Faculty of science and technology

Assignment Coversheet

Student ID Number & Student Name	Abdul Rafay Cheema (u3241012)
Unit Name	Software Technology 1
Unit Number	4483
Unit Tutor	Pranav Gupta
Assignment Name	ST1 Capstone Project – Semester 1 2023
Due Date	12th May 2023
Date Submitted	04th May 2023

I have kept a photocopy or electronic copy of my assignment as instructed.

Student Declaration:

I certify that the attached assignment is my own work. Material drawn from other sources has been appropriately and fully acknowledged as to author/creator, source, and other bibliographic details.

Sign of Student: *Abdul Rafay Cheema* *Date Created: 04th May 2023*

Table of Contents

Introduction

This report describes the details of Python Capstone Project for ST1 unit within the scope of the project requirements provided in the assignment handout [1]. I have decided to work on the project using a zoo animals dataset available in Kaggle data repositories [2].

A system has been created by scientists to classify every organism in the animal kingdom, but there are some exceptions that do not fit into the general guidelines of animal classification. The classification of animals is mainly based on their physical and developmental features, including their body structure [3].

Vertebrates are animals that have a backbone. They include fish, amphibians, birds, and mammals. Clearly, most animal species do not have a backbone. There are about 50,000 vertebrate species. They are placed in nine different classes. Five of the classes are fish. The other classes are amphibians, reptiles, birds, and mammals[4].

The classification of this dataset is essentially based on natural classification. Natural classification refers to a method of categorizing organisms where similarities between them are identified first, and then their shared characteristics are determined. This type of classification system is based on the idea that all members of a specific group share a common ancestor, making it possible to predict the traits that are shared among species within a group. However, one disadvantage of this scheme is that it is subject to change as new information about organisms is discovered [5].

Methodology

The methodology used for developing the software platform involves 3 stages as outlined below:

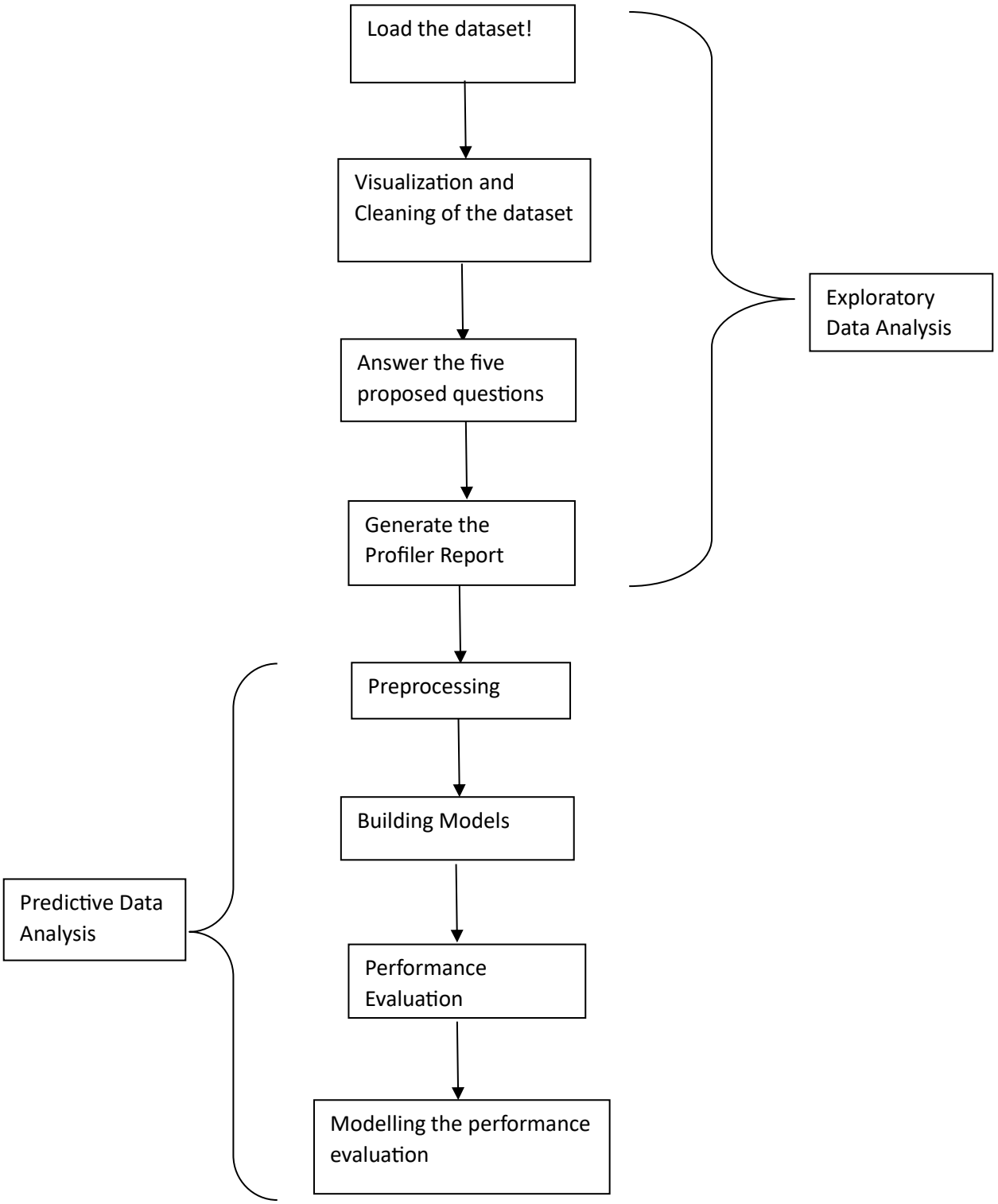
1. Design and development of decision support algorithms based on exploratory data analysis and predictive analytics, for identifying the best performing algorithm for solving a real world problem.

2. Implementation of best performing algorithm as a desktop Tkinter software tool.

3. Deployment of the tool as a web or cloud enabled platform tool.

Stage 1: Algorithm Design Stage

The initial stage, also known as Stage 1, is crucial and its importance cannot be overstated. The type of algorithms designed for exploratory data analysis and predictive analytics will vary based on the intricacy of the problem and the data set being used. Nevertheless, the process for developing these algorithms will follow the same basic steps as illustrated in the flowchart presented below:



Dataset Description

- There is only one dataset used for this project and it is publicly available from Kaggle [2].

- It is a simple database containing 17 Boolean-valued attributes describing animals. The "type" attribute appears to be the class attribute.
- The dataset consists of 102 observations, 17 features and 1 target/class attribute.
- The 17 features include the results of the features and attributes for the animals.
- The target/class variable includes the result of the answers of the several features of the animals.
- Therefore, the task at hand is to develop a software tool to predict the type of the animal, using the results of the afore mentioned questions about the features and the attributes of the different animals.
- The data was originally created by ULRIK THYGE PEDERSEN.

Exploratory Data Analysis

The first phase of the software development activity involved understanding the data, basic exploratory data analysis and visualisation. PyCharm was chosen as the experimental environment, and a code based on a module was used to open the profiler report on a web page in the browser. The python languages used to create the scripts which would give us the information regarding the data, answer the five asked questions and generate the profiler report.Please see the following pieces of code and the corresponding answers relating to EDA:

Here we mount the file and analyse the data to begin with:

```
import pandas as pd

data = pd.read_csv('C:\\Users\Abdul Rafay Cheema\\Desktop\\Zoo.csv')

# Understanding and visualizing the dataset to begin with
print(data.head())
print(data.tail())

print(data.shape)

print(data.columns)

print(data.nunique())

print(data.describe())
```

```
# Next, we remove all the null and duplicate rows to clean our dataset and start performing our EDA.
print(data.dropna())
print(data.drop_duplicates())
```

The above code gives the following result

```
        animal       hair   feathers  ...  domestic   catsize        type
0   b'aardvark'    b'true'   b'false'  ...   b'false'    b'true'   b'mammal'
1   b'antelope'    b'true'   b'false'  ...   b'false'    b'true'   b'mammal'
2       b'bass'   b'false'   b'false'  ...   b'false'   b'false'     b'fish'
3       b'bear'    b'true'   b'false'  ...   b'false'    b'true'   b'mammal'
4       b'boar'    b'true'   b'false'  ...   b'false'    b'true'   b'mammal'

[5 rows x 18 columns]
         animal       hair   feathers  ...  domestic   catsize             type
96    b'wallaby'    b'true'   b'false'  ...   b'false'    b'true'         b'mammal'
97       b'wasp'    b'true'   b'false'  ...   b'false'   b'false'         b'insect'
98       b'wolf'    b'true'   b'false'  ...   b'false'    b'true'         b'mammal'
99       b'worm'   b'false'   b'false'  ...   b'false'   b'false'  b'invertebrate'
100      b'wren'   b'false'    b'true'  ...   b'false'   b'false'           b'bird'

[5 rows x 18 columns]
(101, 18)
Index(['animal', 'hair', 'feathers', 'eggs', 'milk', 'airborne', 'aquatic',
       'predator', 'toothed', 'backbone', 'breathes', 'venomous', 'fins',
       'legs', 'tail', 'domestic', 'catsize', 'type'],
      dtype='object')
animal       100
hair           2
feathers       2
eggs           2
milk           2
airborne       2
aquatic        2
predator       2
toothed        2
backbone       2
breathes       2
venomous       2
fins           2
legs           6
tail           2
domestic       2
catsize        2
type           7
dtype: int64
             legs
count  101.000000
mean     2.841584
std      2.033385
min      0.000000
25%      2.000000
50%      4.000000
75%      4.000000
max      8.000000
```

```
          animal       hair   feathers  ...   domestic    catsize              type
0     b'aardvark'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
1     b'antelope'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
2         b'bass'   b'false'   b'false'  ...   b'false'   b'false'            b'fish'
3         b'bear'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
4         b'boar'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
..            ...        ...        ...  ...         ...         ...                ...
96     b'wallaby'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
97        b'wasp'    b'true'   b'false'  ...   b'false'   b'false'          b'insect'
98        b'wolf'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
99        b'worm'   b'false'   b'false'  ...   b'false'   b'false'  b'invertebrate'
100       b'wren'   b'false'    b'true'  ...   b'false'   b'false'           b'bird'

[101 rows x 18 columns]
          animal       hair   feathers  ...   domestic    catsize              type
0     b'aardvark'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
1     b'antelope'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
2         b'bass'   b'false'   b'false'  ...   b'false'   b'false'            b'fish'
3         b'bear'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
4         b'boar'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
..            ...        ...        ...  ...         ...         ...                ...
96     b'wallaby'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
97        b'wasp'    b'true'   b'false'  ...   b'false'   b'false'          b'insect'
98        b'wolf'    b'true'   b'false'  ...   b'false'    b'true'          b'mammal'
99        b'worm'   b'false'   b'false'  ...   b'false'   b'false'  b'invertebrate'
100       b'wren'   b'false'    b'true'  ...   b'false'   b'false'           b'bird'

[101 rows x 18 columns]
```

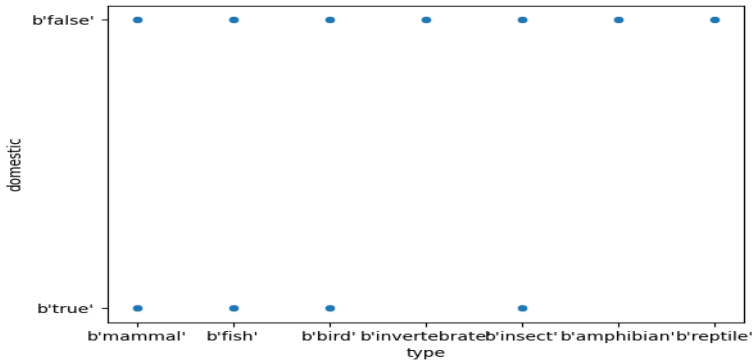
Now we import modules to start our EDA:

```
# Now we start performing Exploratory Data Analysis based on our 5 Questions
import matplotlib.pyplot as plt
import seaborn as sns
```

Now we start answering our five questions:

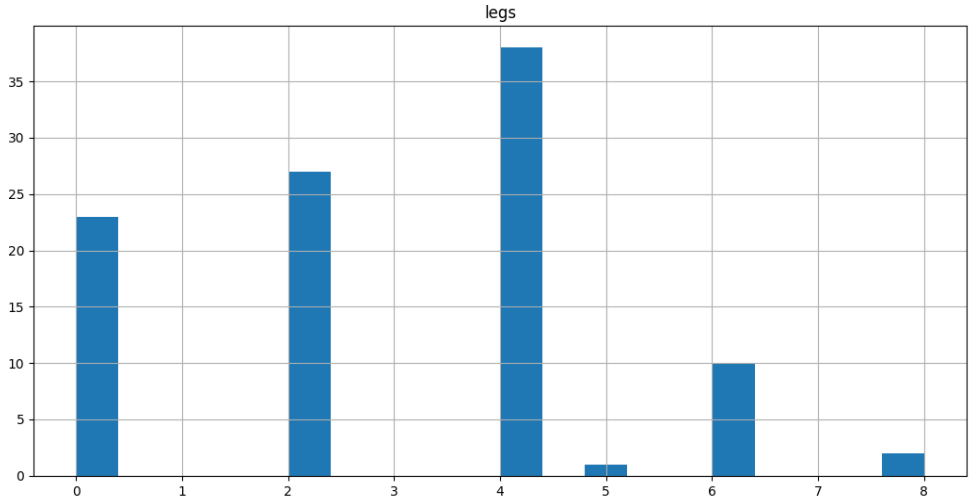
Question no.1: Show the trend of domestication with regards to the type of animals

```
sns.scatterplot(x='type', y='domestic', data=data)
plt.show()
```



Question 2: For each specific number of legs, how many animals from the dataset posses them?

```
fig = plt.figure(figsize=(8, 8))
ax = fig.gca()
data.hist(ax=ax, bins=20)
plt.show()
```



Question 3: What is the average number of legs present in the dataset?

```
average = round(data['legs'].mean())
print('The average number of legs amongst all the animals is', ave
```

```
The average number of legs amongst all the animals is 3
```

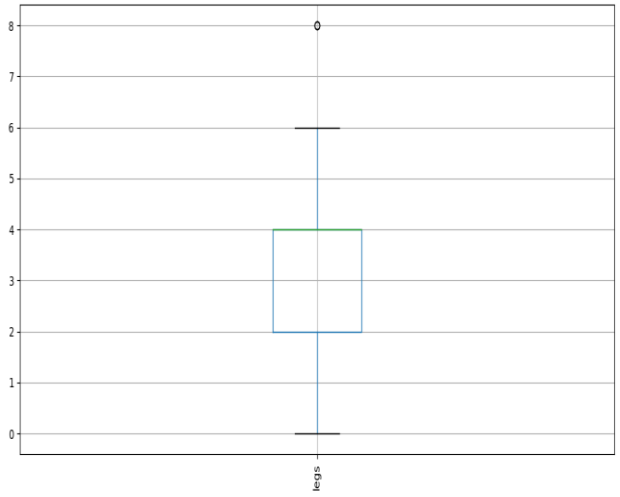
Question 4: What are the data types of each column (e.g., numerical, categorical, text)?

```
data_types = data.dtypes
print('Data types of each column:')
print(data_types)
```

```
Data types of each column:
animal        object
hair          object
feathers      object
eggs          object
milk          object
airborne      object
aquatic       object
predator      object
toothed       object
backbone      object
breathes      object
venomous      object
fins          object
legs           int64
tail          object
domestic      object
catsize       object
type          object
dtype: object
```

Question no.5: Are there any missing values or outliers in the dataset?

```
missing_values = data.isna().sum()
print('Missing values:')
print(missing_values)

plt.figure(figsize=(10, 8))
data.boxplot()
plt.xticks(rotation=90)
plt.show()
```




```
Missing values:
animal        0
hair          0
feathers      0
eggs          0
milk          0
airborne      0
aquatic       0
predator      0
toothed       0
backbone      0
breathes      0
venomous      0
fins          0
legs          0
tail          0
domestic      0
catsize       0
type          0
dtype: int64
```

Next, we generate the profiler report:

```
import pandas as pd
import numpy as np
from pandas_profiling import ProfileReport

profile = ProfileReport(data, title="Zoo EDA", html={'style':
{'full_width': True}})
# profile.to_notebook_iframe()
profile.to_file("EDA")
import webbrowser
import os

chrome_path = r"C:\Program Files\Google\Chrome\Application\chrome.exe"
url = "file:/" + os.path.realpath("EDA.html")
webbrowser.register('chrome', None,
webbrowser.BackgroundBrowser(chrome_path))
webbrowser.get('chrome').open_new_tab(url)
```

This code generates a browser webpage on google chrome, it does take 40 secs to 60 secs for the code to run through and generate the webpage:

Click (Ctrl+Click) on the following link to see the webpage:

file:///C:/Users/Abdul%20Rafay%20Cheema/Desktop/EDA.html

Predictive Data Analytics Stage

To carry out predictive analytics, a number of procedures must be followed. These steps consist of pre-processing, comparing classifiers to determine the most suitable machine learning classifier, and assessing performance using various objective metrics like accuracy, classification report, confusion matrix, and prediction report. The Python scikit-learn package was utilized to obtain these metrics.

• **Pre-processing**: Since the dataset consists of a combination of continuous and categorical attributes/variables, there is a need to pre-process the data with attribute transformation, standardization and normalisation. We used scikit-learn's OrdinalEncoder() function to perform attribute transformation.

• **Normalisation** of the independent values of the dataframe by was done by dropping the target from the dataframe, normalising it, and then reattaching the target to the data frame.

```
# Now we start our Predictive Data Analysis (PDA)
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

from sklearn.exceptions import DataDimensionalityWarning
from sklearn import preprocessing
from sklearn.preprocessing import OrdinalEncoder

for col in data:
    if data[col].dtype == 'object':
```

```
        data[col] =
OrdinalEncoder().fit_transform(data[col].values.reshape(-1, 1))
data

class_label = data['type']
data = data.drop(['type'], axis=1)
data = (data - data.min()) / (data.max() - data.min())
data['type'] = class_label
data

# pre-processing
zoo_data = data.copy()
le = preprocessing.LabelEncoder()
animal = le.fit_transform(list(zoo_data["animal"]))
hair = le.fit_transform(list(zoo_data["hair"]))
feathers = le.fit_transform(list(zoo_data["feathers"]))
eggs = le.fit_transform(list(zoo_data["eggs"]))
milk = le.fit_transform(list(zoo_data["milk"]))
airborne = le.fit_transform(list(zoo_data["airborne"]))
aquatic = le.fit_transform(list(zoo_data["aquatic"]))
predator = le.fit_transform(list(zoo_data["predator"]))
tooth = le.fit_transform(list(zoo_data["toothed"]))
backbone = le.fit_transform(list(zoo_data["backbone"]))
breathes = le.fit_transform(list(zoo_data["breathes"]))
venomous = le.fit_transform(list(zoo_data["venomous"]))
fins = le.fit_transform(list(zoo_data["fins"]))
legs = le.fit_transform(list(zoo_data["legs"]))
tail = le.fit_transform(list(zoo_data["tail"]))
domestic = le.fit_transform(list(zoo_data["domestic"]))
catsize = le.fit_transform(list(zoo_data["catsize"]))
family = le.fit_transform(list(zoo_data["type"]))
# Modeling and preparation

# Predictive analytics model development by comparing different Scikit-
learn classification algorithms
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import accuracy_score
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier

x = list(
    zip(hair, feathers, eggs, milk, airborne, aquatic, predator, tooth,
backbone, breathes, venomous, fins, legs, tail,
        domestic, catsize))
y = list(animal)
# Test options and evaluation metric
```

