MACHINE LEARNING HOMEWORK THREE

MATTHEW FERGUSON

3/1/2022

Information Gain First Level

- Entropy of Edible: $-\frac{16}{24}\log_2\left(\frac{16}{24}\right)-\frac{8}{24}\log_2\left(\frac{8}{24}\right)$ =0.9183 bits
- Remainder of White: $\frac{10}{24}\left(-\frac{7}{10}\log_2\left(\frac{7}{10}\right)-\frac{3}{10}\log_2\left(\frac{3}{10}\right)\right)+\frac{14}{24}\left(-\frac{9}{14}\log_2\left(\frac{9}{14}\right)-\frac{5}{14}\log_2\left(\frac{5}{14}\right)\right)$ =0.9118 bits
- Remainder of Tall: $\frac{14}{24}\left(-\frac{10}{14}\log_2\left(\frac{10}{14}\right)-\frac{4}{14}\log_2\left(\frac{4}{14}\right)\right)+\frac{10}{24}\left(-\frac{6}{10}\log_2\left(\frac{6}{10}\right)-\frac{4}{10}\log_2\left(\frac{4}{10}\right)\right)$ =0.9080 bits
- Remainder of Frilly: $\frac{8}{24}\left(-\frac{3}{8}\log_2\left(\frac{3}{8}\right)-\frac{5}{8}\log_2\left(\frac{5}{8}\right)\right)+\frac{16}{24}\left(-\frac{13}{16}\log_2\left(\frac{13}{16}\right)-\frac{3}{16}\log_2\left(\frac{3}{16}\right)\right)$ =0.7823 bits

Frilly is the feature with the highest information gain (IG.f = 0.1360 bits)

Information Gain Second Level (Frilly = Yes)

- Entropy of Frilly = Yes: $-\frac{3}{8}\log_2\left(\frac{3}{8}\right)-\frac{5}{8}\log_2\left(\frac{5}{8}\right)$ =0.9544 bits
- Remainder of White: $\frac{3}{8}\left(-\frac{3}{3}\log_2\left(\frac{3}{3}\right)\right)+\frac{5}{8}\left(-\frac{3}{5}\log_2\left(\frac{3}{5}\right)-\frac{2}{5}\log_2\left(\frac{2}{5}\right)\right)$ =0.6068 bits
- Remainder of Tall: $\frac{5}{8}\left(-\frac{\log_2\left(\frac{1}{5}\right)}{5}-\frac{4}{5}\log_2\left(\frac{4}{5}\right)\right)+\frac{3}{8}\left(-\frac{2}{3}\log_2\left(\frac{2}{3}\right)-\frac{1}{3}\log_2\left(\frac{1}{3}\right)\right)$ =0.7956 bits

