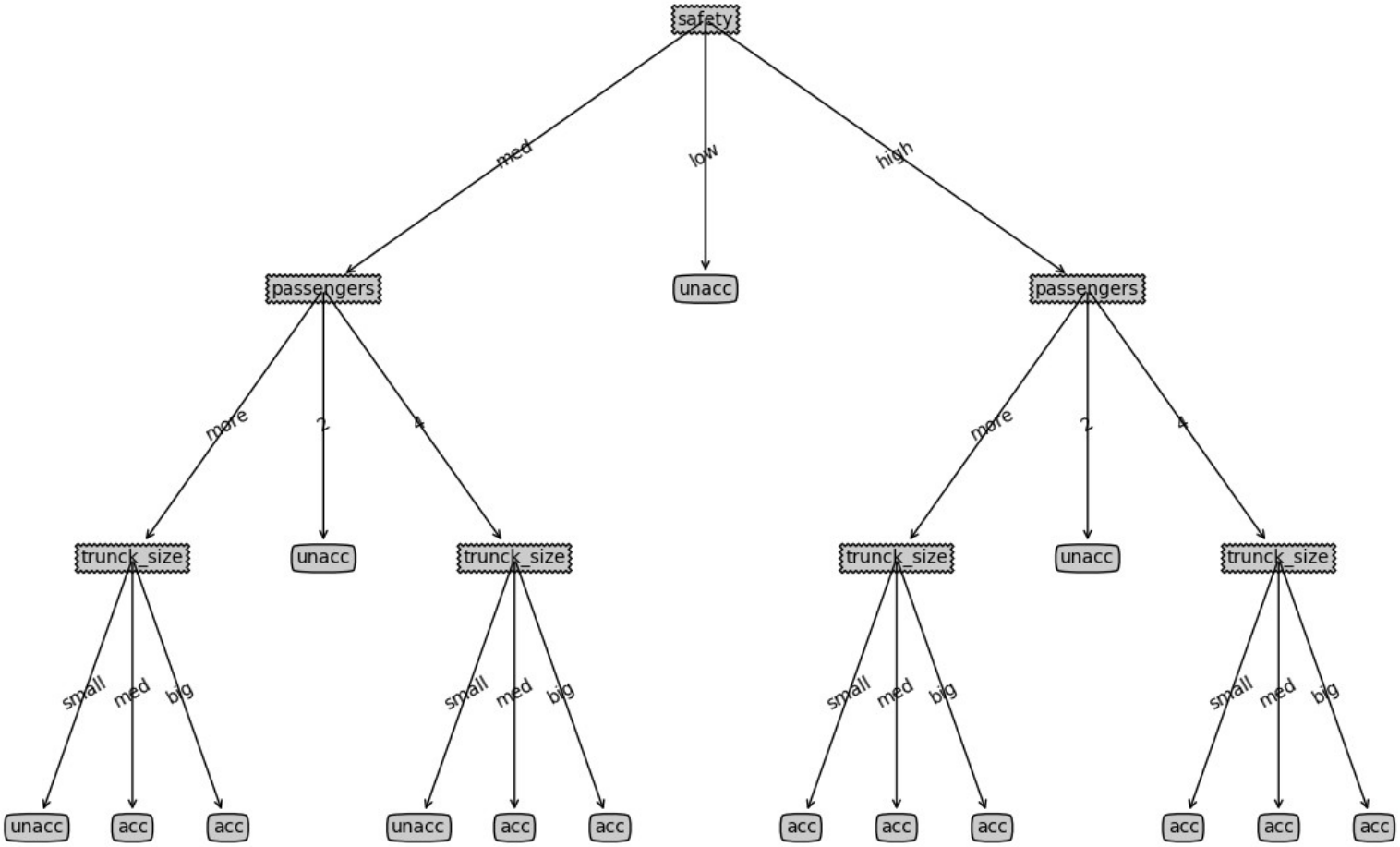
Nate Dauterman
ECE 49595
April 23rd, 2022

Car.csv Decision Tree



Code for stopCriteria function

```
def stopCriteria(dataSet):
    '''
    Criteria to stop splitting:
    1) if all the classe labels are the same, then return the class label;
    2) if there are no more features to split, then return the majority label of the subset.

    Parameters
    ----------
    dataSet: 2-D list
        [n_sampels, m_features + 1]
        the last column is class label

    Returns
    -------
    assignedLabel: string
        if satisfying stop criteria, assignedLabel is the assigned class label;
        else, assignedLabel is None
    '''
    assignedLabel = None

    lastColumn = [row[len(row) - 1] for row in dataSet]

    if lastColumn.count(lastColumn[0]) == len(lastColumn):
        assignedLabel = lastColumn[0]
    elif len(dataSet[0]) == 1:
        assignedLabel = max(lastColumn, key = lastColumn.count)

    #print(assignedLabel)
    return assignedLabel
```

Code for chooseBestFeature function

```
def chooseBestFeature(dataSet):
    '''
    choose best feature to split based on Gini index

    Parameters
    ----------
    dataSet: 2-D list
        [n_sampels, m_features + 1]
        the last column is class label

    Returns
    -------
    bestFeatId: int
        index of the best feature
```

```
    '''

    uniqueItems = []
    for i in range(len(dataSet[0])):
        uniqueItems.append(set([row[i] for row in dataSet]))
    #print(uniqueItems)

    masterGini = 1
    for label in uniqueItems[len(uniqueItems) - 1]:
        labelCol = ([row[len(row) - 1] for row in dataSet])
        masterGini -= (labelCol.count(label) / len(labelCol)) ** 2
    #print(masterGini)

    gains = []
    for colInd, col in enumerate(uniqueItems[:-1]):
        copyMasterGini = masterGini
        for item in col:
            rowLabels = [dataSet[i][len(uniqueItems) - 1] for i, x in enumerate(dataSet) if x[colInd] ==
item]

            gini = 1
            for label in uniqueItems[len(uniqueItems) - 1]:
                gini -= (rowLabels.count(label) / len(rowLabels)) ** 2
            #print(gini)

            copyMasterGini -= ([row[colInd] for row in dataSet].count(item) / len([row[colInd] for row in
dataSet])) * gini

        gains.append(copyMasterGini)

    #print(gains)
    maxGains = max(gains)
    bestFeatId = gains.index(maxGains)

    return bestFeatId
```