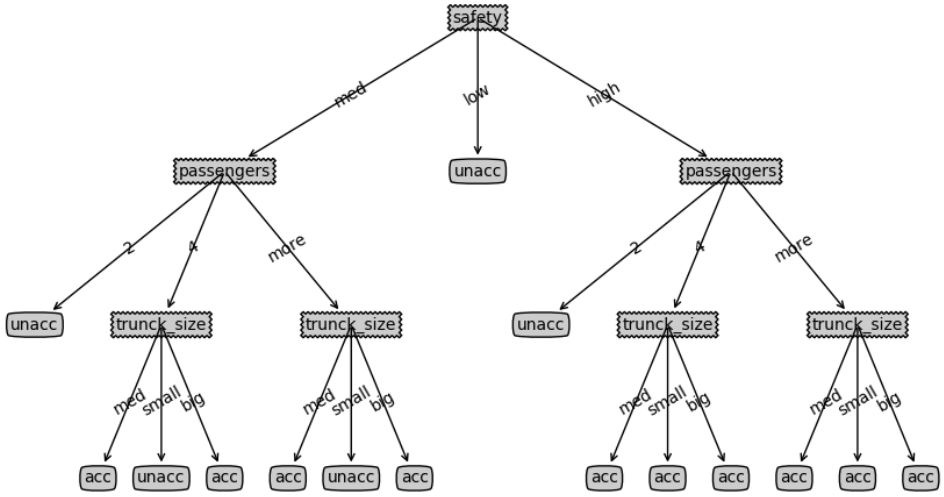
CSE 469: Assignment 4
Decision Tree Classification

Peter M. VanNostrand

11/25/2019

ASSIGNMENT 4: DECISION TREE CLASSIFICATION

Classification Tree: Car Dataset



Decision Tree Code

Stop Criteria

```
def stopCriteria(dataSet):
    # Count the frequency of all labels
    label_freq = {}
    for sample in dataSet:
        # Get the label of the sample
        label = sample[-1]
        # If its a novel label, start with frequency 0
        if not (label in label_freq):
            label_freq[label] = 0
        # Increment the frequency
        label_freq[label] += 1

    # Find the most frequent label
    max_freq = 0
    assignedLabel = None
    for label, freq in label_freq.items():
        if freq > max_freq:
            max_freq = freq
            assignedLabel = label

    # If there is theres more than one label and features to be split
    if max_freq!=len(dataSet) and len(dataSet[0])>1:
        assignedLabel = None

    return assignedLabel
```

Choose Best Feature

```
def chooseBestFeature(dataSet):
    n_samples = len(dataSet) # total number of samples
    m_features = len(dataSet[0]) - 1 # number of features
    gain = [1] * m_features
    for i in range(0, m_features):
        gain[i] = calcGini(dataSet)
        for value in getVals(dataSet, i):
            subset = splitData(dataSet, i, value)
            n_subset = len(subset)
            gini_subset = calcGini(subset)
            gain[i] -= (n_subset / n_samples) * gini_subset

    bestFeatId = gain.index(max(gain))
    return bestFeatId
```

Calc Gini

```
def calcGini(dataSet):
    label_freq = {}
    for sample in dataSet:
        label = sample[-1]
        if not (label in label_freq):
            label_freq[label] = 0
        label_freq[label] += 1

    total_freq = len(dataSet)
    gini = 1
    for freq in label_freq.values():
        gini -= (freq/total_freq)**2

    return gini
```

Get Vals

```
def getVals(dataSet, feature_id):
    dict_vals = {}
    for sample in dataSet:
        value = sample[feature_id]
        dict_vals[value] = 0
    return dict_vals.keys()
```