Decision Tree Classification Experiment Report

Manikanta Kalyan Gokavarapu
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260
mgokavar@buffalo.edu

1 Decision Tree Classification

A Decision tree is a binary tree that recursively splits the dataset until we are left with pure leaf nodes that is the data with only one type of class. some basic terminology in decision tree is a below.

- Root Node: It can be found at the start of a decision tree. It represents the whole sample of data that is split into two sets under the specified condition.
- Splitting: splitting a node into two or more smaller nodes
- Decision node: the node we obtain following the division of the root nodes.
- Leaf node: Leaf nodes are nodes in which further splitting is not feasible.
- Sub-Tree: The sub-section of this decision tree.
- Parent and Child Node: A node that has sub-nodes is referred to as the parent node of sub-nodes, while sub-nodes are the children of the parent node.

The basic working of decision tree is as follows, the data is feed into root node of a decision tree and based on a specific splitting condition the data is split into two parts like for example data points satisfying a specific split condition goes into one node and which doesn't satisfy the condition goes to other node and if we get a node with a mix of both types of classes data we iterate further down the decision tree until we get a leaf node with only one type of class data. But if we have more complex data, we cannot achieve 100% pure leaf nodes. In those cases, we opt for "majority voting" and we will assign the majority class of the data points to the test point.

As there would be many possible splitting conditions our model needs to learn which features to choose and the corresponding correct threshold values to optimally split the data. The optimal split condition is chosen based on a goal of achieving pure leaf nodes. So, to achieve the split conditions, the model will choose the split that maximizes the information gain. To calculate the information gain we need to use entropy. Entropy is a measure of randomness of a certain point in the given set of data points or it's measure of information contained in a state. The formula for Entropy is given as below.

$$Entropy = \sum -p_k \log(p_k)$$

$$p_k = probability\ of\ class\ k$$

A Pure node is identified if it has minimum entropy value. The Information gain is given by combined Entropy of the parent nodes minus combined entropy of child nodes.

$$Information\ gain = Entropy\ of\ parent\ nodes - \sum w_k\ Entropy\ (child\ \ nodes_k$$

So, this IG value is used for choosing the split conditions in a decision tree. Priority is given to the split condition with high Information gain. The model iterates through all the possible split conditions and finds the best condition using information gain for classifying the input data.

2 Experiment

For the Experiment purposes I have taken a dataset which contain 14 attributes. The attributes data is as follows.

- Age - age in years
- Sex -1=male, 0=female
- Cp - chest pain type
- Trestbps - resting blood pressure (in mm Hg on admission to the hospital)
- Chol - serum cholesterol in mg/dl
- Fbs - (fasting blood sugar > 120 mg/dl) (1=true; 0=false)
- Restecg – resting electrocardiographic results
- Thalach – maximum heart rate achieved.
- Exang – exercise induced angina (1=Yes ; 0=No)
- Oldpeak – ST depression induced by exercise
- Slope – the slope of the peak exercise ST segment
- Ca – number of major vessels (0-3) colored by flourosopy
- Thal – 3 = normal; 6= fixed defect; 7= reversable defect
- Target - have disease or not (1=yes, 0=no)

The first five rows of the dataset are given in the figure [4.1]

	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

Figure [4.1]

And the attribute information such as count, mean, std, min of the dataset are given in the figure [4.2].

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000

thal	target
303.000000	303.000000
2.313531	0.544554
0.612277	0.498835
0.000000	0.000000
2.000000	0.000000
2.000000	1.000000
3.000000	1.000000
3.000000	1.000000

Figure [4.2]

So, As I want to build a Decision tree classification model to classify the people who will have heart disease or not (target) using some input features. First, I need to choose the input parameters or features to classify. Based on the correlation matrix output I have chosen all the features as all the features are co-related with the target attribute to classify whether a person has heart disease or not. The correlation matrix output is given the below figure [4.3]

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522	0.096801	0.210013	-0.168814	0.276326	0.068001	-0.225439
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020	0.141664	0.096093	-0.030711	0.118261	0.210041	-0.280937
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762	-0.394280	-0.149230	0.119717	-0.181053	-0.161736	0.433798
restbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698	0.067616	0.193216	-0.121475	0.101389	0.062210	-0.144931
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940	0.067023	0.053952	-0.004038	0.070511	0.098803	-0.085239
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567	0.025665	0.005747	-0.059894	0.137979	-0.032019	-0.028046
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123	-0.070733	-0.058770	0.093045	-0.072042	-0.011981	0.137230
thalach	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000	-0.378812	-0.344187	0.386784	-0.213177	-0.096439	0.421741
exang	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812	1.000000	0.288223	-0.257748	0.115739	0.206754	-0.436757
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187	0.288223	1.000000	-0.577537	0.222682	0.210244	-0.430696
slope	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784	-0.257748	-0.577537	1.000000	-0.080155	-0.104764	0.345877
ca	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177	0.115739	0.222682	-0.080155	1.000000	0.151832	-0.391724
thal	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439	0.206754	0.210244	-0.104764	0.151832	1.000000	-0.344029
target	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741	-0.436757	-0.430696	0.345877	-0.391724	-0.344029	1.000000

Figure [4.3]

To build the decision tree classification model, I have split the data into two parts in which 80% I used for train data and remaining 20% I used for test data. The test results are in below figure [4.4] plotted via confusion matrix and the true negative values are 22 and true positive are 27 values.

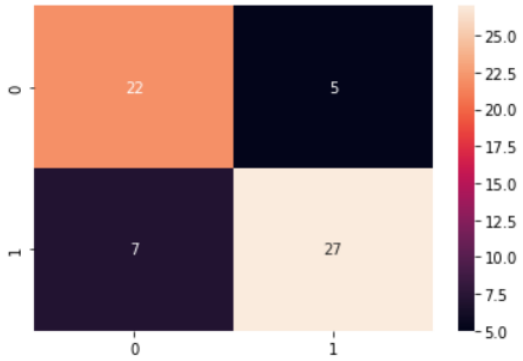

Figure [4.4]

The decision tree plotted by the model to classify using split conditions of the features is given in below figure [4.5]

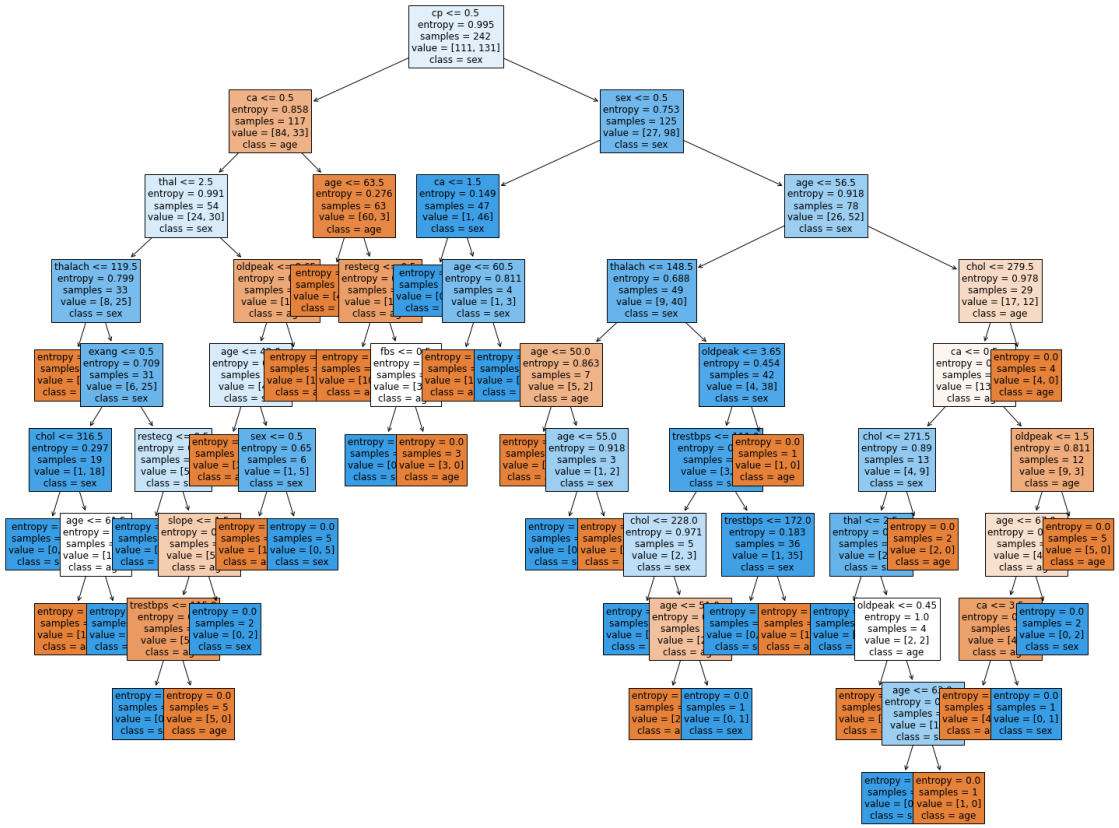

Figure [4.5]

The overall accuracy or classifier score of the model is given as 80%. And remaining results of model like Roc_Auc score, precision score, recall score, F1 score are given in the below figure [4.6].

```
print ('Classifier',classifier.score(X_test, y_test))
print('roc_auc: ', roc_auc_score(y_test, y_pred))
print('precision: ', precision_score(y_test, y_pred))
print('recall: ', recall_score(y_test, y_pred))
print('f1: ', f1_score(y_test, y_pred))
```

```
Classifier 0.8032786885245902
roc_auc:  0.8044662309368191
precision:  0.84375
recall:  0.7941176470588235
f1:  0.8181818181818182
```

Figure [4.6]

References

[1] Devanshi Srivastava, Programs Buzz, Decision Tree: Introduction, 06/04/2021

[2] Charan kakararaparthi, Heart_Diseases, Prediction of Heart Disease by Effective Machine Learning techniques.