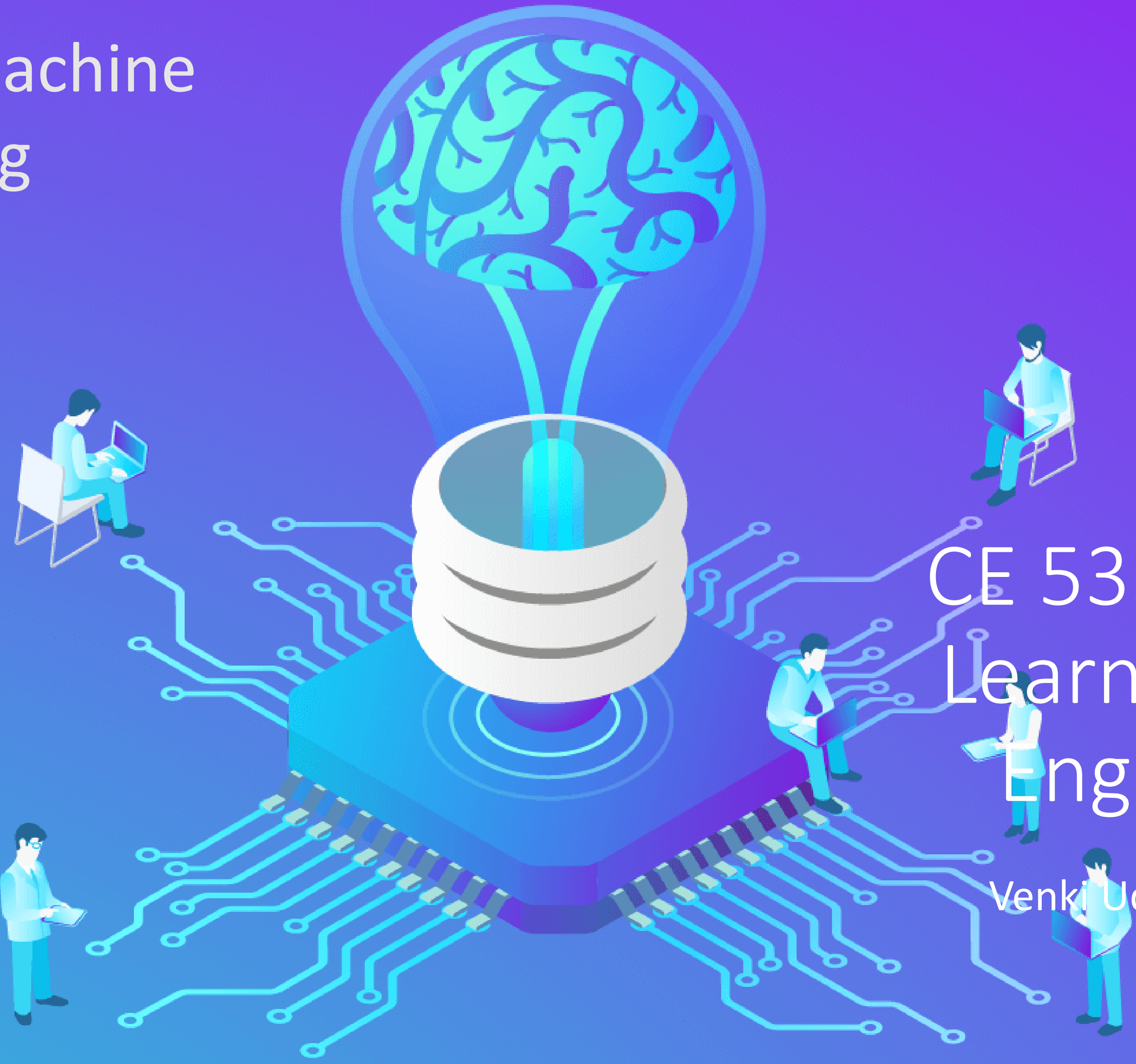
Python for Machine Learning

Decision Trees
Introduction



CE 5331 Machine Learning for Civil Engineers

Venki Uddameri, Ph.D. , P.E.

Recap

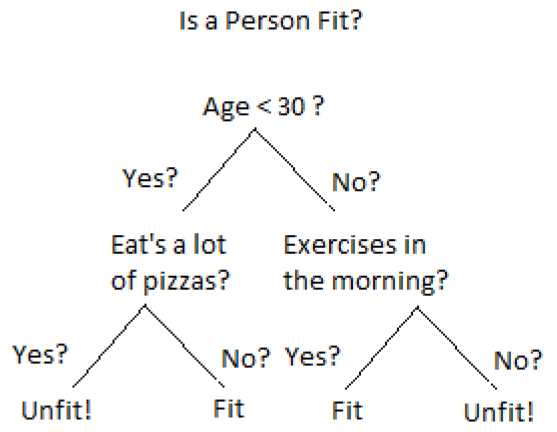
- What is Machine Learning
- How is it useful for Civil Engineers
- Overview of Machine Learning Methods
- Linear Regression
  - Bivariate
  - Regression interpretation
  - Multivariate
- Logistic Regression
  - Maximum likelihood estimation
  - Regularization (introduction)
- Naïve Bayesian Classifier
  - What is it
  - What makes it naïve
  - Bayes theorem
  - Prior, likelihood and posterior
- K-Nearest Neighbor
  - How does the algorithm work
  - Why is it a lazy learner
  - How to do regression and classification

Python – Introduction
Python – Functions
Python - Pandas
Python – np, scipy, statsmodels
Python – Scikit learn – linear, metrics
Python – Matplotlib, seaborn
Python – Mixed_Naive_Bayes
Python – scikit learn neighbors module

In this module we shall look at Classification Trees

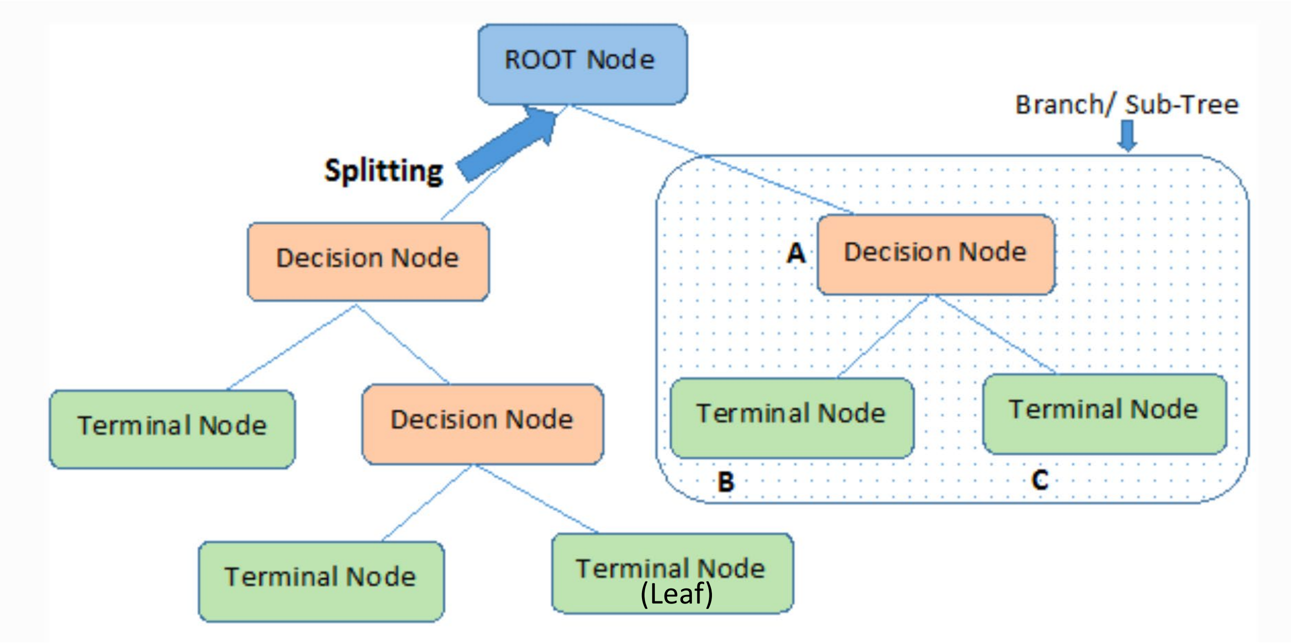
Decision Trees

- A supervised machine learning algorithm for classification and regression
- Useful when transparency is important
- The algorithm encodes the underlying relationship as a set of IF-THEN rules
  - IF-THEN rules are nested
- The IF-THEN rules can be visualized as a tree
  - Branches and leaves



Decision Trees - Terminology

- A Decision Tree comprises of 3 elements
  - Root node
  - Decision nodes
  - Leaf/Terminal nodes
- Branch /sub-tree
  - A sub-tree is a portion of a tree



Splitting: The process of splitting a node into two branches
Pruning: The process of removing sub-nodes (opposite of splitting)

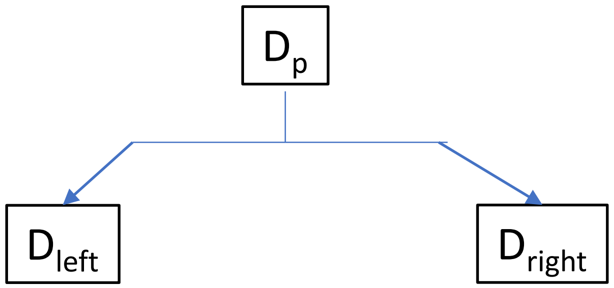
A node divided into sub nodes is called a parent node and nodes that stem from these parent nodes are called child nodes

Decision Tree Construction – Decisions

- Decision tree algorithms prefer the features (input) to be discrete
  - If they are continuous then they are made discrete first
  - How should this discretization take place?
- Which feature (input) should be used as the root node?
  - What do we gain (or lose) by making a feature (input) a root node?
  - How should we measure this loss or gain?
- Which feature (input) should be used as a decision node?
  - What do we gain (or lose) by making a feature (input) a root node?
  - How should we measure this loss or gain?
- When do we stop adding branches?
  - How much splitting is necessary?
- Is the final tree structure too complex?
  - Can we simplify the structure without losing much?
    - Model parsimony
  - Can we avoid overfitting
    - Is the tree able to generalize the data

Statistical criteria are used to make these decisions

How to Fit a Decision Tree Model

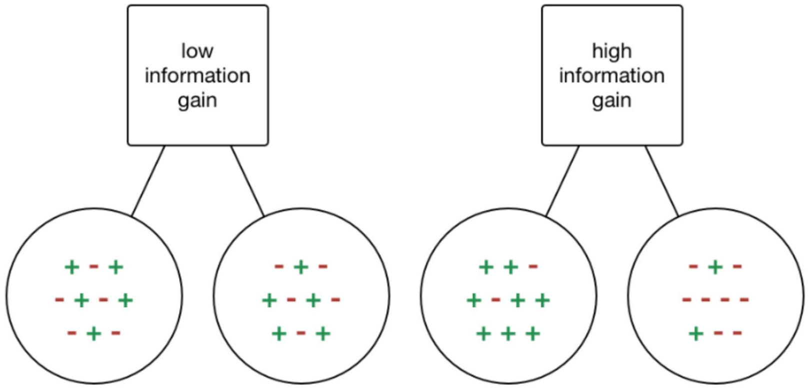


- The typical objective is to maximizing information gain
  - Information gain is maximized at each split
- Identify the feature that provides the maximum information gain at each split
  - While multiple features can be simultaneously used to make a split only one feature is used at a time
- For simplicity each parent node is split into two nodes
  - Binary classification trees

IG is the information Gain
I is the Impurity Measure

Parent

$$IG\left(D_p, f\right) = I(D_p) - \underbrace{\frac{N_{left}}{N_p} I\left(D_{left}\right) - \frac{N_{right}}{N_p} I\left(D_{right}\right)}_{\text{Child Node}}$$



Starting with the Root node → loop through each feature to compute the IG (combinatorial optimization problem)

Impurity Measures

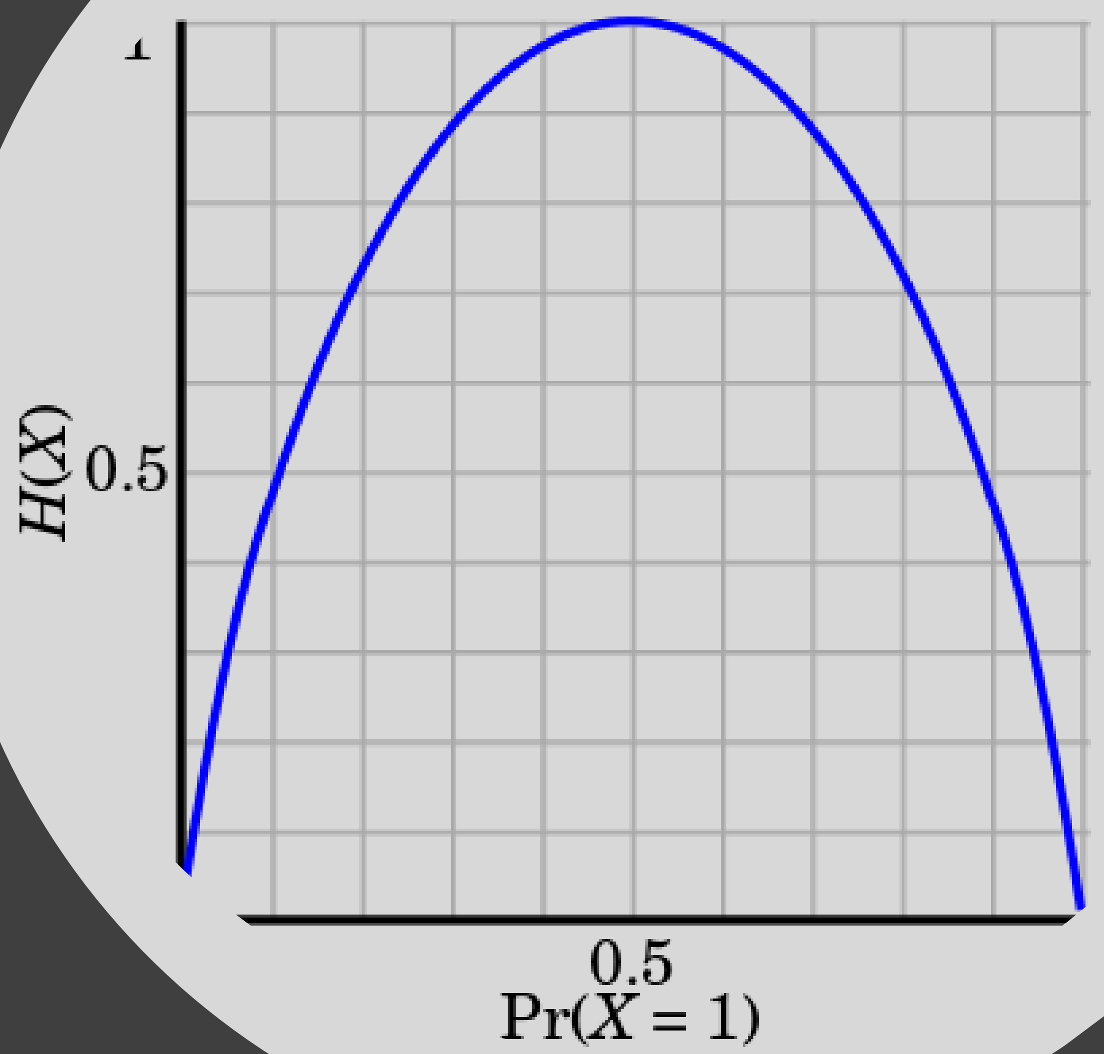
- Different Impurity Measures have been prescribed in the literature
  - Entropy
  - Gini Impurity
  - Classification Error
  - Gain Ratio
  - CHAID –
    - Chi-Square Automatic Interaction Detector
  - Variance Minimization
    - Use for continuous variable

