CS60050 : Machine Learning

Report
ASSIGNMENT - 1 [E]

Prepared by:
Group No. 51
Shristi Singh (19CS10057)
Mayank Kumar (19CS30029)

September 19, 2021

I. Analysis of the dataset

Shape of dataset: (270, 14)

Basic details about dataset:

The dataset has 13 attributes that decide the target, i.e. absence (1) or presence (2) of heart disease.

- Age (age)
- Sex (sex)
- Chest Pain Type (chest_pain)
- Resting Blood Pressure (resting_bp)
- Serum Cholesterol in mg/dl (serum_chol)
- Fasting blood sugar > 120 mg/dl (blood_sugar)
- Resting electrocardiographic results (rest_ecg)
- Maximum heart rate achieved (max_heart_rate)
- Exercise induced angina (induced_ang)
- Oldpeak = ST depression (oldpeak)
- Slope of peak exercise ST segment (peak_st_seg)
- Number of major vessels (major_vessels)
- Thal (thal)

Overview of the dataset:

The target class label is an integer, and no value is missing.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 270 entries, 0 to 269
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   age             270 non-null    float64
 1   sex             270 non-null    float64
 2   chest_pain      270 non-null    float64
 3   resting_bp      270 non-null    float64
 4   serum_chol      270 non-null    float64
 5   blood_sugar     270 non-null    float64
 6   rest_ecg        270 non-null    float64
 7   max_heart_rate  270 non-null    float64
 8   induced_ang     270 non-null    float64
 9   oldpeak         270 non-null    float64
 10  peak_st_seg     270 non-null    float64
 11  major_vessels   270 non-null    float64
 12  thal            270 non-null    float64
 13  target          270 non-null    int64  
dtypes: float64(13), int64(1)
memory usage: 29.7 KB
```

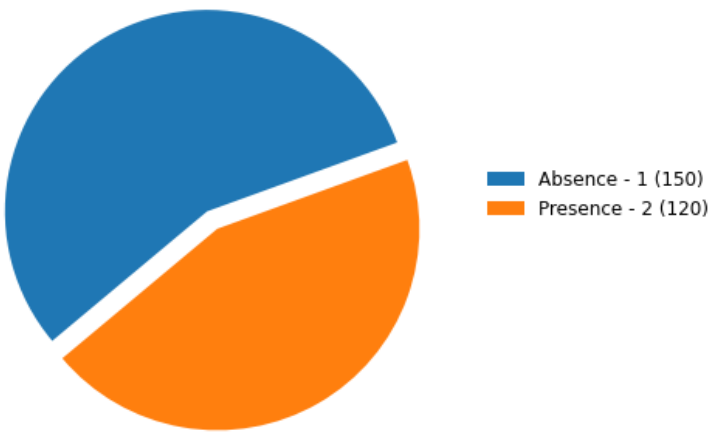
Descriptive Statistics of the dataset:

	age	sex	chest_pain	resting_bp	serum_chol	blood_sugar	rest_ecg	max_heart_rate	induced_ang	oldpeak	peak_st_seg	major_vessels	thal	target
count	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.00000	270.000000	270.000000	270.000000	270.000000
mean	54.433333	0.677778	3.174074	131.344444	249.659259	0.148148	1.022222	149.677778	0.329630	1.05000	1.585185	0.670370	4.696296	1.444444