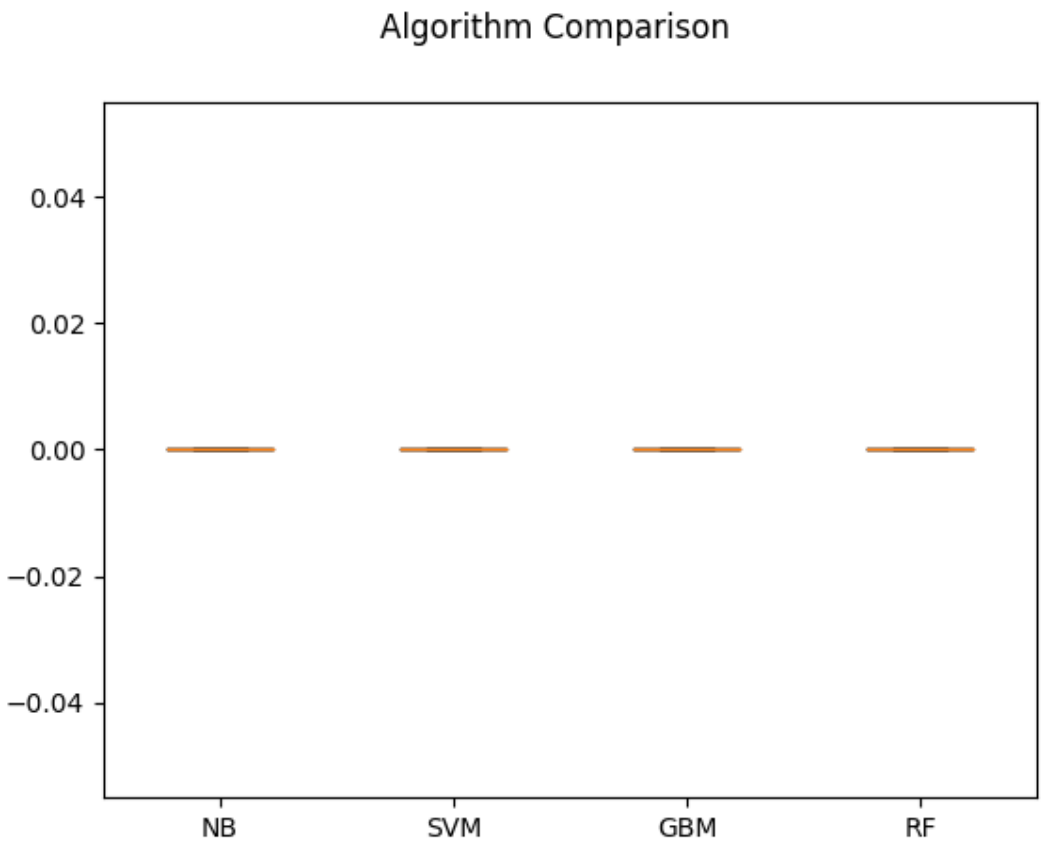
```
num_folds = 5
seed = 7
scoring = 'accuracy'
# Model Test/Train
# Splitting what we are trying to predict into 4 different arrays -
# X train is a section of the x array(attributes) and vise versa for
Y(features)
# The test data will test the accuracy of the model created
import sklearn.model_selection

x_train, x_test, y_train, y_test =
sklearn.model_selection.train_test_split(x, y, test_size=0.20,
random_state=seed)

np.shape(x_train), np.shape(x_test)
models = []
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))
models.append(('GBM', GradientBoostingClassifier()))
models.append(('RF', RandomForestClassifier()))
# evaluate each model in turn
results = []
names = []
print("Performance on Training set")
for name, model in models:
    kfold = KFold(n_splits=num_folds, shuffle=True, random_state=seed)
    cv_results = cross_val_score(model, x_train, y_train, cv=kfold,
scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    msg += '\n'
    print(msg)
```

The above code will print these performance evaluation sets:

```
Performance on Training set
NB: 0.000000 (0.000000)

SVM: 0.000000 (0.000000)

GBM: 0.000000 (0.000000)

RF: 0.000000 (0.000000)
```

The figures should not be zero, but it has been concluded that the values are zero when the dataset is faulty.

```
# Compare Algorithms' Performance
fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```



- Now we perform Model Evaluation by testing with independent/external test data set.
- Then we make predictions on validation/test dataset.
- After which we perform Model Evaluation by testing with independent/external test data set.
- Then we Make predictions on validation/test dataset **.**

```
models.append(('DT', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))
models.append(('GBM', GradientBoostingClassifier()))
models.append(('RF', RandomForestClassifier()))
dt = DecisionTreeClassifier()
nb = GaussianNB()
gb = GradientBoostingClassifier()
```

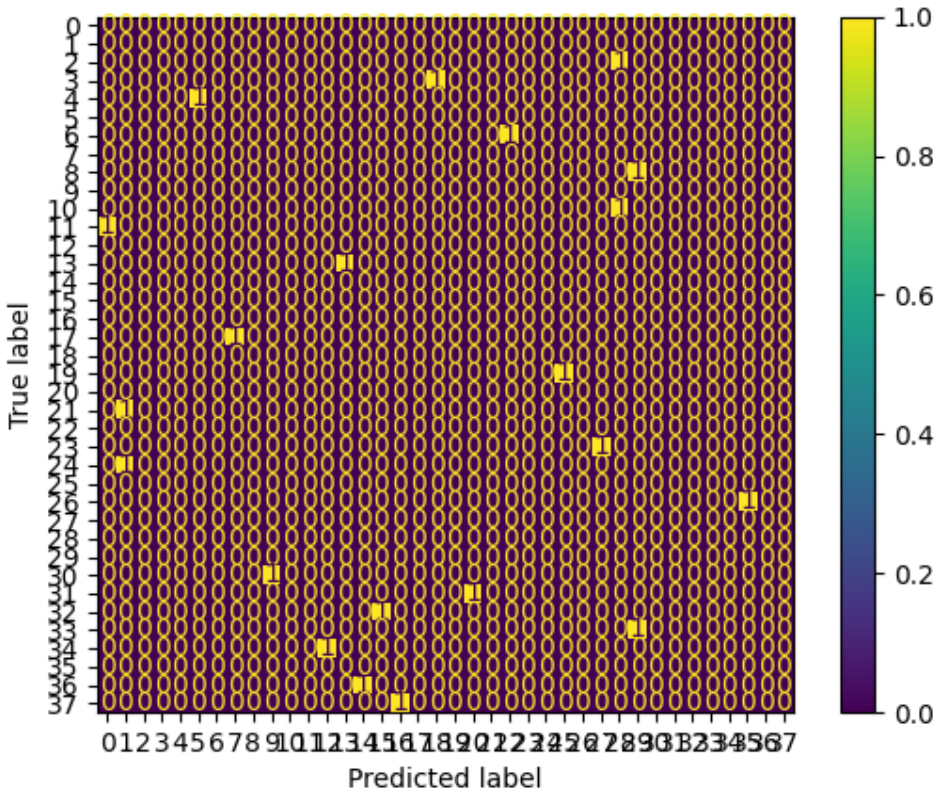
```
rf = RandomForestClassifier()
best_model = rf
best_model.fit(x_train, y_train)
y_pred = best_model.predict(x_test)
print("Best Model Accuracy Score on Test Set:", accuracy_score(y_test,
y_pred))

print(classification_report(y_test, y_pred))
```

This code will give:

```
Best Model Accuracy Score on Test Set: 0.047619047619047616
```

- **Model Performance Evaluation Metric 2-Confusion matrix**

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot()
plt.show()
```



	precision	recall	f1-score	support
1	0.00	0.00	0.00	0
10	0.00	0.00	0.00	0
11	0.00	0.00	0.00	1
12	0.00	0.00	0.00	1
13	0.00	0.00	0.00	1
14	0.00	0.00	0.00	0
15	0.00	0.00	0.00	1
16	0.00	0.00	0.00	0
18	0.00	0.00	0.00	1
19	0.00	0.00	0.00	0
20	0.00	0.00	0.00	1
22	0.00	0.00	0.00	1
24	0.00	0.00	0.00	0
25	1.00	1.00	1.00	1
26	0.00	0.00	0.00	0
32	0.00	0.00	0.00	0
35	0.00	0.00	0.00	0
36	0.00	0.00	0.00	1
37	0.00	0.00	0.00	0
39	0.00	0.00	0.00	1
40	0.00	0.00	0.00	0
44	0.00	0.00	0.00	1
45	0.00	0.00	0.00	0
48	0.00	0.00	0.00	1
49	0.00	0.00	0.00	1
50	0.00	0.00	0.00	0
51	0.00	0.00	0.00	1
53	0.00	0.00	0.00	0
56	0.00	0.00	0.00	0
59	0.00	0.00	0.00	0
65	0.00	0.00	0.00	1
70	0.00	0.00	0.00	1
77	0.00	0.00	0.00	1
85	0.00	0.00	0.00	1
87	0.00	0.00	0.00	1
90	0.00	0.00	0.00	0
92	0.00	0.00	0.00	1
93	0.00	0.00	0.00	1
accuracy			0.05	21
macro avg	0.03	0.03	0.03	21
weighted avg	0.05	0.05	0.05	21

- **Model Evaluation Metric 4- Prediction report**

```
for x in range(len(y_pred)):
    print("Predicted: ", y_pred[x], "Actual: ", y_test[x], "Data: ",
x_test[x], )
```

This will give:

```
Predicted:  16 Actual:  36 Data:  (0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0)
Predicted:  25 Actual:  25 Data:  (0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 2, 0, 0, 0)
Predicted:  32 Actual:  77 Data:  (0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0)
Predicted:  35 Actual:  93 Data:  (1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 2, 1, 0, 0)
Predicted:  53 Actual:  48 Data:  (1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 2, 1, 0, 0)
Predicted:  45 Actual:  15 Data:  (0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 4, 0, 0, 0)
Predicted:  26 Actual:  92 Data:  (1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0)
Predicted:  40 Actual:  70 Data:  (0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1)
Predicted:  59 Actual:  85 Data:  (0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1)
Predicted:  1 Actual:  22 Data:  (1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 2, 1, 0, 1)
Predicted:  14 Actual:  13 Data:  (0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
Predicted:  50 Actual:  39 Data:  (1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 4, 0, 0, 0)
Predicted:  90 Actual:  51 Data:  (0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 2, 1, 0, 0)
Predicted:  37 Actual:  12 Data:  (0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0)
Predicted:  24 Actual:  87 Data:  (0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 4, 0, 0, 0)
Predicted:  10 Actual:  44 Data:  (1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 2, 1, 0, 1)
Predicted:  56 Actual:  11 Data:  (0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0)
Predicted:  19 Actual:  65 Data:  (0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1)
Predicted:  56 Actual:  20 Data:  (0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0)
Predicted:  59 Actual:  18 Data:  (0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1)
Predicted:  10 Actual:  49 Data:  (1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 2, 1, 0, 1)

Process finished with exit code 0
```

Stage 2: Implementation/Deployment Stage

After finding out the best performing algorithm and machine learning model for prediction of the zoo dataset, we now move forward to design and make a GUI program which will use the code we designed earlier to implement the code.

The GUI has been designed in an interactive way where the customer will be asked to enter the details, yes or no, about the features, characteristics, and attributes of the animals.

Based on these answers the code will predict which type of animal we are entering the details about.

The following link will take you to the google drive file which contains the dataset itself, the GUI code and the dataset model it is based on.

GUI:

https://drive.google.com/drive/folders/1UCZUqgQUB1WvSHJyAZKpeY3HALXX8PWa?usp=share_link

Conclusions

The following report outlines the progress made on the ST1 capstone project, which involves designing, developing, implementing, and deploying a software platform that uses Python to create data-driven models for predicting zoo animal behaviour. The report provides detailed information on the zoo dataset and analyses its variables, resulting in the creation of a platform that uses historical data and insights to predict future risks or probabilities related to animal behaviour. This predictive analytics tool is available as a desktop software making it accessible to students, animal experts, and other curious individuals seeking to expand their knowledge about animals.

References:

1. https://uclearn.canberra.edu.au/courses/13571/assignments/105232

2. Ulrik Thyge Pedersen, “Zoo Animals,” *Kaggle.com*, 2023.https://www.kaggle.com/datasets/uciml/zoo-animal-classification

3. “Features Used to Classify Animals | Biology II,” *Lumenlearning.com*, 2023. https://courses.lumenlearning.com/suny-biology2xmaster/chapter/features-used-to-classify-animals/ (accessed May 04, 2023).

4. CK-12 Foundation, “Welcome to CK-12 Foundation | CK-12 Foundation,” *CK-12 Foundation*, 2023. https://www.ck12.org/book/ck-12-third-grade-science/section/3.5/ (accessed May 04, 2023).

5. “Classification | BioNinja,” *Bioninja.com.au*, 2023. https://ib.bioninja.com.au/standard-level/topic-5-evolution-and-biodi/53-classification-of-biodiv/classification.html (accessed May 04, 2023).

5. https://towardsdatascience.com/creating-restful-apis-using-flask-and-python-655bad51b24