Most common in classification trees

Most common in regression problems

Entropy

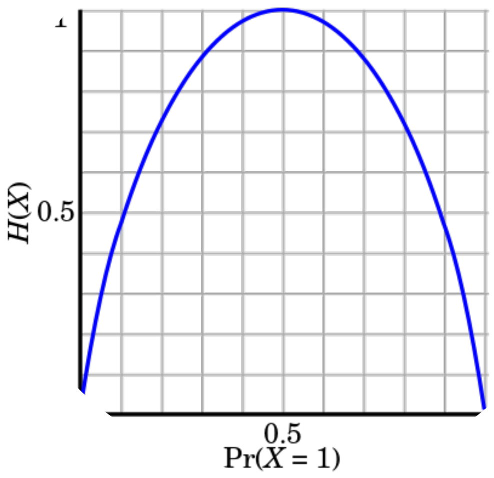
- Entropy is a measure of information content (or lack thereof) in a dataset
- When the entropy is high there is very little information known
  - Truly stochastic process
  - Hard to predict which state is more (or less preferred)
- Entropy is also a measure of diversity
  - More entropy → greater diversity



Information entropy is similar in spirit to the thermodynamic concept of entropy

Entropy - Calculations

- There are many definitions of Entropy
  - The Shannon's entropy is most common
  - Entropy is maximal when the classes are perfectly mixed
    - Uniform distribution



Number of classes

$$I_H = -\sum_{i=1}^{c} p(i|t) log_2 p(i|t)$$

Entropy

Proportion of samples at node 't' that belong to class "i"

For a binary (2 class) case:
Entropy is zero when probability is zero or one
Entropy is maximum when the probability is 0.5

Impurity – Gini Coefficient

- Gini Coefficient
  - A Measure of probability of misclassification
- Gini Coefficient is similar to Entropy
  - Can pick one or the other as the impurity measure
  - The cut-off thresholds have greater sensitivity then the impurity measures
- Gini Coefficient
  - Maximum is the classes are perfectly mixed
    - Uniform distribution across classes
      - For binary case p = 0.5

Gini Impurity Coefficient

$$I_g = \sum_{i=1}^{c} p(i|t)\,(1 - p(i|t)) = 1 - \sum_{i=1}^{N} p(i|t)^2$$

Proportion of values in ith class at node t

Impurity – Classification Error

- A measure of de minimis error of the classifier
- Classification error is not good at the splitting stage
  - Not sensitive to changes in probabilities at nodes
- Classification error is used to prune the tree

Classification Error

$$I_E = 1 - max\left[p(i|t)\right]$$

Proportion of values in ith class at node t

Example

- Compute the Entropy, Gini Coefficient, Classification Error corresponding to the following splits
- Compute the Information gain (IG) based on the above impurities

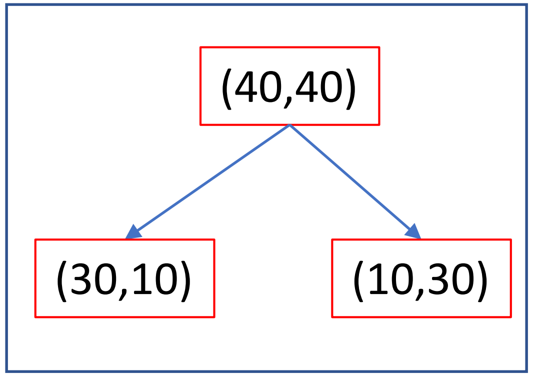
Relevant Equations

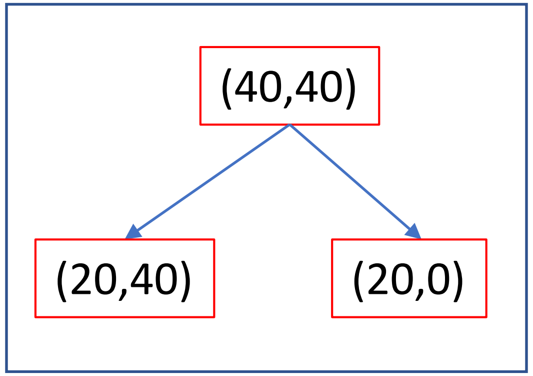
$$IG\left(D_p, f\right) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

$$I_H = \sum_{i=1}^{c} p(i|t) log_2 p(i|t)$$

$$I_g = \sum_{i=1}^{c} p(i|t)\left(1 - p(i|t)\right) = 1 - \sum_{i=1}^{N} p(i|t)^2$$

$$I_E = 1 - max\left[p(i|t)\right]$$





Building Decision Trees

- Complex Decision Trees can be built by dividing the feature space into rectangles
- However a complex tree has a very high potential of overfitting the data
  - Memorizing the training data with poor generalization abilities
- You can control the depth of the Decision tree
  - You will have to play with this to find an optimal pruned tree

Decision Tree Algorithms

- D3 (obsolete)
- ID3 → Successor of D3
- C4.5 → Successor of ID3
- Classification and Regression Trees (CART)
- Multi-Adaptive Regression Splines

Methods	CART	C4.5	CHAID	QUEST
Measure used to select input variable	Gini index; Twoing criteria	Entropy info-gain	Chi-square	Chi-square for categorical variables; J-way ANOVA for continuous/ordinal variables
Pruning	Pre-pruning using a single-pass algorithm	Pre-pruning using a single-pass algorithm	Pre-pruning using Chi-square test for independence	Post-pruning
Dependent variable	Categorical/ Continuous	Categorical/ Continuous	Categorical	Categorical
Input variables	Categorical/ Continuous	Categorical/ Continuous	Categorical/ Continuous	Categorical/ Continuous
Split at each node	Binary; Split on linear combinations	Multiple	Multiple	Binary; Split on linear combinations

Features	*ID3*	*C4.5*	*CART*
Type of data	Categorical	Continuous and Categorical	continuous and nominal attributes data
Speed	Low	Faster than ID3	Average
Boosting	Not supported	Not supported	Supported
Pruning	No	Pre-pruning	Post pruning
Missing Values	Can't deal with	Can't deal with	Can deal with
Formula	Use information entropy and information Gain	Use split info and gain ratio	Use Gini diversity index

Decision Tree Algorithm

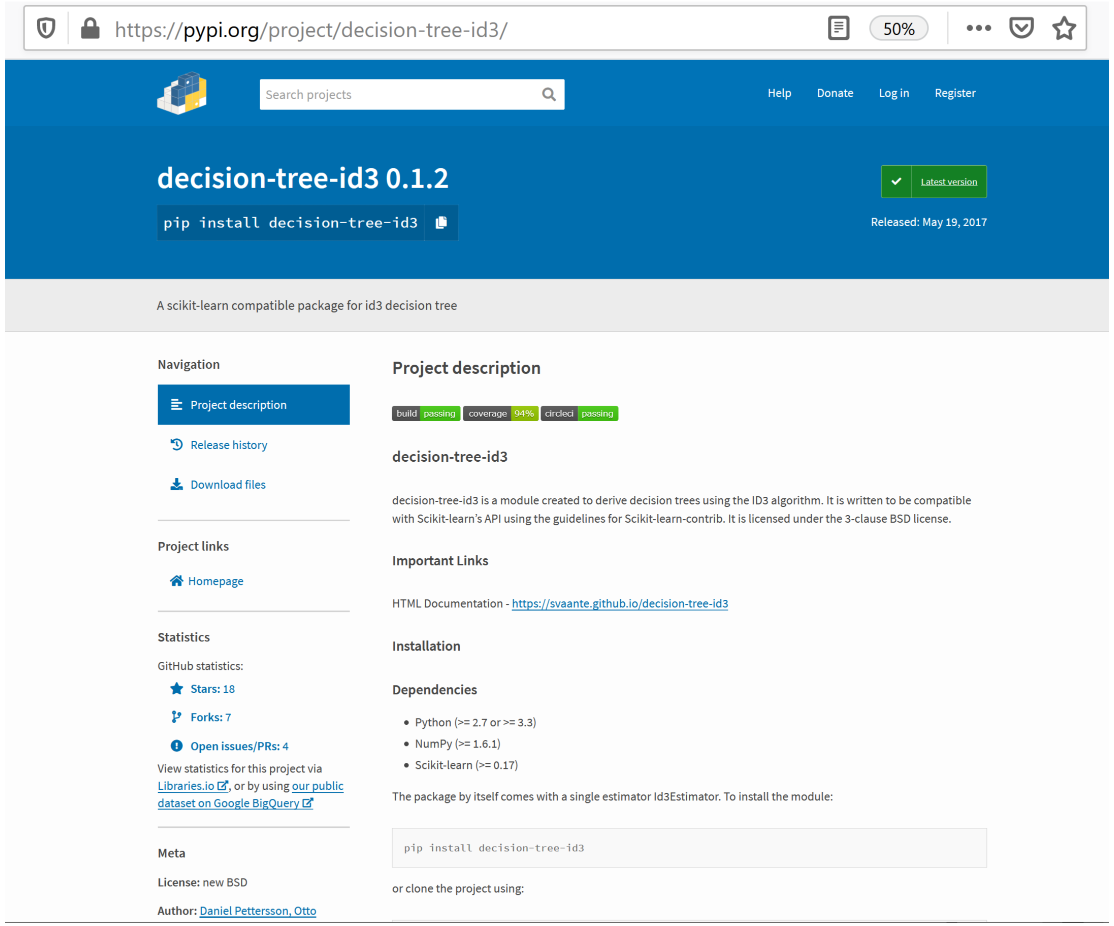
- ID3 is a very basic algorithm
- Works only with categorical features
  - You have to discretize continuous features prior to applying the method
- Does not allow for automatic pruning
  - You can do trial and error
  - Entropy and Information Gain

Steps in ID3 algorithm:

1. It begins with the original set S as the root node.
2. On each iteration of the algorithm, it iterates through the very unused attribute of the set S and calculates **Entropy(H)** and **Information gain(IG)** of this attribute.
3. It then selects the attribute which has the smallest Entropy or Largest Information gain.
4. The set S is then split by the selected attribute to produce a subset of the data.
5. The algorithm continues to recur on each subset, considering only attributes never selected before.

ID3 Algorithm in Python

- The library sklearn uses CART as default
- There is a scikit learn type package on PIP
  - You can install by going to the anaconda prompt
- There are a number of different default parameters to control
  - the growth of the tree: - max_depth, the max depth of the tree. –
  - the minimum number of samples in a split to be considered. – min_samples_split,
  - prune, if the tree should be post-pruned to avoid overfitting and cut down on size



Note all your features must be categorical and this model is more for developing a basic understanding of Decision Trees

You should know

- What are decision trees
- What are their advantages
- How do decision tree algorithms work
- Basic elements necessary for building decision trees
  - Entropy
  - Gini Index
  - Information Gain
- Various algorithms for building decision trees
- ID3 algorithm workings

Python
decision_tree_ID3
package