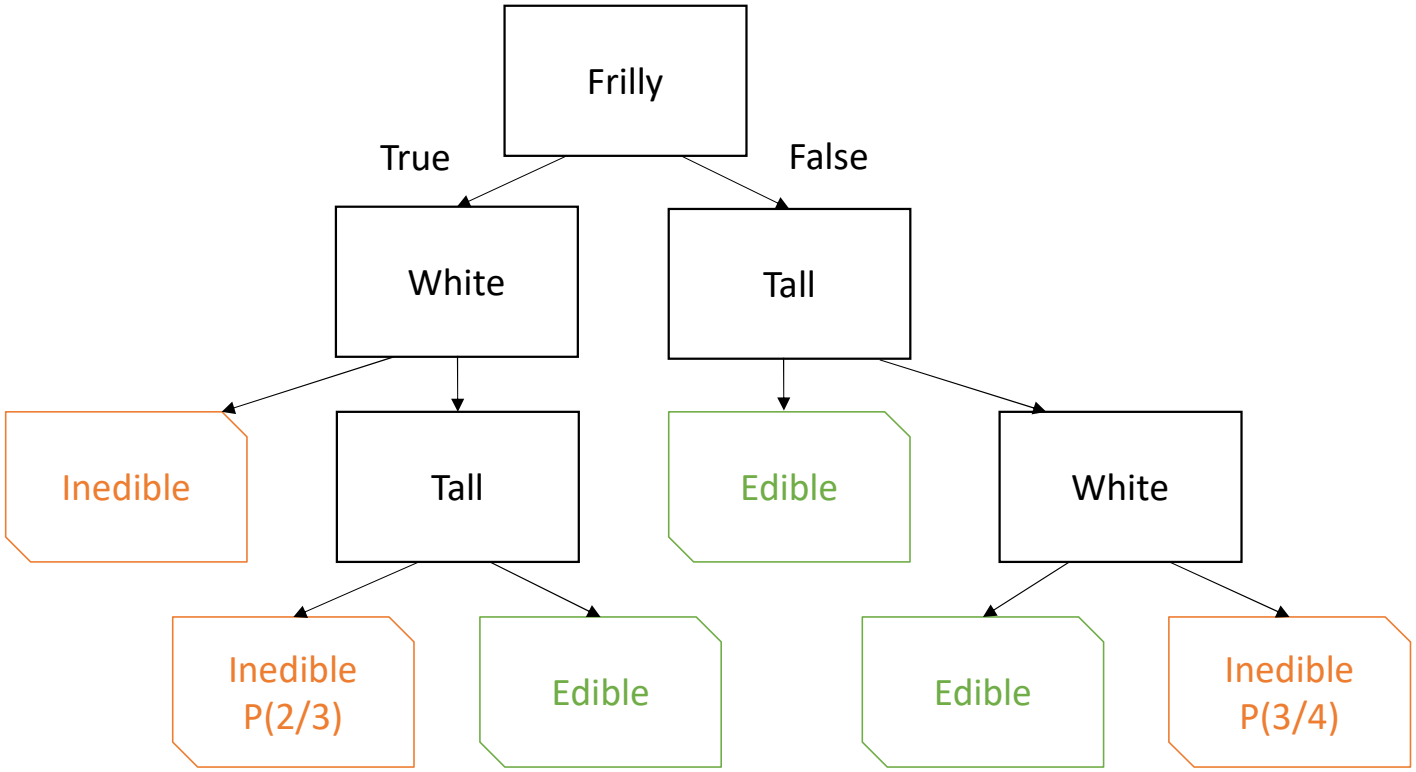
White is the feature with the highest information gain when Frilly = Yes (IG.w = 0.3476 bits)

Information Gain Second Level (Frilly = No)

- Entropy of Frilly = No: $-\frac{13}{16}\log_2\left(\frac{13}{16}\right)-\frac{3}{16}\log_2\left(\frac{3}{16}\right)$ =0.6962 bits
- Remainder of White: $\frac{7}{16}\left(-\frac{7}{7}\log_2\left(\frac{7}{7}\right)\right)+\frac{9}{16}\left(-\frac{6}{9}\log_2\left(\frac{6}{9}\right)-\frac{3}{9}\log_2\left(\frac{3}{9}\right)\right)$ =0.5165 bits
- Remainder of Tall: $\frac{9}{16}\left(-\frac{9}{9}\log_2\left(\frac{9}{9}\right)\right)+\frac{7}{16}\left(-\frac{4}{7}\log_2\left(\frac{4}{7}\right)-\frac{3}{7}\log_2\left(\frac{3}{7}\right)\right)$ =0.4310 bits

Tall is the feature with the highest information gain when Frilly = No (IG.t = 0.2652 bits)

Hand Calculated Tree

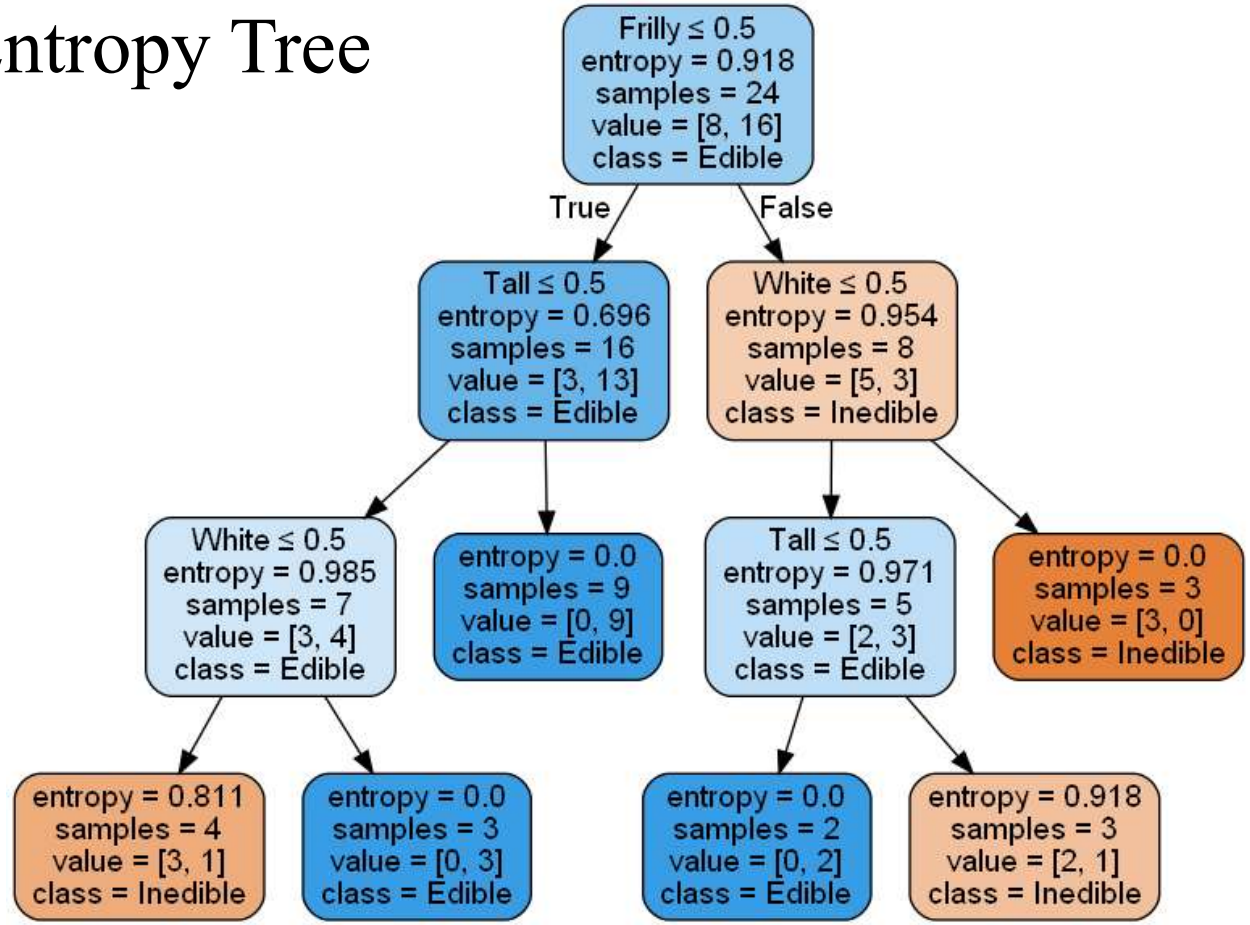


Part Two Code

```
from sklearn import tree
import pandas as pd
import pydot

df = pd.read_excel(r'C:\Users\Matt\Desktop\AlienMushrooms.xlsx')
cm=df.corr()
Feature=df[['White','Tall','Frilly']].values.tolist()
Target=df['Edible'].tolist()

clf=tree.DecisionTreeClassifier(criterion="entropy")
clf=clf.fit(Feature, Target)

Fnames=['White','Tall','Frilly']
Tnames='Edible'

dot_data=tree.export_graphviz(clf, out_file=None, feature_names=Fnames, class_names=Tnames, filled=True, rounded=True, special_characters=True)

(graph,) =pydot.graph_from_dot_data(dot_data)
graph.write_png(r'C:\Users\Matt\Desktop\AlienMushrooms.png')
```

