

Demo: Decision trees

Fraida Fund

This is a simple demo notebook that demonstrates a decision tree classifier.

Attribution: Parts of this notebook are slightly modified from [this tutorial from “Intro to Data Mining”](#).

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import sklearn
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier, RandomForestClassifier, AdaBoostClassifier
```

```
df = pd.read_csv('http://www.cse.msu.edu/~ptan/dmbook/tutorials/tutorial6/vertebrate.csv')
df
```

	Name	Warm-blooded	Gives Birth	Aquatic Creature	\
0	human	1	1	0	
1	python	0	0	0	
2	salmon	0	0	1	
3	whale	1	1	1	
4	frog	0	0	1	
5	komodo	0	0	0	
6	bat	1	1	0	
7	pigeon	1	0	0	
8	cat	1	1	0	
9	leopard shark	0	1	1	
10	turtle	0	0	1	
11	penguin	1	0	1	
12	porcupine	1	1	0	
13	eel	0	0	1	
14	salamander	0	0	1	

	Aerial Creature	Has Legs	Hibernates	Class
0	0	1	0	mammals
1	0	0	1	reptiles
2	0	0	0	fishes
3	0	0	0	mammals
4	0	1	1	amphibians
5	0	1	0	reptiles
6	1	1	1	mammals
7	1	1	0	birds
8	0	1	0	mammals
9	0	0	0	fishes
10	0	1	0	reptiles
11	0	1	0	birds
12	0	1	1	mammals
13	0	0	0	fishes
14	0	1	1	amphibians

We'll make it a binary classification problem:

```
df['Class'] = df['Class'].replace(['fishes','birds','amphibians','reptiles'],'non-mammals')
df
```

	Name	Warm-blooded	Gives Birth	Aquatic Creature	\
0	human	1	1	0	
1	python	0	0	0	
2	salmon	0	0	1	
3	whale	1	1	1	
4	frog	0	0	1	
5	komodo	0	0	0	
6	bat	1	1	0	
7	pigeon	1	0	0	
8	cat	1	1	0	
9	leopard shark	0	1	1	
10	turtle	0	0	1	
11	penguin	1	0	1	
12	porcupine	1	1	0	
13	eel	0	0	1	
14	salamander	0	0	1	

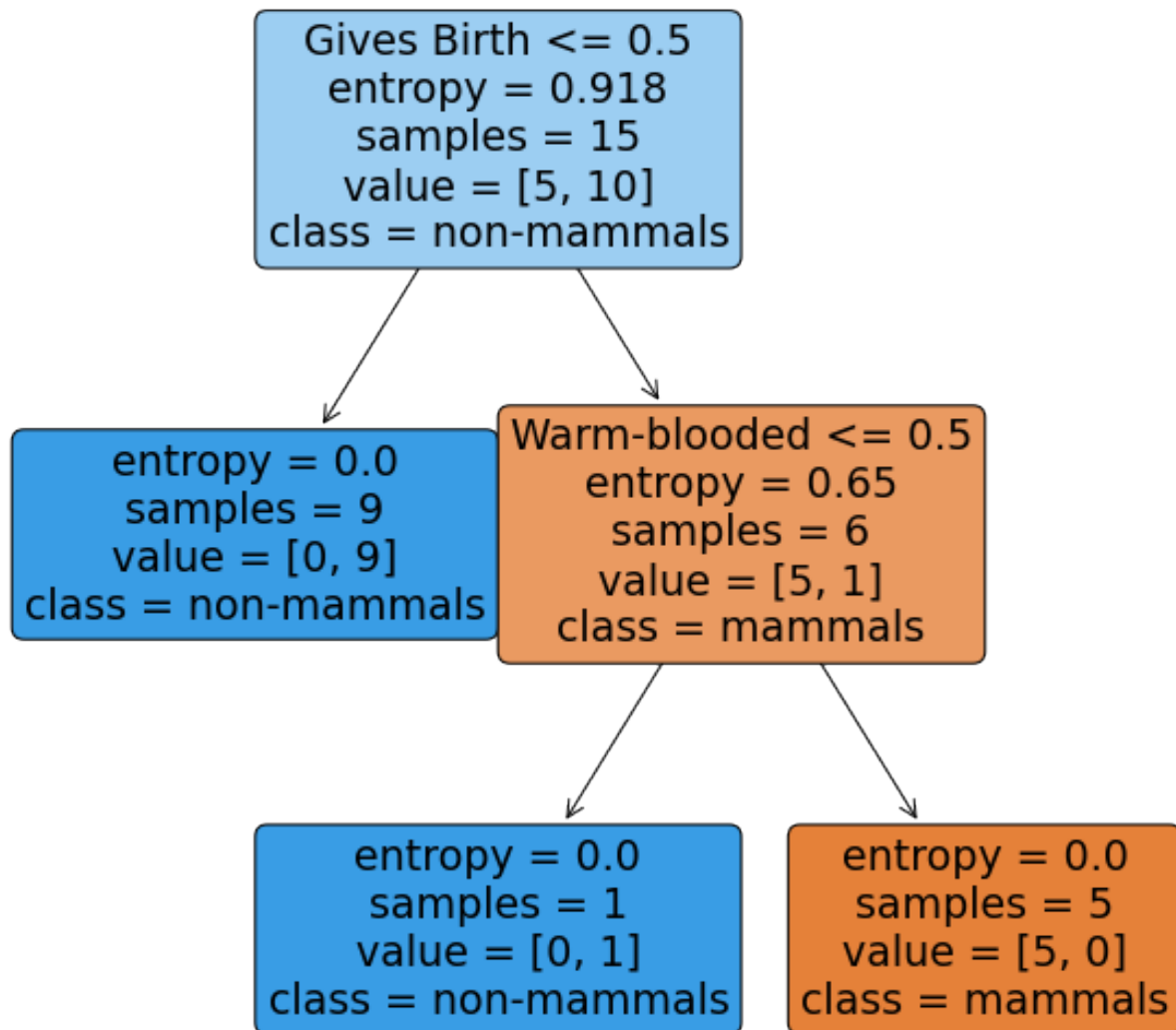
	Aerial Creature	Has Legs	Hibernates	Class
0	0	1	0	mammals
1	0	0	1	non-mammals
2	0	0	0	non-mammals
3	0	0	0	mammals
4	0	1	1	non-mammals
5	0	1	0	non-mammals
6	1	1	1	mammals
7	1	1	0	non-mammals
8	0	1	0	mammals
9	0	0	0	non-mammals
10	0	1	0	non-mammals
11	0	1	0	non-mammals
12	0	1	1	mammals
13	0	0	0	non-mammals
14	0	1	1	non-mammals

Decision tree

```
y = df['Class']
X = df.drop(['Name','Class'],axis=1)

clf_dt = DecisionTreeClassifier(criterion='entropy')
clf_dt = clf_dt.fit(X, y)
```

```
plt.figure(figsize=(10,10))
sklearn.tree.plot_tree(clf_dt,
                        feature_names = df.columns.drop(['Name', 'Class']),
                        class_names = ["mammals", "non-mammals"],
                        filled=True, rounded=True);
```



Feature importance

```
df_importance = pd.DataFrame({'feature': df.columns.drop(['Name', 'Class']),
                             'importance': clf_dt.feature_importances_})
df_importance
```

	feature	importance
0	Warm-blooded	0.283143
1	Gives Birth	0.716857
2	Aquatic Creature	0.000000
3	Aerial Creature	0.000000
4	Has Legs	0.000000
5	Hibernates	0.000000