In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df= pd.read_csv('Mammal.csv')
```

Checking the unique value from the feature

In [3]:
```
df.nunique()
```

Out[3]:
```
Name                  15
Body Temperature       2
Skin Cover             6
Give Birth             2
Aquatic Creature       3
Aerial Creature        2
Has Legs               2
Hibernates             2
Class                  5
dtype: int64
```

In [4]:
```
df['Body Temperature'].value_counts()
```

Out[4]:
```
Cold    8
Warm    7
Name: Body Temperature, dtype: int64
```

Rename the columns name for better understanding of the data.

In [5]:
```
df.rename(columns={"Body Temperature":"Temperature"},inplace=True)
```

In [6]:
```
df.rename(columns={"Skin Cover":"Skin","Aquatic Creature":"Aquatic"},inplace=True
```

Change the Data Type from Object to Integer.

In [7]:
```
df.Temperature= df.Temperature.replace({'Warm':1,'Cold':0})
```

In [8]:
```
df.Skin= df.Skin.replace({'none':5,'fur':4,'quills':3,'feathers':2,'scales':1,'ha
```

In [9]:
```
df.Aquatic= df.Aquatic.replace({'semi':2,'yes':1,'no':0})
```

Drop Column

In [10]:

```
df.drop('Name',axis=1,inplace=True)
```

Data After Cleaning

In [11]:

```
df
```

Out[11]:

	Temperature	Skin	Give Birth	Aquatic	Aerial Creature	Has Legs	Hibernates	Class
0	1	0	1	0	0	1	0	Mammal
1	0	1	0	0	0	0	1	Reptile
2	0	1	0	1	0	0	0	Fish
3	1	0	1	1	0	0	0	Mammal
4	0	5	0	2	0	1	1	Amphibian
5	0	1	0	0	0	1	0	Reptile
6	1	0	1	0	1	1	1	Mammal
7	1	2	0	0	1	1	0	Bird
8	1	4	1	0	0	1	0	Mammal
9	0	1	1	1	0	0	0	Fish
10	0	1	0	2	0	1	0	Reptile

In [12]:

```
df.dtypes
```

Out[12]:

```
Temperature          int64
Skin                 int64
Give Birth           int64
Aquatic              int64
Aerial Creature      int64
Has Legs             int64
Hibernates           int64
Class               object
dtype: object
```

In [13]:

```
df.shape
```

Out[13]:

```
(15, 8)
```

In [14]:

```
df.columns
```

Out[14]:

```
Index(['Temperature', 'Skin', 'Give Birth', 'Aquatic', 'Aerial Creature',
       'Has Legs', 'Hibernates', 'Class'],
      dtype='object')
```

In [15]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Temperature      15 non-null     int64 
 1   Skin             15 non-null     int64 
 2   Give Birth       15 non-null     int64 
 3   Aquatic          15 non-null     int64 
 4   Aerial Creature  15 non-null     int64 
 5   Has Legs         15 non-null     int64 
 6   Hibernates       15 non-null     int64 
 7   Class            15 non-null     object
dtypes: int64(7), object(1)
memory usage: 1.1+ KB
```

In [16]:

```
df.isnull().sum()
```

Out[16]:

```
Temperature        0
Skin               0
Give Birth         0
Aquatic            0
Aerial Creature    0
Has Legs           0
Hibernates         0
Class              0
dtype: int64
```

In [17]:

```
df.head()
```

Out[17]:

	Temperature	Skin	Give Birth	Aquatic	Aerial Creature	Has Legs	Hibernates	Class
0	1	0	1	0	0	1	0	Mammal
1	0	1	0	0	0	0	1	Reptile
2	0	1	0	1	0	0	0	Fish
3	1	0	1	1	0	0	0	Mammal
4	0	5	0	2	0	1	1	Amphibian

In [18]:

```
df.tail()
```

Out[18]:

	Temperature	Skin	Give Birth	Aquatic	Aerial Creature	Has Legs	Hibernates	Class
10	0	1	0	2	0	1	0	Reptile
11	1	2	0	2	0	1	0	Bird
12	1	3	1	0	0	1	1	Mammal
13	0	1	0	1	0	0	0	Fish
14	0	5	0	2	0	1	1	Amphibian

In [19]:
```
df.describe()
```

Out[19]:

	Temperature	Skin	Give Birth	Aquatic	Aerial Creature	Has Legs	Hibernates
count	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000
mean	0.466667	1.800000	0.400000	0.800000	0.133333	0.666667	0.333333
std	0.516398	1.698739	0.507093	0.861892	0.351866	0.487950	0.487950
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	1.000000	0.000000	1.000000	0.000000	1.000000	0.000000
75%	1.000000	2.500000	1.000000	1.500000	0.000000	1.000000	1.000000
max	1.000000	5.000000	1.000000	2.000000	1.000000	1.000000	1.000000

