediction.

Introduction

Data mining is the process of finding previously unknown patterns and trends in the large databases and using that information in order to build predictive models. In health care, data mining is becoming increasingly popular. Healthcare industry today generates large amount of complex data about patients, hospital resources, disease diagnosis, electronic patient records, medical devices, etc. The large amount of data is a key resource to be processed and analyzed for the knowledge extraction that enables support for cost-savings and decision making. Data mining provides a set of tools and techniques that can be applied to this processed data to discover hidden patterns and also provides healthcare professionals an additional source of knowledge for making decisions.

Problem Description

Health care field has a vast amount of data. For processing these data, certain data techniques are used. Heart disease is the leading cause of death worldwide. About 610k people die because of heart disease in the United States every year - that's one in every 4 deaths. Diagonzing the patients correctly in a timely basis is the most challenging task for medical fraternity. Thus, its an implicit necessity to predict the condition at the earliest. The objective of the project is to build classifiers to predict the possibility of a person getting cardiovascular disease. In this project, we are focusing on angiographic disease status in patients based on the diameter narrowing of arteries. We output a value of 0 if diameter narrowing is less than 50% and value of 1 if diameter narrowing is greater than 50%.

Keywords

Heart Disease, Machine Learning Algorithms, Classification Algorithms, Decision Trees, Naive Bayes, Kernel, Entropy, Ensemble learning, Epileptic Seizures.

Previous Work

Previous work we are referring to predicts the possibility of heart disease using two classifier algorithms namely Decision Tree and Naive Bayes algorithm and does the performance analysis of the results [1]. In this project, we have replicated the

previous work and also implemented 3 additional classification algorithms for heart disease prediction and compared the results and performance.

Methodology

In this project, we are implementing five classifier algorithms namely Naive Bayes, Decision Tree, Support Vector Machine, Random Forest and KNN (K Nearest Neighbours) classifiers. We ran these models on a Healthcare dataset. The data set used for predicting heart disease is taken from UCI Machine Learning Repository [6]. UCI is a collection of databases that are used for implementing machine learning algorithms. Specifically we are using data obtained from Cleveland Clinical Foundation. The overall database consists of data records of 297 individuals. The data consists of total 76 attributes, but we are using only 14 important attributes for our analysis as suggested by various Machine Learning researchers. Figure 1 shows the snippet of our data set.

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	output
63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
67	1	4	160	286	0	2	108	1	1.5	2	3	3	0
67	1	4	120	229	0	2	129	1	2.6	2	2	7	0
37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
41	0	2	130	204	0	2	172	0	1.4	1	0	3	0
56	1	2	120	236	0	0	178	0	0.8	1	0	3	0
62	0	4	140	268	0	2	160	0	3.6	3	2	3	1

Figure 1: Sample Data set

These are the 14 features we have selected for our data set.

1. **Age**: age in years
2. **Gender**: (1 = male; 0 = female)
3. **Cp**: chest pain type
4. **Trestbps**: resting blood pressure (in mm Hg on admission to the Hospital)
5. **Chol**: serum cholesterol in mg/dl
6. **Fbs**: fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
7. **Restecg**: 0 = normal; 1 = having ST-T wave abnormality; 2 = showing left ventricular hypertropy
8. **Thalach**: maximum heart rate achieved in beats per minute (bpm)
9. **Exang**: exercise induced angina (1 = yes; 0 = no)
10. **Oldpeak**: ST depression induced by exercise relative to rest
11. **Slope**: 1 = up-sloping; 2 = flat; 3 = down-sloping (Slope of the peak exercise ST segment)
12. **Ca**: number of major vessels (0-3)
13. **Thal**: 3 = normal; 6 = fixed defect; 7 = reversible defect
14. **Output**: 0 = Low chance; 1 = High Chance

We consider that these 14 features are the most essential for prediction of heart diseases. For instance, consider ‘age’ of a person. It is the most important risk factor in developing cardiovascular or heart diseases, with approximately a tripling of risk with each decade of life. Coronary fatty streaks can begin to form in adolescence. It is estimated that 82 percent of people who die of coronary heart disease

are 65 and older. Simultaneously, the risk of stroke doubles every decade after age 55.

Another example is angina (Chest pain) is discomfort caused when one's heart muscle doesn't get enough oxygen-rich blood. Angina pain may even feel like indigestion. Thus, this is one of the critical features to be taken into consideration.

Similarly, we considered the rest of the 12 features for building our model.The last column in our data set from Figure 1, is the classification column which contains two values, 0 representing low chance of getting heart disease and 1 representing having high chance of getting heart disease. We have divided our dataset into Training set (75%) and Testing set (25%). We have also trained another 5 models for the Epileptic seizures dataset in order to analyze the consistency in the performance of these 5 algorithms.

Tools Used

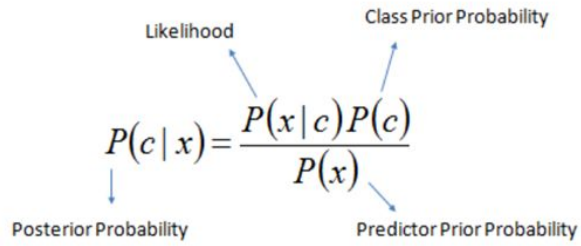
We used Python 3.6 along with Anaconda Spyder to build our models. We ran the models on Acer Predator Helios 300 Laptop. We chose Python as it is stable, flexible and has set of tools and packages easily available. Some of the packages used are NumPy, Keras and Scikit-learn.

Algorithms

We have implemented 5 classification algorithms for the prediction of heart disease on the Cleveland dataset.

1. Naive Bayes Classifier

Naive Bayes is among one of the simplest, but most powerful algorithms for classification. It is based on Bayes' Theorem. Naive Bayes is a classification algorithm for binary and multiclass classification problems. Rather than attempting to calculate the probabilities of each attribute value, they are assumed to be conditionally independent given the class value. It assumes that the presence of a feature in a class is unrelated to any other feature.



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure-2: Bayesian Probability

The entire process is broken down into 5 steps as follows:

1. Separate by class
2. Summarize Dataset
3. Summarize Data by class
4. Gaussian Probability Density Function
5. Evaluate Accuracy

For step 1, we calculate the probability of data by the class they belong to.

For step 2, we have calculated 2 statistics that is the mean and the standard deviation.

The mean is the average value and can be calculated as:

$$mean = sum(x)/n * count(x)$$

where x is the list of values or a column we are looking.

The sample standard deviation is calculated as the mean difference from the mean value. This can be calculated as:

$$Standard\ deviation = \sqrt{\sum_{i=1}^{N} \left(\frac{(x_i - \bar{x})^2}{N-1} \right)}$$

where $\bar{x}$is the mean of all values of x

For step 3, we separate the dataset into rows by class. The results in the form of a list of tuples of statistics are then stored in a dictionary by their class value.

For Step 4, calculating the probability or likelihood of a given real value like X1 is difficult. One way to do this is that assuming the values are drawn from a distribution like a bell curve or a Gaussian distribution.

The gaussian distribution can be calculated as:

$$f(x) = (1/sqrt(2 * PI) * sigma) * exp(-((x - mean)^2/(2 * sigma^2)))$$

For step 5, we calculate the probabilities separately for each class. If P(c = 0) > P(c = 1) then, the patient has low chances of getting heart disease, else vice versa.

Figure-3 shows the accuracy obtained for Naive Bayes.

```
Accuracy: 80.00%
Confusion matrix:
 [[ 8  4]
 [ 8 40]]
Classification Report:
             precision    recall  f1-score   support

High Chance       0.50      0.67      0.57        12
  No chance       0.91      0.83      0.87        48

avg / total       0.83      0.80      0.81        60
```

Figure-3: Naive Bayes results

2. Decision Tree Classifier :

Decision Tree classifier is the most efficient supervised machine learning algorithm that uses a tree-like graph or model of decisions and their possible consequences,including chance event outcomes,resource costs, and utility. It is an algorithm that contains conditional statements.

It is a flowchart-like structure in which each internal node represents a test and each branch represents the outcome of this test and each leaf node represents the class label. The path from root to leaf represent classification rules.

It is a tree based learning algorithm and considered to be one of the best and mostly used supervised learning methods. It empowers predictive models with high accuracy,stability. Unlike linear models, it maps nonlinear relationships quite well. Decision Tree algorithms are referred as CART(Classification and Regression Trees).

Information Gain:

Less impure node requires less information to describe it whereas more impure node requires more information. Information theory is a measure to define this degree of disorganization in a system known as Entropy. If the sample is completely homogeneous, then the entropy is zero and if the sample is equally divided, it has entropy of one.

Entropy can be calculated using the formula :

$$E(S) = \sum_{i=1}^{c} -p_i log_2 p_i$$

Where p is the probability

E(S) is the entropy

Entropy is also used with categorical target variable. It chooses the split which has the lowest entropy compared to parent node and other splits. The lesser the entropy the better it is.

Steps to calculate the entropy for a split:

1. Calculate the entropy of parent node.
2. Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.

We can derive information gain from entropy as **1-Entropy.**

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

A snippet of the entropy calculation is shown in FIgure-4.

Sl No	Attributes	Description	Number Of Patients	Has Disease	No Disease	Entropy
1	Age	<=50 >50	32 68	8 48	24 20	0.811 0.873
2	Gender	Male Female	51 49	39 12	12 37	0.781 0.796
3	Chest Pain	Less Pain More Pain	33 67	10 50	23 17	0.884 0.817
4	Trbps	>120 <=120	71 29	21 12	50 17	0.876 0.978
5	Cholesterol	High Normal	57 43	24 23	33 20	0.981 0.996
6	Heredity	Yes No	57 43	23 14	34 29	0.972 0.910
7	Fbs	Yes No	33 67	27 15	6 52	0.684 0.752
8	Smoking	Yes No	22 78	17 21	5 57	0.773 0.840
9	Thal	Normal High	52 48	12 32	40 16	0.77 0.917

Figure-4: Entropy Calculation for Cleveland dataset [8]

```
Accuracy: 86.67%
Confusion matrix:
 [[ 2  2]
 [ 4 37]]
Classification Report:
             precision    recall  f1-score   support

High Chance       0.33      0.50      0.40         4
  No chance       0.95      0.90      0.92        41

avg / total       0.89      0.87      0.88        45
```

Figure-5 : Decision Tree Results

3. K-Nearest Neighbour Classifier

The K-nearest neighbours (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. This algorithm assumes that similar things exists in closer proximity.

It captures the idea of similarity also called proximity considering the minkowski distance between the points.

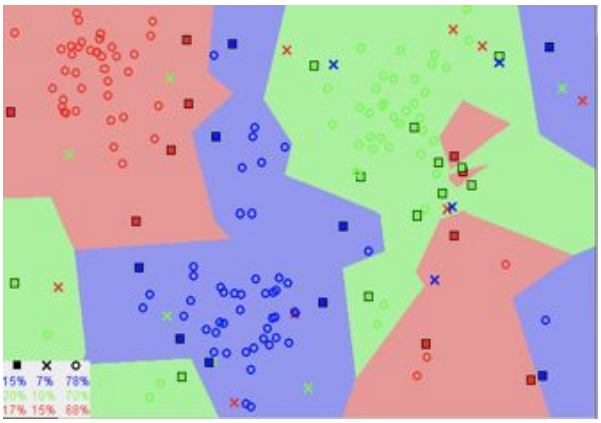


Figure-6: Sample KNN clusters

KNN Algorithm :

1. Load the data.
2. Initialize the K to your chosen number of neighbors.
3. For each sample in the data.

i) Calculate the distance between the query sample and the current sample from the data.

ii) Add the distance and the index of the example to an ordered collection.

4. Sort the ordered collection of distances and indices in ascending order by distances.
5. Pick the first K entries from the sorted collection.
6. Get the labels of the selected K entries.
7. Return the mode of the K labels.

Choosing the Hyperparameter K value:

In order to select the K that is right, we need to run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithms ability to accurately make predictions when it is given data it hasn't seen before. After set of trial and error experiments we ended up with a K value of 7.

```
Accuracy: 84.00%
Confusion matrix:
 [[ 0 11]
 [ 1 63]]
Classification Report:
             precision    recall  f1-score   support

High Chance       0.00      0.00      0.00        11
  No chance       0.85      0.98      0.91        64

avg / total       0.73      0.84      0.78        75
```

Figure-7 : KNN Results

4. Random Forest Classifier

Random Forest classifier is an ensemble algorithm that creates a set of decision trees from randomly selected subset of training set.It then considers the votes from different decision trees to decide the final class of the input.

This is an efficient algorithm because a single decision tree may be prone to noise, but the aggregate of many decision trees reduce the effect of noise giving more accurate results. In general, this

can also be implemented using weight concept where the decision tree output with high error rate are given the low weight value whereas the output with low error rate are given with higher weight value going with the greedy approach and ending up getting more accurate output.

General input parameters for this classifier is total number of trees to be generated and the decision tree parameters like minimum split,split criteria etc.

Random Forest Classifier output is not subject to Overfitting and output is less prone to noise and there is no correlation between the decision trees.

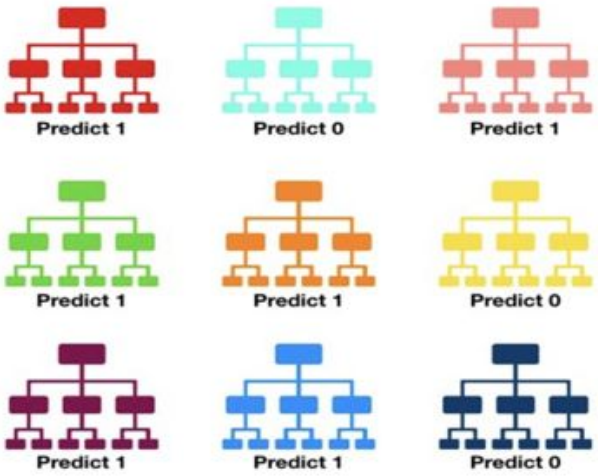


Figure-8: Random Forest intuition [9]

In the above Random Forest Classifier example, the output of 9 different decision trees are taken into consideration and the same input is fed into all the classifiers. Six Decision trees confirmed the output as 1 where as the three decision trees confirmed the output as 0. Based on the majority voting concept Random forest classifier ends up in choosing the final output as 1. The above example clearly defines the functionality of Random forest classifier in general.

```
Accuracy: 84.00%
Confusion matrix:
 [[ 8  9]
 [ 3 55]]
Classification Report:
             precision    recall  f1-score   support

High Chance       0.73      0.47      0.57        17
  No chance       0.86      0.95      0.90        58

avg / total       0.83      0.84      0.83        75
```

Figure-9: Random Forest results

5. Support Vector Machine:

Support Vector Machine is a discriminative classifier defined by a separating hyperplane. Given labelled training data the algorithm outputs an optimal hyperplane which categorizes new examples. In 2-D plane this hyperplane is a line dividing a plane in two parts where each class lay in either side. In scenarios, where we cannot draw a hyperplane to separate the data into clusters, we end up doing transformation and add one more dimension and do the operation. Once the operation is performed, we transform back this line to original plane. These transformations are called **Kernels.** As part of this project we have considered Radial Basis Function kernel (RBF) for building the model.

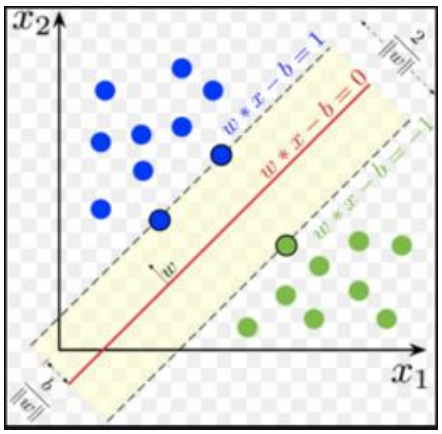


Figure-10: Sample Support Vector Machine

Margin:

A Margin is a separation of line to the closest class points. A good margin is one where this separation is larger for both the classes. A good margin allows the

points to be in their respective classes without crossing to other class.

```
Accuracy: 84.44%
Confusion matrix:
 [[ 6  2]
 [ 5 32]]
Classification Report:
             precision    recall  f1-score   support

High Chance       0.55      0.75      0.63         8
  No chance       0.94      0.86      0.90        37

avg / total       0.87      0.84      0.85        45
```

Figure-11: Support Vector Machine results

Interpretation of results

From our models, we can clearly see that Decision Tree Algorithm outperforms all the other algorithms with an accuracy of **86.687%**. The accuracy recorded for Naive Bayes, KNN,SVM and Random Forest algorithms are **80%, 84%, 84.44%** and **84%** respectively. Since, the training set was of comfortable size, the Decision tree algorithm was stable and didn't overfit the data. This has led to the high accuracy result for the algorithm.

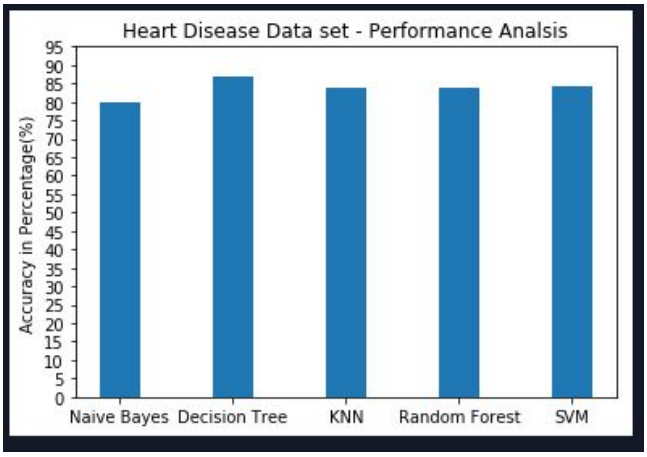


Figure-12: Performance and analysis of algorithms

Performance on the Epileptic Seizure Dataset

Epilepsy is the fourth most common neurological disorder in the world and it affects people of all ages. It is a spectrum condition with a wide range of seizure types and control and varies from person to person. Out of curiosity, we wanted to analyse the performance of the algorithms with the Epileptic Seizure Detection dataset taken from UCI Machine Learning Repository [7].

We observed the following results:

- Naive Bayes: **95.58%** accuracy
- Random Forest: **96.28%** accuracy
- kNN: **87.83%** accuracy
- SVM: **80.07%** accuracy
- Decision Tree: **91.30%** accuracy

More detailed overview of the results can be inferred from figures 13 - 17.

```
NAIVE BAYES ALGORITHM
Accuracy: 95.58%
Confusion matrix:
 [[ 537   59]
 [  68 2211]]
Classification Report:
             precision    recall  f1-score   support

Has seizures       0.89      0.90      0.89       596
  No seizure       0.97      0.97      0.97      2279

 avg / total       0.96      0.96      0.96      2875
```

Figure-13: Naive Bayes results on Epilepsy dataset

```
RANDOM FOREST ALGORITHM
Accuracy: 96.28%
Confusion matrix:
 [[ 537   49]
 [  58 2231]]
Classification Report:
             precision    recall  f1-score   support

Has seizures       0.90      0.92      0.91       586
  No seizure       0.98      0.97      0.98      2289

 avg / total       0.96      0.96      0.96      2875
```

Figure-14: Random forest results on Epilepsy dataset

```
KNN ALGORITHM
Accuracy: 87.83%
Confusion matrix:
 [[ 140  208]
 [   2 1375]]
Classification Report:
             precision    recall  f1-score   support

Has seizures       0.99      0.40      0.57       348
  No seizure       0.87      1.00      0.93      1377

 avg / total       0.89      0.88      0.86      1725
```

Figure-15: KNN results on Epilepsy dataset

```
SVM ALGORITHM
Accuracy: 80.07%
Confusion matrix:
 [[   0  573]
 [   0 2302]]
Classification Report:
              precision    recall  f1-score   support

Has seizures       0.00      0.00      0.00       573
  No seizure       0.80      1.00      0.89      2302

 avg / total       0.64      0.80      0.71      2875
```

Figure-16: SVM results on Epilepsy dataset

```
DECISION TREE ALGORITHM
Accuracy: 91.30%
Confusion matrix:
 [[ 400  161]
 [  89 2225]]
Classification Report:
              precision    recall  f1-score   support

Has seizures       0.82      0.71      0.76       561
  No seizure       0.93      0.96      0.95      2314

 avg / total       0.91      0.91      0.91      2875
```

Figure-17: Decision tree results on Epilepsy dataset

From the observations, it is clear that Random Forest achieved the best accuracy. This is because, Random Forest is an ensemble algorithm comprising of a collection of decision trees. This means that the majority of the results of the decision trees represents the accuracy of the Random Forest algorithm and given sufficient training data, the accuracy of each of the decision trees increases which thereby increases the accuracy of Random forest algorithm.

