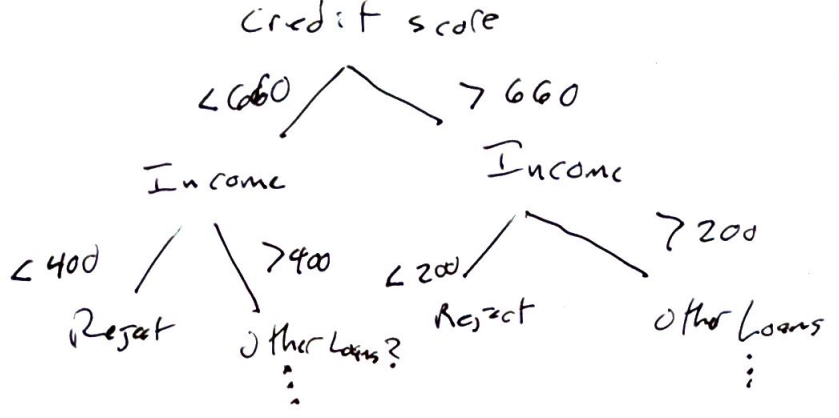
# Supervised Learning: Decision Trees and Random Forests

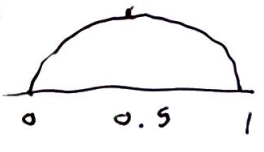
Example: Loan request - will it be approved (yes or no)
Training set with multiple attributes, label




Models how you'd make the decision

Entropy: Relative measure of disorder in the data source

$$H(x) = -\sum_{i=1}^{N} P(x_i) \log(P(x_i))$$




Classification process: looking to reduce disorder
- Group the observations in a data set

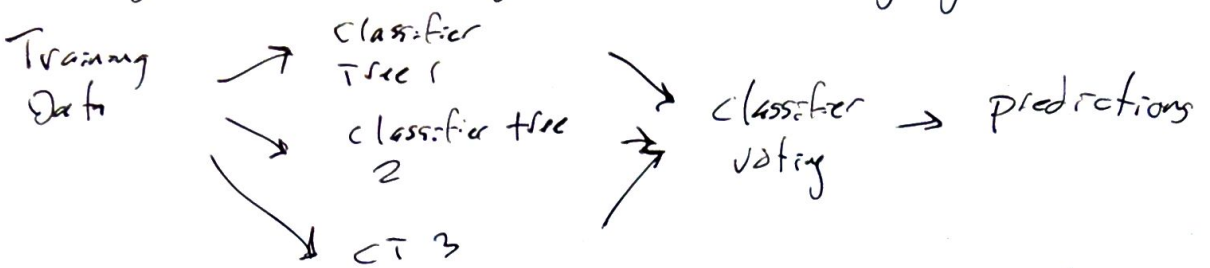
- Nodes are split based on the feature that has the largest ~~ent~~ information gain (IG) between parent node and its split nodes
- One metric to quantify IG is to compare entropy before and after splitting
- In a binary case, entropy is 0 if all samples belong to the same class for a node, entropy is 1 if samples contain below both classes with equal proportion
- The splitting process can go iteratively at each child node until the end-nodes (or leaves) are pure
  - Splitting usually stops at certain criteria to avoid over fitting

Decision Trees

- Train (build the tree) by maximizing IG to choose splits
- Easy to interpret
- Expressive = flexible
- Less Need for feature transformations
- Susceptible to over fitting
- Must prune to avoid overfitting
- scikit-learn: sklearn.tree.DecisionTreeClassifier

# Ensemble Method

Learn multiple models & combine their results, usually via majority vote or averaging




Randomly train models on random subsets and features from the training data

## Random Forest Algorithm

- Set of decision trees, each learned from diff randomly sampled subset with replacement
- Features to split on for each tree, randomly selected subset from original features
- Prediction: Average output probabilities
- Increases diversity through random selection of training dataset and subset of features for each tree
- Reduces variance through averaging
- Each tree typically does not need to be pruned
- More expensive to train and run
- sklkit-learn: sklearn.ensemble.RandomForestClassifier