***K.T.S.P Mandal’s***

***K.M.C College Khopoli***

***DEPARTMENT OF COMPUTER SCIENCE***

***KHOPOLI–410203***

*A Project Report*

*On*

***Zoo Animal Classification***

*Submitted To*

***University of Mumbai***

*By*

***Miss Neha Narendra Ghonge***

*Under Guidance Of*

***Prof. Ashwini Patil***

*2021-2022*

***K.T.S.P MANDAL’S***

***KMC COLLEGE KHOPOLI***

***DEPARTMENT OF COMPUTER SCIENCE***

***CERTIFICATE***

*This is to certify that* ***Neha Narendra Ghonge***  *has successfully completed the project on the topic of*

*“****zoo animal classification****” in Sem-II.*

*During the academic year 2021-2022 as per the guidelines issued by* ***University of Mumbai****.*

***Teacher’s HOD’s Examiner’s***

***Signature Signature Signature***

***Date: Date:***

***ACKNOWLEDGMENT***

*In the accomplishment of this project successfully, many people have best owned upon me their blessings and the heart pledged support, this time I am utilizing to thank all the people who have been concerned with this project.*

*Primarily, I would thank god for being able to complete this project with success. Then I would like to thank my principal* ***Prof****,****Dr.Pratap Patil*** *and my project teacher* ***Prof. Ashwini Patil*** *whose valuable guidance has been the ones that helped me patch this project and make it full proof success. Her suggestions and her instructions have served as the major contributor towards the completion of the project. I am also thankful to my head of department* ***Prof. Dhanashree Pawar*** *who encourage me and gave me moral support during my project.*

*Technologies Used*

***Software requirement:***

*Software requirements for this system are as listed follows:*

* Frontend : Python
* Software : Jupyter Notebook
* Operating System : Windows

***Hardware requirement:***

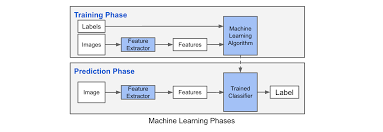
*Minimum Hardware requirements for these system are listed below:*

* C.P.U:- RMD Ryzen.
* R.A.M:- 8 Giga Bytes.
* Hard Disk:- 40 Giga Bytes.
* Type Of System : Single User

Description Of Project

Why Animal classification ?

Animal classification is important because it helps us identify and name all animals on Earth. As machine learning goes wild , In order to study the ecology of animals in natural habitats, such behaviour classification models need to be transferred to wild individuals. However, at present, the development of those models usually requires direct observation of the target animals. The goal of this study was to infer the behaviour of wild, free-roaming animals from acceleration data by training behaviour classification models on captive individuals, without the necessity to observe their wild conspecifics. We further sought to develop methods to validate the credibility of the resulting behaviour extrapolations.

Labeled Classification:

About Dataset

This dataset consists of 101 animals from a zoo.  
There are 16 variables with various traits to describe the animals.  
The 7 Class Types are: Mammal, Bird, Reptile, Fish, Amphibian, Bug and Invertebrate

The purpose for this dataset is to be able to predict the classification of the animals, based upon the variables.  
It is the perfect dataset for those who are new to learning Machine Learning.

zoo.csv

Attribute Information: (name of attribute and type of value domain)

1. animal\_name: Unique for each instance
2. hair Boolean
3. feathers Boolean
4. eggs Boolean
5. milk Boolean
6. airborne Boolean
7. aquatic Boolean
8. predator Boolean
9. toothed Boolean
10. backbone Boolean
11. breathes Boolean
12. venomous Boolean
13. fins Boolean
14. legs Numeric (set of values: {0,2,4,5,6,8})
15. tail Boolean
16. domestic Boolean
17. catsize Boolean
18. class\_type Numeric (integer values in range [1,7])

class.csv

This csv describes the dataset

1. Class\_Number Numeric (integer values in range [1,7])
2. Number*Of*Animal*Species*In\_Class Numeric
3. Class\_Type character -- The actual word description of the class
4. Animal\_Names character -- The list of the animals that fall in the category of the class

Zoo Animal Classification:

Classification is used to categorize different objects. It is a supervised problem in machine learning where we have a labeled dataset.This dataset consists of 101 animals from a zoo.  
There are 16 variables with various traits to describe the animals.  
The 7 Class Types are: Mammal, Bird, Reptile, Fish, Amphibian, Bug and Invertebrate.First, we will import the data from zoo.data file using read\_csv() function. And then will provide attribute names using names parameter of the read\_csv() function. The imported data is transferred to dataframe df. A dataframe is a two-dimensional data structure which stores the data in tabular form i.e. rows and columns. And now we will have a look at the data. head() is used to display the first five rows of the dataframe by default.

Decision Tree classifier:-

Decision tree classifiers provide a readable classification model that is potentially accurate in many different application contexts, including energy-based applications. The decision tree classifier creates the classification model by building a decision tree.

Linear Discriminant Analysis:-

Linear Discriminant Analysis as its name suggests is a linear model for classification and dimensionality reduction. Most commonly used for feature extraction in pattern classification problems.

*Coding And Output*

**IN1:-**

import numpy as np

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

from sklearn.preprocessing import LabelEncoder

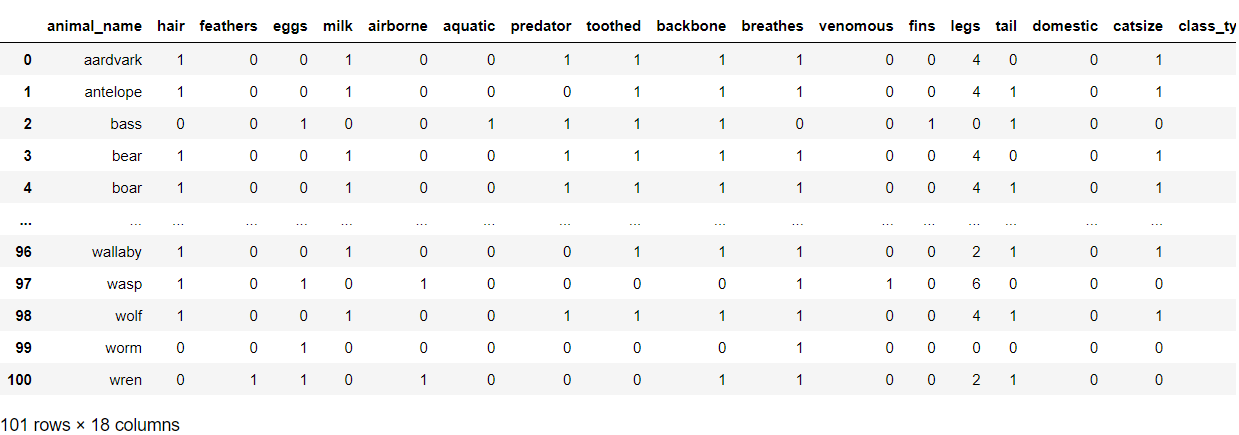
from sklearn.model\_selection import train\_test\_split

**IN2:-**

data = pd.read\_csv('C:\\All csv files\\archive\\zoo.csv')

**IN3:-**

data

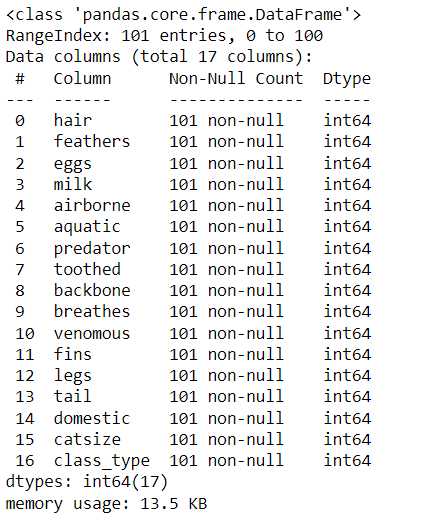
**OUT3:-**

**IN4:-**

data.drop('animal\_name', axis=1, inplace=True)

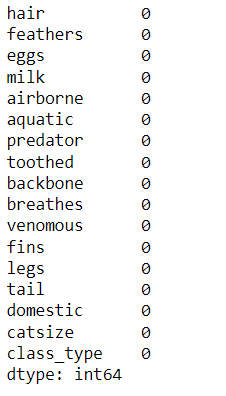
**IN5:-**

data.info()

**OUT5:-**

**IN6:-**

data.isna().sum()

**OUT6:-**

**IN7:-**

y = data['class\_type']

X = data.drop('class\_type', axis=1)

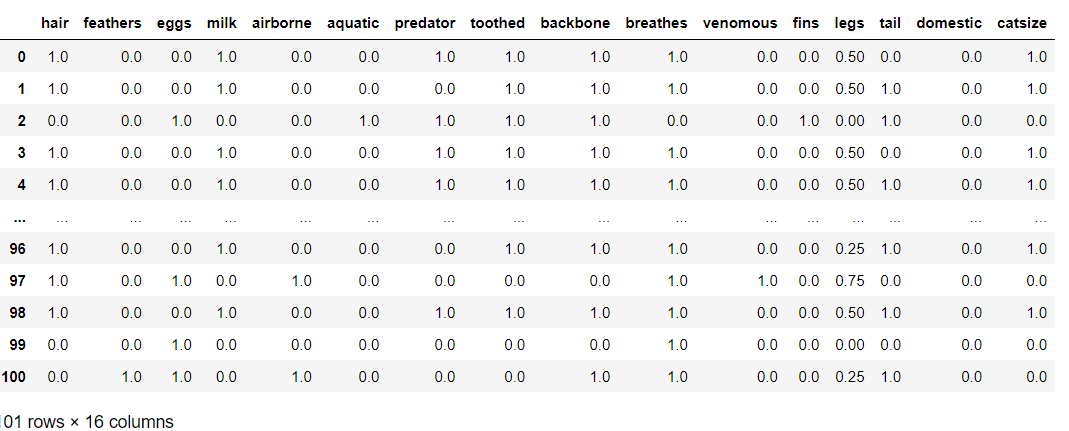
**IN8:-**

scaler = MinMaxScaler()

X = pd.DataFrame(scaler.fit\_transform(X), columns=X.columns)

**IN9:-**

X

**OUT9:-**

**IN10:-**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, train\_size=0.7)

**IN11:-**

from sklearn.linear\_model import LogisticRegression

log\_model = LogisticRegression()

log\_model.fit(X\_train, y\_train)

**OUT11:-**

LogisticRegression()

**IN12:-**

from sklearn.svm import SVC

svm\_model = SVC(C=10.0)

svm\_model.fit(X\_train, y\_train)

**OUT12:-**

SVC(C=10.0)

**IN13:-**

from sklearn.neural\_network import MLPClassifier

nn\_model = MLPClassifier(hidden\_layer\_sizes=(64, 64))

nn\_model.fit(X\_train, y\_train)

**OUT13:-**

MLPClassifier(hidden\_layer\_sizes=(64, 64))

**IN14:-**

log\_acc = log\_model.score(X\_test, y\_test)

svm\_acc = svm\_model.score(X\_test, y\_test)

nn\_acc = nn\_model.score(X\_test, y\_test)

print("Accuracy Results\n" + "\*"\*16)

print(" Logistic Model:", log\_acc)

print(" SVM Model:", svm\_acc)

print("Neural Network Model:", nn\_acc)

**OUT14:-**

Accuracy Results

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Logistic Model: 0.9032258064516129

SVM Model: 0.967741935483871

Neural Network Model: 1.0

**IN15:-**

from mpl\_toolkits.mplot3d import Axes3D

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt # plotting

import numpy as np # linear algebra

import os # accessing directory structure

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

**IN16:-**

nRowsRead = 1000 # specify 'None' if want to read whole file

df1 = pd.read\_csv('C:\\All csv files\\archive\\class.csv', delimiter=',', nrows = nRowsRead)

df1.dataframeName = 'class.csv'

nRow, nCol = df1.shape

print(f'There are {nRow} rows and {nCol} columns')

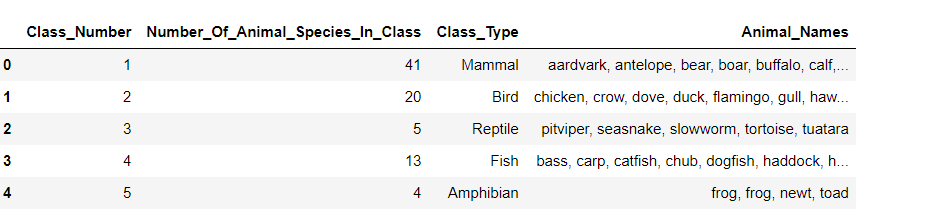
**OUT16:-**

There are 7 rows and 4 columns

**IN17:-**

df1.head(5)

**OUT17:-**



**IN 18:-**

zoo = pd.read\_csv('C:\\All csv files\\archive\\zoo.csv')

classes = pd.read\_csv('C:\\All csv files\\archive\\class.csv')

**IN19:-**

print(zoo.shape)

**OUT19:-**

(101, 18)

**IN20:-**

print(zoo.columns)

**OUT20:-**

Index(['animal\_name', 'hair', 'feathers', 'eggs', 'milk', 'airborne',

'aquatic', 'predator', 'toothed', 'backbone', 'breathes', 'venomous',

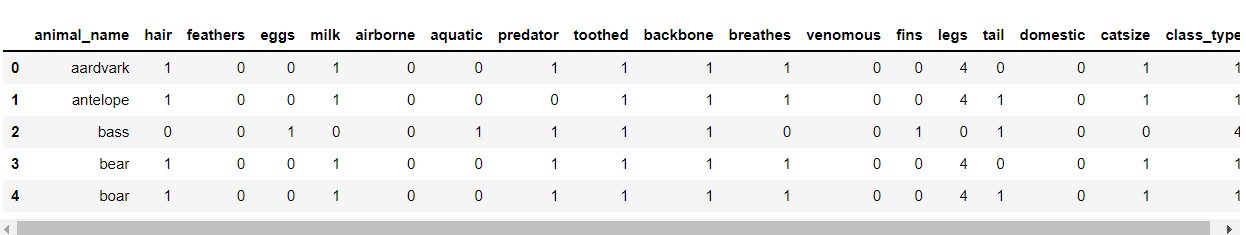
'fins', 'legs', 'tail', 'domestic', 'catsize', 'class\_type'],

dtype='object')

**IN21:-**

zoo.head()

**OUT21:-**



**IN22:-**

# Perform an assessment of how much missing data there is in each column of the dataset

null\_ser = zoo.isnull().sum()

null\_ser\_perc = (zoo.isnull().sum()/len(zoo))\*100

null\_dict = {'num of missing':null\_ser.values,'perc of missing':null\_ser\_perc}

zoo\_null\_df = pd.DataFrame(data=null\_dict,index=null\_ser.index)

zoo\_null\_df.sort\_values(by='num of missing',ascending=False,inplace=True)

**IN23:-**

# Dataframe with Number of missings and % of missings for each columns (represented as index)

zoo\_null\_df

**OUT23:-**

**IN24:-**

import seaborn as sns

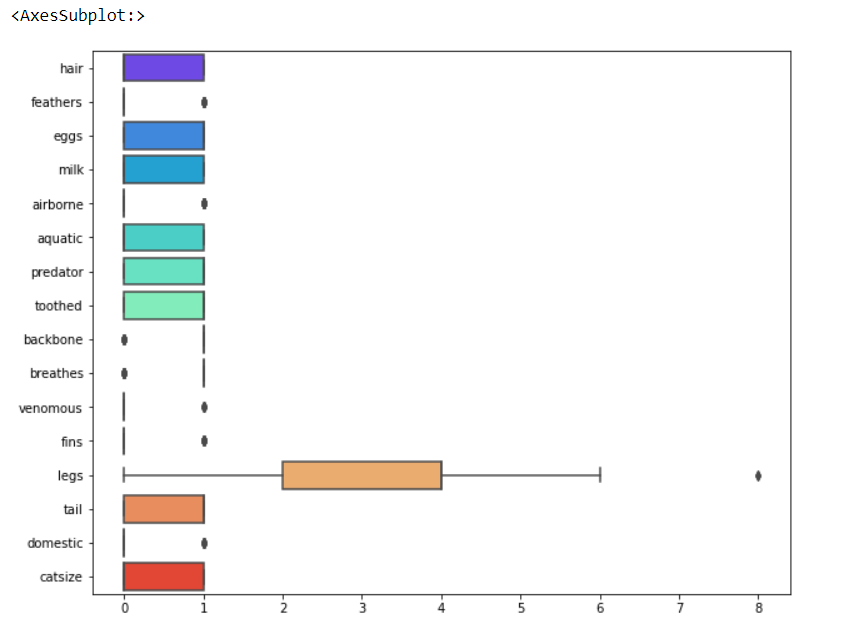
**IN25:-**

fig\_dims = (10, 8)

fig, ax = plt.subplots(figsize=fig\_dims)

sns.boxplot(ax=ax, data=zoo.loc[:,zoo.columns != 'class\_type'], orient='h', palette='rainbow')

