Appendix-A2

October 28, 2024

[11]:
```
import os
import numpy as np
import pandas as pd
import math
import matplotlib.pylab as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, StratifiedKFold,␣
 ↪cross_val_score

from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

%matplotlib inline
sns.set(style='ticks', palette='Set2')
!pip install graphviz
```

```
Requirement already satisfied: graphviz in c:\users\16099\anaconda3\lib\site-
packages (0.20.3)
```

[12]:
```
path = "./churn2.csv"
df = pd.read_csv(path)[["COLLEGE", "INCOME", "OVERAGE", "LEFTOVER",␣
 ↪"HOUSE","HANDSET_PRICE","OVER_15MINS_CALLS_PER_MONTH","AVERAGE_CALL_DURATION","REPORTED_SAT
 ↪dropna()
```

[13]:
```
df["COLLEGE2"] = (df.COLLEGE == "one").astype(int)
```

[14]:
```
df.REPORTED_SATISFACTION = df.REPORTED_SATISFACTION.astype('str')
df.REPORTED_USAGE_LEVEL = df.REPORTED_USAGE_LEVEL.astype('str')
df.CONSIDERING_CHANGE_OF_PLAN = df.CONSIDERING_CHANGE_OF_PLAN.astype('str')
```

[15]:
```
df["LEAVE2"] = (df.LEAVE == "STAY").astype(int)
```

[16]:
```
df.dtypes
```

[16]:
```
COLLEGE                         object
INCOME                           int64
```

```
OVERAGE                          int64
LEFTOVER                         int64
HOUSE                            int64
HANDSET_PRICE                    int64
OVER_15MINS_CALLS_PER_MONTH      int64
AVERAGE_CALL_DURATION            int64
REPORTED_SATISFACTION           object
REPORTED_USAGE_LEVEL            object
CONSIDERING_CHANGE_OF_PLAN      object
LEAVE                           object
COLLEGE2                         int32
LEAVE2                           int32
dtype: object
```

[17]:
```
predictor_cols = ["INCOME",␣
 ↪"OVERAGE","LEFTOVER","HOUSE","OVER_15MINS_CALLS_PER_MONTH","AVERAGE_CALL_DURATION","COLLEGE
target_col = "LEAVE2"
```

[18]:
```
predictor_cols = ["INCOME",␣
 ↪"OVERAGE","LEFTOVER","HOUSE","OVER_15MINS_CALLS_PER_MONTH","AVERAGE_CALL_DURATION","COLLEGE
target_col = "LEAVE2"
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =␣
 ↪train_test_split(df[predictor_cols],df[target_col],test_size = 0.
 ↪25,random_state = 0)
```

[19]:
```
print("X_train shape: {}".format(X_train.shape))
print("X_test shape: {}".format(X_test.shape))
print("y_train shape: {}".format(y_train.shape))
print("y_test shape: {}".format(y_test.shape))
```

```
X_train shape: (15000, 7)
X_test shape: (5000, 7)
y_train shape: (15000,)
y_test shape: (5000,)
```

[21]:
```
scale = StandardScaler()
X_train_scaled = scale.fit_transform(X_train)

kfold = StratifiedKFold(n_splits = 10, shuffle = True, random_state = 0)

lr_regression = LogisticRegression(max_iter = 100000, random_state = 0)
lr_accuracy_train = cross_val_score(lr_regression, X_train_scaled, y_train, cv␣
 ↪= kfold, scoring='accuracy')

accuracy = np.mean(lr_accuracy_train)
std = np.std(lr_accuracy_train)
```

```
print("10-Fold Accuracy:", lr_accuracy_train)
print("Mean Accuracy:", accuracy)
print("Mean Standard Deviation:", std)
```

```
10-Fold Accuracy: [0.65266667 0.65       0.654      0.648      0.652
0.61266667
 0.62333333 0.61266667 0.634      0.63333333]
Mean Accuracy: 0.6372666666666668
Mean Standard Deviation: 0.015627468693866508
```

[34]:
```
from sklearn.tree import DecisionTreeClassifier

decision_tree = DecisionTreeClassifier(max_depth=6,␣
 ↪criterion="entropy",max_leaf_nodes = 12, min_samples_leaf = 1)
decision_tree.fit(X_train, y_train)
```

[34]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=6, max_leaf_nodes=12)
```

[35]:
```
scale = StandardScaler()
X_train_scaled = scale.fit_transform(X_train)

kfold = StratifiedKFold(n_splits = 10, shuffle = True, random_state=0)

dt_accuracy_train = cross_val_score(decision_tree, X_train_scaled, y_train, cv␣
 ↪= kfold, scoring = 'accuracy')

accuracy = np.mean(dt_accuracy_train)
std = np.std(dt_accuracy_train)

print("10-Fold Accuracy:", dt_accuracy_train)
print("Mean Accuracy:", accuracy)
print("Standard Deviation:", std)
```

```
10-Fold Accuracy: [0.69733333 0.71666667 0.702      0.69066667 0.70933333 0.686
 0.692      0.712      0.702      0.70933333]
Mean Accuracy: 0.7017333333333333
Standard Deviation: 0.009634198346400067
```

[33]:
```
from IPython.display import Image
from sklearn.tree import export_graphviz

def visualize_tree(decision_tree, feature_names, class_names, directory="./
 ↪images", name="tree",proportion=True):
    # Export our decision tree to graphviz format
    directory1 = directory[2:]
    os.system("mkdir %s" %(directory1))
    dot_name = "%s/%s.dot" % (directory, name)
```

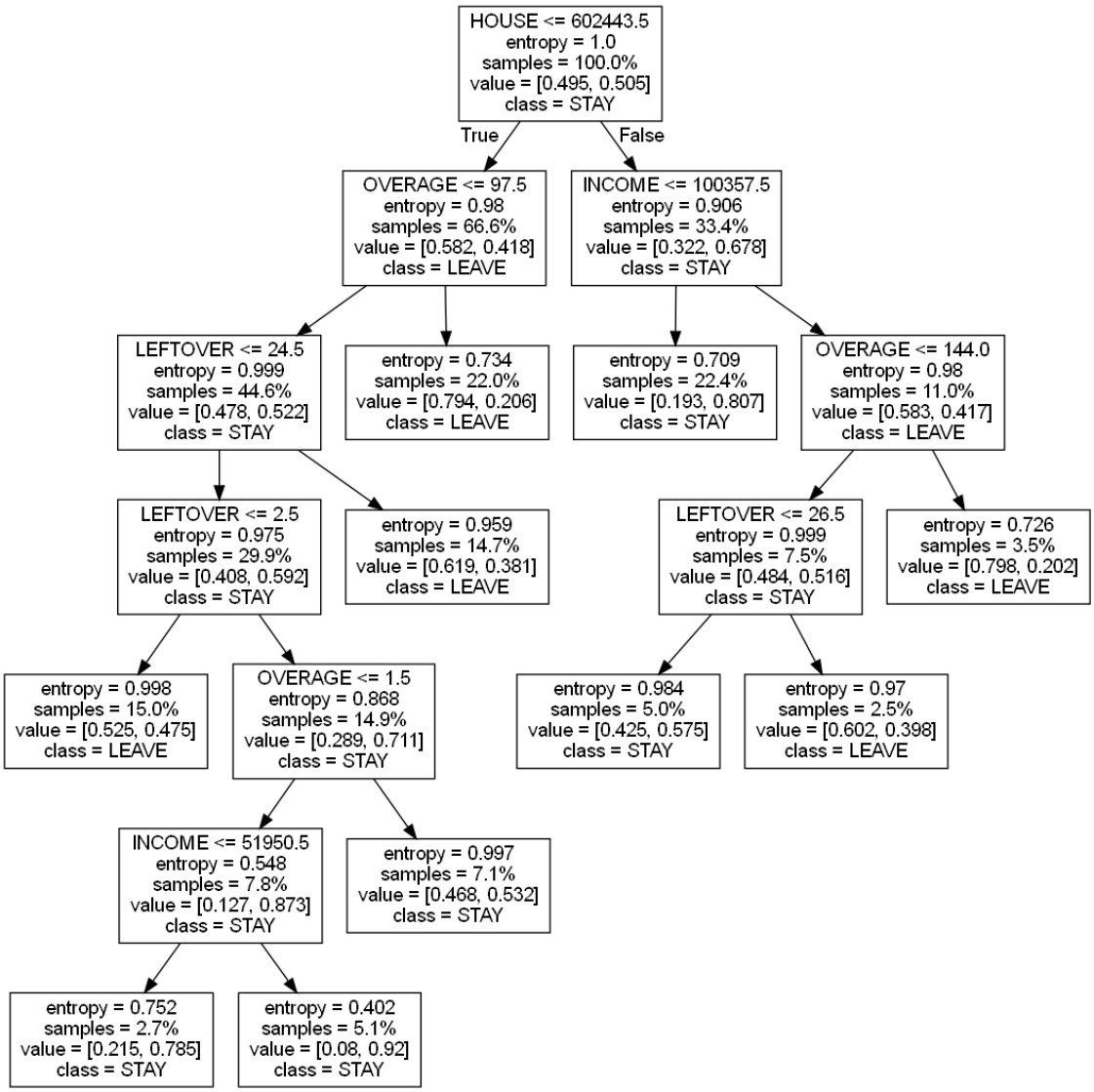
```
    dot_file = export_graphviz(decision_tree,␣
 ↪out_file=dot_name,feature_names=feature_names,class_names=class_names,
                               proportion=proportion)
    # Call graphviz to make an image file from our decision tree
    image_name = "%s/%s.png" % (directory, name)
    print(dot_name)
    print(image_name)
    #os.system("dot -Tpng %s  -o %s" % (dot_name, image_name))
    os.system("dot -Tpng %s  -o %s" % (dot_name, image_name))
    # os.system("cd %s" % (directory1))
    #subprocess.run("dot -T png %s -o %s" % (dot_name, image_name))
    # Return the .png image so we can see it