In [20]:
```
df.corr()
```

Out[20]:

	Temperature	Skin	Give Birth	Aquatic	Aerial Creature	Has Legs	Hibe
Temperature	1.000000	-0.130281	6.000992e-01	-4.172615e-01	0.419314	3.779645e-01	-9.4491
Skin	-0.130281	1.000000	-2.321753e-01	3.610143e-01	-0.191200	4.308637e-01	4.3086
Give Birth	0.600099	-0.232175	1.000000e+00	-4.576043e-01	0.080064	4.006172e-17	-1.60
Aquatic	-0.417261	0.361014	-4.576043e-01	1.000000e+00	-0.376845	-1.885620e-17	3.7712
Aerial Creature	0.419314	-0.191200	8.006408e-02	-3.768446e-01	1.000000	2.773501e-01	1.3867
Has Legs	0.377964	0.430864	4.006172e-17	-1.885620e-17	0.277350	1.000000e+00	2.0000
Hibernates	-0.094491	0.430864	-1.602469e-17	3.771240e-17	0.138675	2.000000e-01	1.0000

In [21]:
```
x= df[['Temperature','Skin','Give Birth','Aquatic','Aerial Creature','Has Legs','
```

```
In [22]: x
```

```
Out[22]: array([[1, 0, 1, 0, 0, 1, 0],
                [0, 1, 0, 0, 0, 0, 1],
                [0, 1, 0, 1, 0, 0, 0],
                [1, 0, 1, 1, 0, 0, 0],
                [0, 5, 0, 2, 0, 1, 1],
                [0, 1, 0, 0, 0, 1, 0],
                [1, 0, 1, 0, 1, 1, 1],
                [1, 2, 0, 0, 1, 1, 0],
                [1, 4, 1, 0, 0, 1, 0],
                [0, 1, 1, 1, 0, 0, 0],
                [0, 1, 0, 2, 0, 1, 0],
                [1, 2, 0, 2, 0, 1, 0],
                [1, 3, 1, 0, 0, 1, 1],
                [0, 1, 0, 1, 0, 0, 0],
                [0, 5, 0, 2, 0, 1, 1]], dtype=int64)
```

```
In [23]: y= df['Class']
```

```
In [24]: y
```

```
Out[24]: 0        Mammal
         1       Reptile
         2          Fish
         3        Mammal
         4     Amphibian
         5       Reptile
         6        Mammal
         7          Bird
         8        Mammal
         9          Fish
         10      Reptile
         11         Bird
         12       Mammal
         13         Fish
         14    Amphibian
         Name: Class, dtype: object
```

```
In [25]: y.value_counts()
```

```
Out[25]: Mammal       5
         Reptile      3
         Fish         3
         Amphibian    2
         Bird         2
         Name: Class, dtype: int64
```

Train Split Model

```
In [26]: from sklearn.model_selection import train_test_split
```

In [27]:

```
x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.3)
```

- Split Model into Train= 70% , Test= 30% Ratio.

SKlearn Model For decision Tree

In [28]:

```
from sklearn.tree import DecisionTreeClassifier
```

In [29]:

```
model = DecisionTreeClassifier()
```

In [30]:

```
model.fit(x_train,y_train)
```

Out[30]:

```
DecisionTreeClassifier()
```

In [31]:

```
y_predict = model.predict(x_test)
```

In [32]:

```
y_predict
```

Out[32]:

```
array(['Reptile', 'Bird', 'Fish', 'Fish', 'Fish'], dtype=object)
```

Impory Model For Checking Score.

In [33]:

```
from sklearn.metrics import accuracy_score
```

In [34]:

```
accuracy_score(y_test,y_predict)*100
```

Out[34]:

```
40.0
```

Creating The Table

In [35]:

```
new_df= pd.DataFrame({'actual':y_test,'predicted':y_predict})
```

In [36]:

```
new_df
```

Out[36]:

	actual	predicted
1	Reptile	Reptile
11	Bird	Bird
14	Amphibian	Fish
10	Reptile	Fish
4	Amphibian	Fish

Confusion matrix

In [37]:
```
from sklearn.metrics import confusion_matrix
```

In [38]:
```
performance= confusion_matrix(y_test,y_predict)
```

In [39]:
```
performance
```

Out[39]:
```
array([[0, 0, 2, 0],
       [0, 1, 0, 0],
       [0, 0, 0, 0],
       [0, 0, 1, 1]], dtype=int64)
```

Decision Tree

In [40]:
```
from sklearn import tree
```

In [41]:
```
feature_name= ['Temperature','Skin','Give Birth','Aquatic','Aerial Creature','Has
```

In [42]:
```
feature_name
```

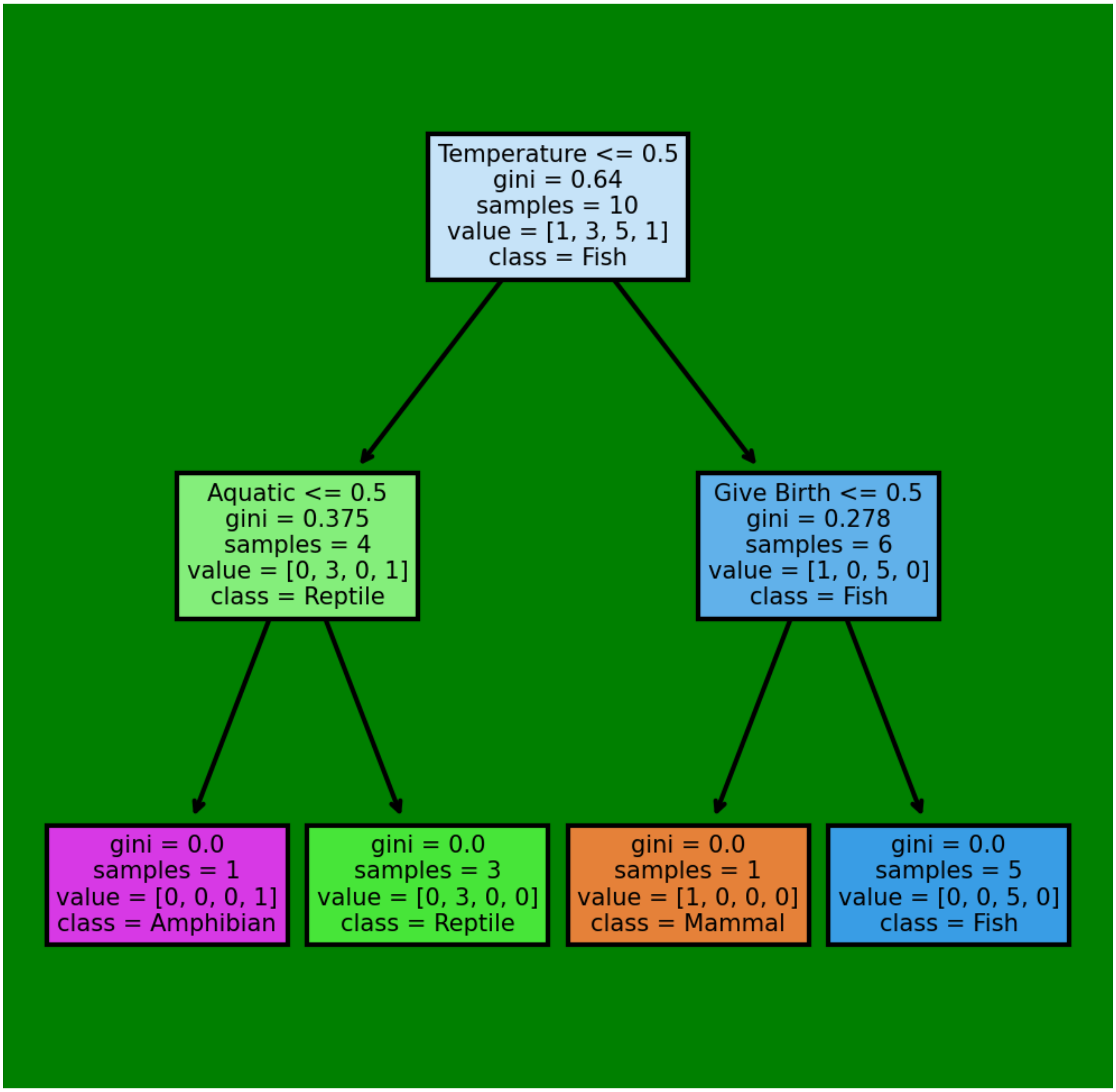
Out[42]:
```
['Temperature',
 'Skin',
 'Give Birth',
 'Aquatic',
 'Aerial Creature',
 'Has Legs',
 'Hibernates']
```

In [43]:
```
class_name= ['Mammal','Reptile','Fish','Amphibian','Bird']
```

In [44]:
```
class_name
```

Out[44]:
```
['Mammal', 'Reptile', 'Fish', 'Amphibian', 'Bird']
```

In [45]:
```
fig,ax= plt.subplots(nrows=1, ncols=1, figsize=(4,4), dpi=300, facecolor='g')
tree.plot_tree(model,feature_names=feature_name, class_names=class_name, filled=T
plt.show()
```



- Task is to find the class of the Creature.

- Creatures are 'Mammal','Reptile','Fish','Amphibian','Bird'

In [ ]: