Kyle Calabro

Dr. Kumar

Artificial Intelligence – Project 4

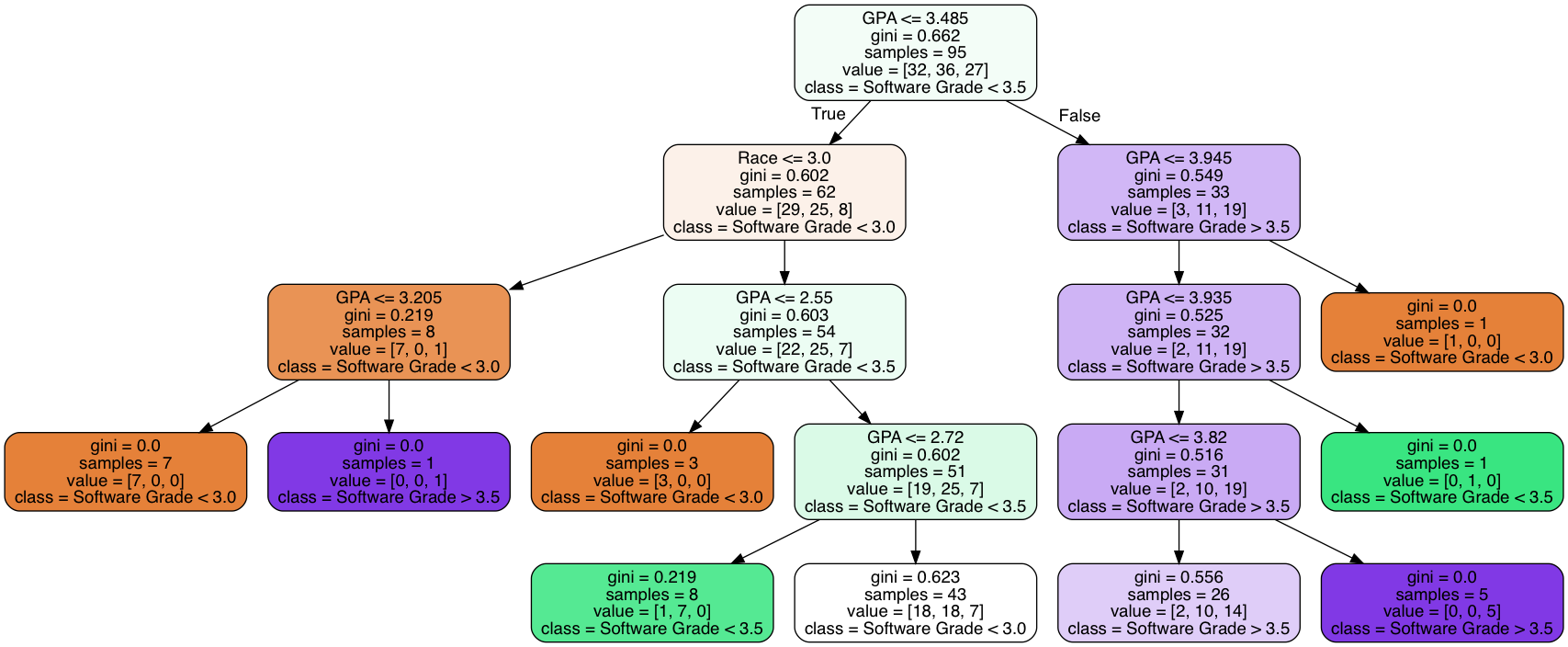
15 April 2018

Preface

* Records were selected/filtered via Microsoft Excel filters
* The Sex category was converted to binary values for every hypothesis
  + 0 represents Males
  + 1 represents Females
* For hypotheses with grades or cumulative GPA as the outcome variable:
  + Values were typically divided up into three groups

Hypothesis One

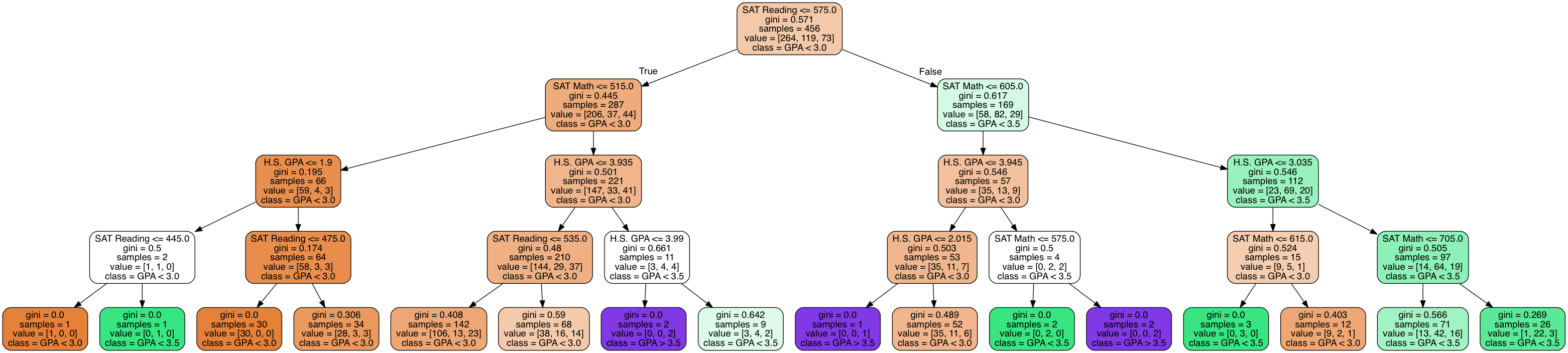
* Who does well in the Software course?
  + Students with higher Cumulative GPA’s perform better in the upper level Software course; perhaps Race and Sex also play a factor.
  + Input Variables
    - 2. Sex
    - 3. Race
    - 9. Cumulative GPA, greater than or equal to 2.5
  + Hyper-Parameter Settings
    - Random State: 7
    - Max Depth: 4
    - Test Size: .4
  + Decision Tree



* + Outcome Variables
    - 15. Grade in Software Course
  + Results
    - Random Forest Accuracies
      * Test Data Accuracy: .42
      * Train Data Accuracy: .97
    - The decision tree indeed proves that students with a Cumulative GPA above 3.485 largely performed well in the Software Course with a grade greater than 3.5. On the other hand students with a Cumulative GPA less than or equal to 3.485 largely received grades less than 3.0 in the Software Course. In these cases, Race also plays a role with races less than or equal to 3.0 receiving grades lower than 3.0 in the Software Course. Whereas, races greater than 3.485 largely received grades between 3.0 and 3.5 in the Software Course. In conclusion, the hypothesis is correct as students with a higher cumulative GPA performed well in the Software Course and race does play a factor in certain cases.

Hypothesis Two

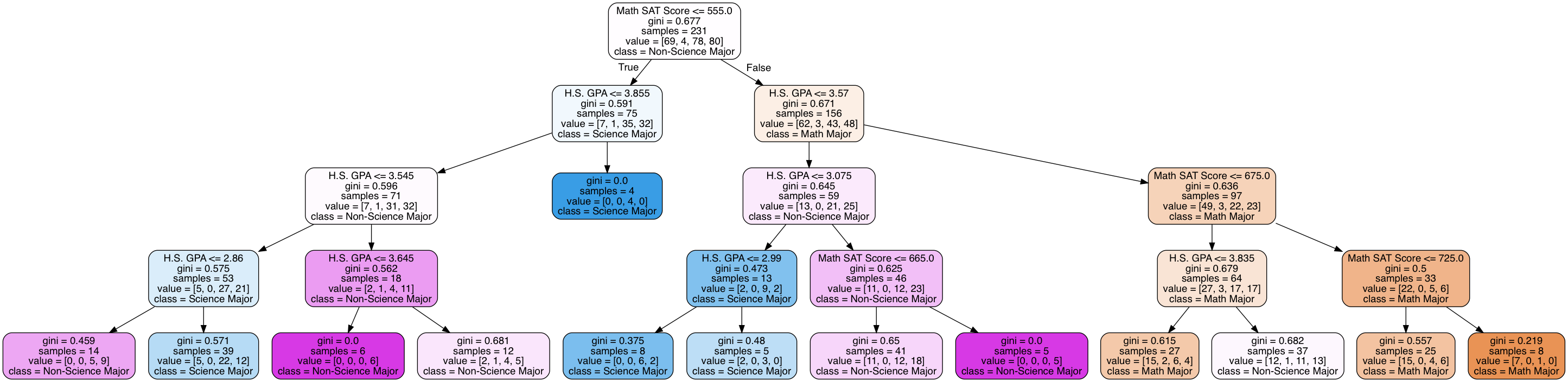
* Who does well in college?
  + Students with above average SAT Math and Reading scores as well as average High School GPA’s will perform well in college.
  + Input Variables
    - 2. Sex
    - 4. First Generation
    - 5. SAT Reading Score
    - 6. SAT Math Score
    - 7. High School GPA
  + Hyper-Parameter Settings
    - Random State: 10
    - Max Depth: 4
    - Test Size: .5
  + Decision Tree



* + Outcome Variables
    - 9. Cumulative GPA
  + Results
    - Random Forest Accuracies
      * Test Data Accuracy: .61
      * Train Data Accuracy: .99
    - The decision tree shows a strong correlation between both SAT Math and Reading scores and a students performance in college. The decision tree shows that students with a SAT Reading score greater than 575 and an SAT Math score greater than 605 largely perform well in college with GPA’s mostly in the 3.0 to 3.5 ranges. Students with low SAT Math and Reading scores however largely did not perform well in college with cumulative GPA’s under 3.0. Students with poor SAT Math and Reading scores but outstanding High School GPA’s performed well in college largely with cumulative GPA’s of 3.5 or greater. Overall, SAT scores and High School GPA are the largest factors in a student’s college performance; Sex and First Generation did not play any role.

Hypothesis Three

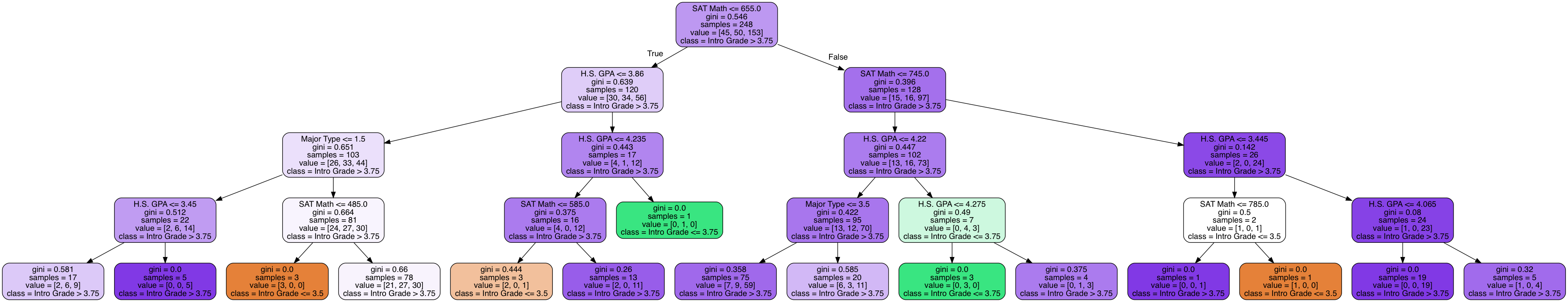
* What majors do females with above average SAT Math scores go into?
  + Females with good Math SAT scores and High School GPA’s will go into either Math or Science Majors.
  + Input Variables
    - 2. Sex, only females were considered with this hypothesis
    - 6. SAT Math Score
    - 7. High School GPA
  + Hyper-Parameter Settings
    - Random State: 6
    - Max Depth: 4
    - Test Size: .3
  + Decision Tree



* + Outcome Variables
    - 10. Major Type
  + Results
    - Random Forest Accuracies
      * Test Data Accuracy: .29
      * Train Data Accuracy: .96
    - The decision tree shows that Female students with good SAT Math scores and High School GPA’s largely enroll in college as Math Majors. The tree also shows that Females with good SAT Math scores and lower High School GPA’s largely enroll in college as Non-Science Majors. On the other hand, low SAT Math scores amongst Female’s results in a balanced enrollment of both Science and Non-Science Majors. Interestingly, no Computer Science Majors were found amongst this group. In conclusion, the hypothesis is partially correct as no Science Majors were found in the group of female’s with good SAT Math scores and High School GPA’s.

Hypothesis Four

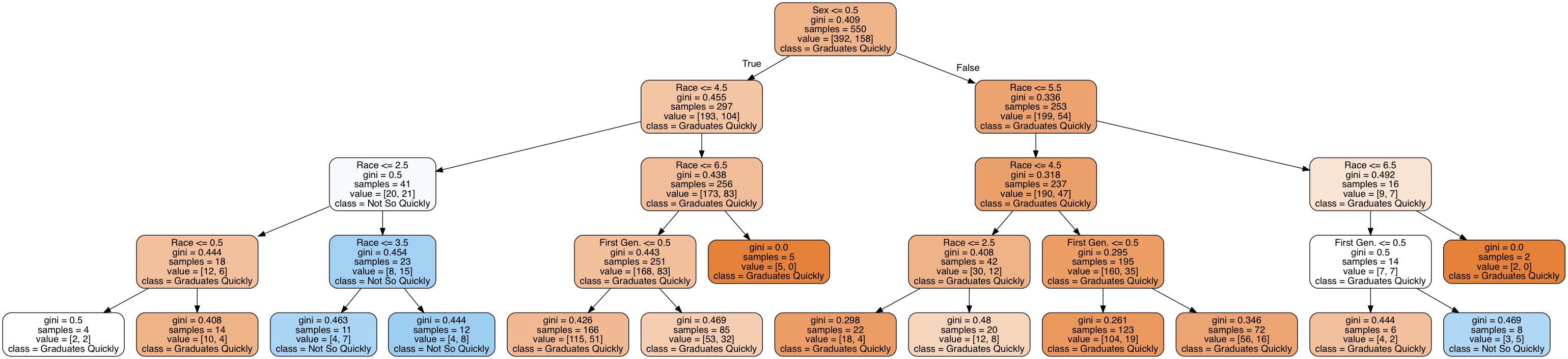
* Who does well in the Intro course?
  + Students with higher SAT Math scores and High School GPA’s will perform better in the Intro Course. Computer Science Majors will also perform better in the Intro Course regardless of SAT Math Score and High School GPA.
  + Input Variables
    - 6. SAT Math Score
    - 7. High School GPA
    - 10. Major Type
  + Hyper-Parameter Settings
    - Random State: 5
    - Max Depth: 4
    - Test Size: .2
  + Decision Tree



* + Outcome Variables
    - 11. Grade in Intro Course
      * Only examining those with a grade greater than 3.25
  + Results
    - Random Forest Accuracies
      * Test Data Accuracy: .58
      * Train Data Accuracy: .99
    - The decision tree shows that Students with good SAT Math scores largely perform very well in the Intro Course with a grade of greater than 3.75, regardless of High School GPA. The tree also shows that students with poor SAT Math scores who are enrolled as Computer Science Majors perform very well in the Intro Course with a grade of 3.75 or better, regardless of High School GPA. In conclusion, the hypothesis is largely correct as Computer Science Majors perform well in the Intro Course as well as students with good Math SAT scores. The hypothesis is slightly off as High School GPA does not play a significant factor in a students grade for the Intro Course.

Hypothesis Five

* Who graduates college quickly?
  + Females will largely graduate quickly; Males of certain races may not graduate so quickly. Whether or not a student is First Generation will not play a large role.
  + Input Variables
    - 2. Sex
    - 3. Race
    - 4. First Generation
  + Hyper-Parameter Settings
    - Random State: 20
    - Max Depth: 4
    - Test Size: .3
  + Decision Tree



* + Outcome Variables
    - 8. Semesters Taken To Graduate
      * Less than or equal to 16 is quickly
      * Greater than 16 is not so quickly
  + Results
    - Random Forest Accuracies
      * Test Data Accuracy: .73
      * Train Data Accuracy: .74
    - The decision tree shows that Males of races less than or equal to 2.5 do not graduate quickly. It also shows that Males of races > 4.5 graduate quickly. Being a First Generation student in large does not affect a Male’s time taken to graduate. The decision tree also shows that Females largely graduate quickly regardless of race or being a first generation student. In conclusion, the hypothesis is correct as females largely graduate quickly, males of certain races do not graduate so quickly, and being a First Generation student largely does not affect the outcome for either sex.

Modified Program

* The following program only performs in conjunction with the aforementioned fifth hypothesis.
  + Programs for the other hypotheses are identical in every sense other than the input and outcome variables chosen for testing and training, and the class and feature names used in the actual decision trees.

# Kyle Calabro

# Dr. Kumar

# Artificial Intelligence - Project 4

# 13 April 2018

# --------------------------- Hypothesis 5 ---------------------------

from sklearn import \_\_version\_\_ as sklearn\_version

from distutils.version import LooseVersion

if LooseVersion(sklearn\_version) < LooseVersion('0.18'):

raise ValueError('Please use scikit-learn 0.18 or newer')

from IPython.display import Image

from sklearn import datasets

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.tree import export\_graphviz

import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

from pydotplus import graph\_from\_dot\_data

from csv import reader

from matplotlib.colors import ListedColormap

import matplotlib.pyplot as plt

from numpy import array

import numpy as np

# To retrieve the dataset from a given csv file.

def getDataset(filename):

file = open(filename)

lines = reader(file)

dataset = list(lines)

return dataset

# To retrieve the outcome variable's column as a dictionary.

def outcomeVariable(dataset, colNo):

classVal = [row[colNo] for row in dataset]

unique = set(classVal)

lookup = dict()

for i, value in enumerate(unique):

lookup[value] = i

for row in dataset:

row[colNo] = lookup[row[colNo]]

return lookup

# To convert any given column in a given dataset to float values.

def stringToFloat(dataset, colNo):

for row in dataset:

row[colNo] = float(row[colNo].strip())

# The name of the file.

filename = 'Hypothesis5.csv'

dataset = getDataset(filename)

stringToFloat(dataset, 1)

stringToFloat(dataset, 2)

stringToFloat(dataset, 3)

stringToFloat(dataset, 7)

lookup = outcomeVariable(dataset, 7)

dataset = array(dataset)

X = dataset[:, [1, 2, 3]]

y = dataset[:, [7]]

print("Class Labels: ", np.unique(y))

# Building a decision tree.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.30, random\_state = 20, stratify = y)

sc = StandardScaler()

sc.fit(X\_train)

X\_train\_std = sc.transform(X\_train)

X\_test\_std = sc.transform(X\_test)

tree = DecisionTreeClassifier(criterion = 'gini',

max\_depth = 4,

random\_state = 20)

tree.fit(X\_train, y\_train)

# Output the actual decision tree as an image.

dot\_data = export\_graphviz(tree,

filled = True,

rounded = True,

class\_names = ['Graduates Quickly',

'Not So Quickly'],

feature\_names = ['Sex', 'Race', 'First Gen.'],

out\_file = None)

graph = graph\_from\_dot\_data(dot\_data)

graph.write\_png('Hypothesis5.png')

# Constructing a random forest and getting the accuracies of the hypothesis.

forest = RandomForestClassifier(criterion='gini',

n\_estimators=25,

random\_state=20,

n\_jobs=2)

forest.fit(X\_train, y\_train.ravel())

# Calculate the accuracy of the test data.

predicted = forest.predict(X\_test)

accuracy = accuracy\_score(y\_test, predicted)

print("Accuracy for Test Data: ", accuracy)

# Calculate the accuracy of the training data.

predicted = forest.predict(X\_train)

accuracy = accuracy\_score(y\_train, predicted)

print("Accuracy for Train Data: ", accuracy)