std	9.109067	0.468195	0.950090	17.861608	51.686237	0.355906	0.997891	23.165717	0.470952	1.14521	0.614390	0.943896	1.940659	0.497827
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.00000	1.000000	0.000000	3.000000	1.000000
25%	48.000000	0.000000	3.000000	120.000000	213.000000	0.000000	0.000000	133.000000	0.000000	0.00000	1.000000	0.000000	3.000000	1.000000
50%	55.000000	1.000000	3.000000	130.000000	245.000000	0.000000	2.000000	153.500000	0.000000	0.80000	2.000000	0.000000	3.000000	1.000000
75%	61.000000	1.000000	4.000000	140.000000	280.000000	0.000000	2.000000	166.000000	1.000000	1.60000	2.000000	1.000000	7.000000	2.000000
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.20000	3.000000	3.000000	7.000000	2.000000

First 5 rows in dataset:

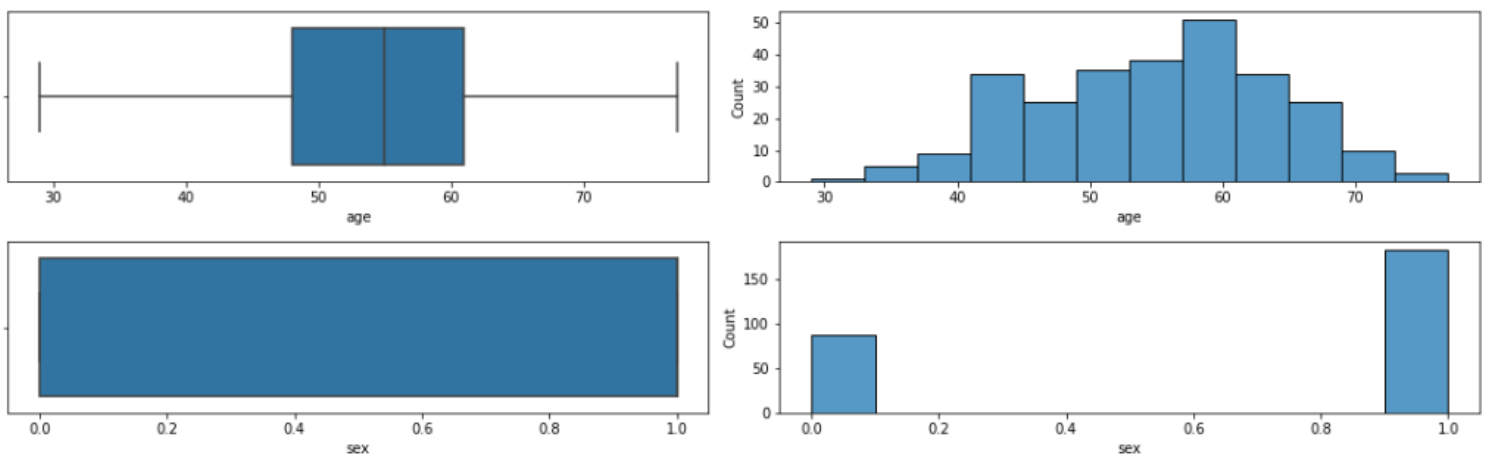
	age	sex	chest_pain	resting_bp	serum_chol	blood_sugar	rest_ecg	max_heart_rate	induced_ang	oldpeak	peak_st_seg	major_vessels	thal	target
0	70.0	1.0	4.0	130.0	322.0	0.0	2.0	109.0	0.0	2.4	2.0	3.0	3.0	2
1	67.0	0.0	3.0	115.0	564.0	0.0	2.0	160.0	0.0	1.6	2.0	0.0	7.0	1
2	57.0	1.0	2.0	124.0	261.0	0.0	0.0	141.0	0.0	0.3	1.0	0.0	7.0	2
3	64.0	1.0	4.0	128.0	263.0	0.0	0.0	105.0	1.0	0.2	2.0	1.0	7.0	1
4	74.0	0.0	2.0	120.0	269.0	0.0	2.0	121.0	1.0	0.2	1.0	1.0	3.0	1

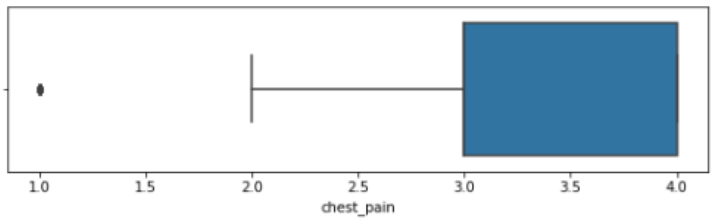
Count for Target Values (Class Labels)

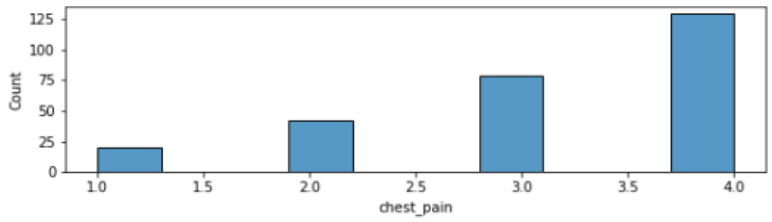


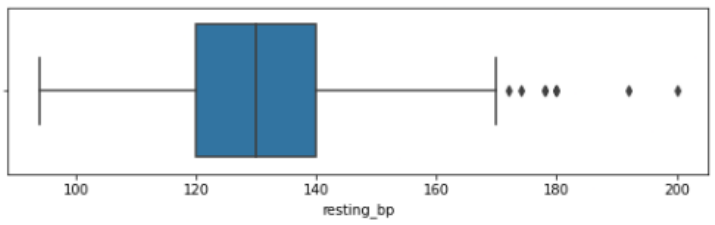
Here, we can see that the ‘Absence’ and ‘Presence’ have counts in the same range, occupying 55.56% and 44.44% of the values, respectively.

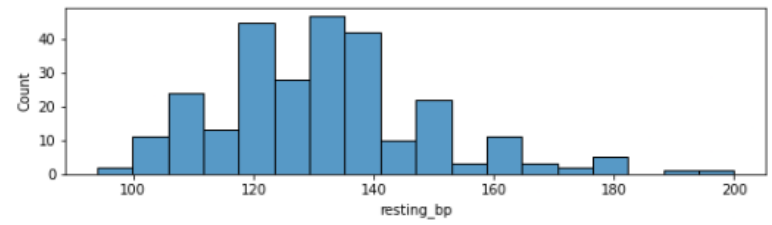
Distribution of attribute values (univariate):

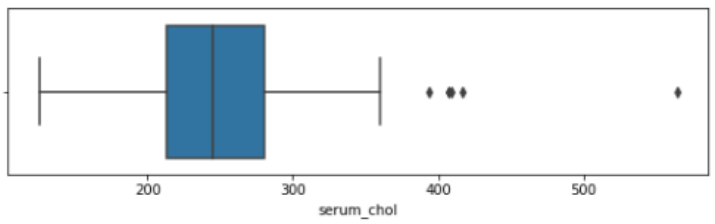


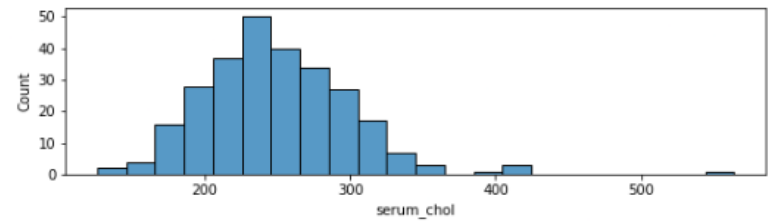

chest_pain

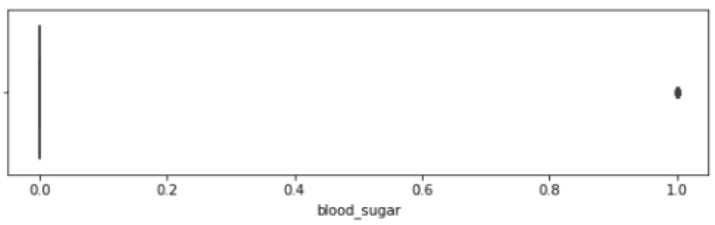

Count
chest_pain

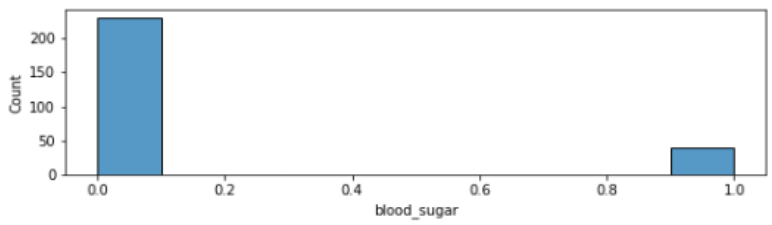

resting_bp

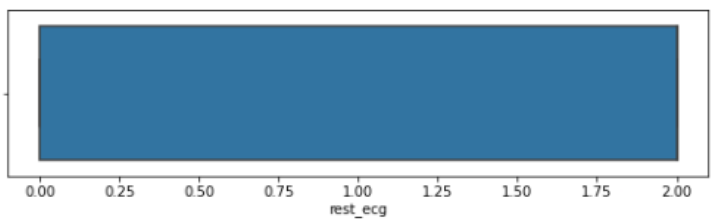

Count
resting_bp

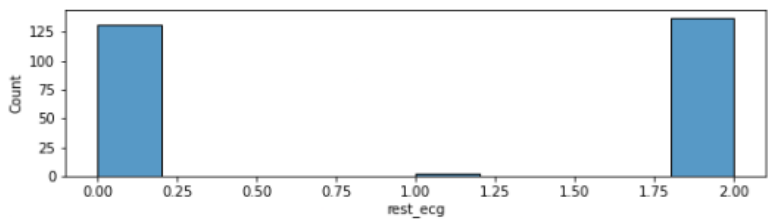

serum_chol

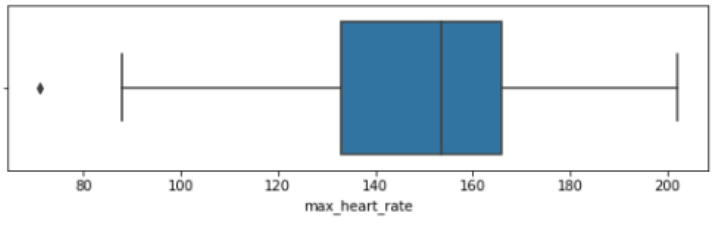

Count
serum_chol

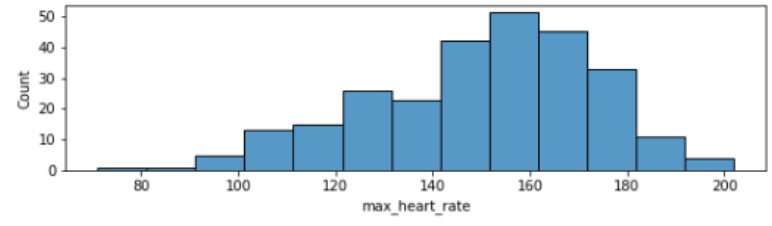

blood_sugar

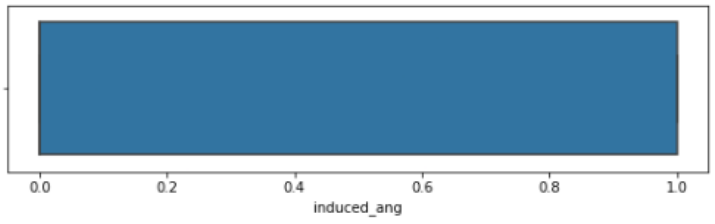

Count
blood_sugar

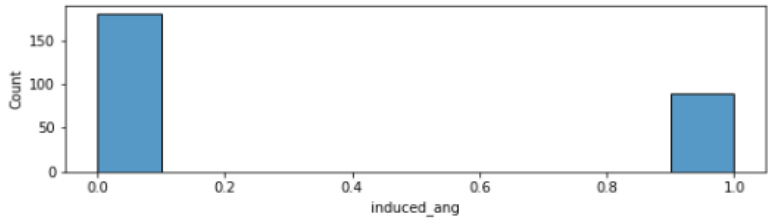

rest_ecg

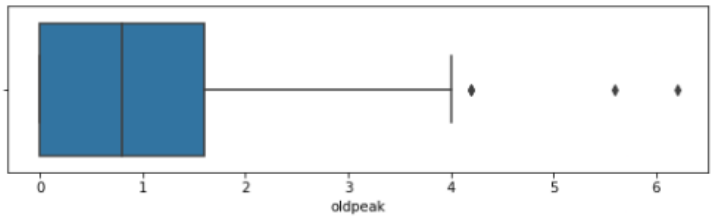

Count
rest_ecg

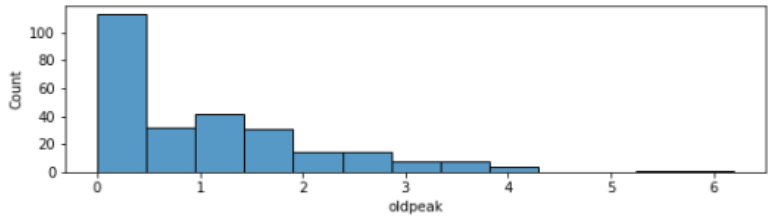

max_heart_rate

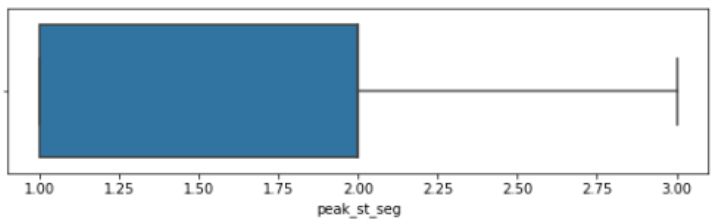

Count
max_heart_rate

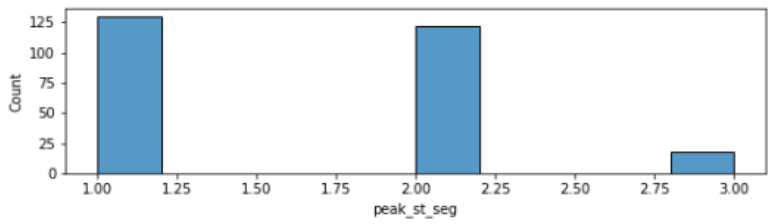

induced_ang

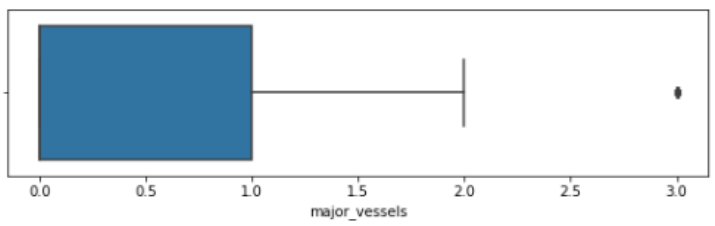

Count
induced_ang

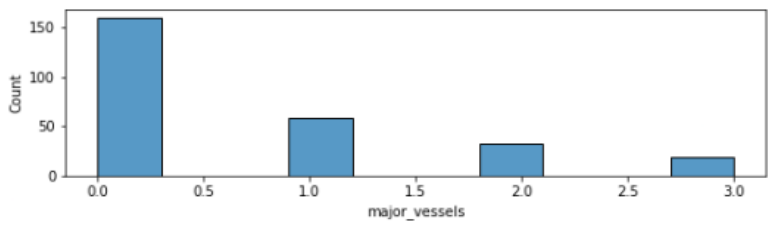

oldpeak

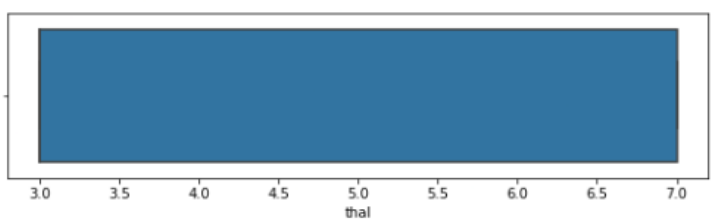

Count
oldpeak

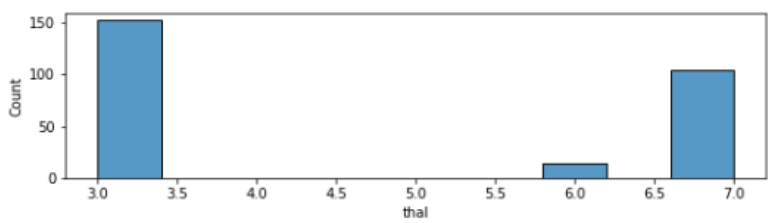

peak_st_seg


Count
peak_st_seg


major_vessels


Count
major_vessels


thal


Count
thal

The boxplot displays the distribution of data for an attribute based on a five number summary - minimum, first quartile, median, third quartile and maximum. It helps in analyzing if data is symmetrical or skewed, or how they are grouped.
The histplot shows distribution of the dataset using a histogram. It helps in analyzing how the count for values attained is varying over different attributes.

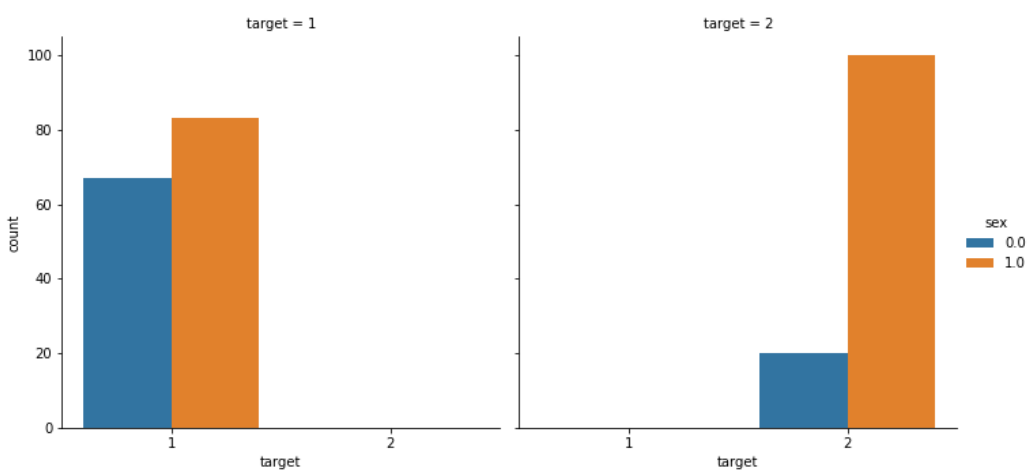
A few points to be noted from these univariate plots are:

1. Most people have ages in the range of 50 and 60.
2. The count for chest pain is continuously increasing for its four types.
3. Serum cholesterol values are mostly clustered at 200 to 300 mg/dl.
4. People with fasting blood sugar below 120 mg/dl have much higher counts than ones above it.
5. Age, resting blood pressure, serum cholesterol, maximum heart rate and oldpeak have continuous values, while others are categorical.
6. The oldpeak (ST depression induced by exercise relative to rest) and the number of major vessels (0-3, coloured by fluoroscopy) is skewed towards the left.
7. The ecg results, induced angina and thal are well-distributed over their respective range.

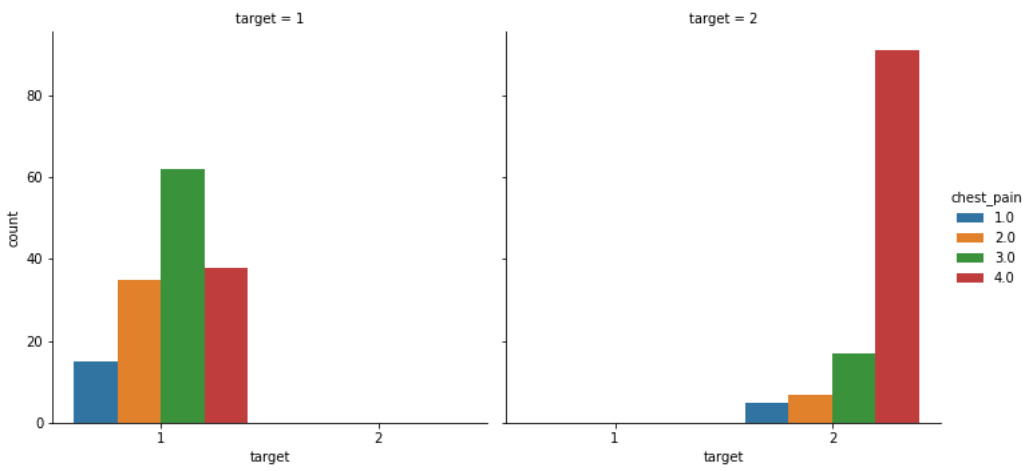
Distribution of attribute values (bivariate):

Catplot for categorical attributes:

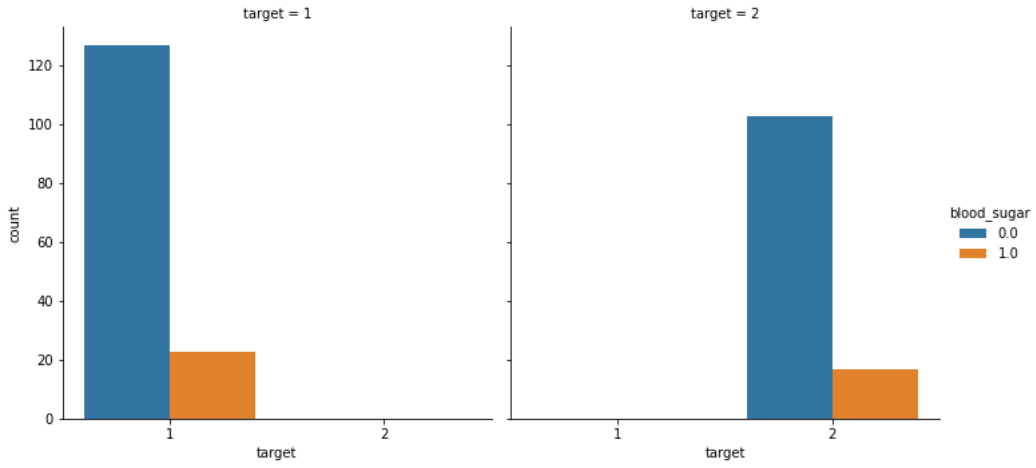
Sex : For most females (0.0), heart disease is absent, while for most males (1.0), it is present.



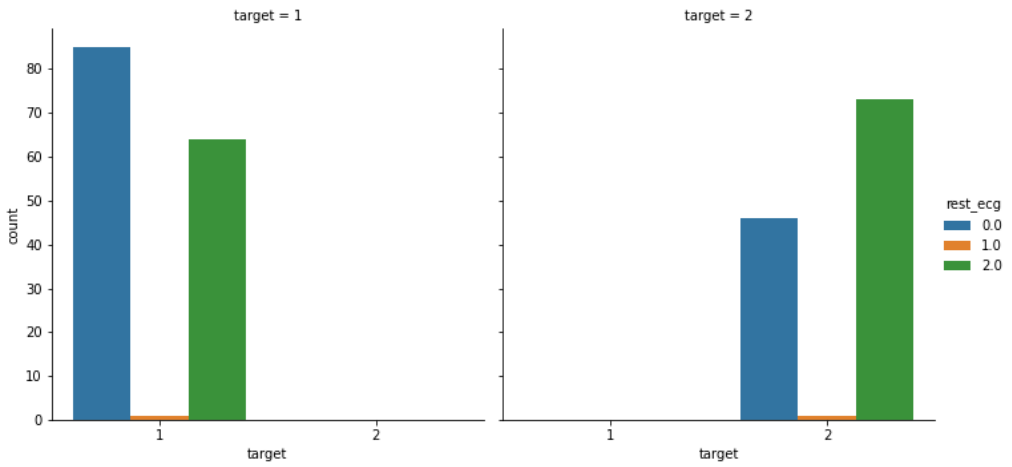
Chest Pain : Heart disease has a very high probability of being present if chest pain type is 4.



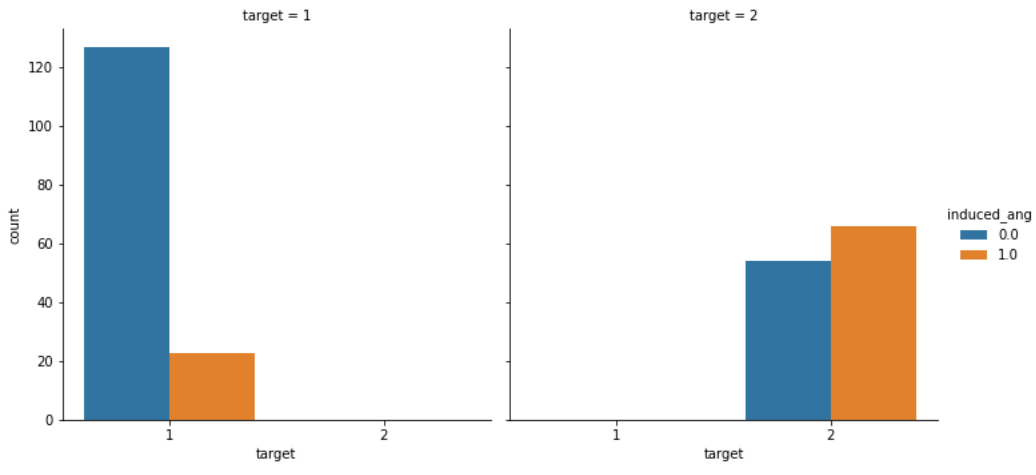
Fasting Blood Sugar : It has similar distribution in both presence and absence of the disease.



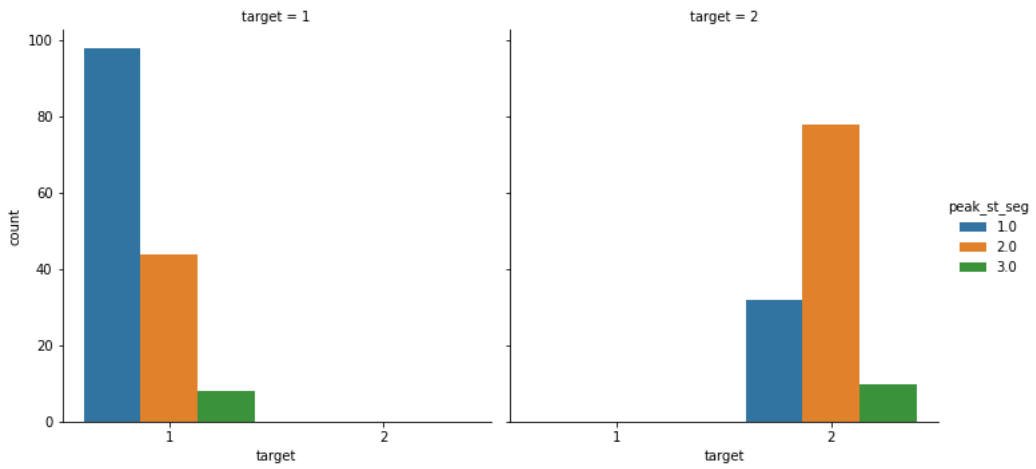
Resting Electrocardiographic Results : People having the result value 2 are more likely to have the disease. Also, only a handful of people have its value 1.



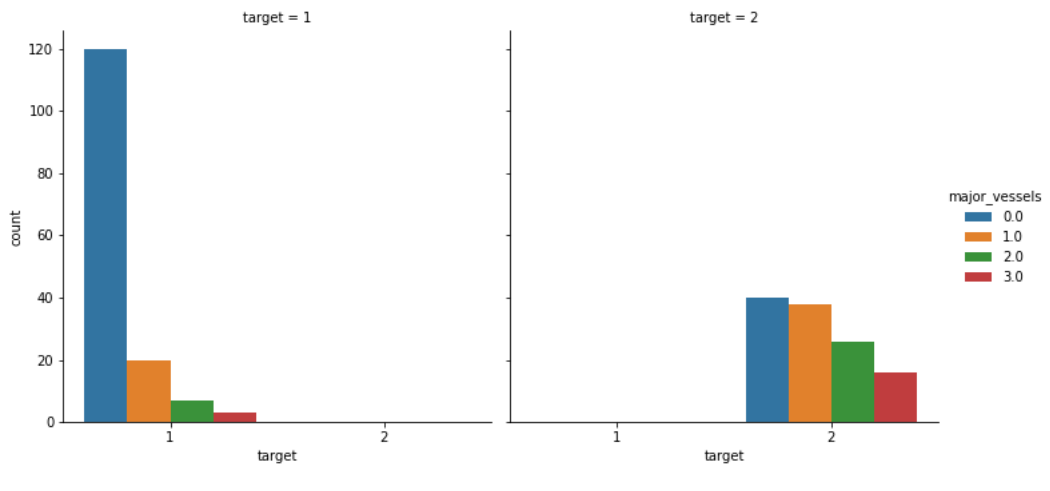
Exercise Induced Angina : People having its value 0 are more likely to not have the disease.



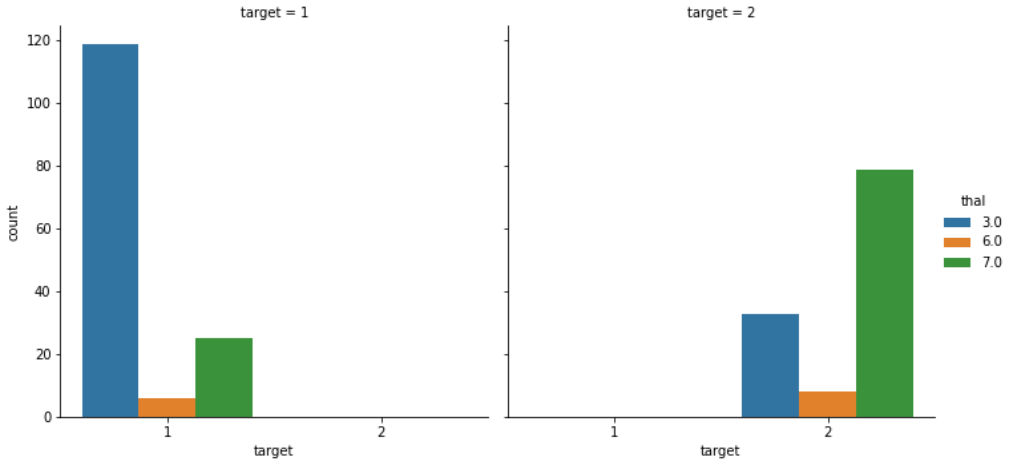
Slope of the peak exercise ST segment : Heart disease is mostly absent if its value is 1, and mostly present if the value is 2. Not much can be said here if the value is 3.



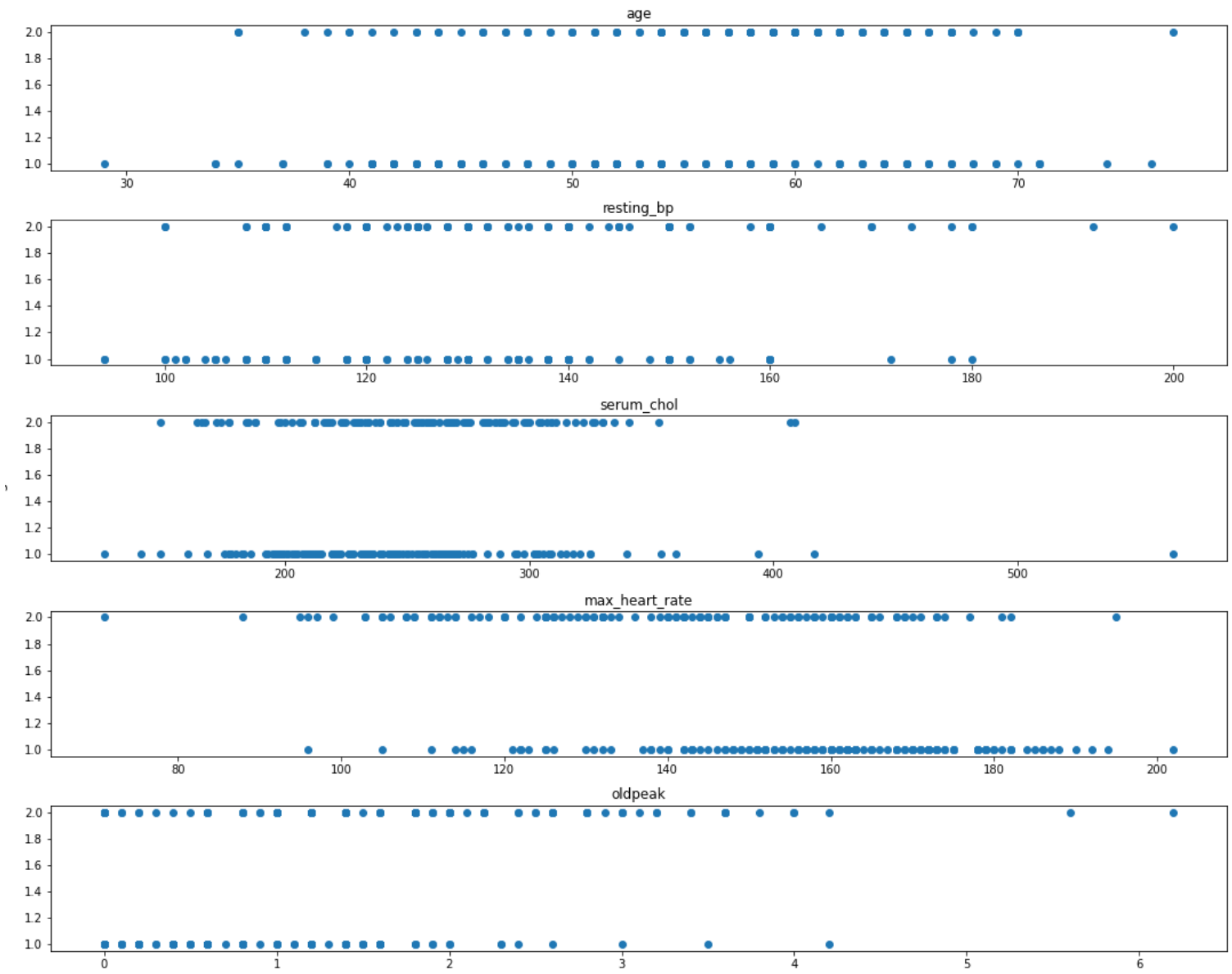
Number of major vessels : There is a high chance that the disease is absent if its value is 0.



Thal : Not many people have its value 6. Probability of having heart disease is high if it is 7, and low if it is 3.



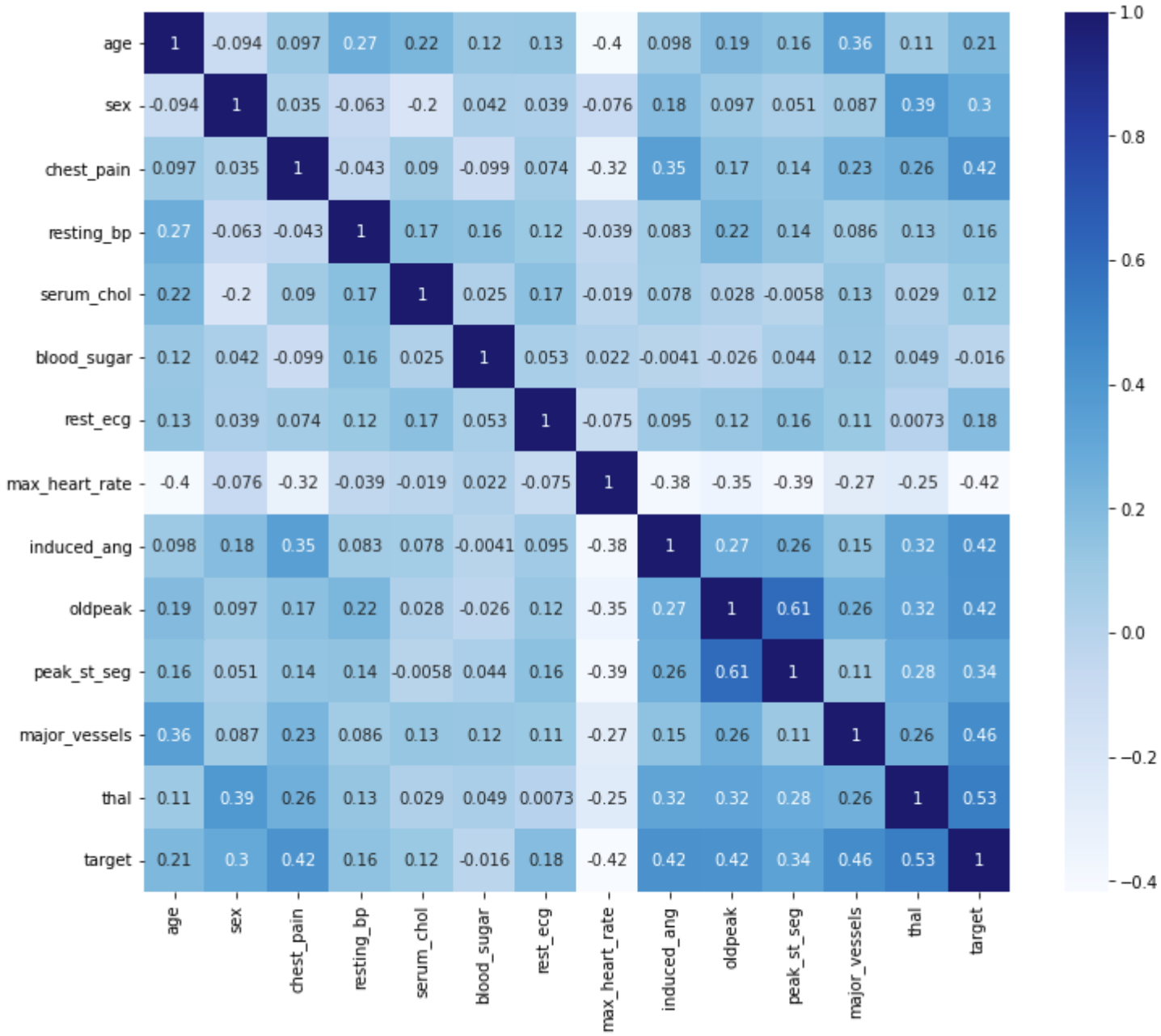
Scatter Plot for continuous attributes:



1. The distribution is similar for both presence and absence in most of the attributes.
2. The frequency of any attribute value is not taken into consideration in the scatter plot, so not much can be said about these continuous attributes.

Heatmap (Multi-Collinearity of Attributes):

Heatmap is used to better visualize the magnitude of a phenomenon in a dataset and assist by directing towards areas that matter the most through colour-codings.



Among the attributes, it can be seen that there is a strong relationship between peak_st_seg & oldpeak, thal & sex, age & major_vessels, and induced_ang and chest_pain.

For the target column, the relationship is best with thal, and almost equally strong with major_vessels, chest_pain, induced_ang and oldpeak. Also, the max_heart_rate is negatively impacting the target value by a huge amount.

II. Procedure

The decision tree was designed using the ID3 algorithm. The dataset provided in .dat format was converted into the .csv format before designing the decision tree.

1. The data provided in the form of a .dat file is converted to a .csv file. The dataset along with its attributes were deeply analysed before proceeding to build a Decision Tree. Then, we separate the categorical attributes from the continuous attributes as the two of them will have different calculations for splitting and predictions. On the basis of all the plots made, we also determine priority order for the attributes, which is used for the case when the condition for splitting is equal for two attributes.

