Pune Institute of Computer Technology
Dhankawadi, Pune

DEPARTMENT OF COMPUTER ENGINEERING

Academic Year 2020-21

DMW MINI PROJECT REPORT ON

ZOO ANIMAL DATASET CLASSIFICATION

SUBMITTED BY

Swapnil Markhedkar	41172
Nirvi Vakharia	41175
Apurva Wani	41177

Under the guidance of
Prof. Vijayendra Gaikwad

Contents

List of Figures

1 INTRODUCTION

1.1 Title: Zoo Animal Dataset Classification

1.2 Problem Statement

Consider a labeled dataset belonging to an application domain. Apply suitable data preprocessing steps such as handling of null values, data reduction, discretization. For prediction of class labels of given data instances, build classifier models using different techniques (minimum 3), analyze the confusion matrix and compare these models. Also apply cross validation while preparing the training and testing datasets.

1.3 Objectives

1. Learn how to apply preprocessing steps on a labelled dataset.
2. Learn to build various data classifier models.
3. Learn to split dataset into train and test set and apply cross validation.
4. Learn multiclass prediction and analysis of confusion matrix.

1.4 Outcomes

1. The dataset was analysed using bar charts and correlation matrices.
2. Different models were tested with variations in their parameters.
3. A Graphical User Interface was created over the selected model to make the testing more interactive.

1.5 Software and Hardware Requirements

Any working laptop, internet connection, Kaggle account, Python.

2 DATASET DETAILS

2.1 Data Analysis

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

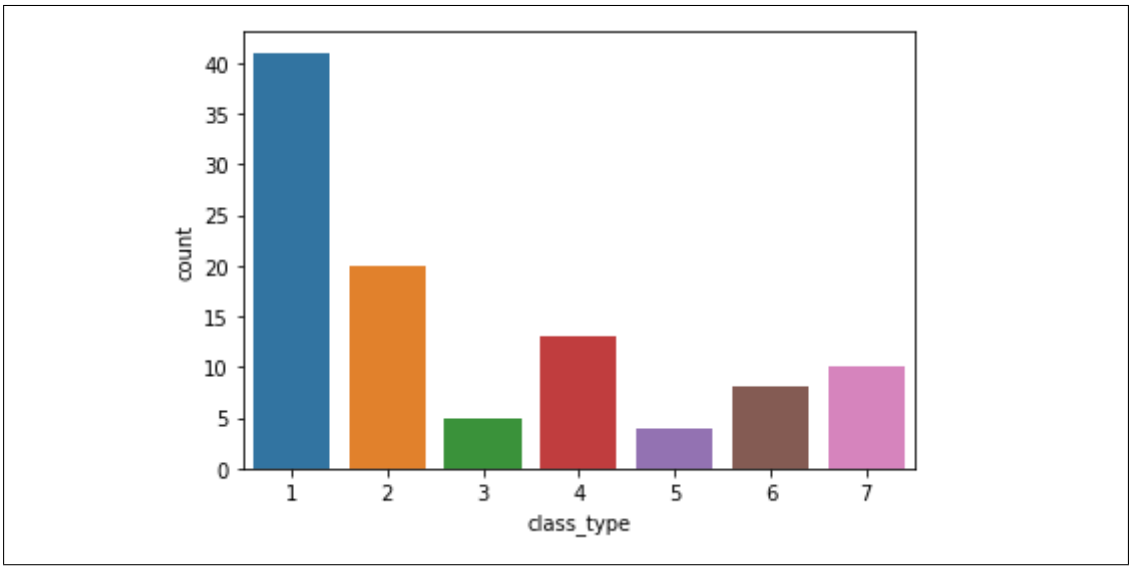

Figure 1: Dataset Distribution

2.2 Data Preparation

As seen in Fig. 1, the class ratio of the first 2 classes is nearly 2:1 (41:20). This can cause a problem of class imbalance while training and selecting a suitable model for testing. Two approaches to make a balanced dataset out of an imbalanced one are under-sampling and over-sampling.

