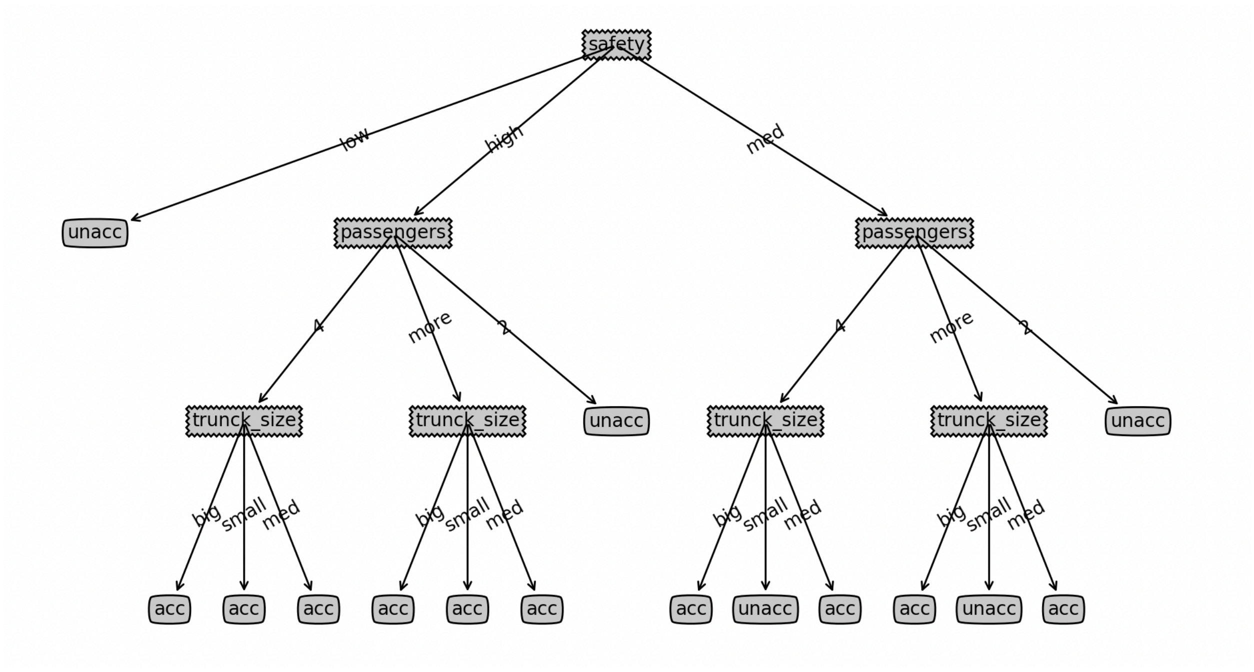
Assignment4

1. The tree drawn based on the output obtained from the Car Dataset using your algorithm.



2. The code of the two functions that you implement.

```
def calGini(dataSet):
    # get number of unique class labels
    labelCounts = {}
    for instance in dataSet:
        label = instance[-1]
        if label not in labelCounts:
            labelCounts[label] = 0
        labelCounts[label] += 1

    # print(labelCounts) ## {'No': 5, 'Yes': 9}....
    Gini_subtrahend = 0.0
    totalInstances = len(dataSet)
    for label, count in labelCounts.items():
        # print(label, count) ## No 3
        Gini_subtrahend += (count / totalInstances) ** 2

    score = 1 - Gini_subtrahend

    return score

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
    '''
    #TODO

    # for each feature i in the dataset
    #      calculate gini index on dataset
    #      for each value of the feature
    #              subset = splitData(dataset, i, value)
    #              calculate gini index on the subset
    #      calculate Gain for feature i
    #  Find the bestGain and the corresponding feature id

    # print(dataSet[0]) ## ['Rainy', 'Hot', 'High', 'FALSE', 'No'],
    feature_scores = {}

    for i in range(len(dataSet[0]) - 1):
        Gini = calGini(dataSet)

        # find unique values of the current feature column
        feature_values = set() # set ensure uniqueness
        for instance in dataSet:
            feature_values.add(instance[i])
        # print(feature_values) ## {'Overcast', 'Sunny', 'Rainy'}


        Gini_subtrahend = 0.0
        for feature_val in feature_values:
            data_subset = splitData(dataSet, i, feature_val)
            # print(data_subset) ## ['Hot', 'High', 'FALSE', 'Yes'], ...

            gini = calGini(data_subset)
            Gini_subtrahend += (len(data_subset) / len(dataSet)) * gini

        feature_scores[i] = Gini - Gini_subtrahend

    bestFeatId = max(feature_scores, key=feature_scores.get) # key=feature_s
cores.get means get the value of the dict
    return bestFeatId


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
# TODO

# if all the classe labels are the same, then return
# the class label
allSame = True
for i in range(len(dataSet)-1):
    # print the class labels
    # print(dataSet[i][4],"\n")
    if dataSet[i][-1] != dataSet[i+1][-1]:
        # class labels are not the same
        allSame = False
        break
if allSame:
    assignedLabel = dataSet[0][-1]
    return assignedLabel



# if there are no more features to split, then return
# the majority label of the subset.
# noMoreFeatures = True
# for i in range(len(dataSet)-1):
#     for j in range(len(dataSet[i])-1):
#         # print the feature values
#         print(dataSet[i][j],end=" ")
#         if dataSet[i][j] is None:
    #           # no more features to split
    #           break
    #     print("")

    # only class label column left
    if len(dataSet[0]) == 1:
        labelCounts = {}
        for instance in dataSet:
            label = instance[-1]
            if label not in labelCounts:
                labelCounts[label] = 0
            labelCounts[label] += 1
        # majority vote
        sortedLabelCounts = sorted(labelCounts.items(), key=lambda x: x[1], reve
rse=True)
        assignedLabel = sortedLabelCounts[0][0]



    return assignedLabel
```

3. Did you use any AI tools (such as ChatGPT, Microsoft CoPilot, or similar) in completing this assignment? If so, please briefly describe how you used them (e.g., writing the codes, writing the report, etc.) and which specific AI tools you used. If not, simply answer ‘No’.

No.