**OUT25:-**



**IN26:-**

zoo['animal\_name'].unique()

**OUT26:-**

array(['aardvark', 'antelope', 'bass', 'bear', 'boar', 'buffalo', 'calf',

'carp', 'catfish', 'cavy', 'cheetah', 'chicken', 'chub', 'clam',

'crab', 'crayfish', 'crow', 'deer', 'dogfish', 'dolphin', 'dove',

'duck', 'elephant', 'flamingo', 'flea', 'frog', 'fruitbat',

'giraffe', 'girl', 'gnat', 'goat', 'gorilla', 'gull', 'haddock',

'hamster', 'hare', 'hawk', 'herring', 'honeybee', 'housefly',

'kiwi', 'ladybird', 'lark', 'leopard', 'lion', 'lobster', 'lynx',

'mink', 'mole', 'mongoose', 'moth', 'newt', 'octopus', 'opossum',

'oryx', 'ostrich', 'parakeet', 'penguin', 'pheasant', 'pike',

'piranha', 'pitviper', 'platypus', 'polecat', 'pony', 'porpoise',

'puma', 'pussycat', 'raccoon', 'reindeer', 'rhea', 'scorpion',

'seahorse', 'seal', 'sealion', 'seasnake', 'seawasp', 'skimmer',

'skua', 'slowworm', 'slug', 'sole', 'sparrow', 'squirrel',

'starfish', 'stingray', 'swan', 'termite', 'toad', 'tortoise',

'tuatara', 'tuna', 'vampire', 'vole', 'vulture', 'wallaby', 'wasp',

'wolf', 'worm', 'wren'], dtype=object)

**IN27:-**

# target value

target\_col = 'class\_type'

zoo[target\_col].unique()

**OUT27:-**

array([1, 4, 2, 7, 6, 5, 3], dtype=int64)

**IN28:-**

zoo[target\_col].value\_counts()

**OUT28:-**

1 41

2 20

4 13

7 10

6 8

3 5

5 4

Name: class\_type, dtype: int64

**IN29:-**

y = zoo[target\_col]

X = zoo.drop(target\_col, axis = 1)

**IN30:-**

# split data

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y)

**IN31:-**

#paramter

pars = list(range(3,12,2))

print(pars)

**OUT31:-**

[3, 5, 7, 9, 11]

**IN32:-**

# import some libaray

from sklearn.neighbors import KNeighborsClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, accuracy\_score, f1\_score

for dirname, \_, filenames in os.walk('C:\\All csv files\\archive'):

for filename in filenames:

print(os.path.join(dirname, filename))

**OUT32:-**

C:\All csv files\archive\class.csv

C:\All csv files\archive\zoo.csv

**IN33:-**

from sklearn.model\_selection import train\_test\_split

**IN34:-**

#Classifiying data and target

N = zoo.iloc[:,1:17].values # Not considering the name of the animal placing it

A = zoo.iloc[:,17].values # Class number to be assigned (labels)

# Separating into test and training data

N\_train, N\_test, A\_train, A\_test = train\_test\_split(N, A, test\_size=0.4, random\_state=0)

**IN35:-**

k = zoo.columns.values

index = [0,17]

feature\_names = np.delete(k,index)

feature\_names

**OUT35:-**

array(['hair', 'feathers', 'eggs', 'milk', 'airborne', 'aquatic',

'predator', 'toothed', 'backbone', 'breathes', 'venomous', 'fins',

'legs', 'tail', 'domestic', 'catsize'], dtype=object)

**IN36:-**

target\_names = np.array(classes['Class\_Type'])

target\_names

**OUT36:-**

array(['Mammal', 'Bird', 'Reptile', 'Fish', 'Amphibian', 'Bug',

'Invertebrate'], dtype=object)

**IN37:-**

#Standardizing data

from sklearn.preprocessing import StandardScaler

N\_std = StandardScaler().fit\_transform(N\_train)

**IN38:-**

#svm

from sklearn.metrics import accuracy\_score

from sklearn import svm

from sklearn import metrics

model = svm.LinearSVC()

model.fit(N\_train, A\_train)

# make predictions

expected = A\_test

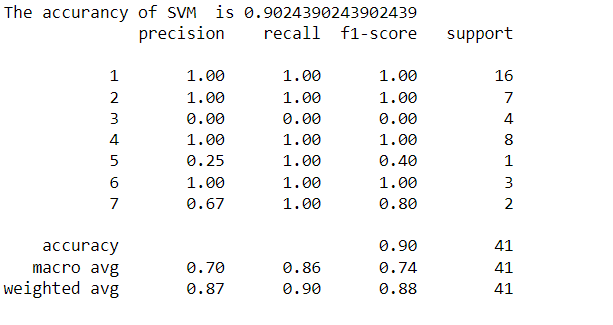
predicted = model.predict(N\_test)

print ('The accurancy of SVM is ' + str(accuracy\_score(expected, predicted)))

# summarize the fit of the model

print(metrics.classification\_report(expected, predicted))

**OUT38:-**



**IN39:-**

#decision tree

from sklearn.metrics import accuracy\_score

from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier()

model.fit(N\_train,A\_train)

# make predictions

expected = A\_test

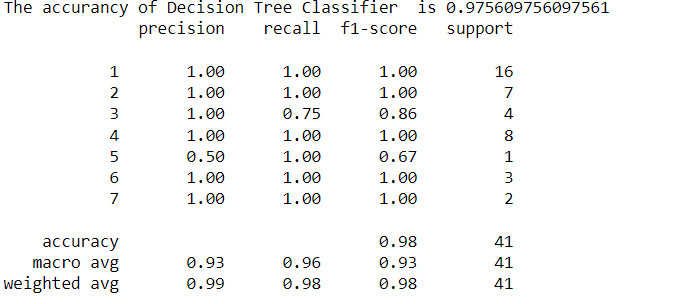
predicted = model.predict(N\_test)

print ('The accurancy of Decision Tree Classifier is ' + str(accuracy\_score(expected, predicted)))

# summarize the fit of the model

print(metrics.classification\_report(expected, predicted))

**OUT39:-**



**IN40:-**

#linear Discriminant Analysis

from sklearn.metrics import accuracy\_score

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

model = LinearDiscriminantAnalysis()

model.fit(N\_train,A\_train)

# make predictions

expected = A\_test

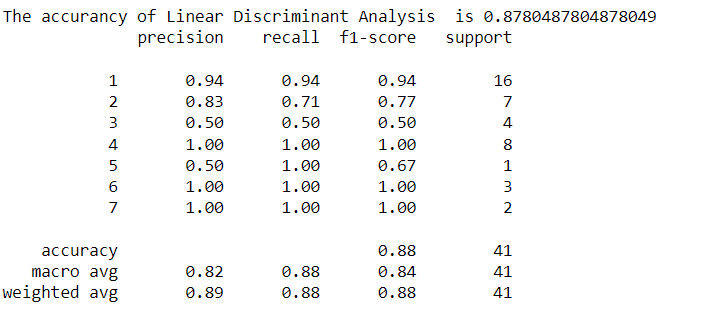
predicted = model.predict(N\_test)

print ('The accurancy of Linear Discriminant Analysis is ' + str(accuracy\_score(expected, predicted)))

# summarize the fit of the model

print(metrics.classification\_report(expected, predicted))

**OUT40:-**



*Conclusion And Future use*

**Conclusion:-**

The conclusion of project is the animals are categorized according to their properties and features ,features such as feathers,egg, milk, leg ,tails etc.

The SVM is also used to predict the accuracy score.

And then the decision tree is used as well as linear discriminant analysis is used to predict the accuracy .

Then according to accuracy score the animals are predicted .

**Future Use:-**

* Artificial Neural Network (ANN) model can be applied.
* Developing a detector that analyses the video images of animals from camera traps in real-time.
* Pre- processing& Animal Discovery.
* Convolutional neural network is the best approach for animal classification. As a future scope one can take low-resolution datasets, the image size with 32 x 32 pixels and plan to train the model for 6 to 8 categories.

*References*

* <https://www.kaggle.com/datasets/uciml/zoo-animal-classification>
* <https://github.com/arshit-b/Zoo-Animal-Classification>
* <https://pranavkumar623.medium.com/zoo-animal-classification-e4cc070a5d44>
* <https://res.infoq.com/articles/book-review-handson-genetic-algorithms/en/resources/Hands_On_Genetic_Algorithms_with_Python-%20chapter%207-01-1589611146347.pdf>