James Gray
graymatter@u.northwestern.edu

Introduction

The purpose of this assignment is to inform a marketing plan that will target future visitors to one of the Wisconsin Dells seven attractions referenced in the Harrington (2007) case study. This study will outline a specific set of recommendations to drive more visitors to "The Ducks" tour. A supervised learning model was developed using a 1,698 in-person survey conducted at various locations and attractions across the Wisconsin Dells area to predict those visitors who will take the Duck tour given a specific set of attributes.

The predictors and response variables are as follows (from Exhibit 1):

nnights = length of stay
nadults = number of adults in party
nchildren = number of children under 18 in the party
planning = how far in advance the vacation was planned
sex = sex of survey respondent
age = survey respondent age category
education = highest level of education completed
income = level of total household income
region = zip code of region
rideducks = yes/no if a visitor would take the Duck tour (response)

An EDA is conducted to evaluate the predictive accuracy of the attributes of the variables to determine those that are appropriate for the model. A classification tree is then used to determine the attributes associated with visitors who are likely to take The Ducks tour.

Results

The first step of the study conducts an Exploratory Data Analysis (EDA) by calculating a distribution for each of the categorical variables to determine those variables that may be valid for predicting those visitors most likely to ride the duck tour.

- age – the age of survey respondent had minimal variation on those visitors that took the Duck tour and therefore this variable will not be used in the model.
- sex – the sex of the survey respondent does not show a significant difference in those that did or did not take the Ducks tour so it will not be used in the model.
- education – there are imbalance showing correlations between age groups that are more likely to ride the ducks.
- income – the amount of household income had minimal variation across the categories on those groups that did and did not take the Ducks tour. This variable will not be used in the model
- region – there are imbalances showing correlations between certain regions that are more likely to ride the ducks. This variable will be used in the model.
- nchildren – there is variation across the distribution that the number of children influences those that took the Ducks tour so this variable will be used in the model.
- nadults – the number of the adults in the group that take the Ducks tour has minimal variation across the categories and therefore will not be used in the model.

- nnights – there is a strong correlation that visitors that are staying 2 or more nights are more likely to ride the ducks. This variable will be used in the model.
- planning – there is a correlation that the more days that the vacation is planned in advance the more likely visitors are to take the Ducks tour.

Distribution of Ducks by age--

rideducks	NO	YES
age		
25-34	0.379747	0.620253
35-44	0.438543	0.561457
45-54	0.429907	0.570093
55-64	0.325000	0.675000
65+	0.406780	0.593220
LT 25	0.523077	0.476923

Distribution of Ducks by sex---

rideducks	NO	YES
sex		
Female	0.437884	0.562116
Male	0.394828	0.605172

Distribution of Ducks by Education --

rideducks	NO	YES
education		
College Grad	0.481366	0.518634
HS Grad or Less	0.410314	0.589686
Post Grad	0.414508	0.585492
Some College	0.386574	0.613426

Distribution of Ducks by Income --

rideducks	NO	YES
income		
Lower Income	0.405367	0.594633
Middle Income	0.443124	0.556876
Upper Income	0.385417	0.614583

Distribution of Ducks by Region --------

rideducks	NO	YES
region		
Chicago	0.326034	0.673966
Madison	0.606335	0.393665
Milwaukee	0.442231	0.557769
Minneapolis/StPaul	0.354839	0.645161
Other	0.350943	0.649057
Other Wisconsin	0.526316	0.473684

Distribution of Ducks by Children

rideducks	NO	YES
nchildren		
1	0.431818	0.568182
2	0.371002	0.628998
3	0.366255	0.633745
4	0.471154	0.528846
5+	0.557692	0.442308
No kids	0.468013	0.531987

Distribution of Ducks by Adults

rideducks	NO	YES
nadults		
1	0.476190	0.523810
2	0.405375	0.594625
3	0.472603	0.527397
4	0.393701	0.606299
5+	0.435644	0.564356

Distribution of Ducks by Nights

rideducks	NO	YES
nnights		
0	0.569132	0.430868
1	0.562130	0.437870
2	0.375000	0.625000
3	0.325658	0.674342
4+	0.317269	0.682731

