Motivation

- Decision tree is one of the important algorithms of machine learning. But the process of pruning and features selection is not so obvious by the results and parameters themselves. Also, some features might be given high priority due to some unintended biases of the training data. This project combines the advantages of machine learning and human judgement for decision making to optimize problem.
- Challenges for this project:
  - The design of the pruning process visualization which should clearly show the changes of tree structure and features used at each decision node.
  - The front-end implementation of visualization with D3, which should demonstrate the transitions as wanted.
  - The back-end should allow changing of the feature at a given node appointed by user with as few modifications on the original tree as possible.

Problem

This project aims to provide users with the functionality to prune the decision tree in various ways and see the transition process of the tree being pruned. So that when users are not satisfied with the training result from the machine learning algorithm, they could modify the trained decision tree via the graphical interface and see how their modifications affect factors such as training accuracy, structure of the tree and the partition of training samples by each decision node.

Approach

- **Programming language:** javascript
- **Major packages:** D3.js and node.js machine learning package
- **Algorithms involved:**
  - **Decision Tree Building:**
    - Decision tree is built in a top-down fashion. To decide which feature to split at each node, maximum entropy decrease for children nodes is calculated for each feature, the one with the largest decrease becomes the feature at that node and the corresponding split to the maximum decrease becomes the threshold.
  - **Basic pruning:**
    - Make the node chosen by user a leaf node and prune its subtree if any.
  - **Automatic pruning:**
    - Given an entropy gain threshold specified by user, prune the decision node if replacement of it by a leaf node predicting majority class has an entropy gain below threshold.
  - **Pruning with Constrain(Advanced pruning):**
    - Given a decision node user is not satisfied with, use samples partitioned to that node to retrain the subtree with that decision node as root with the decision tree building algorithm. Refrain the algorithm from using the feature of the unsatisfying decision node for root.

Results

- **Visualization Designs:**
  - Color of each node signifies major class for current node.
  - Mouse hover over a node triggers tooltip and stacked bar for sample partition information, aids user to make decision on manual pruning.
  - Animated the process of decision tree being pruned/unpruned as collapsing/expanding of nodes, making the changes easier to follow.
  - A thumbnail plot to highlight pruning result of previous step with fully expanded tree in the background, helps user to keep track of the pruning process.
  - Training accuracy simultaneous updated as decision tree being modified

- **Interactions:**
  - Basic pruning triggered by left click on decision node.
  - Advanced pruning triggered by right click on decision node.
  - A slider bar for setting the pruning threshold for automatic pruning.

Future Work

- Explore effective visualization of changed conditions for decision nodes.
- In-depth study about the optimizing the pruning algorithm, current algorithms applied requires a full pass through all the data partitioned to the nodes of interest and all the features, necessary for dealing with situations involving large amount of data.
- More metrics for critical value pruning choices.
- Visualization of other pruning methods like reduced-error pruning, minimum error pruning and randomized pruning.