Classification of Animals into Mammals or Non-Mammals

Name: Chandrashekar Akkenapally

Email:chandrashekar.akkenapally@mst.edu (mailto:chandrashekar.akkenapally@mst.edu)

Course: CS 5402

Assignment: Programming Assignment 3

Date:07/18/2022

Concept Description

Classification of animals into Mammals or Non-Mammals using the 1R classification and Discretization. Building a model Considering the attributes like Legs of the animals, if they lay eggs, presence of tail if they nurse their young with milk, and gestation period to classify animals into Mammals or Non-Mammals. Using the scatter plots, bar charts and various other visualization will understand the attributes clearly.

Data Collection

The client has provided the data set on animal-taxonamy which they had collected in the form of a comma separated file. The provided data set consists of known animals to train and test the classification system with.

Example Description

Animal-taxonamy data set has 19 attributes which will help us to predict the animals into Mammal or Non-Mammal and explore different techniques using the various attributes.

- **animal name:** Displays the animal name .
  Example: aardvark
- **hair** The presence of hair on animals.
  Example: True
- **feathers:** Presence of feathers.
  Example: False
- **eggs:** If they lay eggs or not .
  Example: False
- **milk:** If the animals nurse their young with milk.
  Example: True
- **airborne:** If they are airborne.
  Example: False
- **aquatic:** If they are aquatic.
  Example: False
- **predator:** If they are predatory.
  Example: True
- **toothed:**If the animals have teeth.
  Example: True
- **backbone:** If they have a backbone.
  Example: True
- **breathes:**If the animals breath air.
  Example: True
- **venomous:**If they are venomous.
  Example: False
- **fins:** If they have fins.
  Example: False
- **legs:**The number is legs the animal have.
  Example:4
- **tail:** If the tail present or not.
  Example: False
- **domestic:** If they are domesticated.
  Example: False
- **catsize:**If they are cat sized.
  Example: True
- **gestation:** Gestation period.
  Example: 213.0
- **type:**Type of the animal.
  Example: Mammal

Level of measurement

Nominal

Nominal data are those items which are distinguished by a simple naming system. Animal name and type are name-only, All the other attributes have True or False as the values so, we have to only two outcoms which come under nominal level of measurement.

- animal name
- hair
- feathers
- eggs
- milk
- airborne
- aquatic
- predator
- toothed
- backbone
- breathes
- venomous
- fins
- tail
- domestic
- catsize
- type

Ratio

Mainly the ratio attributes has true zero point. The attributes legs and gestation have zero values.

- Legs
- gestation

Data Import and Wrangling

- Animal-taxonomy csv files are read using pandas.
- Importing all the required functions for data mining and analysis.
- Displaying the top 5 rows from the animal-taxonomy csv files.

In [4]:

```
import numpy as np
import pandas as pd
import collections
import matplotlib.cm as cm
import matplotlib as mpl
from matplotlib import rcParams
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from pandas.plotting import scatter_matrix
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
from sklearn.metrics import confusion_matrix


animal_data = pd.read_csv(r"C:\Users\MYPC\Documents\Introduction to Data Mining\Assignment\
                   ,skip_blank_lines=True,na_filter=True,encoding='latin-1')
animal_data.head() # displays top 5 rows
```

Out[4]:

	animal name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	brea
0	aardvark	True	False	False	True	False	False	True	True	True	
1	anole	False	False	True	False	False	False	False	True	True	
2	antelope	True	False	False	True	False	False	False	True	True	
3	axolotl	False	False	True	False	False	True	False	True	True	F
4	bass	False	False	True	False	False	True	True	True	True	F

In [5]:

```
animal_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 132 entries, 0 to 131
Data columns (total 19 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   animal name  132 non-null    object 
 1   hair         132 non-null    bool   
 2   feathers     132 non-null    bool   
 3   eggs         132 non-null    bool   
 4   milk         132 non-null    bool   
 5   airborne     132 non-null    bool   
 6   aquatic      132 non-null    bool   
 7   predator     132 non-null    bool   
 8   toothed      132 non-null    bool   
 9   backbone     132 non-null    bool   
 10  breathes     132 non-null    bool   
 11  venomous     132 non-null    bool   
 12  fins         132 non-null    bool   
 13  legs         132 non-null    int64  
 14  tail         132 non-null    bool   
 15  domestic     132 non-null    bool   
 16  catsize      132 non-null    bool   
 17  gestation    126 non-null    float64
 18  type         132 non-null    object 
dtypes: bool(15), float64(1), int64(1), object(2)
memory usage: 6.2+ KB
```

In [6]:

```
animals = animal_data.copy()
animals["type"] = np.where(animals["type"] == "mammal", "mammal", "non mammal") #creating a
animals.head()
```

Out[6]:

	animal name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	brea
0	aardvark	True	False	False	True	False	False	True	True	True	
1	anole	False	False	True	False	False	False	False	True	True	
2	antelope	True	False	False	True	False	False	False	True	True	
3	axolotl	False	False	True	False	False	True	False	True	True	F
4	bass	False	False	True	False	False	True	True	True	True	F

Discretization

The attribute gestation is consider for discretization.

In [7]:

```
temp = pd.crosstab(index=animals["gestation"],columns=animals["type"],margins_name="all",ma
temp
```

Out[7]:

type gestation	mammal	non mammal	all
0.0	0	1	1
1.0	0	1	1
2.0	0	3	3
3.0	0	2	2
5.0	0	2	2
...	...	...	...
450.0	1	1	2
540.0	0	2	2
645.0	1	0	1
720.0	0	1	1
all	50	76	126

81 rows × 3 columns

In [8]:

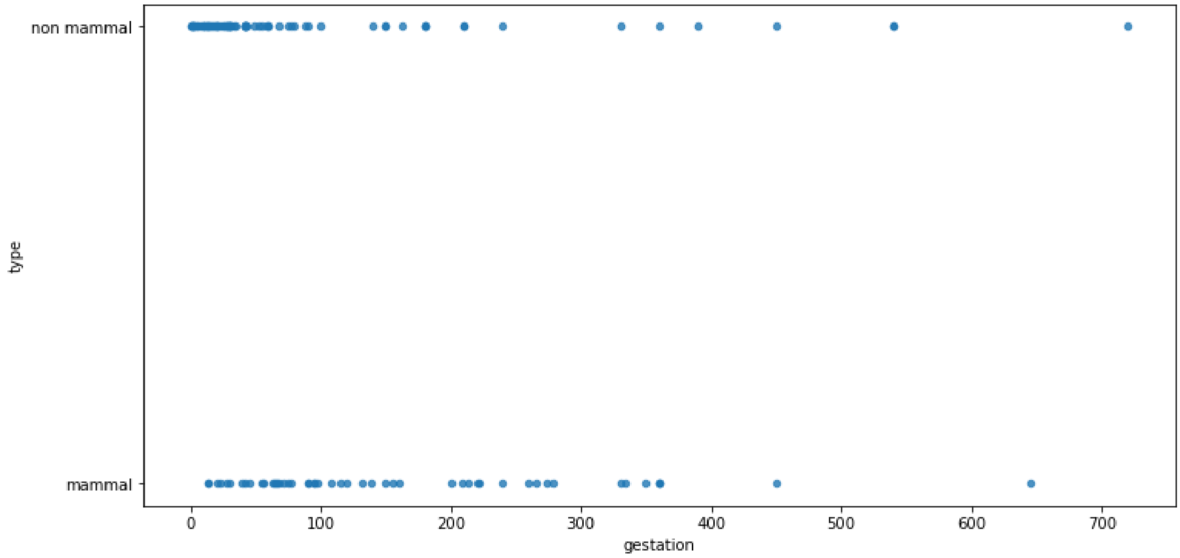
```
from IPython.display import display
display(temp.head(34))
display(temp.tail(46))
```

45.0	1	0	1
49.0	0	1	1
53.0	0	1	1
55.0	1	1	2
56.0	1	0	1
59.0	0	1	1
60.0	0	2	2

Head= Non mammal=52, Mammal=11 Tail= Non Mammal= 24,Mammal=39

In [10]:

```
animals.plot(kind="scatter", x="gestation", y="type",alpha=0.8, figsize=(12,6))
plt.show()
```



- If the gestation period is greater than 60, then they are mammals.
- If the gestation period is less than or equal to 60 they are Non-mammals.
- Error rate of 35 of 126.

```
<hr style="border:2px solid gray"> </hr>

<font size="5" face="Times New Roman"><b>Exploratory Data Analysis:</b></font><br><br>

<ul>
  <li><font size="3" face="Times New Roman">Splitting the animal data set into training data and test data.</font></li>
  <li><font size="3" face="Times New Roman">Handling the missing values.</font></li>
</ul>
```

In [11]:

```
animals.describe()
```

Out[11]:

	legs	gestation
count	132.000000	126.000000
mean	2.916667	119.460317
std	2.261668	141.358588
min	0.000000	0.000000
25%	1.500000	26.500000
50%	4.000000	61.500000
75%	4.000000	162.250000
max	12.000000	720.000000

Handling Missing Values:

In [22]:

```
animals["gestation"].fillna(value=animals["gestation"].mean(), inplace=True) # replacing th
animals.describe()
```

Out[22]:

	legs	gestation
count	132.000000	132.000000
mean	2.916667	119.460317
std	2.261668	138.083427
min	0.000000	0.000000
25%	1.500000	28.000000
50%	4.000000	66.500000
75%	4.000000	156.250000
max	12.000000	720.000000

In [23]:

```
# dividing the data into test and training sets
training_data, test_data = train_test_split(animals, test_size=0.2, random_state=42)
work_set = training_data.copy() # assigning a copy of train set to work_set
```

In [14]:

```
work_set.head()
```

Out[14]:

	animal name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	br
95	rhea	False	True	True	False	False	False	True	False	True	
96	scorpion	False	False	False	False	False	False	True	False	False	
0	aardvark	True	False	False	True	False	False	True	True	True	
12	catfish	False	False	True	False	False	True	True	True	True	
126	wallaby	True	False	False	True	False	False	False	True	True	

In [15]:

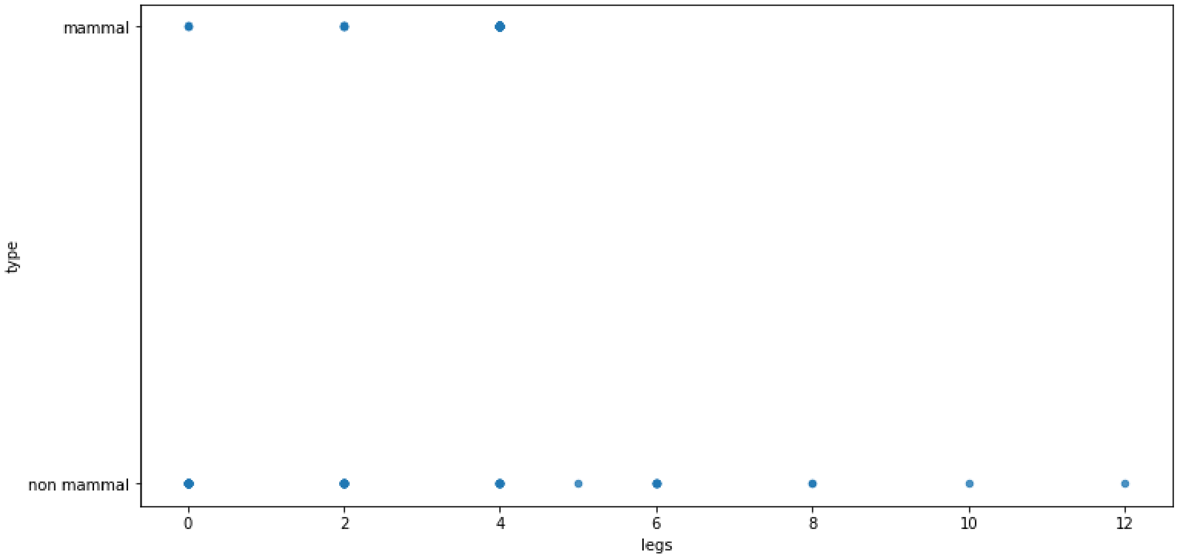
```
work_set.groupby('legs')['type'].value_counts(ascending=True)
```

Out[15]:

```
legs  type      
0     mammal         3
      non mammal    24
2     mammal         5
      non mammal    17
4     non mammal    12
      mammal        31
5     non mammal     1
6     non mammal     8
8     non mammal     2
10    non mammal     1
12    non mammal     1
Name: type, dtype: int64
```

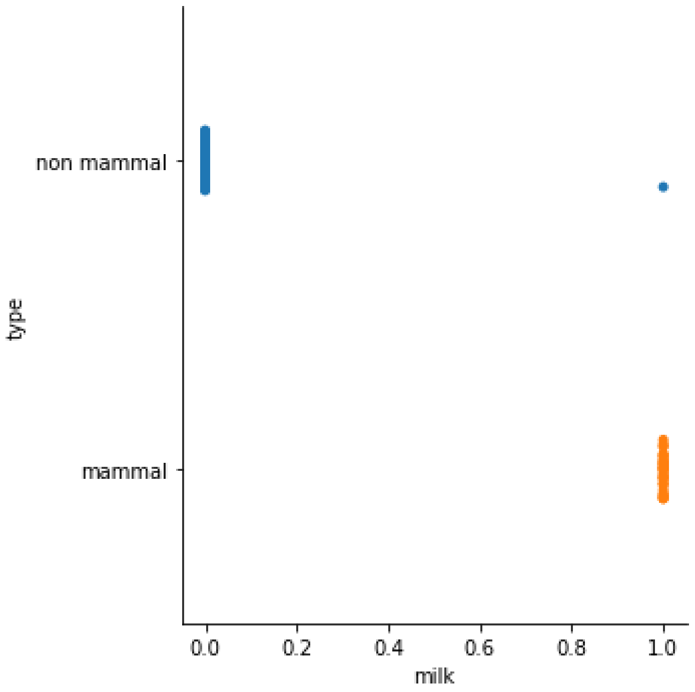
In [16]:

```
work_set.plot(kind="scatter", x="legs", y="type",alpha=0.8, figsize=(12,6))
plt.show()
```



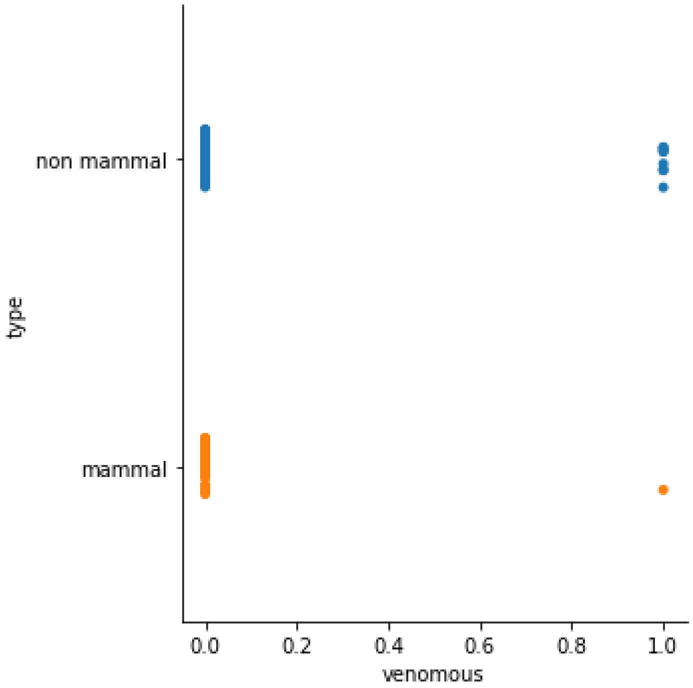
In [17]:

```
sns.catplot(x="milk", y="type", data=work_set)
plt.show()
```



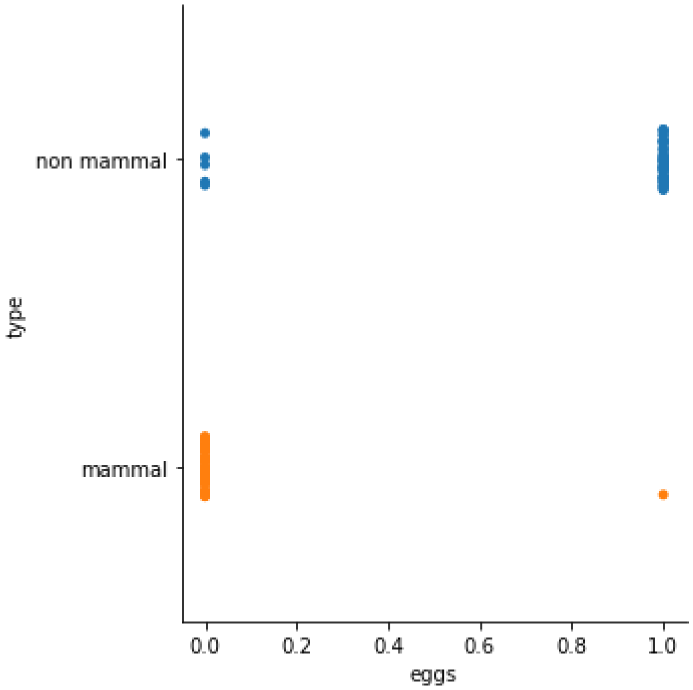
In [18]:

```
sns.catplot(x="venomous", y="type", data=work_set)
plt.show()
```



In [19]:

```
sns.catplot(x="eggs", y="type", data=work_set)
plt.show()
```



Mining or Analytics:

- Let's find out the attribute which is best fit for from 1R classifier.
- Here we consider four attributes that are legs, milk, aquatic, eggs.
- The error rate for each attribute is calculated.

In [20]:

```
atr1= pd.crosstab(index=work_set["legs"],columns=work_set["type"],margins_name='Total',marg
atr2= pd.crosstab(index=work_set["milk"],columns=work_set["type"],margins_name='Total',marg
atr3=pd.crosstab(index=work_set["aquatic"],columns=work_set["type"],margins_name='Total',ma
atr4=pd.crosstab(index=work_set["eggs"],columns=work_set["type"],margins_name='Total',margi
```

In [253]:

```
display(atr1,atr2,atr3,atr4)
```

type	mammal	non mammal	Total
legs			
0	3	24	27
2	5	17	22
4	31	12	43
5	0	1	1
6	0	8	8
8	0	2	2
10	0	1	1
12	0	1	1
Total	39	66	105

type	mammal	non mammal	Total
milk			
False	0	65	65
True	39	1	40
Total	39	66	105

type	mammal	non mammal	Total
aquatic			
False	34	31	65
True	5	35	40
Total	39	66	105

type	mammal	non mammal	Total
eggs			
False	38	5	43
True	1	61	62
Total	39	66	105

Legs

- If,Legs =0 then Non-Mammal(Error=3)
- Legs =2 then Non-Mammal(Error=5)

- Legs =4 then Mammal (Error=12)
- Legs =5 then Non-Mammal (Error=0)
- Legs =6 then Non-Mammal (Error=0)
- Legs =8 then Non-Mammal (Error=0)
- Legs =10 then Non-Mammal (Error=0)
- Legs =12 then Non-Mammal (Error=0)

Milk

- If the Animal nurse their young with milk i.e., represented by True then it is a Mammal(Error=1)
- If the Animal doesn't nurse their young with milk i.e., represented by False then it is a Non Mammal(Error=0)

Aquatic

- If the animal is aquatic i.e., represented by True then it is a Non Mammal(Error=5)
- If the animal isn't aquatic i.e., represented by False then it is a Mammal(Error=31)

Eggs

- If the animal lay eggs ie. True then, it is a Non Mammal (Error=1)
- If the animal doesn't lay eggs i.e., Represented with False then, it is a Mammal(Error=5)

In [21]:

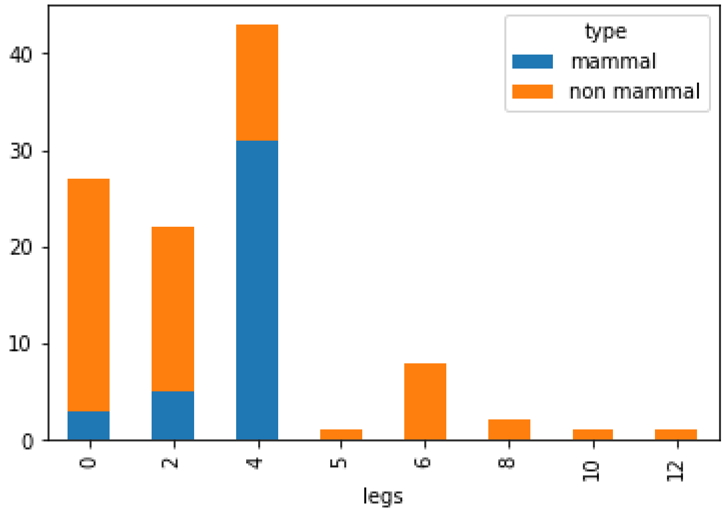
```python
Total_value = 105
Legs_error=20
Milk_error=1
Aquatic_error=36
Eggs_error=6

Legs_errorrate= Legs_error/Total_value
print('legs error rate is '+str(Legs_errorrate))
Milk_errorrate=Milk_error/Total_value
print('Milk error rate is '+str(Milk_errorrate))
Aquatic_errorrate=Aquatic_error/Total_value
print('Aquatic error rate is '+str(Aquatic_errorrate))
Eggs_errorrate=Eggs_error/Total_value
print('Eggs error rate is '+str(Eggs_errorrate))
```

```
legs error rate is 0.19047619047619047
Milk error rate is 0.009523809523809525
Aquatic error rate is 0.34285714285714286
Eggs error rate is 0.05714285714285714
```

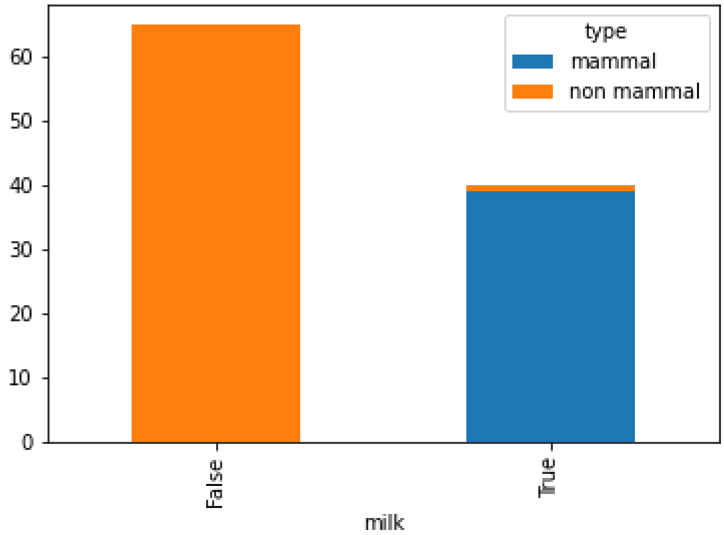
```html
<font size="4" face="Times New Roman"><b>Considering all the error rates from the above
attributes, we can see that the Milk attribute has the lowest error rate.</b></font><br>
<br>
<font size="4" face="Times New Roman"><b>1R classifier rule set is:</b></font><br><br>
<ul>
  <li><font size="3" face="Times New Roman">If the Animal nurse their young with milk
i.e., represented by True then it is a Mammal(True->Mammal)</font></li>
  <li><font size="3" face="Times New Roman">If the Animal doesn't nurse their young with
milk i.e., represented by False then it is a Non Mammal(False->Non Mammal)</font></li>
</ul>
```

In [255]:

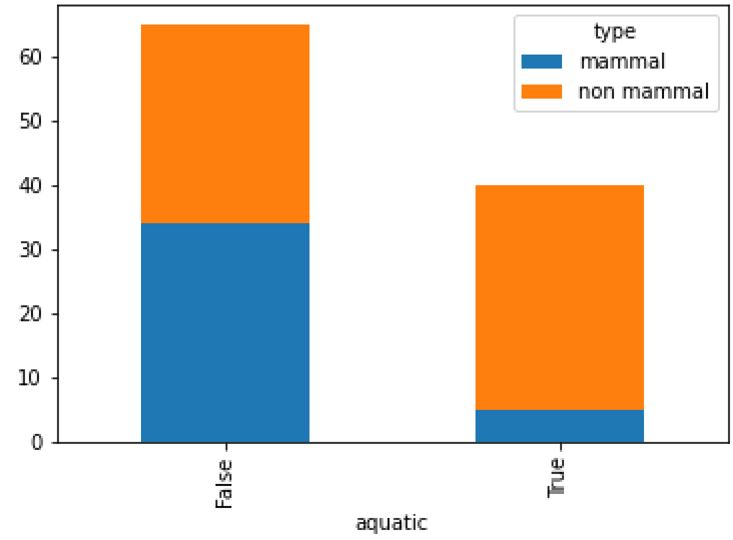
```
pd.crosstab(index=work_set["legs"],columns=work_set["type"]).plot.bar(stacked=True)
display(plt.show())
pd.crosstab(index=work_set["milk"],columns=work_set["type"]).plot.bar(stacked=True)
display(plt.show())
pd.crosstab(index=work_set["aquatic"],columns=work_set["type"]).plot.bar(stacked=True)
display(plt.show())
pd.crosstab(index=work_set["eggs"],columns=work_set["type"]).plot.bar(stacked=True)
display(plt.show())
```



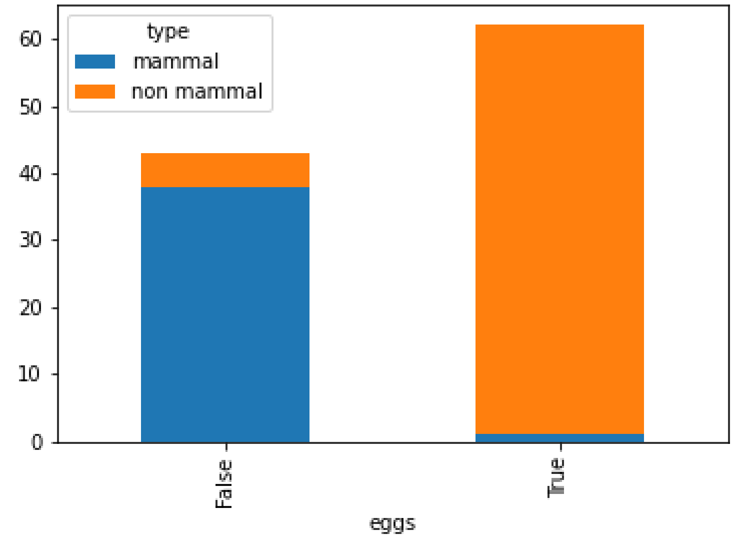
```
None
```



```
None
```



```
None
```



```
None
```

```
<hr style="border:2px solid gray"> </hr>
<font size="5" face="Times New Roman"><b>Evaluation:</b></font><br><br>
<ul>
  <li><font size="3" face="Times New Roman">Here we use the test data set to create a
confusion matrix for the manually created 1R classifier.</font></li>
  <li><font size="3" face="Times New Roman">Accuracy and F1-Score for the model is
determined.</font></li>
</ul>
```

In [256]:

```
M_true = test_data.copy()
M_pred = test_data.copy()
```

In [257]:

```
M_true.head()
```

Out[257]:

	animal name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	l
56	human	True	False	False	True	False	False	True	True	True	
83	pheasant	False	True	True	False	True	False	False	False	True	
19	chupacabra	True	False	False	True	False	False	True	True	True	
31	duck	False	True	True	False	True	True	False	False	True	
76	opossum	True	False	False	True	False	False	True	True	True	

In [258]:

```
M_pred.head()
```

Out[258]:

	animal name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	l
56	human	True	False	False	True	False	False	True	True	True	
83	pheasant	False	True	True	False	True	False	False	False	True	
19	chupacabra	True	False	False	True	False	False	True	True	True	
31	duck	False	True	True	False	True	True	False	False	True	
76	opossum	True	False	False	True	False	False	True	True	True	

In [259]:

```
M_true.loc[M_true["milk"]=="True","type"]= "mammal"
M_true.loc[M_true["milk"]=="False","type"]= "non mammal"
```

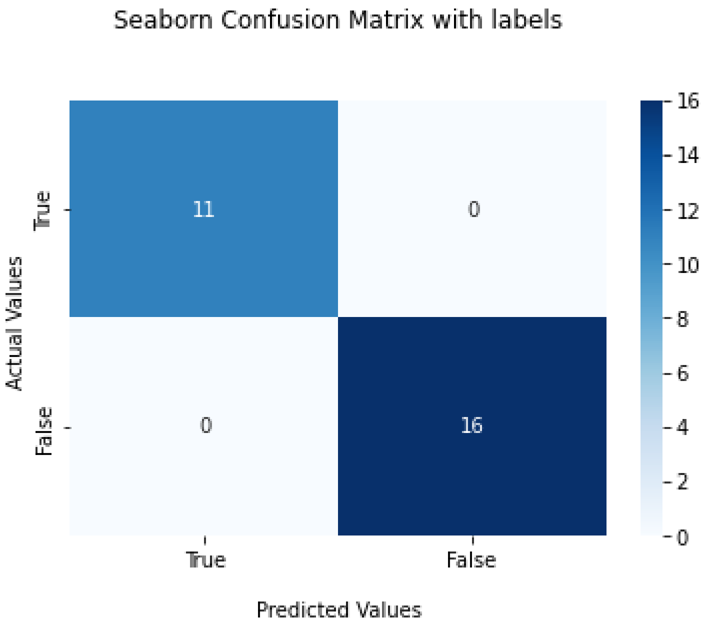
In [260]:

```

from sklearn.metrics import confusion_matrix
binary_confusion_matrix = confusion_matrix(M_true["type"], M_pred["type"])
print("Binary confusion matrix:\n%s" % binary_confusion_matrix)
```

```
Binary confusion matrix:
[[11  0]
 [ 0 16]]
```

In [261]:

```
ax = sns.heatmap(binary_confusion_matrix, annot=True, cmap='Blues')

ax.set_title('Seaborn Confusion Matrix with labels\n\n');
ax.set_xlabel('\nPredicted Values')
ax.set_ylabel('Actual Values ');

ax.xaxis.set_ticklabels(['True','False'])
ax.yaxis.set_ticklabels(['True','False'])

## Display the visualization of the Confusion Matrix.
plt.show()
```



In [262]:

```
TP=11 # True positive
FP=0 # False positive
FN=0 # False Negative
TN=16 # True Negative
Total=27 # Total number of values
Accuracy = (TP+TN)/27
print(Accuracy)
```

```
1.0
```

In [263]:

```
Precision = TP/(TP+FP)
Recall = TP/(TP+FN)
print("Precision = " +str(Precision))
print("Recall = "+ str(Recall))
```

```
Precision = 1.0
Recall = 1.0
```

In [264]:

```
F_measure= (2*Recall*Precision)/(Recall+Precision)
print("F_measure = "+str(F_measure))
```

```
F_measure = 1.0
```

Results:

- After choosing the different attributes for classification of Mammal or Non Mammal, we can see that MIlk attribute showed best result when compared to other attributes..
- In 1R classifier, the error rate of milk attribute is 0.00952,legs error rate is 0.19047, Aquatic error rate is 0.34285,Eggs error rate is 0.05714.
- The error rate for the milk attribute was very low in 1R classifier.
- After testing the test data with the 1R classifier data, the milk attribute Accuracy is 1 which is 100%.
- The classification of animals as Mammals or Non mammals can be done based on the attribute Milk.
- If the animals nurse their young with milk then it's mammal and if not it's a Non mammal based on our classifier model.

Reference:

- https://pandas.pydata.org/docs (https://pandas.pydata.org/docs) .

- https://realpython.com/ (https://realpython.com/).

- https://towardsdatascience.com/ (https://towardsdatascience.com/).

- https://pythongeeks.org/python-scatter-plot/ (https://pythongeeks.org/python-scatter-plot/).

- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html).