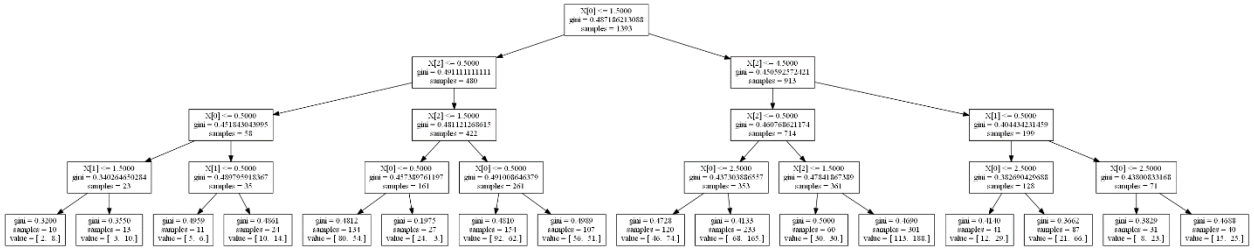
Distribution of Ducks by vacation planned in advance

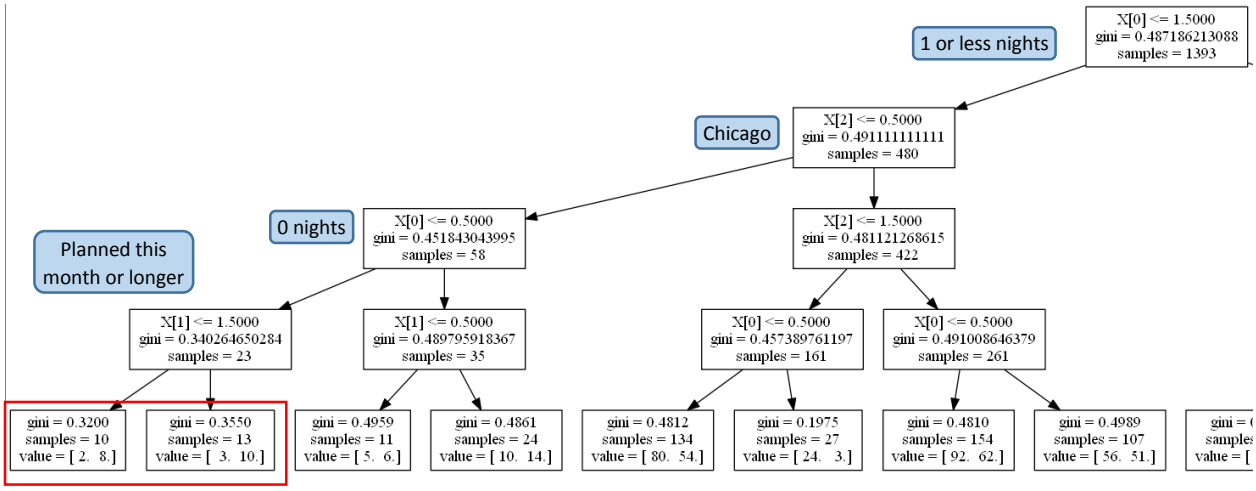
rideducks	NO	YES
nnights		
0	0.569132	0.430868
1	0.562130	0.437870
2	0.375000	0.625000
3	0.325658	0.674342
4+	0.317269	0.682731

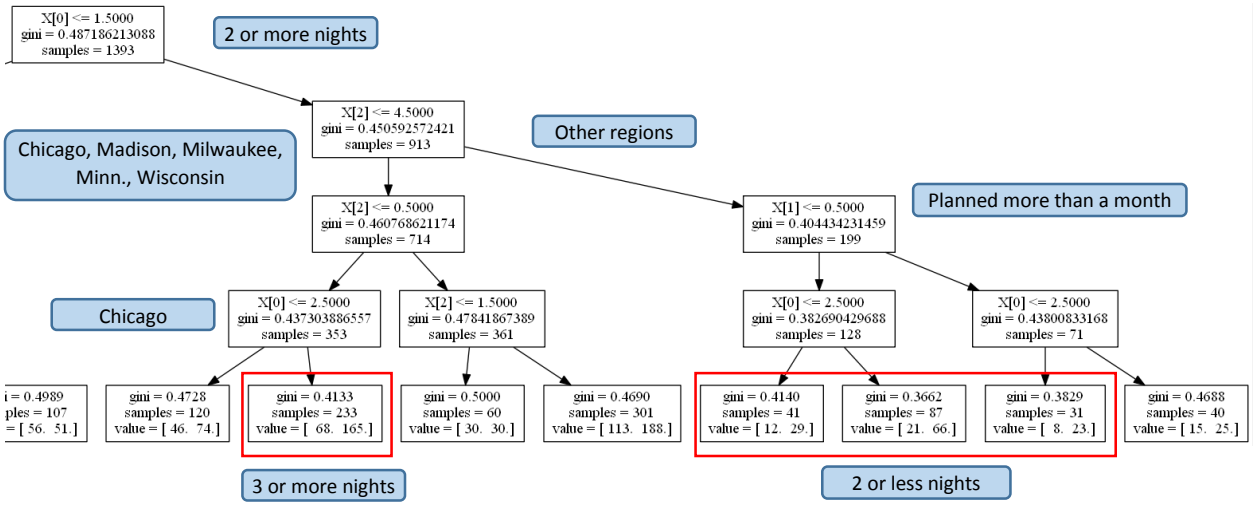
The model was initially run with selected variables above and the cross validation mean was 60.6%. Different permutations of all of the variables were modeled to determine if the prediction could be

improved. The highest predictive accuracy was obtained using the three predictors of nnights, region and planning. The model accuracy was also slightly improved by setting the max tree height to 4 levels. A ten-folds cross-validation mean accuracy was 61.6%.

X[0] = nights, X[1] = planning, X[2] = region







Conclusions

We can infer a few basic conclusions that will guide The Ducks marketing plan. The leaf nodes in the tree that are highlighted by the red boxes with the lower gini indices are combinations of factors that increase the probability that visitors will take the Duck tour. Recommendations for the marketing plan include:

- Heavily target visitors from the Chicago region.
- Make campaign offers for Chicago residents who will be making a day trip for date a month or more out.
- Target visitors from Chicago who will be staying 3 or more nights.
- Target other regional areas beyond Chicago, Madison, Milwaukee, Minnesota and Wisconsin for visitors who would stay for 2 or less nights.

Code

```
# Assignment 1: James Gray - graymatter@u.northwestern.edu - July 12, 2014
#
# This Python program will use the Wisconsin Dells case study data to target
# potential Wisconsin Dells visitors to "The Ducks" tour.

# here are a few of the packages we rely upon for work in predictive analytics
# import os # operating system module
# import pandas as pd # pandas for data frame operations
# import numpy as np # arrays an math functions
# import scipy as sp # statistics and more for science
# import sklearn as sk # machine learning tools
# import statsmodels.formula.api as smf # for working with R-style formulas
# import statsmodels.api as sm # for working with numpy arrays
# import matplotlib.pyplot as plt # 2D plotting
# import networkx as nx # software for network analysis

# these modules should prove especially useful in this first assignment
import pandas as pd # pandas for data frame operations
import numpy as np # arrays an math functions
import sklearn as sk # machine learning tools
import matplotlib.pyplot as plt # 2D plotting

# use scikit-learn as our toolset for machine learning
import sklearn as sk

pd.options.display.max_rows = 20 # set max rows for Pandas output

# ------------------------------------------------------------------------
# DATA PREPARATION NOTES
# ------------------------------------------------------------------------
#
# Data from the Wisconsin Dells case study were collected as part of a
# survey study in the summer of 1995. The study and survey data are described in

# Harrington, J. C. (2007). Wisconsin Dells. Madison, Wisconsin:
# Research Publishers.
#
# the work below uses pandas... which provides the DataFrame data structure,
# a DataFrame is a 2-dimensional labeled data structure with
# columns of potentially different types... like an R data frame or SQL table

# ------------------------------------------------------------------------
# READ IN DATA... Wisconsin Dells survey data
# ------------------------------------------------------------------------
# read in Wisconsin Dells survey data after we ensure that
# the comma-delimited text file is in our working directory
```

```
# this creates a pandas DataFrame object
wi_dells_data = pd.read_csv("wisconsin_dells.csv")

print("\nContents of wi_dells_data object ----------")
# examine the structure of this DataFrame object
print(wi_dells_data)

# ---------------------------------------------------------------------------
# Focus on the Wisconsin Dells visitors who "rideducks"
# ---------------------------------------------------------------------------

# to see the data we need to do a little work due to the large number of columns
# documentation for what we are doing here is available at
# http://pandas.pydata.org/pandas-docs/stable/basics.html?highlight=set_option
pd.set_option('display.max_columns', 11) # to allow at most 11 columns of data output

# the Wisconsin Dells demographic data are provided in the first ten columns
# let's select those columns along with the column for rideducks,
# this will give us 11 columns of data for our analysis
# let's use df as a shorthand for our working data frame
# a Python list is used to specify the columns of interest

df = pd.DataFrame(wi_dells_data,
columns = ["id", "nnights", "nadults", "nchildren", "planning",
"sex", "age", "education", "income", "region", "rideducks"])
print("\nContents of the working data frame df----------")
# examine the structure of this DataFrame object
print(df)
# print the first five rows of the DataFrame
print(pd.DataFrame.head(df)) # note NaN values for missing data

# ---------------------------------------------------------------------
# Data Cleansing and Transformation
# ---------------------------------------------------------------------

# Remove rows with missing data
df_complete = df.dropna()
print(df_complete.shape) # inspect how many rows were deleted

# Transform the binary target into numerical data using Dictionary
ducks_mapper = {'NO': -1,'YES' : 1}
y = df_complete['rideducks'].map (ducks_mapper)
print("\nTarget variable array")
print(y[0:10])

# Transform explanatory nominal vars into numerical data using Dictionary
# Use Dictionary to store mapping
# Create new DF
```

```
sex_mapper = {'Female': 0, 'Male': 1}
sex = df_complete['sex'].map(sex_mapper) #new DF to hold sex
print (sex[0:10]) # inspect sex transformation

age_mapper = {'LT 25' : 1,
'25-34' : 2,
'35-44' : 3,
'45-54' : 4,
'55-64' : 5,
'65+' : 6}
age = df_complete['age'].map(age_mapper)
print (age[0:10])
education_mapper = {'HS Grad or Less':0,
'Some College': 1,
'College Grad': 2,
'Post Grad':3}
education = df_complete['education'].map(education_mapper)
income_mapper = {'Lower Income':0,
'Middle Income': 1,
'Upper Income': 2 }
income = df_complete['income'].map(income_mapper)

planning_mapper = { 'One Month or More Ago': 0,
'This Month': 1,
'This Week': 2}
planning = df_complete['planning'].map(planning_mapper)
children_mapper = {'No kids': 0,
'1': 1,
'2': 2,
'3': 3,
'4': 4,
'5+': 5}
children = df_complete['nchildren'].map(children_mapper)

nights_mapper = { '0': 0,
'1': 1,
'2': 2,
'3': 3,
'4+': 4}
nights = df_complete['nnights'].map(nights_mapper)

adults_mapper = { '1': 1,
'2': 2,
'3': 3,
'4': 4,
'5+': 5}
adults = df_complete['nadults'].map(adults_mapper)
```

```
region_mapper = { 'Chicago': 0,
'Madison': 1,
'Milwaukee':2,
'Minneapolis/StPaul':3,
'Other Wisconsin': 4,
'Other': 5 }
region = df_complete['region'].map(region_mapper)

# ---------------------------------------------------------------------
# Perform EDA to evaluate predictive accuracy of each categorical variable
#----------------------------------------------------------------------

ducksbyage = pd.crosstab(df_complete.age, df_complete.rideducks)
ducksbyagepct = ducksbyage.div(ducksbyage.sum(1).astype(float), axis =0)
ducksbyagepct.plot (kind='barh', stacked = True)
print("\nDistribution of Ducks by respondent age ----------")
print (ducksbyagepct)

ducksbysex = pd.crosstab(df_complete.sex, df_complete.rideducks)
ducksbysexpct = ducksbysex.div(ducksbysex.sum(1).astype(float), axis =0)
ducksbysexpct.plot (kind='barh', stacked = True)
print("\nDistribution of Ducks by respondent sex----------")
print (ducksbysexpct)

ducksbyeducation = pd.crosstab(df_complete.education, df_complete.rideducks)
ducksbyeducationpct = ducksbyeducation.div(ducksbyeducation.sum(1).astype(float), axis =0)
ducksbyeducationpct.plot (kind='barh', stacked = True)
print("\nDistribution of Ducks by respondent education ----------")
print (ducksbyeducationpct)

ducksbyincome = pd.crosstab(df_complete.income, df_complete.rideducks)
ducksbyincomepct = ducksbyincome.div(ducksbyincome.sum(1).astype(float), axis =0)
ducksbyincomepct.plot (kind='barh', stacked = True)
print("\nDistribution of Ducks by Household Income ----------")
print (ducksbyincomepct)

ducksbyregion = pd.crosstab(df_complete.region, df_complete.rideducks)
ducksbyregionpct = ducksbyregion.div(ducksbyregion.sum(1).astype(float), axis =0)
ducksbyregionpct.plot (kind='barh', stacked = True)
print("\nDistribution of Ducks by Region ----------")
print (ducksbyregionpct)

ducksbychildren = pd.crosstab(df_complete.nchildren, df_complete.rideducks)
ducksbychildrenpct = ducksbychildren.div(ducksbychildren.sum(1).astype(float), axis =0)
ducksbychildrenpct.plot (kind='barh', stacked = True)
print("\nDistribution of Ducks by Children in party ----------")
print (ducksbychildrenpct)
```

```
ducksbyadults = pd.crosstab(df_complete.nadults, df_complete.rideducks)
ducksbyadultspct = ducksbyadults.div(ducksbyadults.sum(1).astype(float), axis =0)
ducksbyadultspct.plot (kind='barh', stacked = True)
print("\nDistribution of Ducks by Adults in party ----------")
print (ducksbyadultspct)

ducksbynights = pd.crosstab(df_complete.nnights, df_complete.rideducks)
ducksbynightspct = ducksbynights.div(ducksbynights.sum(1).astype(float), axis =0)
ducksbynightspct.plot (kind='barh', stacked = True)
print("\nDistribution of Ducks by Nights in stay ----------")
print (ducksbynightspct)

ducksbyplanning = pd.crosstab(df_complete.nnights, df_complete.rideducks)
ducksbynightspct = ducksbynights.div(ducksbynights.sum(1).astype(float), axis =0)
ducksbynightspct.plot (kind='barh', stacked = True)
print("\nDistribution of Ducks by vacation planned in advance ----------")
print (ducksbynightspct)
# gather all explanatory variables into a numpy array
# here we use .T to obtain the transpose for the structure we want.
# The EDA showed that region, nights, planning had the most
# significant impact on taking the Ducks tour
x = np.array([np.array(nights),
np.array(planning), np.array(region)]).T
print(x.shape) # check the structure of the array of explanatory variables
print("\nInput variable array")
print(x[0:10,]) # examine the first 10 rows of data


# ------------------------------------------------------------------------
# build a simple tree classifier for these data
#-------------------------------------------------------------------------

# define modeling method with random number seed for reproducibility
from sklearn import tree
model = tree.DecisionTreeClassifier(min_samples_split = 20, min_samples_leaf = 10,
max_depth = 4, random_state = 9999)
my_tree = model.fit(x, y) # defines tree classifier object

# use tab completion to see methods available for this object
# my_tree.<tab>
# use my_tree? to obtain information about the object

# multi-fold cross-validation with ten folds to calculate predictive accuracy
from sklearn.cross_validation import cross_val_score
cv_results = cross_val_score(model, x, y, cv=10)
print(round(cv_results.mean(),3)) # cross-validation average proportion correct
print (cv_results)
```

```
# lets see what the fitted tree looks like
from sklearn.externals.six import StringIO
with open('ducks.dot', 'w') as f:
f = tree.export_graphviz(my_tree, out_file=f)
```

References

Harrington, J. C. (2007). *Wisconsin Dells.* Madison, Wisconsin: Research Publishers.