Heart Disease Classification using Decision Trees

Riken Patel
ECE 407: Pattern Recognition
University of Illinois at Chicago
rpate322@uic.edu
Jakub Marek
ECE 407: Pattern Recognition
University of Illinois at Chicago
jmarek6@uic.edu

***Abstract*— Heart Disease/Cardiovascular Disease (CVD) is one of the main causes of death around the world. In the United States CVD was found to be the leading cause of death among men and women across all ethnic groups. This is a huge problem, new methods for not just treating the illness but also for quick and early detection need to be developed to reduce the numbers of CVD deaths per year. In this paper, a classification model will be used to differentiate from a group of patients who have and do not have CVD. The classification model that will be used is called Decision Tree Classification, which is a model that focuses on nodes and splits to separate out the parameters into checkpoints that it runs the data through. Using this classifier individuals can run large amounts of data rather quickly based on the compact nature of Decision Trees. Using this method, doctors or patients can input their metrics that are required for the dataset parameters and this project will output with around 90% accuracy whether an individual has heart disease. This would be a valuable tool that can be used in the medical setting which can help doctors very quickly and efficiently reach conclusions about their patients regarding cardiovascular health.**

Keyword-Heart disease, CVD, decision tree classification, random forest classification, grid search

I. Introduction

Cardiovascular disease (CVD) is the number one cause of death globally, in 2016 an estimated amount of 17.9 million individuals died due to CVD [1]. Which represents 31% of all deaths that occurred in 2016, where 85% of them were due to a heart attack and stroke [1]. Clearly this is one of the greatest health concerns that we are currently facing in the medical field. CVD can be separated into 6 separate groups that fit under the umbrella as a CVD. These groups are as follows: coronary heart disease, cerebrovascular disease, peripheral arterial disease, rheumatic heart disease, congenital heart disease, and deep vein thrombosis/pulmonary embolism [1]. All these variants have to do with different locations of the body but carry a common theme of restricting blood supply to different parts of the body. The restriction of the blood in the various locations is due to damaged blood vessels or blood clot formations. This leads to strokes and heart attacks. Strokes and heart attacks are acute events that lead to temporary or permanent blockages within the brain or the heart [1]. The most common reason for these events is the buildup of fatty deposits on the inner walls of the blood vessels which supply blood to the brain or heart [1]. In addition to this, blood clots can lead to bleeding from blood vessels that can bleed into the brain causing strokes [1]. These events can lead to permanent damage to these vital organs or even death.

Once an individual has been determined to have a CVD or is in the high-risk category, they can expect a variety of symptoms. These symptoms include numbness on the body, chest pain, dizziness, and shortness of breath just to name a few [1]. Once these symptoms occur, damage begins in the heart or brain due to blockages leading to those adverse effects previously listed. That is why it is so crucial to create diagnostic techniques that can allow for early detection which can save those individuals that are at high risk. Therefore, in this paper, a method is proposed to use current medical metrics and a classification model to create a model which allows for the classification of individuals into positive and negative for having CVD using a dataset acquired from Kaggle [2]. In effect creating a diagnostics tool that can determine whether an individual is suffering from CVD or if they are a high-risk individual that may suffer from it in the future is crucial. The following metrics are included in this dataset: age, sex, chest pain type (4 values), resting blood pressure, serum cholesterol (mg/dl), fasting blood sugar (>120 mg/dl), resting electrocardiographic results (values 0,1,2), maximum heart rate achieved, exercise included angina, oldpeak (ST depression induced by exercise relative to rest), the slope of the peak exercise ST segment, number of major vessels (0-3) colored by fluoroscopy, thal (3 is normal, 6 is fixed defect, and 7 is reversible defect), and our truth table (target: 1 or 0) [2]. All the previously mentioned attributes are used in the medical field to help doctors determine the risk of individuals of having CVD, making them a great base to use in the diagnostics classification model proposed in this paper.

Age was found to be one of the most important factors. It was estimated that 82% of individuals that died from CVD were found to be of 65 years or older [9]. In addition to this the risk of individuals of developing CVD also increases with age making this one of the main markers for the development of

classification diagnostic tools. Furthermore, if an individual is a male, they have a higher propensity to develop CVD compared to females [9]. The next attribute is angina which develops when an individual develops chest pain or discomfort. An angina is discomfort which develops when there is restricted flow of blood through the heart muscles. Resting blood pressure that is high is a symptom of extra particles residing within the bloodstream. This can be due to individuals suffering from obesity, high cholesterol, and diabetes. This can result in the damage of blood vessels which can result in heart attacks and strokes [9]. Serum cholesterol, measured in mg/dl, is the measurement of cholesterol levels found within the body of the individuals. A high level of low-density lipoprotein (LDL) cholesterol tends to restrict the arteries of the heart leading to increased risk of a heart attack. While a high level of high-density lipoprotein (HDL) cholesterol lowers the risk of heart attacks [9]. Fasting blood sugar is a measurement of blood sugar gathered while an individual has not consumed anything for a while. This measurement gives a great reading on the insulin levels found within the body. Low levels within the body increase the risk of individuals having a heart attack. Resting ECG is a measurement of the pulse of the heart under normal conditions. Abnormal readings on the ECG waves show that an individual's heart is having trouble beating properly, which leads to the conclusion that that individual may be suffering from a CVD. Maximum heart rate was found to increase the stress put on the heart, leading to a cardiovascular risk [9]. The higher the maximum heart rate achieved, the higher the risk. In addition to that, if an angina occurred during the max heart rate test was conducted, that was noted under the exercise induced angina attribute. Lastly, the peak exercise ST segment is a note on the ECG wave abnormalities that are found during exercise. In general, these abnormalities include depressions in the peaks that are due to weakened blood vessels [9].

Machine learning and neural networks as classification methods have been used for a very long time and since the recent technology boom, the medical care field is a place that has huge amounts of data ripe for application. Some of the most promising techniques for the modeling of this data are naive Bayesian classifiers, neural networks, and symbolic learning [10]. The most promising under symbolic learning being decision trees and their ability to analyze medical data and conclude a diagnosis. One example is a decision tree model that was applied within the field of oncology focusing on the recurrence of different cancers found in patients [10]. This technique is proven in a multitude of applications spanning from the oncology example above to gynecology and several disease classifications. Due to its prevalence in the medical field and diagnostic prowess in a multitude of applications, this project used a decision tree method for the diagnosis of cardiovascular disease. In addition to this during the creation of the project, other classification techniques were used with reduced accuracy. To increase the accuracy further a Random Forest Classifier is used. This is a type of estimator that fits the total number of decision trees of the dataset into many sub-samples creating a forest , then it averages the predictive accuracy and controls the over-fitting of the data increasing the accuracy of the model.

II. METHOD DESCRIPTION

Dataset

The dataset that was utilized is "Heart Disease UCI" from Kaggle. This dataset contains 76 attributes. However, only 14 of the experimental subsets were utilized in this dataset, which consists of 303 individual's data . The following attributes are included in this dataset: age, sex, chest pain type (4 values), resting blood pressure, serum cholesterol (mg/dl), fasting blood sugar (>120 mg/dl), resting electrocardiographic results (values 0,1,2), maximum heart rate achieved, exercise induced angina, oldpeak (ST depression included by exercise relative to rest), the slope of the peak exercise ST segment, number of major vessels (0-3) colored by fluoroscopy, thal (3 is normal, 6 is fixed defect, and 7 is reversible defect), and our truth table (target: 1 or 0).

Chest pain displays what type of chest pain was experienced by the individuals using a 1-4 classification format. A typical angina is represented as a 1, an atypical angina is represented as a 2, non-anginal pain is represented as a 3 and lastly asymptomatic chest pain is represented by a 4 [9]. The next attribute that needs explaining is the resting ECG, which represents the resting ECG results of the individuals which ranges from 0-2. A normal signal is valued at a 0, ST-T wave abnormalities are valued at a 1 and left ventricular hypertrophy is valued as a 2 [9]. The last attribute that needs extra information is the thal attribute that displays the thalassemia. The values for this attribute are 3 which represent a normal reading, 6 for a fixed defect and a 6 for a reversible defect [9].

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
2	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
3	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
4	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
5	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
6	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
7	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
8	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
9	44	1	1	120	263	0	1	173	0	0	2	0	3	1
10	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
11	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
12	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
13	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
14	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
15	64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
16	58	0	3	150	283	1	0	162	0	1	2	0	2	1
17	50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
18	58	0	2	120	340	0	1	172	0	0	2	0	2	1
19	66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
20	43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
21	69	0	3	140	239	0	1	151	0	1.8	2	2	2	1

Figure 1: Data "heart.xlsx". First 21 samples of the total 303 samples.

Decision Tree Classification

The data was classified using a decision tree classifier algorithm. Decision trees are a hierarchical data structure that represents data by implementing a divide and conquer strategy [3]. It is a non-linear method of classification and regression that uses entropy and pdf-based learning. A decision tree classifies the data in a tree-like structure using the truth table, determining whether the question on hand is supposed to be true (1) or false (0). In this instance, the data was classified by whether a person were to have heart disease or not. The decision tree is drawn upside down. Therefore, the "root" of the decision tree is at the top, the splitting is considered the "branches", and the decision at the end is called the leaf [4]. The feature with

the highest information gain feature becomes the root of the decision tree. The entropy of a certain sub-feature is subtracted by the entropy of the whole feature. This is how information gain is calculated. Thus, the lower the entropy of the features, the higher the information gain. In the case of variance, the smallest variance feature becomes the root. As mentioned earlier, decision tree classification was utilized because it is a good prediction and classification model used for diagnostics.

This function includes the criteria of gini or entropy; splitter which is the strategy used to choose the split at each node; max depth which is the maximum depth of the tree; min samples which is the minimum samples required to split an internal node; min samples leaf which is the minimum number of samples required to be at a leaf node; min weight fraction leaf which is the minimum weighted fraction of the sum total of weights necessary for it to be a leaf node; max features which is the number of features necessary to consider when searching for the best split; random state, which controls the estimators randomness; max leaf nodes which determines the maximum number of leaf nodes to reduce impurity; min leaf nodes, which determines the reduction of the impurity greater than or equal to the set value; min impurity split, which is the threshold for early stopping in tree growth; and class weight, which is associated with the classes in the form [5]. Many of these attributes are default set unless otherwise specified within the function. As such, the set values determine the accuracy of the model once fitted and predicted properly.

Optimization

To optimize the decision tree to increase the accuracy of the model using the Heart Disease UCI dataset, Random Forest Classifier and GridsearchCV were used.

First, grid search was used to find the best parameters for the decision tree classifier. GridsearchCV utilizes a fit and score method to search for the best parameter values for an estimator. Some of the parameters within this function include estimator, param grid, scoring, refit, error score, and return train score. The estimator provides the model with the score function for the scikit-learn estimator interface. Parameter grid is a dictionary of names (strings) that uses keys and lists of parameter settings to try as values enabling the search of any sequence of parameter settings [6]. Scoring is one of the more important attributes as it evaluates the performance of the cross-validated model on the test set. For this dataset, AUC (area under the curve) and accuracy score are used for scoring. Accuracy score computes the subset accuracy of the truth labels and predicted labels [7]. Refit is used to find the best parameters for an estimator on the whole dataset. Error score assigns a value to the score in case there is an error in fitting the estimator. Return train score is a Boolean that is used to get information on the different parameters overfit or underfit the data [6]. The best parameters using grid search on Decision Tree Classifier were the criterion set to entropy, maximum depth and maximum leaf nodes set to 5, and minimum samples split set to a value of 2.

```
In [60]: runfile('C:/Users/Owner/Documents/UIC/ECE 407/Project/
407project.py', wdir='C:/Users/Owner/Documents/UIC/ECE 407/Project')
Best params for DecisionTreeClassifier: {'criterion': 'entropy',
'max_depth': 5, 'max_leaf_nodes': 5, 'min_samples_split': 2}
Accuracy of Decision Tree Classifier: 83.60655737704919 %
```

Figure 2: Best parameters for Decision Tree Classifier

Random forest classifier is a type of estimator that fits the total number of decision trees of the dataset into many sub-samples, then it averages the predictive accuracy and controls the over-fitting of the data. Some of the parameters that are important for the Random Forest Classifier function are number of estimators, criterion, max depth, max samples, min samples split, min samples leaf, max features, max leaf nodes, min impurity decrease, min impurity split, bootstrap, and random state [8]. Number of estimators is the number of decision trees in the forest. The criterion is either gini (gini impurity) or entropy (information gain) which is the function to measure the quality of a split. Max depth is the maximum depth of the decision tree. Bootstrap helps reduce the dataset that is needed to build each tree. Max samples rely on bootstrap; if bootstrap is true, the number of samples to draw from X to train each base estimator [8]. Min samples split is the minimum number of samples that are needed to split at an internal node. Min samples leaf is the minimum number of samples needed to be at a leaf node. Max features the maximum number of features to consider deciding the best split. Max leaf nodes is the maximum number of leaf nodes that help determine the relative reduction in impurity. Min impurity decreases if it makes a node split if that split allows a decrease of the impurity greater than or equal to that value [8]. Min impurity split is the threshold to stop the tree growth. Random state controls the randomness of the bootstrapping of the samples when the decision trees are being built. It also considers the randomness of the sampling of the features when searching for the best split at the nodes [8]. Grid search was also used to obtain the best parameters for Random Forest Classifier. The best parameters include a max depth of 10, the number of estimators to 20, and random state set to 12.

```
In [67]: runfile('C:/Users/Owner/Documents/UIC/ECE 407/Project/
407project.py', wdir='C:/Users/Owner/Documents/UIC/ECE 407/Project')
Best params for RandomForestClassifier: {'max_depth': 10,
'n_estimators': 20, 'random_state': 12}
Accuracy of Random Forest Classifier: 90.1639344262295 %
```

Figure 3: Best parameters for Random Forest Classifier

Programming process

Python 3.0 was used to classify the "Heart Disease UCI" dataset. We also utilized the libraries of NumPy, Pandas, Scikit learn (Sklearn), and Matplotlib to program the algorithm and to plot the decision tree.

First, the excel file of the dataset was read into our IDE, Visual Studio Code. Then, the dataset was separated into two different arrays, labels which are the "target", and the other array are the features that are compared to the labels to construct the decision tree. Decision Tree Classifier was used within the Sklearn library to classify the data. GridsearchCV was utilized to optimize the parameters in Decision Tree

Classifier to obtain a higher accuracy of the model. Overall, the accuracy of Decision Tree Classifier led to a subpar accuracy than what was expected. This may have been due to the lack of data given in the dataset. Therefore, random forest classifiers were used to optimize the decision tree classifier. Grid search was also used to optimize Random Forest Classifier, and that gave us a more desirable outcome for the accuracy of the model.

III. Experimental results

The accuracy was lower than expected with the Decision Tree Classifier. Therefore, GridsearchCV function from Sklearn library was utilized to find the best parameters for the Decision Tree Classifier. Again, an accuracy of 83.6% was the highest that could be achieved. Figure 4 displays the decision tree that was obtained using Decision Tree Classifier. Thus, the Random Forest Classifier function, also from the Sklearn library, was utilized to build multiple decision trees and to predict which decision tree would give the best parameters. GridsearchCV was also applied to Random Forest Classifier to figure out the best parameters for the function to increase the accuracy of the model. The best parameters we obtained for Random Forest Classifier included a max depth of 10, number of trees was set to 20, and random state set to 12. The random forest classification model achieved an accuracy of 90.16%. The decision tree for random forest classifiers is shown in figure 5.

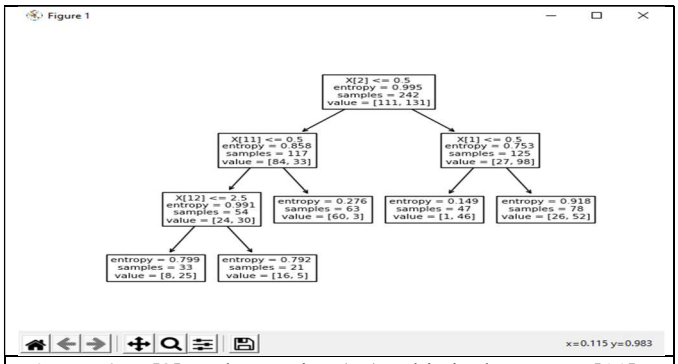

Figure 4: X[2] = chest pains (cp). This is the root. X[11] = number of major vessels (ca). X[1] = sex. X[12] = thal.

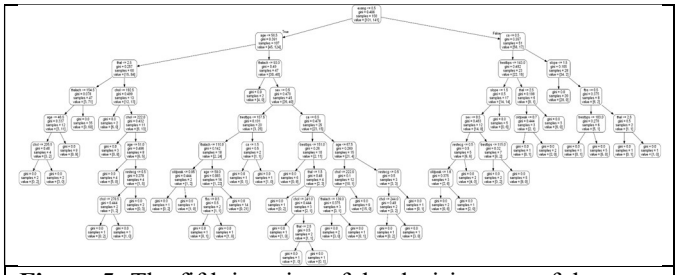

Figure 5: The fifth iteration of the decision tree of the Random Forest Classifier.

IV. Conclusion

Cardiovascular disease is ever present in our society and methods of early detection and diagnosis are extremely important to reduce those high death numbers. In this paper a method was proposed, with 14 patient metrics, that can help doctors and individuals diagnose cardiovascular disease with an accuracy of 90.16% with only 303 samples. The method used was decision trees and a random forest classifier. As this data set grows with the addition of new patient samples the accuracy will grow further, increasing the effectiveness of the model at estimating the risk of individuals. Classification methods like this need to be added more frequently into the field of medicine to aid doctors in their diagnosis of patients. Methods like the one proposed in the paper provide an easy method for doctors to check their diagnosis on patients reducing doctor error. In addition to a model like this, if posted online, the public can give an avenue for people that have not seen a doctor, to check whether they may be at high risk for CVD allowing them time to seek medical attention. In effect leading to a more educated general population and more efficient/accurate doctor diagnosis.

V. CONTRIBUTION OF EACH MEMBER

Riken and Jakub worked on the code together. Jakub wrote the abstract, introduction, and conclusion of the paper. Riken wrote the method description and experimental results. We both worked on the presentation and video together focusing on similar components as the paper. We helped each other out to complete the project, 50:50 work split as a group.

VI. References

[1] "Cardiovascular diseases (CVDs)," *World Health Organization*, 17-May-2017.[Online].Available:https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds). Apr.,19, 2021.

[2] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano. *Heart Disease Data Set*, Cleveland, Hungary, Switzerland, and the VA Long Beach: Center for Machine Learning and Intelligent Systems, 1988: Apr.,19, 2021. Available: https://www.kaggle.com/ronitf/heart-disease-uci.

[3] A. Cetin, Class Lecture, Topic: "Decision Trees." ECE 407, College of Engineering, University of Illinois at Chicago, Apr.,19, 2021.

[4] P. Gupta, "Decision Trees in Machine Learning," *Towards Data Science*, 2017:Apr.,19, 2021. Available: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052.

[5] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E., "Scikit-learn: sklearn.tree.DecisionTreeClassifier," *Journal of Machine Learning Research*., vol. 12, pp. 2825-2830, 2011. Available:
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[6] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E., "Scikit-learn: sklearn.tree.GridSearchCV," *Journal of Machine Learning Research*., vol. 12, pp. 2825-2830, 2011. Available:
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[7] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E., "Scikit-learn: sklearn.metrics.accuracy_score," Journal of Machine Learning Research., vol. 12, pp. 2825-2830, 2011. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

[8] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E., "Scikit-learn: sklearn.tree.RandomForestClassifier," *Journal of Machine Learning Research*., vol. 12, pp. 2825-2830, 2011. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[9] S. Rawat, "Heart Disease Prediction,"*Towards Data Science*, 2019: Apr.,19, 2021. Available: https://towardsdatascience.com/heart-disease-prediction-73468d630cfc

[10] Igor Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective", *Artificial Intelligence in Medicine*, Volume 23, Issue 1, 2001, Pages 89-109, ISSN 0933-3657, https://doi.org/10.1016/S0933-3657(01)00077-X.
(https://www.sciencedirect.com/science/article/pii/S093336570100077X)

VII. Appendix: code list

```
"""
Authors: Riken Patel & Jakub Marek
ECE 407 Project
"""

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import make_scorer
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import export_graphviz
import pydot

## Loading in the data from excel spreadsheet for heart disease UCI:
data = pd.read_excel('heart.xlsx')
attributes = ['age','sex','cp','trestbps','chol','fbs','restecg','thalach','exang','oldpeak','slope','ca','thal']
labels = np.array(data['target'])
x = data[['age','sex','cp','trestbps','chol','fbs','restecg','thalach','exang','oldpeak','slope','ca','thal']]

#Sort data into test and train samples
x_train, x_test, labels_train, labels_test = train_test_split(x, labels, test_size = 0.2, random_state=0)


## Obtaining parameters (GridSearchCV) DecisionTreeClassifier
clf_df = DecisionTreeClassifier()
df_parameters = {'criterion': ['entropy','gini'],'max_depth': [5,10,None],
                 'max_leaf_nodes': [2,5],'min_samples_split': [2,5,10]}
scorer_df  = {'AUC': 'roc_auc', 'Accuracy': make_scorer(accuracy_score)}
clf_GS_df = GridSearchCV(clf_df, df_parameters,scoring=scorer_df, refit='Accuracy', return_train_score=True)
grid_obj = clf_GS_df.fit(x_train, labels_train)
best_params_df = grid_obj.best_params_
print(f"Best params for DecisionTreeClassifier: {best_params_df}")

## Applying the parameters into DecisionTreeClassifier
clf_best_df = grid_obj.best_estimator_
df_fit = clf_best_df.fit(np.array(x_train),labels_train)
pred_df = clf_best_df.predict(np.array(x_test))
accuracy = accuracy_score(labels_test, pred_df)
print("Accuracy of Decision Tree Classifier:",accuracy*100, '%')
tree.plot_tree(df_fit)
plt.show()


## Obtaining parameters (GridSearchCV) RandomForestClassifier
clf_rf = RandomForestClassifier()
rf_parameters = {'n_estimators': [10,15,20],
          'max_depth': [10,None],'random_state': [12,5,None]}
scorer_rf  = {'AUC': 'roc_auc', 'Accuracy': make_scorer(accuracy_score)}
clf_GS_rf = GridSearchCV(clf_rf, rf_parameters,scoring=scorer_rf,refit='Accuracy', return_train_score=True)
grid_obj = clf_GS_rf.fit(x_train, labels_train)
best_params = grid_obj.best_params_
```

```
print(f"Best params for RandomForestClassifier: {best_params}")

## Applying the parameters into RandomForestClassifier
clf_best_rf = grid_obj.best_estimator_
rf_fit = clf_best_rf.fit(np.array(x_train),labels_train)
pred_rf = clf_best_rf.predict(np.array(x_test))
accuracy = accuracy_score(labels_test, pred_rf)
print("Accuracy of Random Forest Classifier:",accuracy*100, '%')

##Plotting the Random Forest Decision Tree using Graphviz
tree_layer = clf_best_rf.estimators_[5]
# Export the image to a dot file
export_graphviz(tree_layer, out_file = 'tree.dot', feature_names = attributes, rounded = True)
# Use dot file to create a graph
(graph, ) = pydot.graph_from_dot_file('tree.dot')
# Write graph to a png file
graph.write_png('tree.png')
```