Part Two Scikit Entropy Tree

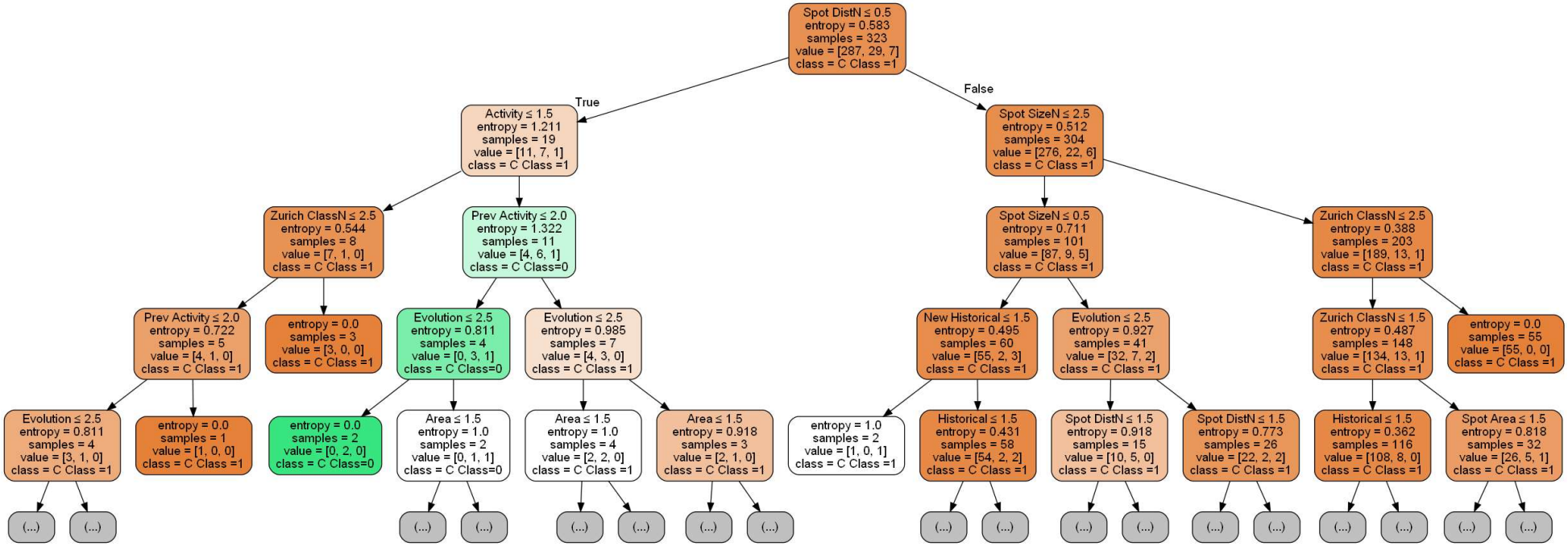
At first glance we see the tree is flipped along the horizontal axis. This occurs because Frilly being less than or equal to 0.5 is the same as asking if Frilly is false. In that sense True and False are flipped from our original decision tree. Overall though we see that the entropy values and sample counts are consistent across our manual calculations. No major differences occur between the hand tabulated decision tree and the scikit decision tree.



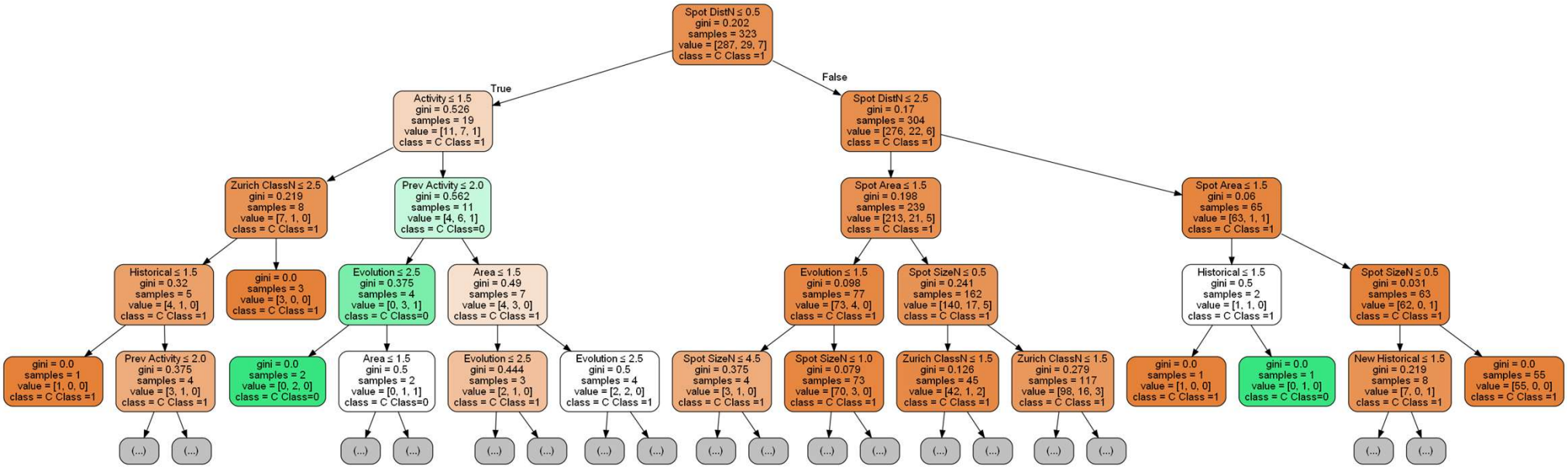
Part Three Code

```
from sklearn import tree
import pandas as pd
import pydot


df = pd.read_excel(r'C:\Users\Matt\Desktop\FlareData.xlsx')
cm=df.corr()
Feature=df[['Zurich ClassN','Spot SizeN','Spot DistN', 'Activity', 'Evolution', 'Prev Activity', 'Historical', 'New Historical', 'Area', 'Spot
Area']].values.tolist()
Target=df['C class'].tolist()


clf=tree.DecisionTreeClassifier(criterion="gini")
clf=clf.fit(Feature, Target)


Fnames=['Zurich ClassN','Spot SizeN','Spot DistN', 'Activity', 'Evolution', 'Prev Activity', 'Historical', 'New Historical', 'Area', 'Spot Area']
Tnames='C Class =1', 'C Class=0'


dot_data=tree.export_graphviz(clf, out_file=None, feature_names=Fnames, class_names=Tnames, filled=True, rounded=True, special_characters=True, max_depth=4)


(graph,) =pydot.graph_from_dot_data(dot_data)
graph.write_png(r'C:\Users\Matt\Desktop\FlareData_Gini.png')
```

Part Three Scikit Entropy Tree



Part Three Scikit Gini Tree



Part Three Discussion

Computationally, Shannon entropy based decision trees rely on logarithms and are more complex to initially compute than Gini based decision trees so the scalability of training with Gini decision trees is better. Gini's maximum impurity is 0.5 and max purity is 0 whereas Shannon entropy's max impurity is 1 and max purity is 0.

We see the Gini and Shannon entropy decision tree shapes overall are similar, however the full Gini tree has less leaves than the full Shannon entropy tree and the Gini has a less complex and more shallow shape. Perhaps the Shannon entropy based tree is more prone to overfitting as a result. The entropy based tree may be more accurate if overfitting is engineered against, however the accuracy gain is probably insignificant especially in comparison to the tradeoff of significantly increase computation time for large scale.