Visualizing Random forest and Decision tree

Ram Kashyap S

May 11, 2017

1 Basic Info

Name: Ram Kashyap S
Email: u1082810@utah.edu
uID: u1082810
Project repository:

`https://github.com/ramkashyap-s/visualizing-random-forest`

2 Overview and Motivation

Decision trees and random forest models are among the most used machine learning models. Since, decision trees are not black box models, they can be understood with the help of visualizations.

Most machine learning packages in python and R output a DOT file which can be visualized using graphviz tool, but they are static images and lack interactive features.

The aim of this project is to build an interactive tool for visualizing the decision trees and to explore how the features are interacting in the random forest model.

3 Visualization Design

3.1 Initial designs

Initial design 1

In the initial design, the idea was to visualize a decision tree along with other components for the given data and parameters. In this design random forest model was not considered. After discussing with the project mentors, I dropped the components for visualizing the accuracy and a plot with depth of leaf nodes for the number of trees as they seemed to have deviated from the objective of the project.

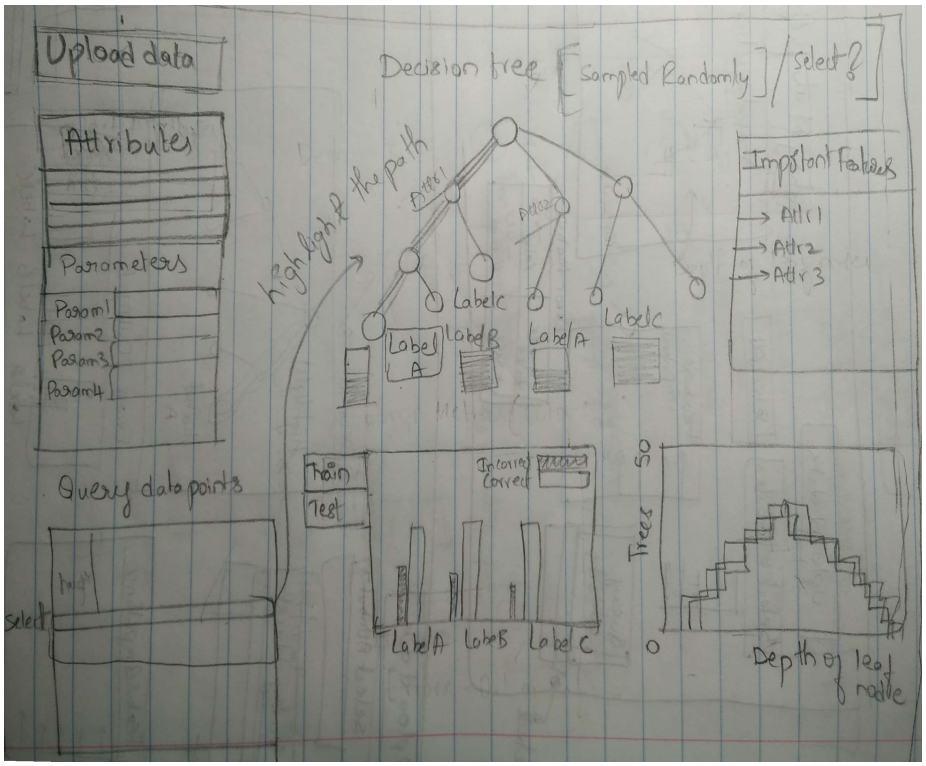

Figure 1: Initial design 1

Initial design 2

In this design, the idea was to create a user guided parameter space exploration. The user can choose the input parameter values for which plots of accuracy and other parameters are visualized as shown in the below figure. Annotations are displayed as a way to guide user to assist in making parameter choices. Later after discussing with the project mentors and reading some background material, I realized that the input space for parameters is huge and the user might be constrained when compared with grid search or other parameter space search algorithms and it incurred high CPU cost in the initial prototype implementation. So, I chose to go with the first design.

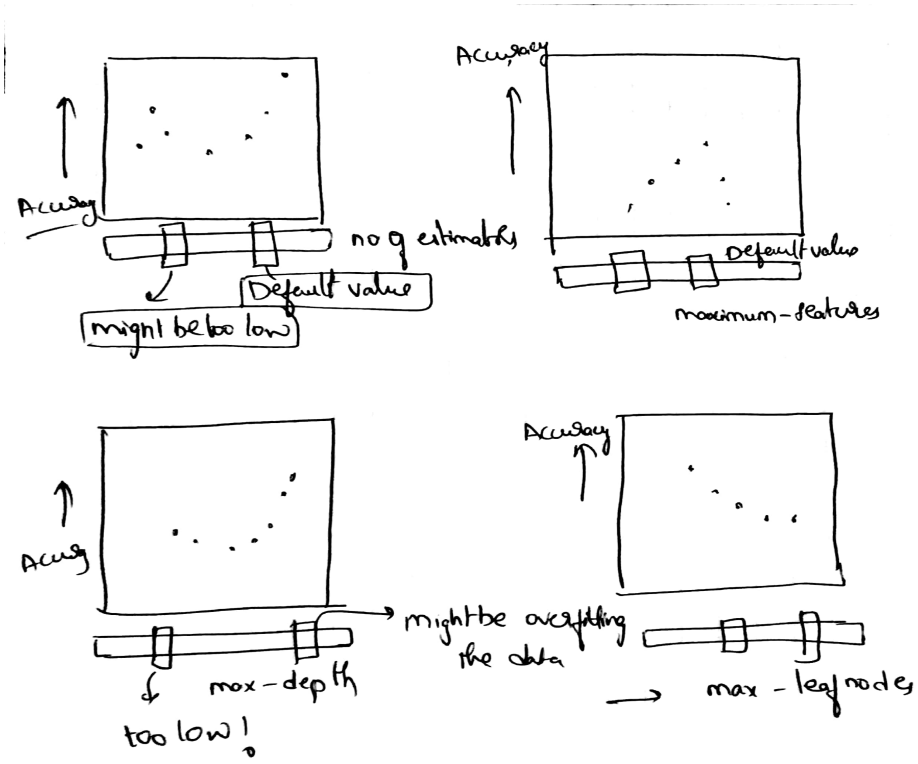

Figure 2: Initial Design 2

3.2 Finalized design

After discussing with mentor, I have finalized on the following designs and used these for the presentation as well.

3.2.1 Decision tree

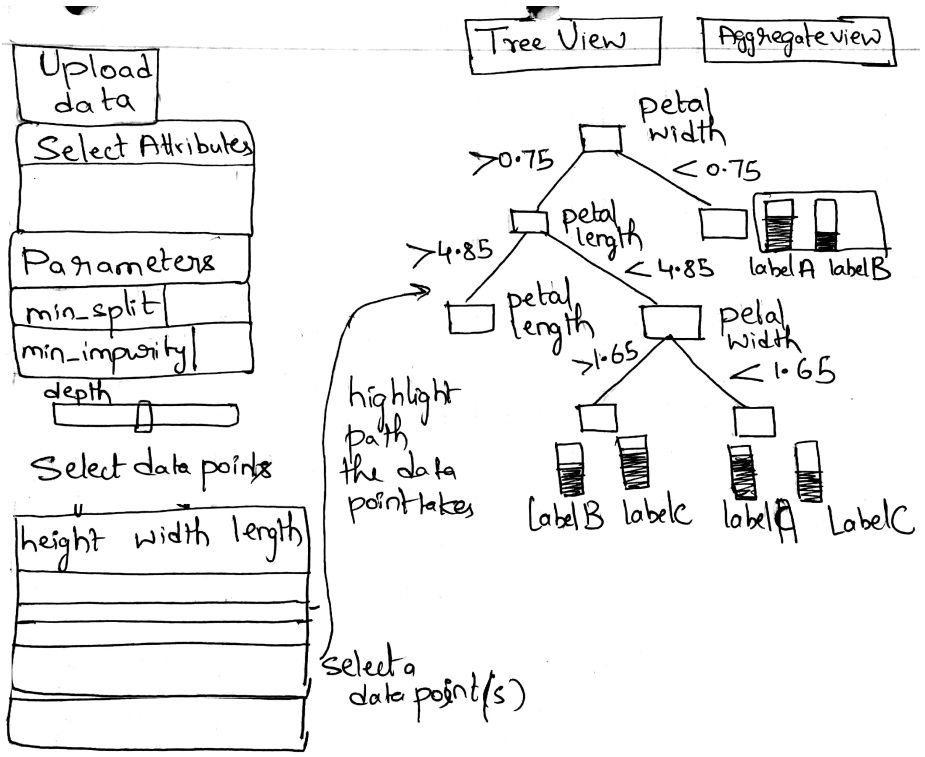

Figure 3: Decision tree view

The above figure displays a single decision tree trained on the data. To make it more interactive, the left panel accepts data, attributes and parameters such as depth of the tree, minimum number of samples required to split an internal node, etc. By changing the parameters we can see how the tree changes.

The label distribution at the leaf node shows how well the rule classified the dataset. For pruning the trees we can use this information and reduce the depth of the model.

I have also proposed to highlight a path for a selected data point. If the label for a data point is already known and a user wanted to see which path it would take, this component would help.

3.2.2 Random forest - Ensemble of Decision trees

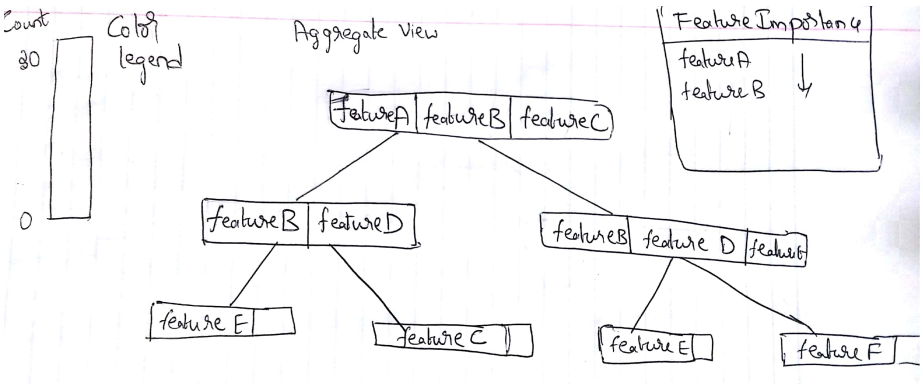

Figure 4: Random forest view

Random forest is an ensemble of different decision trees and provides a higher accuracy than a single decision tree. But, interpreting the model is tricky when compared to a single tree. So, it would be interesting to visualize the model and look at the node interactions and how many times a feature appeared at a position. It can explain the variable importance scores that are given by the model.

4 Implementation

I have used D3.js and Bootstrap for the front end part of the project and flask and python for the server side. When the user uploads a file and selects the model parameters an AJAX request is made and the data is sent in the form of a JSON to the server.

In the back end using sklearn, machine learning models are created. From these models, rules and features are extracted and a tree structure is created along with the 3-fold cross validation accuracy. A response is sent to the front end and the tree structure is rendered using D3.js.

4.0.1 Decision tree

Initially, the tree implementation looked like this.

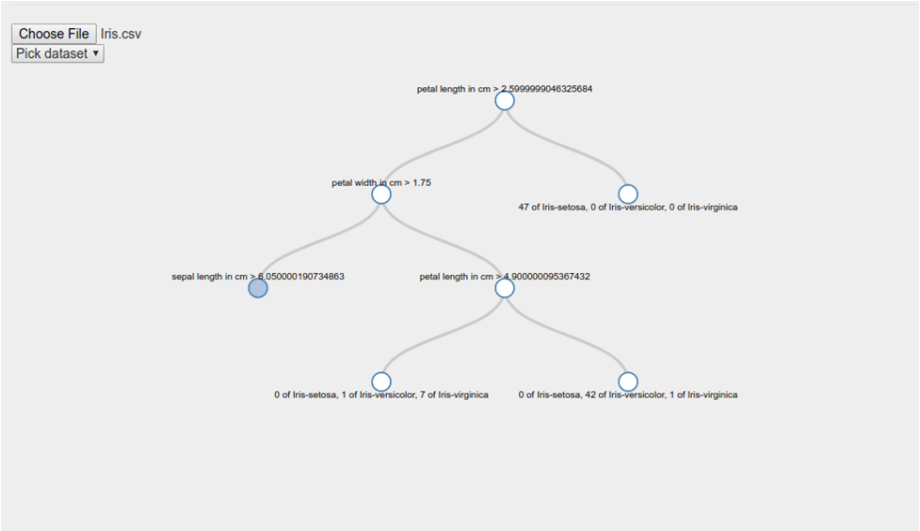

Figure 5: Initial tree

After several enhancements

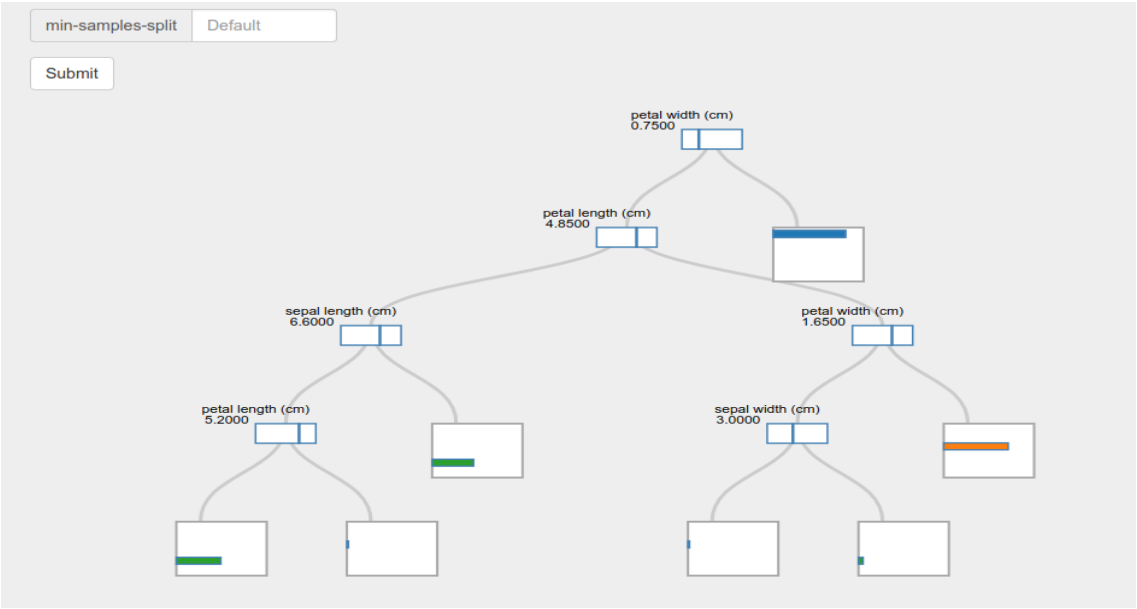

Figure 6: Intermediate progress

In the above tree, the nodes represent the features and the rules are encoded as both text and a line in the rectangle signifying the magnitude of the rule compared to lowest and highest values of that feature. There is also a fixed bounding box for visualizing the distribution of the labels at the leaf nodes.

Problems faced: Due to the large node size, there are scaling problems even for small data sets. So, I have removed the redundant rule encoding feature. Also, I made the bounding box to resize according to the no. of labels in the leaf node.

Finally, the tree view looked like:

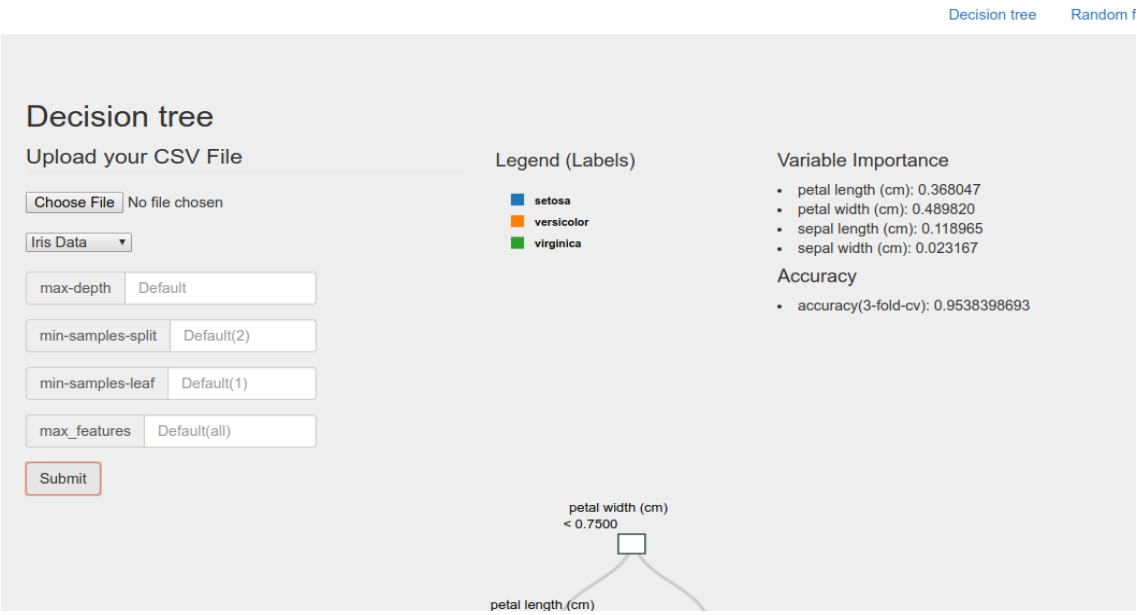

Figure 7: Final view progress

The user can select Iris dataset or upload a csv file which is shown in the figure. The input fields for parameters can be used to change the model parameters. If nothing is specified in those fields, all default values are used.

A legend related to the labels is displayed in the figure. Along with the tree structure, the variable importance scores and accuracy are computed and are displayed in the third column. This section would show how the parameter selection affects the accuracy.

The tree is collapsible and this allows users to inspect only the interested paths/rules. The tool-tip is provided so that users can know the counts of labels in leaf nodes along with the label names. A number scale is not displayed due to the space constraint as the elements are overlapping with text. So counts of labels is shown as a tool-tip

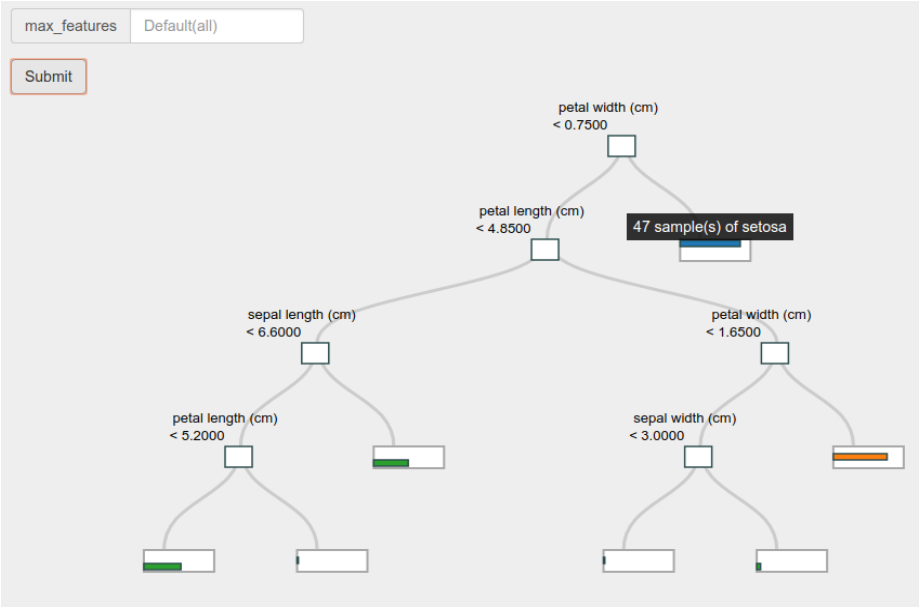

Figure 8: Final view progress

I have tried the data sets such as

	Samples	Attributes	Labels
Iris	150	4	3
Bank Notes	872	4	2
Pima Indians	768	8	2

Un-pruned Bank note data set

Figure 9: Bank note data set

Bank note data set has features of bank note images such as variance, curtosis of image, entropy and skewness of image and 0, 1 as labels. Without pruning the tree looks like the following

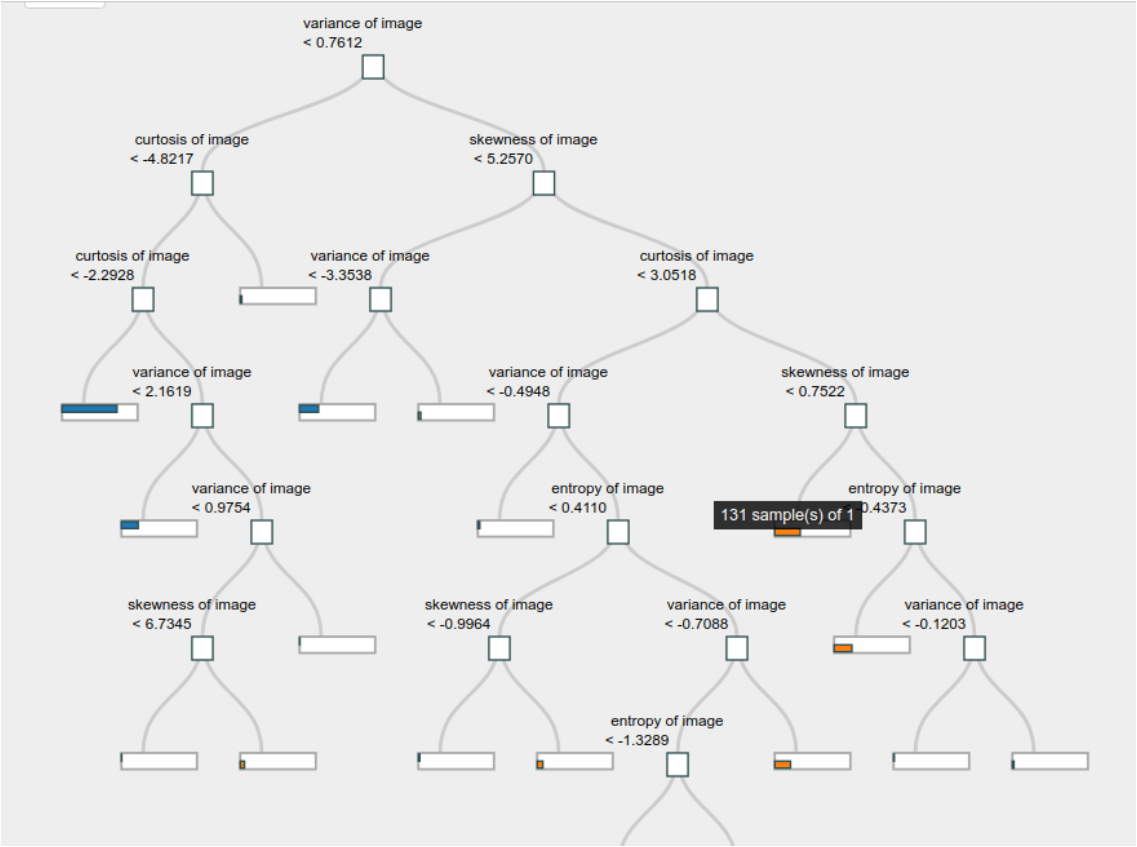

Figure 10: Bank note data set un-pruned

From the figure we can observe that after level 5, the 0 label instances are very less in the leaves. So we can set a depth of 5 and submit, and after pruning the tree looks as the following

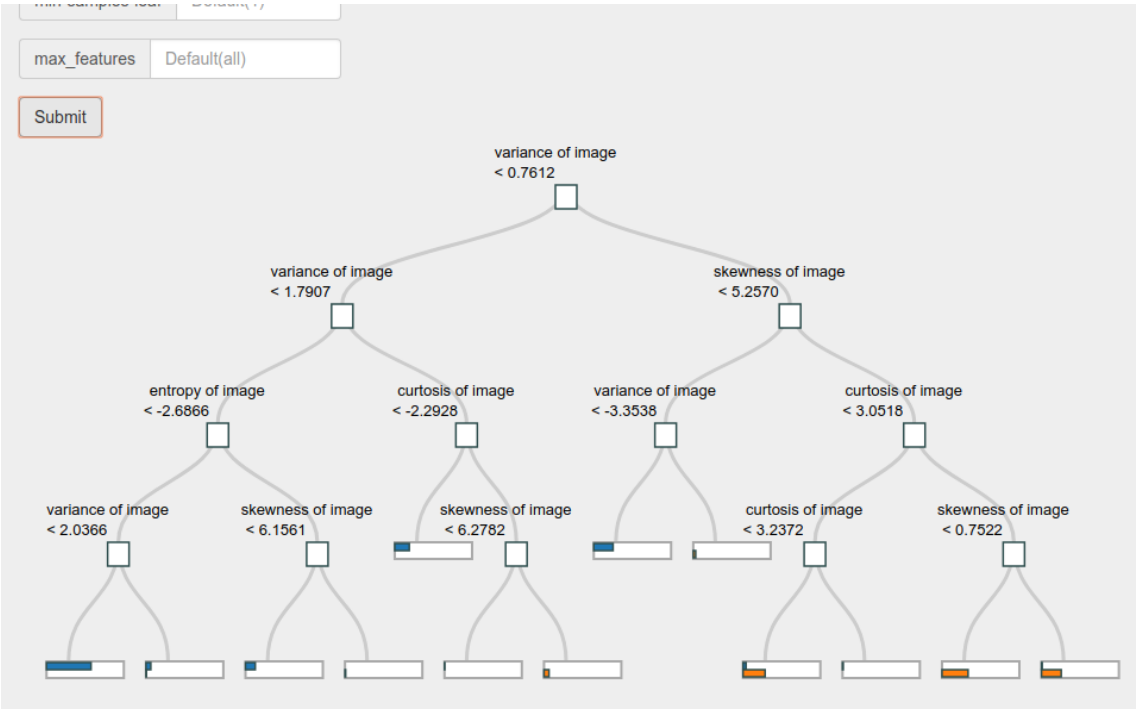

Figure 11: Bank note data set pruned

4.0.2 Random forest

This view also has three columns. First column has data and parameter input fields, second column has legend which shows the color map for attributes instead of labels and third section has feature importance and accuracy.

The main aim for this visualization is to show the feature importance and interaction. For this I traversed through all the trees and incremented the corresponding feature count if it showed up. I used this count to display the number of times a feature has shown up at a particular position with hue representing attributes. On tool-tip, the name of feature along with it's count is displayed.

Un-pruned Random forest on Bank note data set

I used the bank note data set and observed the feature count without pruning and I observed from the figure below that, after fifth level most of the nodes are just leaf nodes (color: green according to legend) meaning further levels are not adding much information to the classifier.

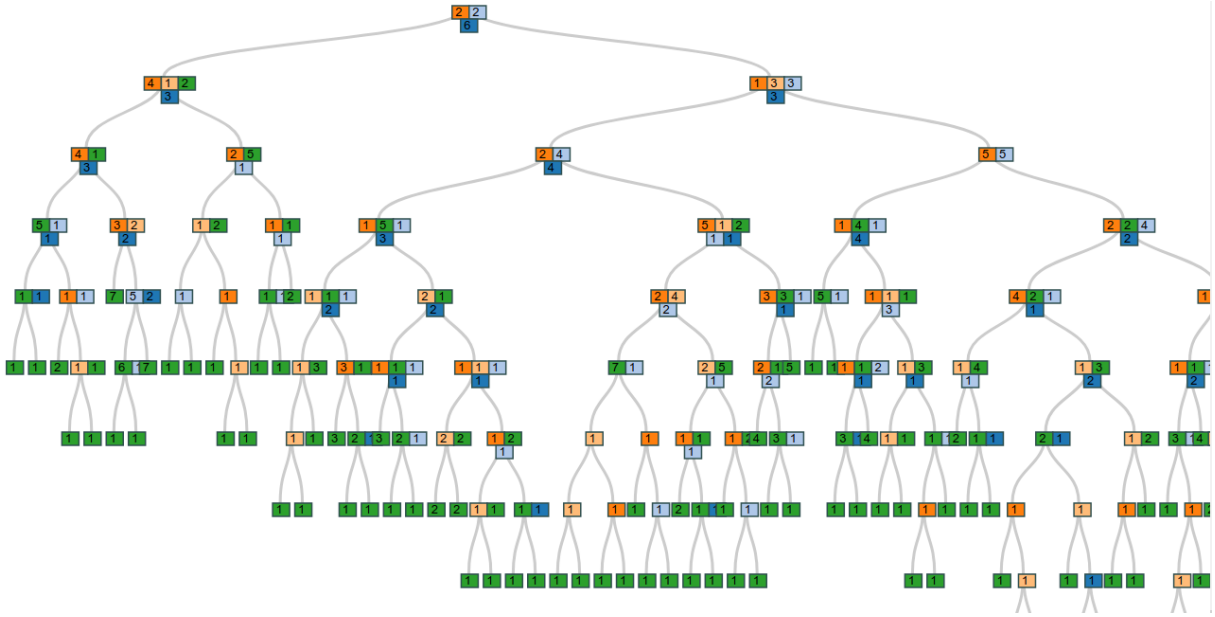

Figure 12: Bank note data - Random forest without pruning

So I have set the number of trees to 10 and maximum depth to 5 and the aggregation of all trees looked like the below figure with a good model accuracy