1. **Under-sampling:**
   Under-sampling balances the dataset by reducing the size of the abundant class. This method is used when quantity of data is sufficient. By keeping all samples in the rare class and randomly selecting an equal number of samples in the abundant class, a balanced new dataset can be retrieved for further modelling.

2. **Over-sampling:**
   Over-sampling is used when the quantity of data is insufficient. It tries to balance dataset by increasing the size of rare samples. Rather than getting rid of abundant samples, new rare samples are generated by using e.g. repetition, bootstrapping or SMOTE (Synthetic Minority Over-Sampling Technique).

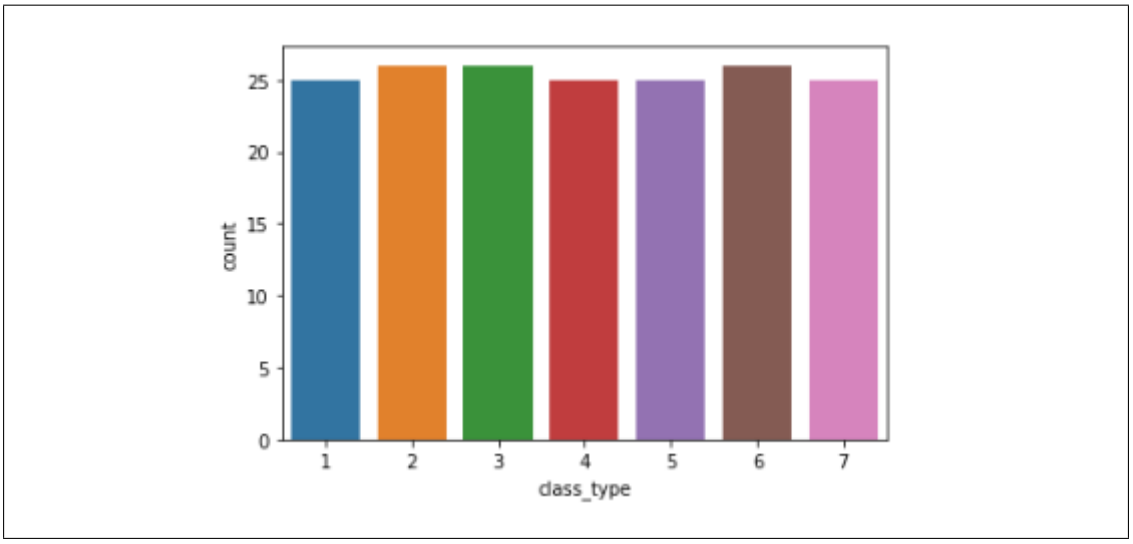

Figure 2: Dataset Distribution after resolving class imbalance problem

Correlation Matrix using Heat Map:

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses.

A heat map is a graphical representation of data in which data values are represented as colors. As visualization is generally easier to understand than reading tabular data, heat maps are typically used to visualize correlation matrices.