2. A Node class is defined with the following parameters:
    - feature_index (index of the feature used for deciding)
    - threshold (maximum or equal threshold used for the feature)
    - left (left child of the current node)
    - right (right child of the current node)
    - gain (Information Gain or Negative Gini Index corresponding to the feature_index for threshold)
    - gini (Gini Index corresponding to the feature_index for threshold)
    - value (Value at the current leaf node)

3. DecisionTreeClassifier class is defined with the parameters: criterion (Gini Index / Information Gain), max_depth (the maximum depth of the tree), and min_sample_split (the minimum number of samples required to split an internal node).

4. In the DecisionTreeClassifier class, a build_tree method is defined. The function works recursively until stopping conditions are met. It finds the best split, calls itself with data for left child and right child, and adds a decision node using minimum Gini Index or maximum Information Gain.

    Gini Index computes the amount of probability of a feature which is classified incorrectly upon its random selection. It varies between 0 and 1. If $p(c_i)$ denotes the probability of an element being classified for a distinct class,

    $$Gini = 1 - \sum_{i=1}^{n} p^2(c_i)$$

    Information Gain (IG) is a measure of information provided by a feature about a class. It is based on the notion of entropy and the splitting through this criterion, in general, intends to decrease the amount of entropy initiating from root node to leaves node. For a feature f, parent node D_p, child nodes D_{left} and D_{right}, entropy I, no. of samples N, no. of samples at child nodes N_{left} and N_{right},

    $$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N} I(D_{left}) - \frac{N_{right}}{N} I(D_{right})$$

For both criterion, i.e., Gini Index and Information Gain, we do the following:

5. Accuracies are calculated for max depths between 1 and 15 of the tree by taking mean of accuracies on 10 random 80-20 train-test splits. The corresponding graph is plotted to determine optimal max depth for the decision tree. Along with this, another graph is plotted for showing the variation of accuracy

with the number of nodes in those 150 cases.

6. The decision tree of a given criterion with optimal depth limit is built and printed.

7. Difference in ground truth target set and predictions made by the classifier is shown through a plot having two line charts.

8. The tree is then pruned using a post-pruning method. If the error at the node is less than the weighted average of the error of children, the tree from the current node is pruned and all the links from the node to the children are removed.

9. The pruned decision tree is then printed.

10. A graph is plotted to show the actual targets, targets predicted from initial decision tree, and targets predicted from pruned decision tree for the given impurity measure.

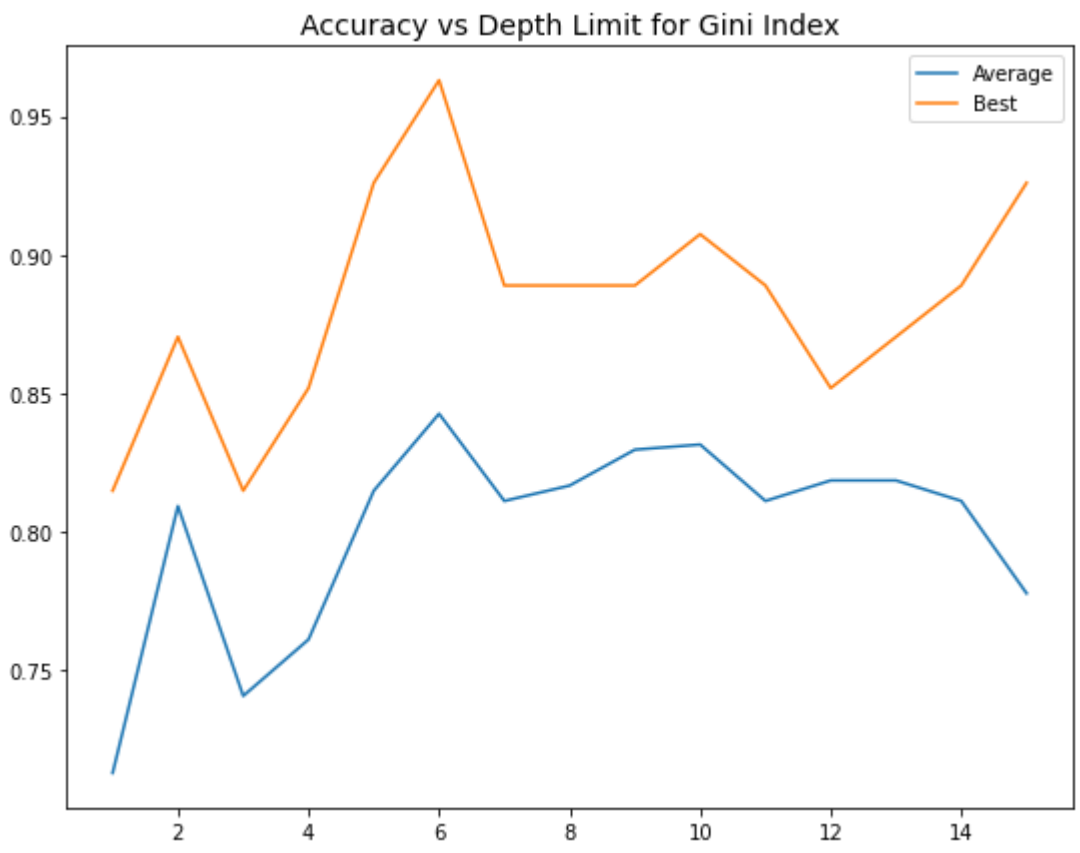
III. Results

A. GINI INDEX

The following results are obtained when Gini Index is taken as the criterion for impurity measure:

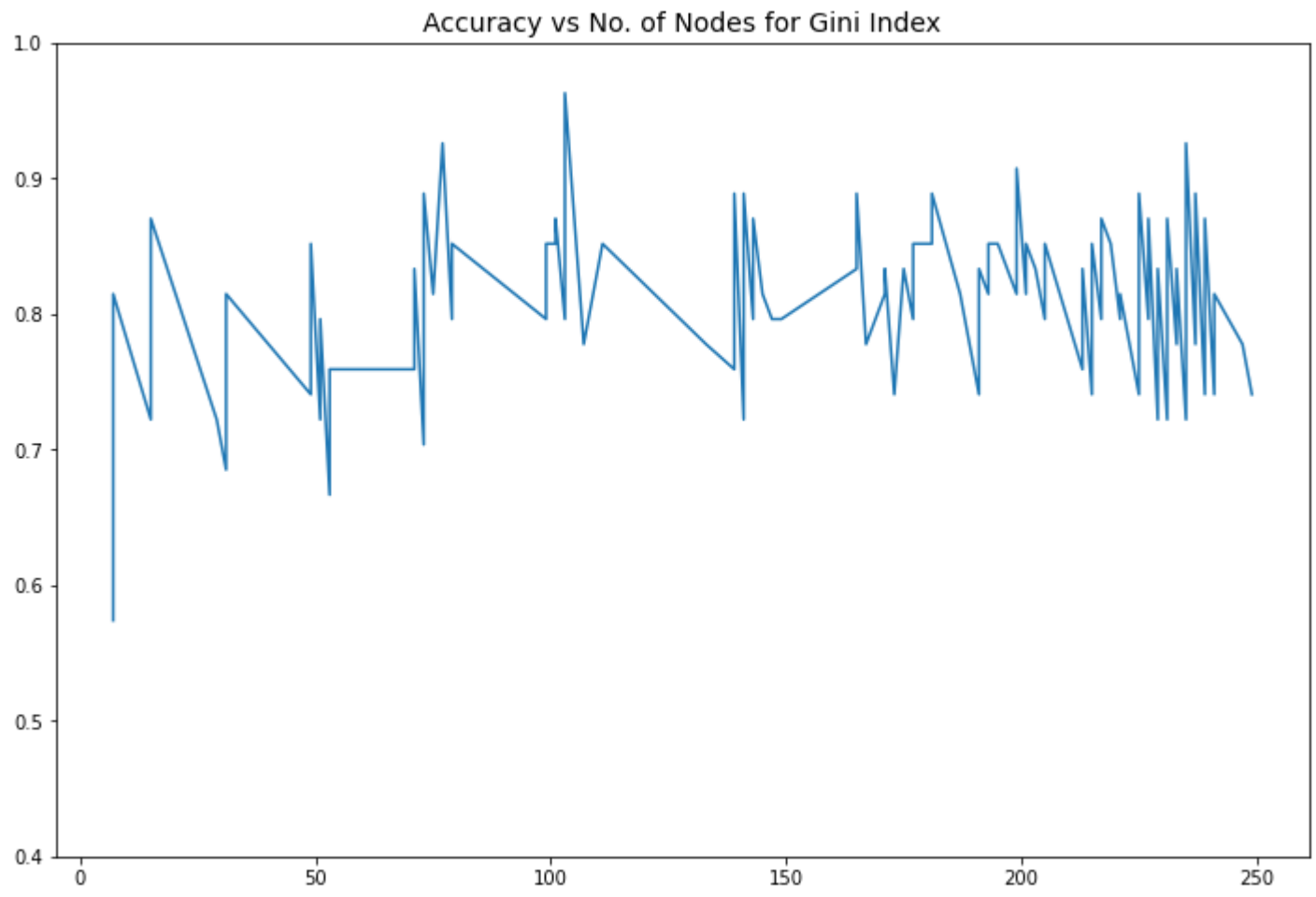
Variation of Accuracy with depth

The following graph is obtained for best accuracy and average accuracy on varying the depth limit.



The **best possible depth limit is 10** and the same is used for the data set. This is because it gives high average accuracy and the difference between the average and the best accuracy is less as compared to some other depth limit having a high best accuracy. For any depth limit, if the best accuracy is much higher as compared to the average accuracy, there has to be some accuracy which is very low and so, this would not be optimal. Hence, keeping such factors in mind, we choose the max_depth = 10.

Variation of Test Accuracy with the total number of nodes in the tree



Although this graph looks very random in this condition, if we think of it by removing some outliers, we can see that accuracy increases with an increase in the number of nodes to top point and then starts to decrease.

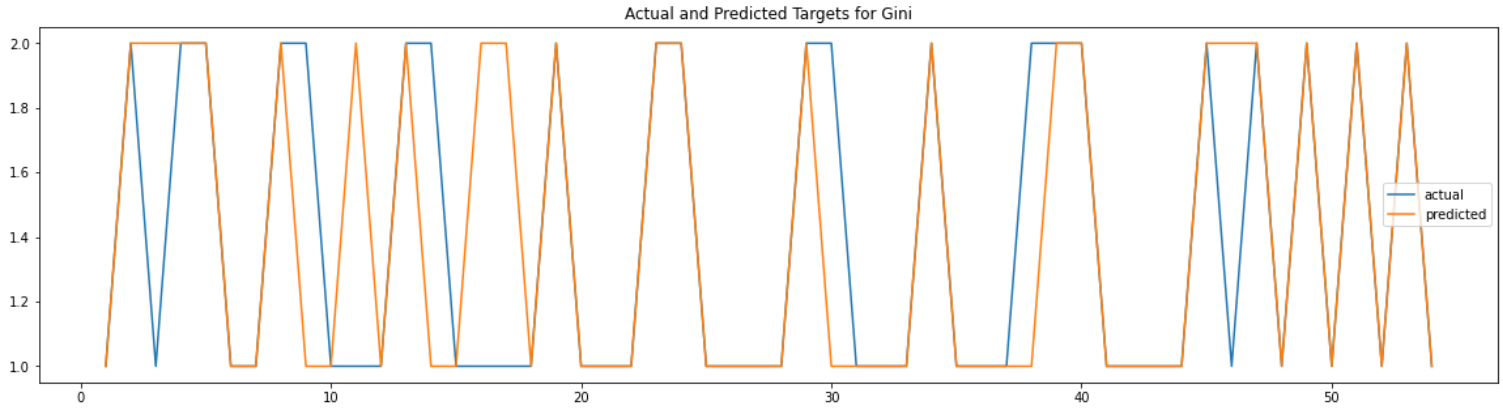
Upon building the decision tree, validation and test accuracies found are:

```
Validation Accuracy before pruning:     0.8055555555555556

   Best Test Accuracy for Gini Index:   0.8333333333333334
Average Test Accuracy for Gini Index:   0.7925925925925925
```

Plot for Actual and Predicted Targets

The plot below compares the actual and predicted targets for the given dataset. For maximum accuracy, the orange line would overlap with the blue line, and we can see this happening in most areas of the graph.



After Pruning

The accuracy increases after pruning the tree because pruning improves predictive accuracy by the reduction of overfitting. It can be confirmed from the outputs attached below:

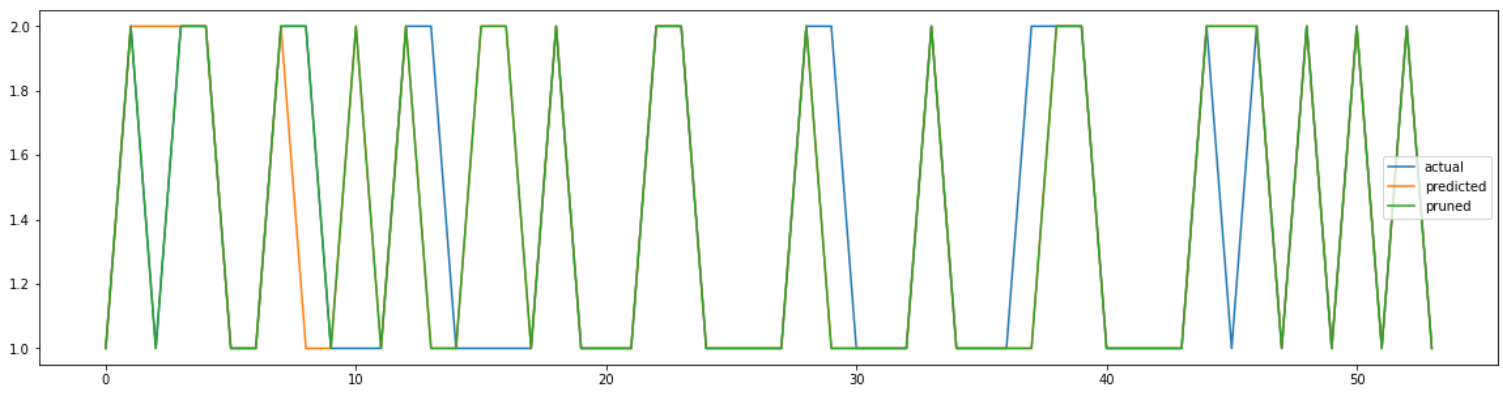
```
Validation Accuracy before pruning:     0.8055555555555556

Validation Accuracy after  pruning:     0.8611111111111112

Test Accuracy before pruning:    0.8333333333333334
Test Accuracy after  pruning:    0.8703703703703703
```

Plot for Actual, Predicted, and Pruned Target for Gini Index

Since the accuracy obtained after pruning the tree is better than that obtained before pruning, thus the plot for the pruned target is slightly closer to the actual target compared to the one obtained before pruning. We can see that the green (pruned predictions) line has almost covered the blue (actual predictions), much more than the orange line, and this says that the post pruned tree is performing better.



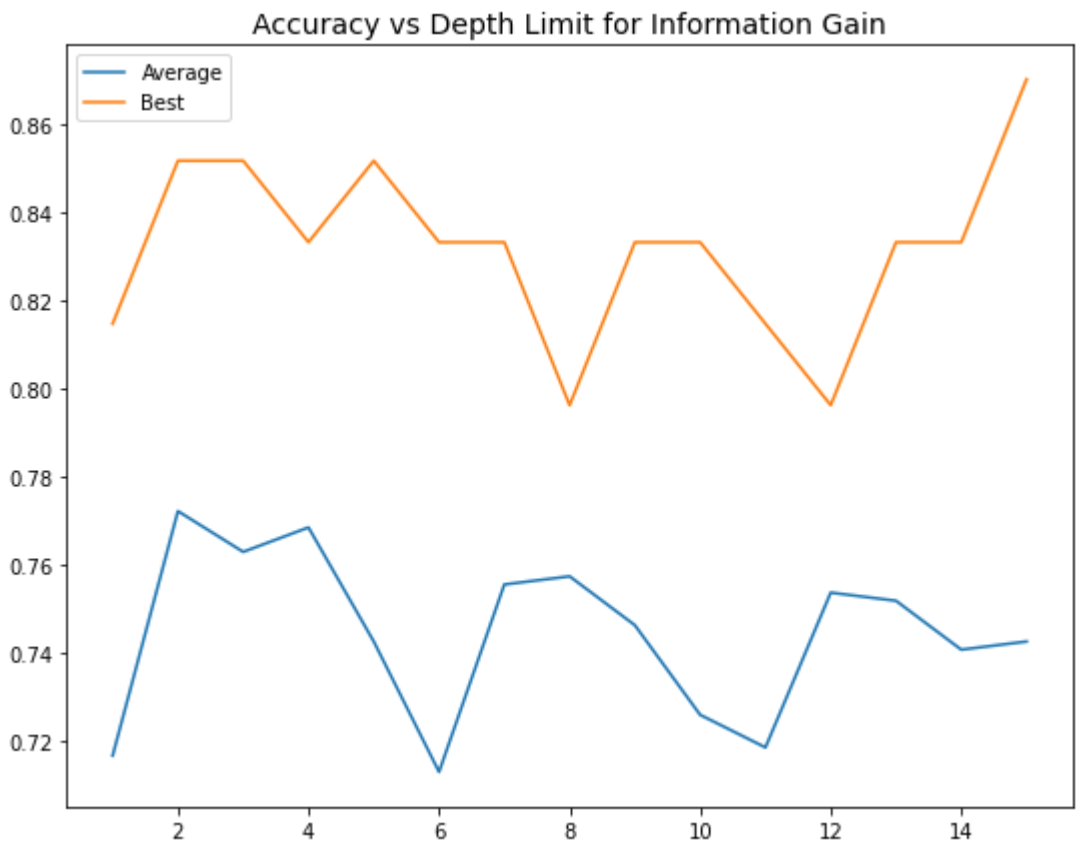
The Gini Index Decision Trees, both before and after pruning, with the hierarchical representation of attributes can be viewed in the files: gini_decision_tree.gv.pdf and gini_pruned_decision_tree.gv.pdf

B. INFORMATION GAIN

The following results are obtained when Information Gain is taken as the impurity measure:

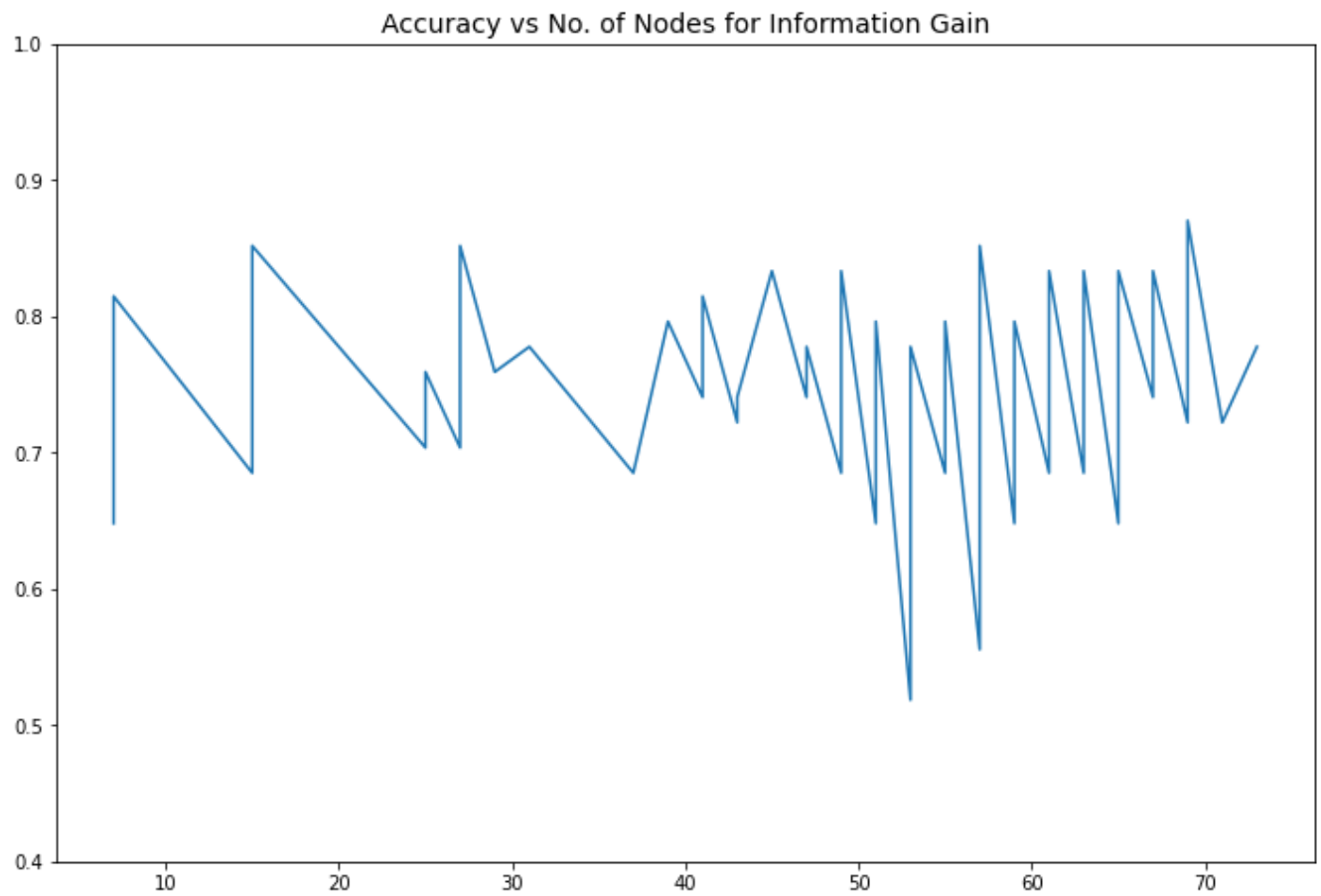
Variation of Accuracy with depth

The following graph is obtained for best accuracy and average accuracy on varying the depth limit.



For the similar reasons mentioned for the Gini Index, here, the **best possible depth limit is 12**. Hence, max_depth = 12.

Variation of Test Accuracy with the total number of nodes in the tree



Just like the graph obtained for the Gini Index, this graph too looks very random in this condition. But if we think of it by removing some outliers, we can see that accuracy decreases with an increase in the number of nodes to the top point, then starts to increase. Mostly, the no. of nodes lies near 50-70.

Upon building the decision tree, validation and test accuracies found are:

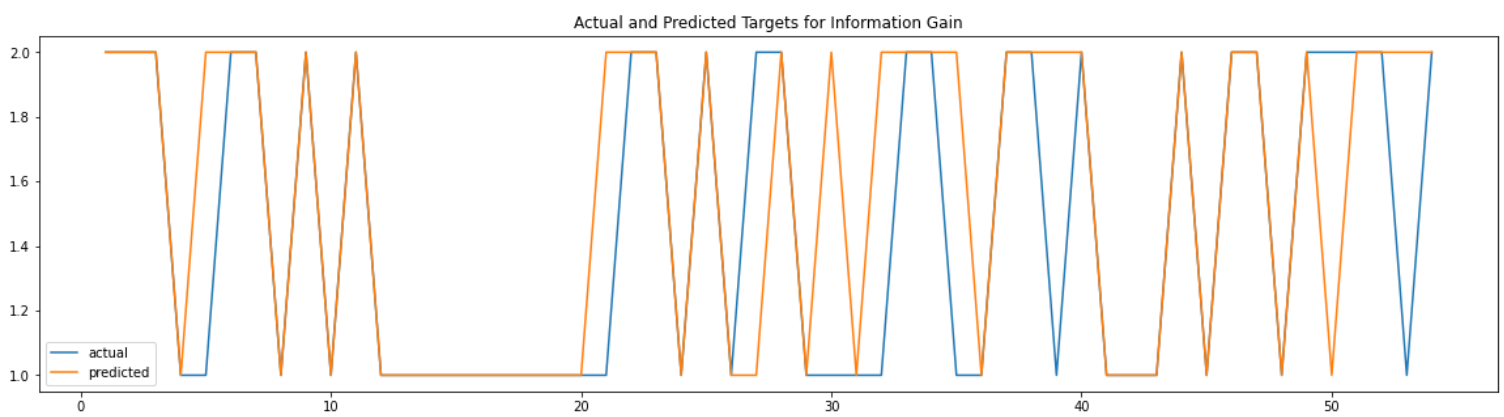
```
Validation Accuracy before pruning:     0.7916666666666666

   Best Test Accuracy for Information Gain:     0.8333333333333334

Average Test Accuracy for Information Gain:     0.7370370370370369
```

Plot for Actual and Predicted Targets

The plot below compares the actual and predicted targets for the given dataset. For maximum accuracy, the orange line would overlap with the blue line, and we can see this happening in most areas of the graph.



After Pruning

The accuracy increases after pruning the tree because pruning improves predictive accuracy by the reduction of overfitting. The same can be observed below:

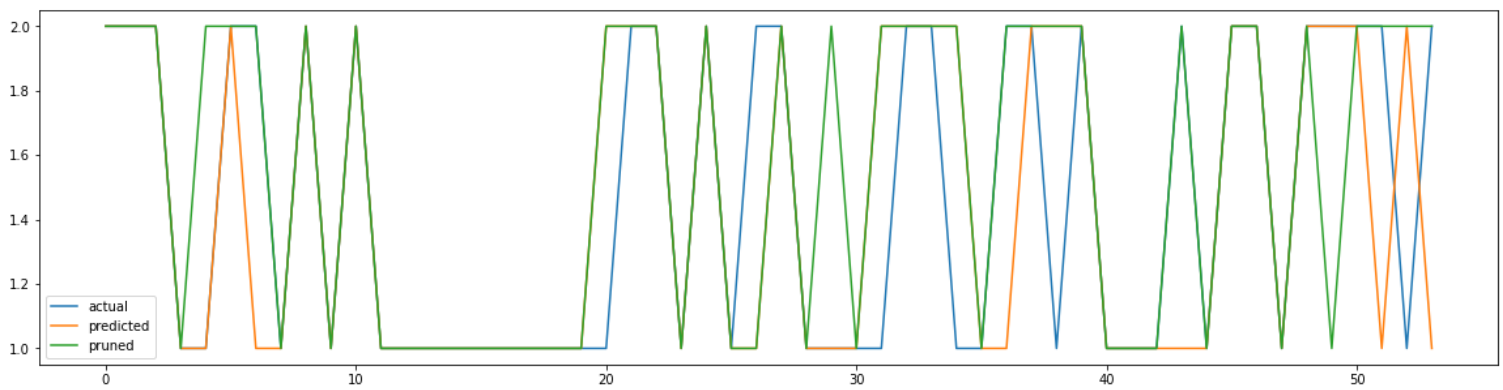
```
Validation Accuracy before pruning:     0.7916666666666666

Validation Accuracy after  pruning:     0.8333333333333334


Test Accuracy before pruning:   0.7962962962962963
Test Accuracy after  pruning:   0.8333333333333334
```

Plot for Actual, Predicted and Pruned Target for Information Gain

Since the accuracy obtained after pruning the tree is better than that obtained before pruning, thus the plot for the pruned target is slightly closer to the actual target compared to the one obtained before pruning.



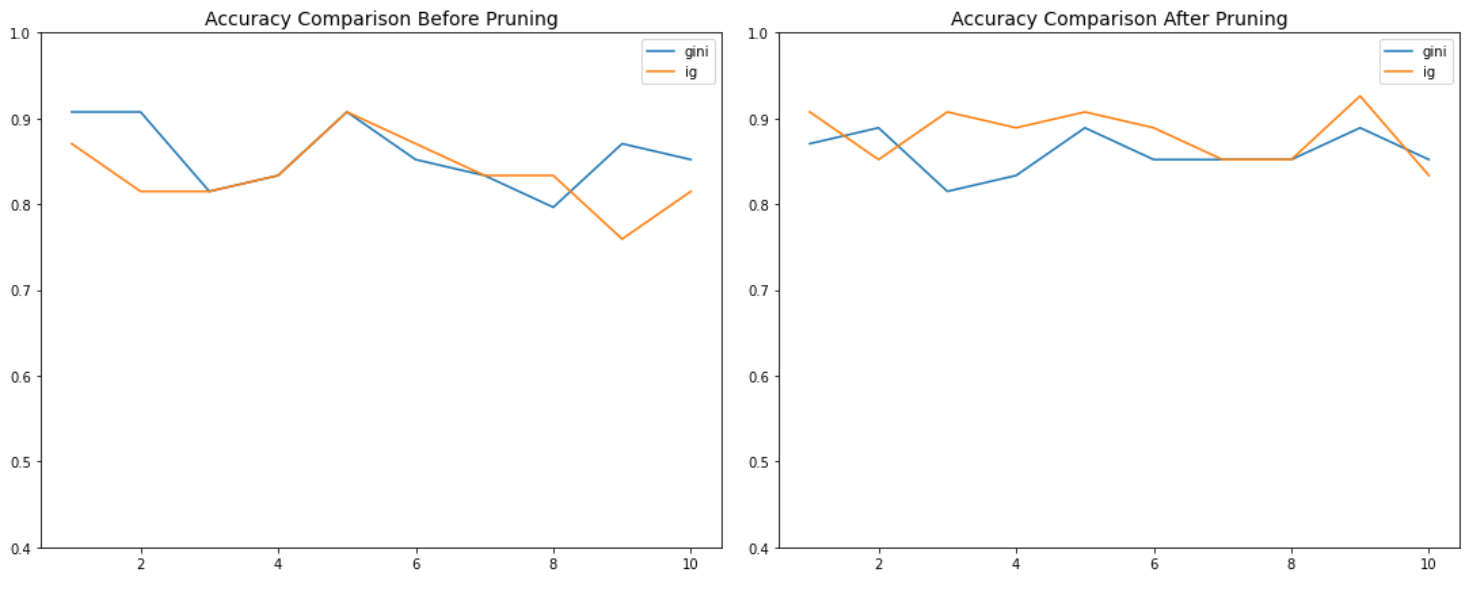
The Information Gain Decision Trees, both before and after pruning, with the hierarchical representation of attributes can be viewed in the files: ig_decision_tree.gv.pdf and ig_pruned_decision_tree.gv.pdf

C. Accuracy Comparison between Gini Index and Information Gain

To compare the accuracy of the two criteria - gini index and information gain in an efficient manner, 10 random 80-20 train-test splits are made, and the predictions are made for the test set for both cases. Here are the outputs for the same.

```
Criterion: GINI INDEX
Average Accuracy over 10 random 80-20 split before pruning: 0.85741
Average Accuracy over 10 random 80-20 split after  pruning: 0.85926

Criterion: INFORMATION GAIN
Average Accuracy over 10 random 80-20 split before pruning: 0.83519
Average Accuracy over 10 random 80-20 split after  pruning: 0.88148
```



It is observed that accuracy results after pruning are better for both of them, and both the criteria have got almost similar accuracy.

References

Dataset Source: https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/heart/heart.dat
Dataset Description: https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/heart/heart.doc