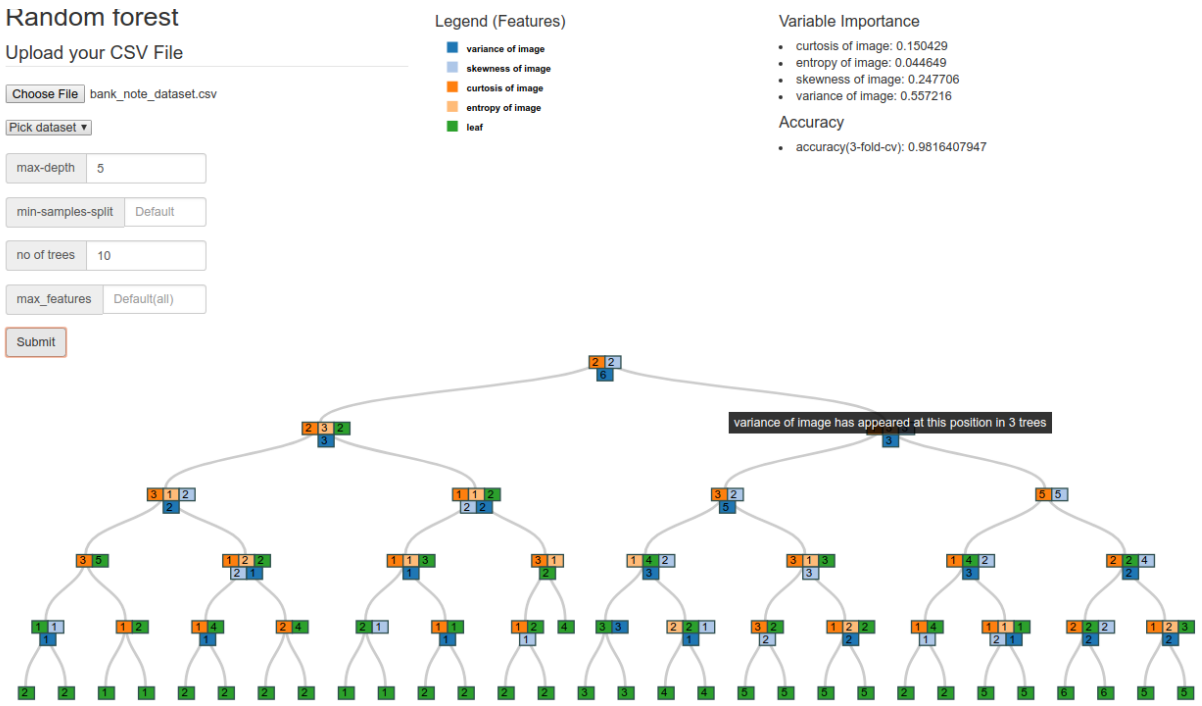

Figure 13: Bank note data - Random forest pruned

Since there is a space constraint on the image size that can be included in the report, I had to shrink the images to show the full view as a consequence their clarity decreased.

As we can observe there are 6 trees that have "variance of image" as their root node and if we see the importance of "variance of image" it has high importance as compared to others. This can be observed about other features at different levels of the layout. Also, we can observe the interactions between different features at different levels.

I have tried Pima Indians data set with 5 tree depth and 15 trees and I made the similar observations about the variable importance as above. I also noted that as I was increasing the trees and decreasing the depth the accuracy went up.

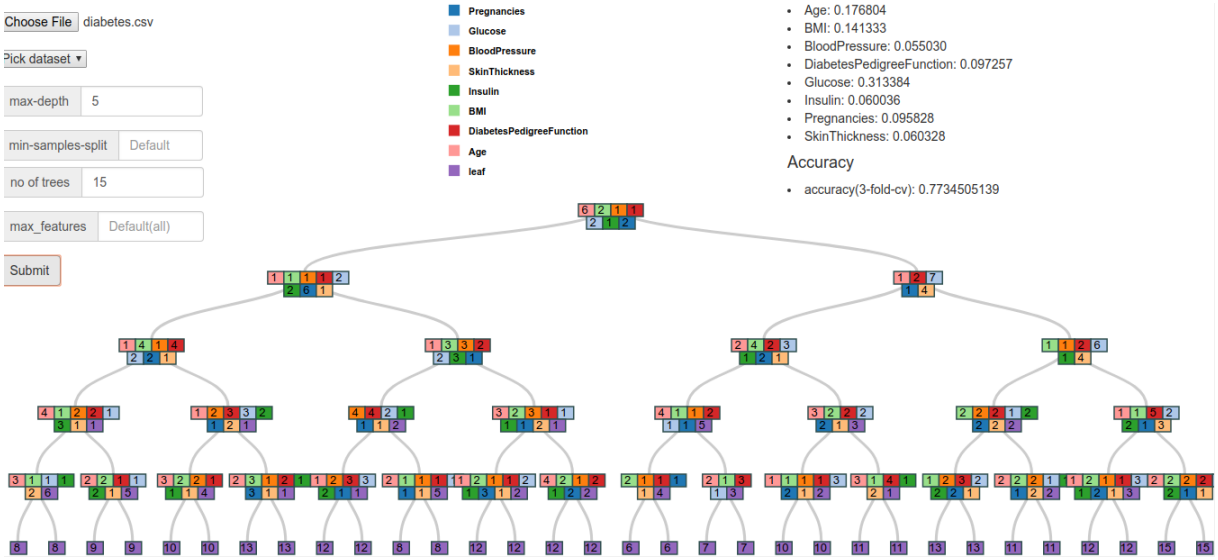

Figure 14: Pima Indians data - Random forest pruned

5 Future work

All the components in the proposed design are implemented except for the path highlighting in the individual tree view. But, few enhancements can be made to the project such as

- As trees and features increase, the separation between nodes should be taken care of.
- For each feature in the random forest view, the rules distribution can be visualized in the tool tip. This can illustrate the distribution of rules in the model.

Since, the basic pipeline is setup, other layouts such as treemap or aggregated hierarchical layouts can be tried as they seem to scale well for more number of features and labels.

6 References

- `http://scikit-learn.org/stable/auto_examples/tree/plot_unveil_tree_structure.html`
- `http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`

- http://www.d3noob.org/2014/01/tree-diagrams-in-d3js_11.html
- https://planspace.org/20151129-see_sklearn_trees_with_d3/
- https://wellecks.wordpress.com/tag/heatmap-tree/