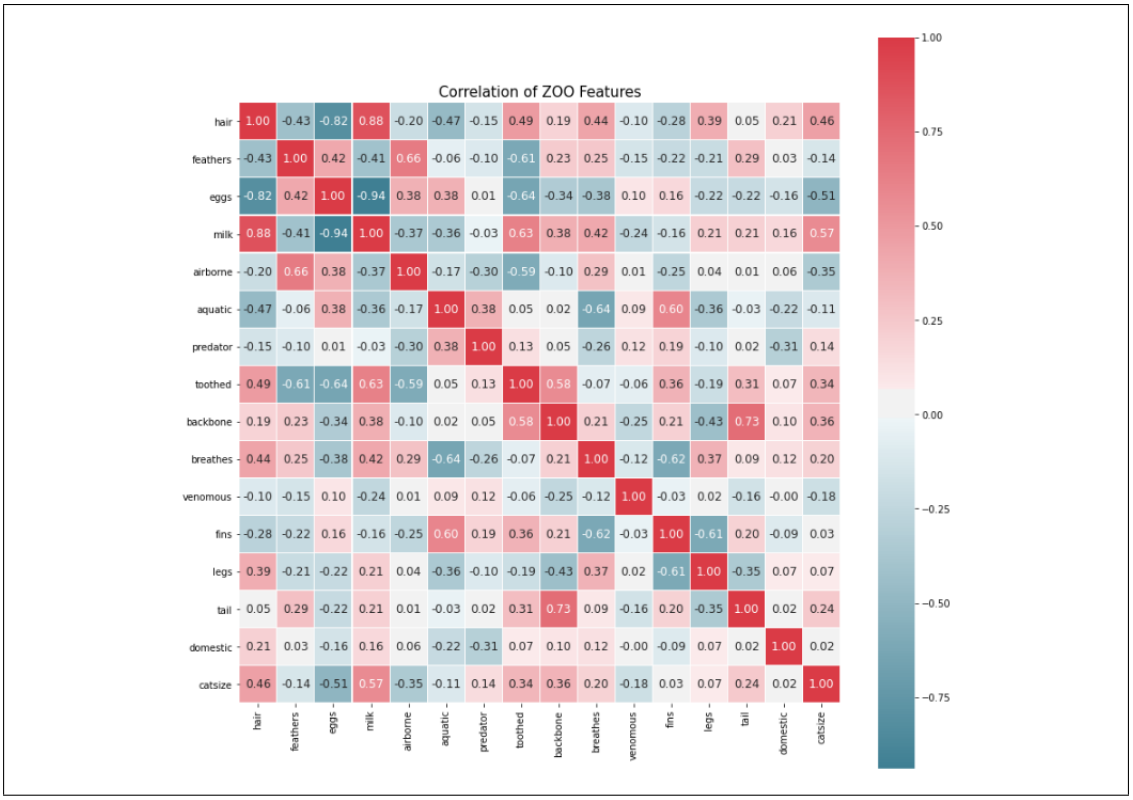

Figure 3: Correlation Matrix using Heat Map

3 THEORY CONCEPTS

3.1 Models Analyzed

Three models were trained and tested for the purpose of classification. They are:

- K-Means Clustering
- Naive Bayes
- K-Nearest Neighbours

Models with multiple parameters were tuned using Grid Search. The performance of these models and results are discussed below.

Classifier	***Accuracy***	***Precision***	***Recall***
K-Means Clustering	0.3333	0.3673	0.3
Naive Bayes	0.8571	0.5555	0.6666
K-Nearest Neighbors	0.9524	0.7999	0.8333
Decision tree	0.9048	0.6667	0.6429

Table 1: Results of the classifiers over original data.

1. **K-Means Clustering:**
   Clustering is a type of unsupervised machine learning which aims to find homogeneous subgroups such that objects in the same group (clusters) are more similar to each other than the others.

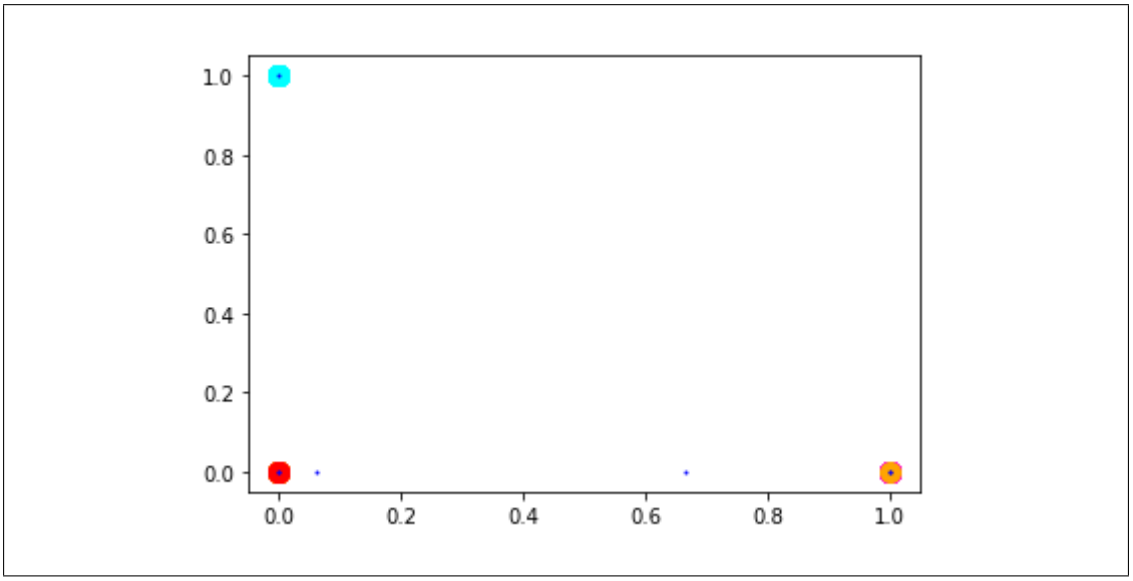

Figure 4: Clusters using K-Means

Here, as we have to classify the animals into 7 classes, we have used the value of K as 7. As observed in Fig. 4 we see that only 3 clusters easily visualized. We hypothesize that this is due to inadequate data related to

individual classes available, thus making their clusters too small, or merging into other clusters.

Furthermore, by using the elbow method, we notice that the suggested number of clusters is 4. However, we know that we need 7 clusters to differentiate between the 7 classes. This could also be due to the class imbalance, as there is not enough data w.r.t classes 3, 5 and 6.

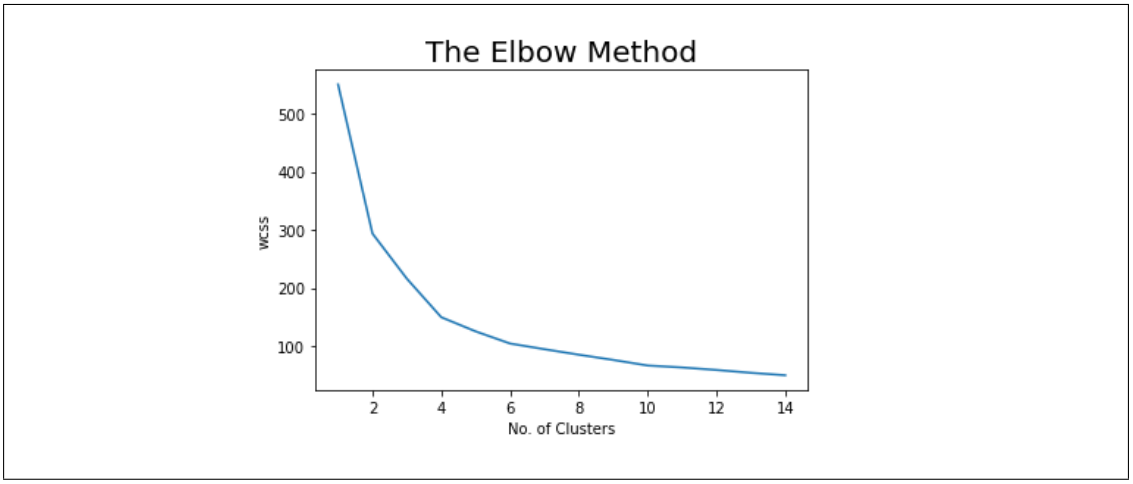

Figure 5: Elbow Method Graph

2. **Naive Bayes:**
   A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem., with an assumption that every pair of attributes are independent of each other. Here the classifier first calculates the probability of each class. Next, it calculates the conditional probability for each attribute for each class and multiplies the same class conditional probability. The product is then multiplied with the prior probability.

   Gaussian Naive Bayes is usually used for continuous discreet data, hence we use Multinomial naive Bayes. Multinomial Naïve Bayes consider a feature vector where a given term represents the number of times it appears or very often i.e. frequency. However, it is very difficult to get the set of independent predictors for developing a model using Naive Bayes.

   For the selected dataset, Naive Bayes performs better than K-means clusters, as it develops a probability score for every attribute and the predictions are based on these scores and the input fed to the model. However, it is possible to improve this result by solving the issue of imbalance of class.

3. **K-Nearest Neighbour:**
   KNN or k-nearest neighbours is the simplest classification algorithm. This classification algorithm does not depend on the structure of the data. Whenever a new example is encountered, its k nearest neighbours from the training data are examined. Distance between two examples can be the euclidean distance between their feature vectors. The majority class among the k nearest neighbours is taken to be the class for the encountered example.

We use grid search with cross validation to tune the parameters. The best estimator found should have leafsize = 2 and k = 1. For our classifier, k = 1 means that the object (vector representation) is simply assigned to the class of that single nearest neighbour.

4. **Decision Tree:**
   Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. Every feature is used to classify the input into the given 7 classes. The decision tree used for this experiment has been visualized below visualized. The parameters for this classifier were selected using Gris Search with cv = 5.

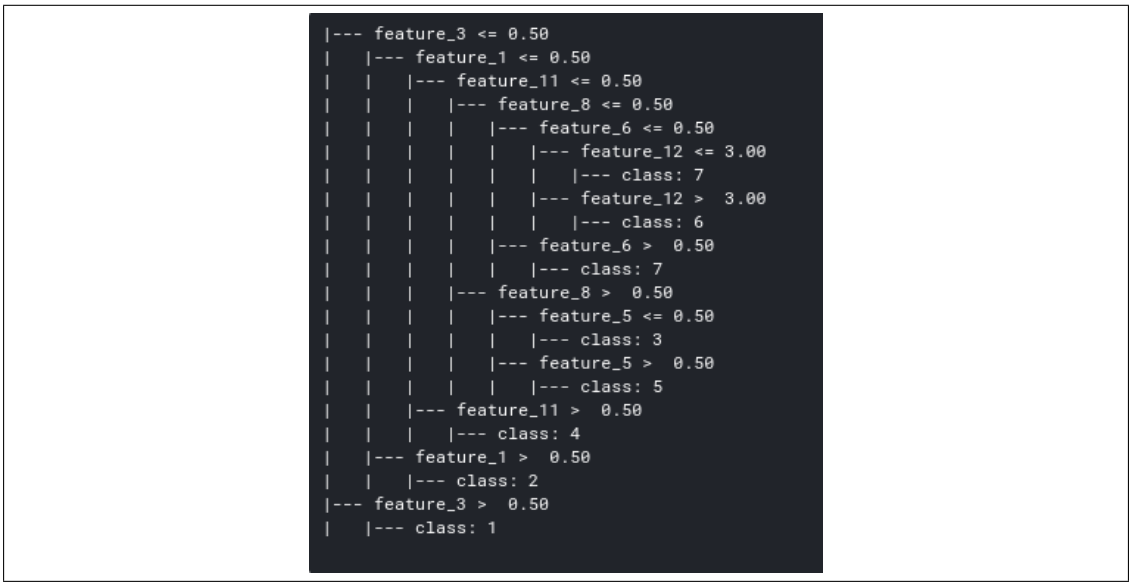

Figure 6: Visual representation of the decision tree before balancing the class distribution

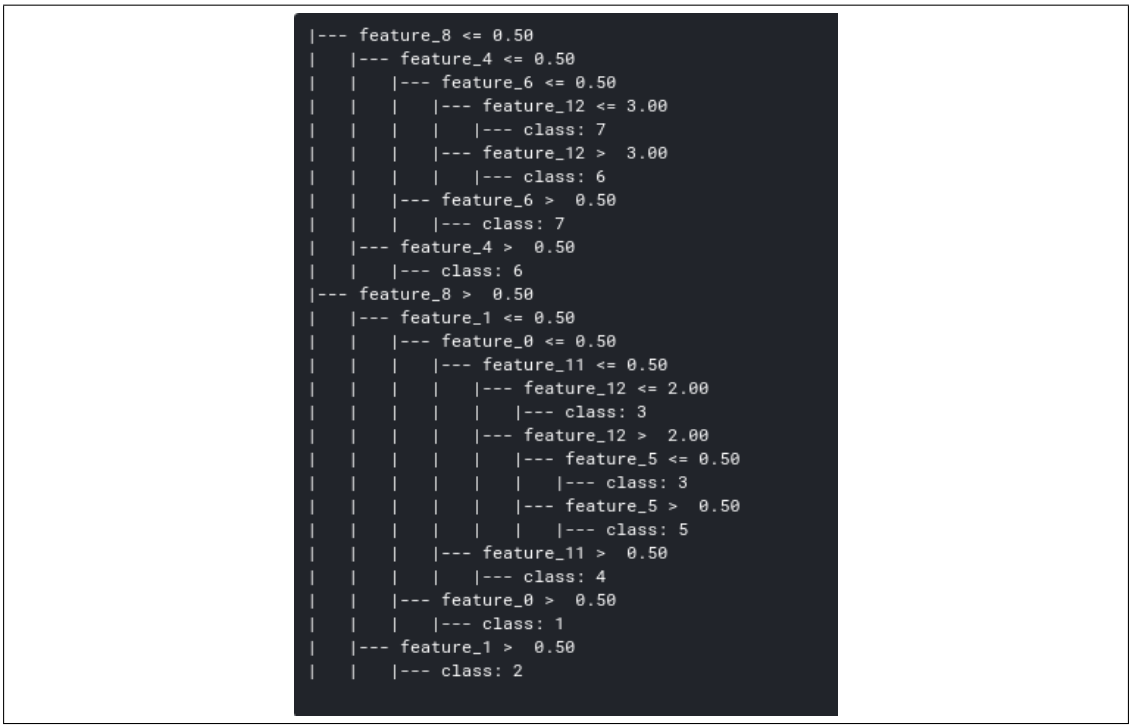

Figure 7: Visual representation of the decision tree used

Further usage of a balanced dataset improved the scores of the models and are tabulated below.

Classifier	***Accuracy***	***Precision***	***Recall***
K-Means Clustering	0.0556	0.025	0.0417
Naive Bayes	0.9722	0.9643	0.9762
K-Nearest Neighbors	1.0000	1.0000	1.0000
Decision tree	0.9722	0.9796	0.9762

Table 2: Results of the classifiers after class balancing.

3.2 Model Selected

As seen in the table above, the K-Nearest Neighbour classifier performs much better after the pre-processing, giving 100% accuracy. This could be because every datapoint is virtually plotted on a vector space during supervised training. Additionally, as there is no 'clustering' or explicit labelling of the datapoints into classes, we overcome the drawbacks similar to that of K-Means Clustering. Here, the model acts like a plotting and retrieval mechanism, where it plots the input, and returns the class of its nearest neighbor.

The K-Means clustering classifier deteriorates after the pre-processing. This could be as the new dataset has copies to balance the class ratios; the repetition of multiple instances causes the model to overfit. Additionally, as the output label is available in the dataset, there is no need to apply an unsupervised learning technique, hence, we do not explore this model further.

For Naive Bayes, the accuracy is close to the first K-NN classifier, however, as the model solely relies on the probabilities of the training dataset, it might not be accurate at all times. This is observed in the test set, when the classifier fails to classify an example with label 5.

We see a larger tree when the class distribution is balanced. This shows that the model is able to understand the data given and make more decision nodes based on features. Further tuning of parameters can help to achieve better results.

4 APPLICATIONS

1. **Classification of new ambiguous animals**

   Every year there are new animals found in remote regions. Recently, there was an untouched underwater cave discovered, with the animals over there having no contact with the outside world for 3000 years. Our model can prove to be very useful in classification of such newly discovered ambiguous animals.
   The Zoo Animal Dataset classification model can be used in schools for learning purposes with an interactive GUI. This will benefit students in understanding seven levels of classification and also compare the classification of animals. This model will not only make learning fun and effective but also help school kids to create new species and classify them according to principles of classification.

   [Ref: Schmidt Ocean Institute. "New species discovered during exploration of abyssal deep sea canyons off Ningaloo." ScienceDaily. ScienceDaily, 12 April 2020.]

2. **Study evolutionary traits**

   Evolution describes changes to the inherited traits of organisms across generations. In our zoo animal dataset classification model we have 16 attributes which describe in-depth details of an organism. Hence if we try to visualize this model using KNN then we could be able to study various inherited traits of organisms and identify animals which have similar characteristics. This may also help us study evolutionary traits of animals.

5 IMPLEMENTATION DETAILS

5.1 GUI Details

Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. It is a thin object-oriented layer on top of Tcl/Tk.

Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

To make the the GUI, Tkinter Widgets Entry, Checkbutton, Button, and Label were used. The four stages to use and display any widget are Create, Configure, Pack, and Bind. The grid() method was used to develop the layout of the GUI by specifying rows and columns within the GUI window.

The GUI was designed such that the user can provide input either in the form of a file, or specify each of the 15 features using the easy-to-use checkboxes. Once the user is done entering the input, the predicted class of animal is shown on clicking the "Predict" button. Other options in the GUI include showing the name of the file being used, as well as the Confusion Matrix for the prediction.

5.2 Integration

Initially, the models built to predict the class of mammal were trained and tested. The next step was to accept user input in the form of a file or text, and then use the trained models to make the prediction.

This was done by exporting the models as a separate .sav file for each model. The models were then loaded as required, and used in conjunction with the predict method on the user input to ultimately determine the class.

The code was broken down into helper functions to further streamline the process, and to handle the different types of user input.

6 RESULTS

6.1 Code Screenshots

```
from tkinter import *
from tkinter import filedialog
root = Tk()

#Custom Input
v1 = IntVar()
v2 = IntVar()

c1 = Checkbutton(root, text = "Has Hair?", variable = v1, \
                 onvalue = 1, offvalue = 0)
c2 = Checkbutton(root, text = "Has Feathers?", variable = v2, \
                 onvalue = 1, offvalue = 0)
c13 = Entry(root)
c13.insert(0, 'Number of Legs')

c1.grid(row=3,column=0)
c2.grid(row=3,column=1)

myButton3 = Button(root, text="Predict", command=predictModel)
myButton3.grid(row=11,column=0, columnspan=2)

# File Input
button_file_open = Button(root, text="Browse File", command=browseFiles)
button_file_open.grid(row=19,column=0)

button_file_open2 = Button(root, text="Show FileName", command=show)
button_file_open2.grid(row=19,column=1)

button_predict = Button(root, text="Show Confusion Matrix", command=showCM)
button_predict.grid(row=21,column=0,columnspan=2)

root.mainloop()
```

Figure 8: Screenshot

6.2 Output Screenshots

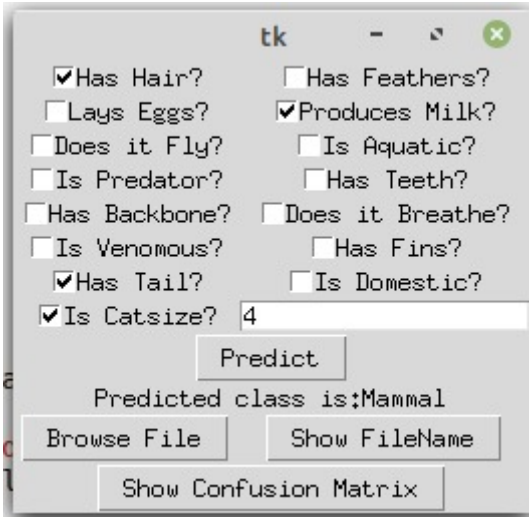

Figure 9: GUI

6.3 Test Cases

Sr. No.	***Input***	***Expected Output***	***K-Means***	***Naive Bayes***	***KNN***	***Decision Tree***
1.	Asian Hornet	Bug	Null Value	**Bug**	**Bug**	**Bug**
2.	Platypus	Mammal	Bird	**Mammal**	**Mammal**	Bird
3.	Wattled Curas-sow	Bird	Mammal	**Bird**	**Bird**	**Bird**
4.	Whale	Mammal	Reptile	Fish	**Mammal**	Fish

Table 3: Test cases

7 CONCLUSION

Successfully implemented a multi-label classification project on Zoo Animal Dataset using various data mining models. The K-Nearest Neighbour (KNN) classifier gave best performance with 100% accuracy. Naive Bayes classifier and Decision Tree model gave 97.22% accuracy.

8 FUTURE SCOPE

The Zoo Animal Dataset Classification model can be visualized using KNN along with an interactive GUI. The GUI would be very useful for school kids to learn and explore animal classification. This model can also prove to be very useful for studying evolutionary traits of organisms and in classification of newly discovered ambiguous species.