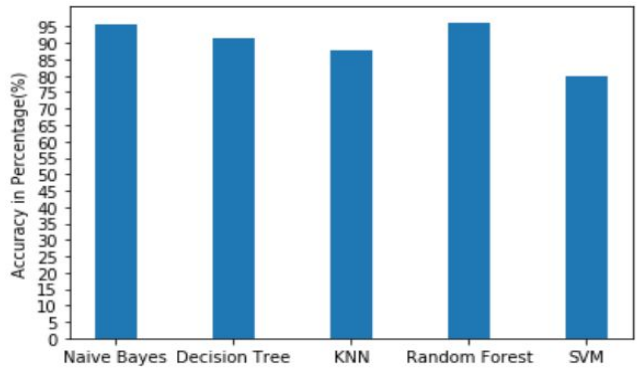


Figure-18: Performance and analysis of algorithms for Epilepsy dataset

Lessons Learnt

In the due course of the project, we have learnt the following lessons:

- With more data points and training data, better accuracy for the models can be achieved. As training data increases, the model can learn efficiently. However, we need to take care that we do not overfit the model.
- Performing Principal Component Analysis (PCA) for feature selection can yield better results. Not all the features contribute equally to the decision. By performing PCA and feature reduction, we can take into account only the essential and most vital features necessary for the model to correctly predict the data.
- Incase of Random Forest algorithm, the chances of predicting a correct label increases as the number of uncorrelated decision trees increases in the model.
- Curse of dimensionality problem: The performance of the classifier increases with the number of features until it reaches a maximum, after which the performance exponentially decreases with increase in the number of features taken into consideration for building the model.

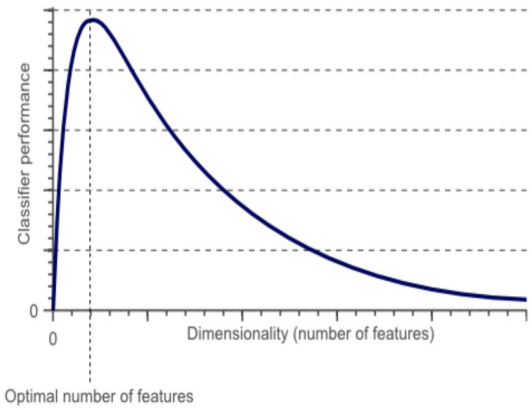


Figure-19: Curse of Dimensionality problem

Conclusions and Future work

In this project, we were able to build five Machine Learning Classifier algorithms - Naive Bayes, Decision Tree, Random Forest, KNN, Support Vector Machine to predict the possibilities of Heart and epilepsy diseases in humans. We were able to achieve a high accuracy of 86.67% using Decision Tree algorithm and 96.28% using Random Forest algorithm. We also conclude that if we have sufficient training data, we may find that Random Forest performs better than Decision Tree for classifying heart diseases. In the future, we can build more models using more classifier algorithms and compare the results.

References

[1] SanthanaKrishnan., J and S Geetha.. "Prediction of Heart Disease Using Machine Learning Algorithms." *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)* (2019): 1-5..

[2] Vijayashree, J.; SrimanNarayanaIyengar, N.C. Heart disease prediction system using data mining and hybrid intelligent techniques: A review. Int. J. Bio-Sci. Biotechnol. 2016, 8, 139–148.

[3] Gandhi, Monika, and Shailendra Narayan Singh. "Predictions in heart disease using techniques of data mining." In 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), pp. 520-525. IEEE, 2015.

[4] Chen, A. H., Huang, S. Y., Hong, P. S., Cheng, C. H., & Lin, E. J. (2011, September). HDPS: Heart disease prediction system. In 2011 Computing in Cardiology (pp. 557-560). IEEE.

[5] Aldallal, A., & Al-Moosa, A. A. A. (2018, September). Using Data Mining Techniques to Predict Diabetes and Heart Diseases. In 2018 4th International Conference on Frontiers of Signal Processing (ICFSP) (pp. 150-154). IEEE.

[6] UCI Machine Learning Repository - Heart Disease Dataset: https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[7] UCI Repository - Epileptic Seizure Recognition Dataset: https://archive.ics.uci.edu/ml/datasets/Epileptic+Seizure+Recognition

[8] "Prediction of Heart Disease Based on Decision Trees - IJRASET." https://www.ijraset.com/fileserve.php?FID=7821.

[9] Image taken from: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

https://www.kdnuggets.com/2016/01/implementing-your-own-knn-using-python.html