    return Image(filename=image_name)

visualize_tree(decision_tree, predictor_cols, ["LEAVE", "STAY"])
```

```
./images/tree.dot
./images/tree.png
```

[33]:


HOUSE <= 602443.5
entropy = 1.0
samples = 100.0%
value = [0.495, 0.505]
class = STAY
True
False
OVERAGE <= 97.5
entropy = 0.98
samples = 66.6%
value = [0.582, 0.418]
class = LEAVE
INCOME <= 100357.5
entropy = 0.906
samples = 33.4%
value = [0.322, 0.678]
class = STAY
LEFTOVER <= 24.5
entropy = 0.999
samples = 44.6%
value = [0.478, 0.522]
class = STAY
entropy = 0.734
samples = 22.0%
value = [0.794, 0.206]
class = LEAVE
entropy = 0.709
samples = 22.4%
value = [0.193, 0.807]
class = STAY
OVERAGE <= 144.0
entropy = 0.98
samples = 11.0%
value = [0.583, 0.417]
class = LEAVE
LEFTOVER <= 2.5
entropy = 0.975
samples = 29.9%
value = [0.408, 0.592]
class = STAY
entropy = 0.959
samples = 14.7%
value = [0.619, 0.381]
class = LEAVE
LEFTOVER <= 26.5
entropy = 0.999
samples = 7.5%
value = [0.484, 0.516]
class = STAY
entropy = 0.726
samples = 3.5%
value = [0.798, 0.202]
class = LEAVE
entropy = 0.998
samples = 15.0%
value = [0.525, 0.475]
class = LEAVE
OVERAGE <= 1.5
entropy = 0.868
samples = 14.9%
value = [0.289, 0.711]
class = STAY
entropy = 0.984
samples = 5.0%
value = [0.425, 0.575]
class = STAY
entropy = 0.97
samples = 2.5%
value = [0.602, 0.398]
class = LEAVE
INCOME <= 51950.5
entropy = 0.548
samples = 7.8%
value = [0.127, 0.873]
class = STAY
entropy = 0.997
samples = 7.1%
value = [0.468, 0.532]
class = STAY
entropy = 0.752
samples = 2.7%
value = [0.215, 0.785]
class = STAY
entropy = 0.402
samples = 5.1%
value = [0.08, 0.92]
class = STAY

[ ]: