Machine Learning

Zoo Animals - Classification
Team-2

195001009 - Aditi Ramprasad

195001016 - Anirudh T E

195001024 - Ashwini Sridar Athreya

195001061 - Mathanggi

195001066 - Mukund Balaji L

III Year, CSE - A

SSN College of Engineering

Problem Statement

Aim:

To classify animals in the given dataset.

Input:

zoo.csv containing 18 variables is used as training data.

Output:

Animals are classified into 7 categories.

Techniques used:

- Random Forest
- Perceptron Model
- Decision Tree

Literature Study

Motivation:

- From the given dataset, we can increase our knowledge base about animals. We learn about living and feeding habitats of particular groups of animals. From a data science point of view, we may also be able to infer similarities between species and genera of animal families.
- From a macroscopic point of view, with the help of these pre-categorized training datasets, classification in machine learning programs leverage a wide range of algorithms to classify future datasets into respective and relevant categories.

Literature Study

Real time Applications of Classification Models:

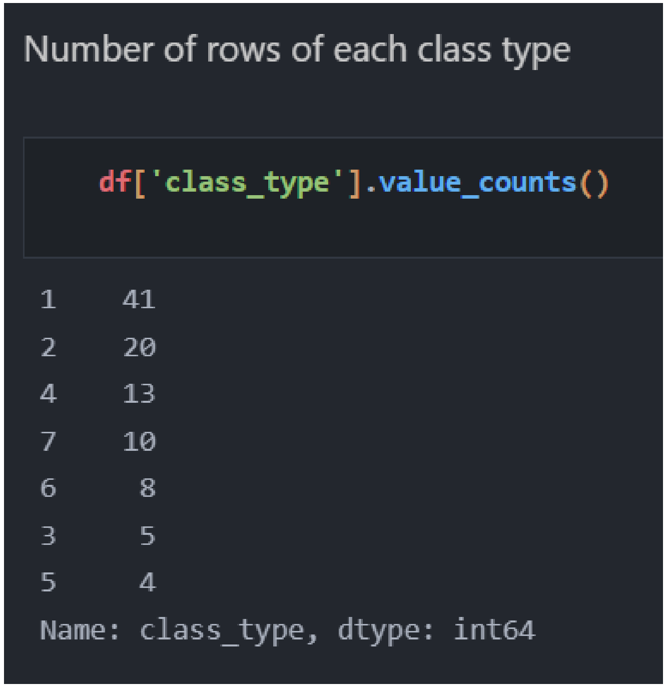
- Image classification
- Web text classification
- Spam filtering
- Ad click-through rate prediction

Reference links:

- https://www.kaggle.com/datasets/uciml/zoo-animal-classification
- https://github.com/arshit-b/Zoo-Animal-Classification/blob/master/zoo.csv

Data Analysis

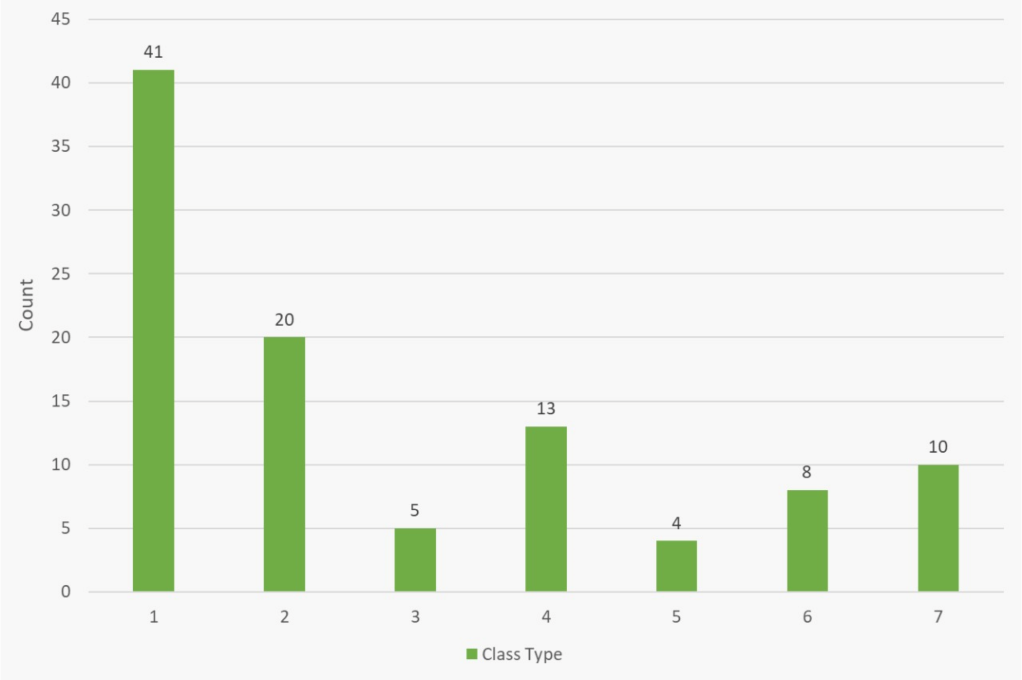
The class types and their counts are as follows:



Number of rows of each class type

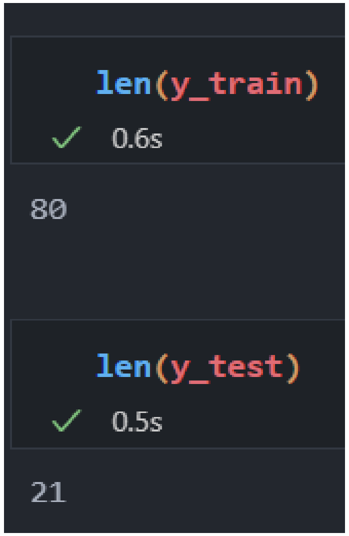
```
df['class_type'].value_counts()
```

```
1    41
2    20
4    13
7    10
6     8
3     5
5     4
Name: class_type, dtype: int64
```

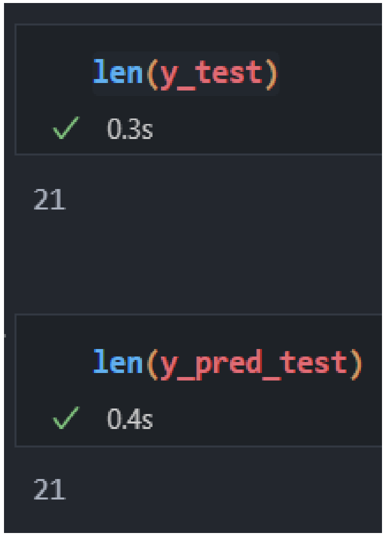


Dataset Split

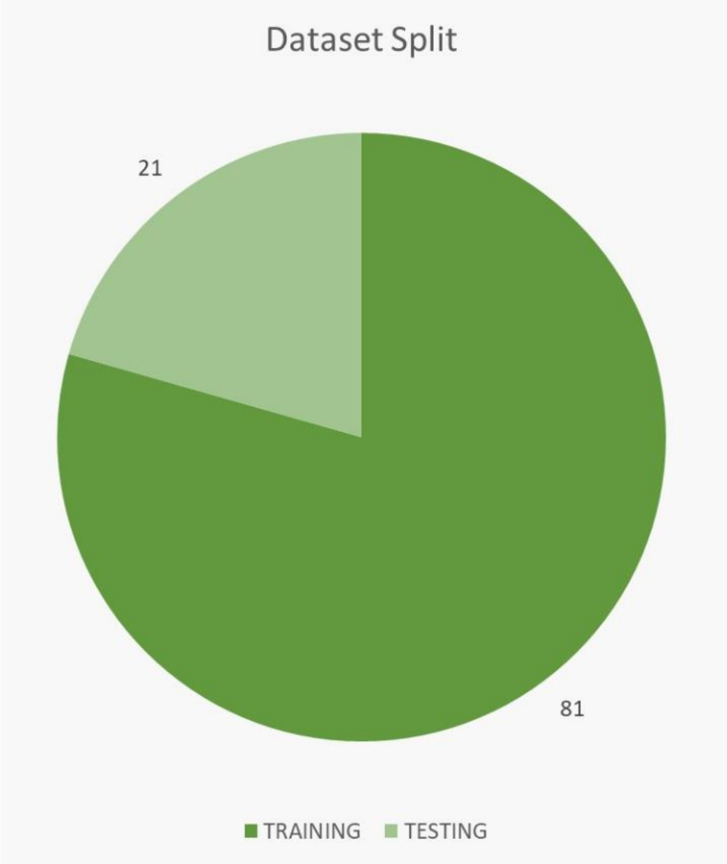
'random_state=42' ensures that the dataset is split in the same manner every time it is reproduced.



```
len(y_train)
✓ 0.6s

80

len(y_test)
✓ 0.5s

21
```

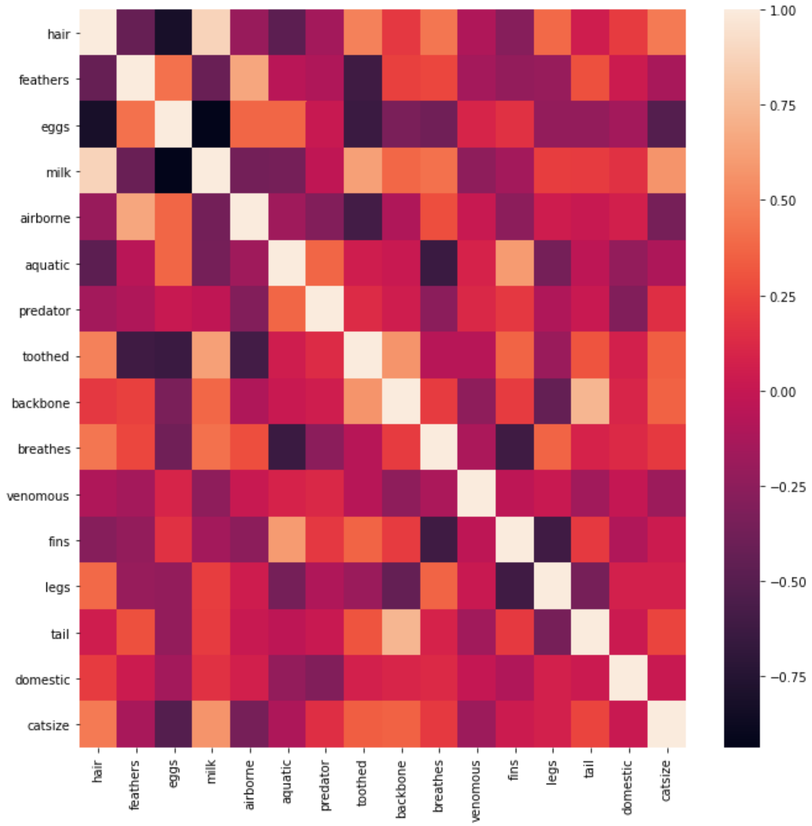


```
len(y_test)
✓ 0.3s

21

len(y_pred_test)
✓ 0.4s

21
```



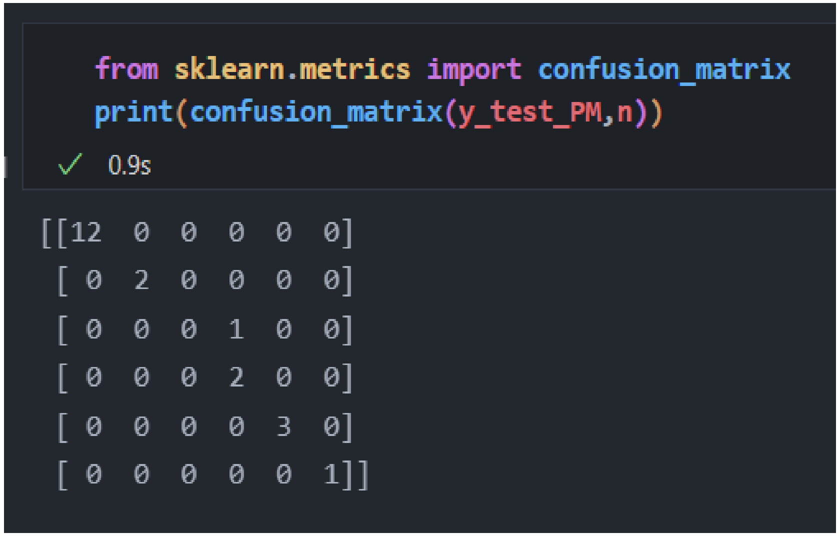
Correlated Feature Removal

- From the performed exploratory data analysis, a heat map was generated.
- Features with correlation factor greater than 0.8 were removed.
- According to the analysis, {'eggs', 'milk’} were dropped from dataset before further analysis.



Confusion Matrices

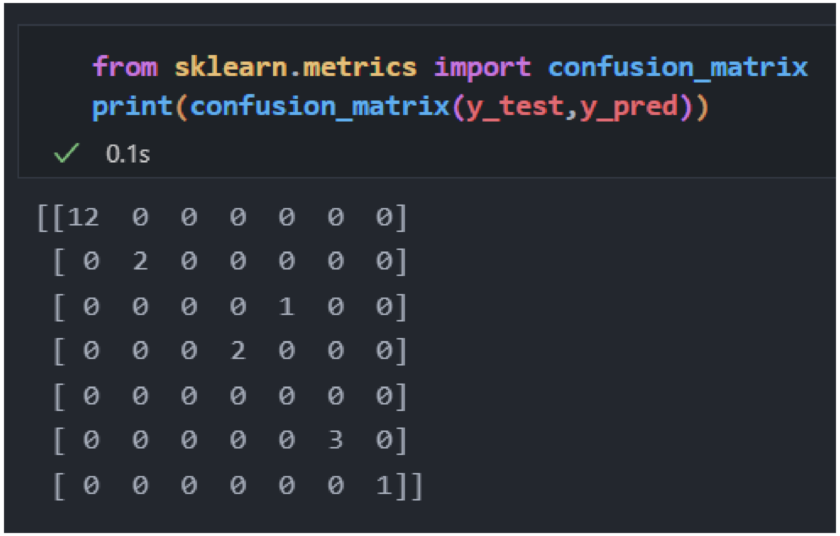
Confusion matrices for Neural Network and Decision Tree models are given below:



```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test_PM,n))
```

✓ 0.9s

```
[[12  0  0  0  0  0]
 [ 0  2  0  0  0  0]
 [ 0  0  0  1  0  0]
 [ 0  0  0  2  0  0]
 [ 0  0  0  0  3  0]
 [ 0  0  0  0  0  1]]
```

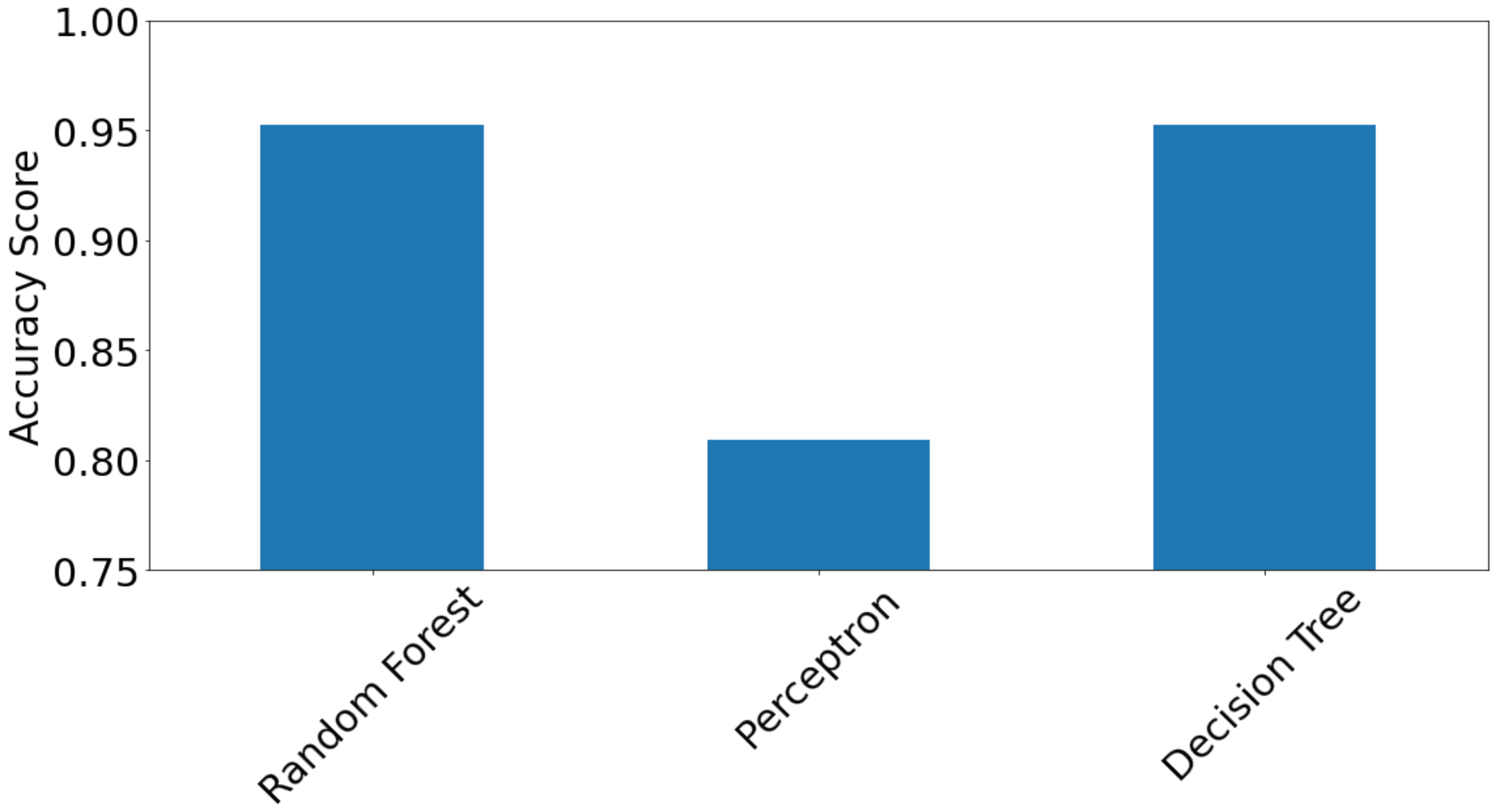


```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred))
```

✓ 0.1s

```
[[12  0  0  0  0  0  0]
 [ 0  2  0  0  0  0  0]
 [ 0  0  0  0  1  0  0]
 [ 0  0  0  2  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  3  0]
 [ 0  0  0  0  0  0  1]]
```

Results


1.00
0.95
0.90
0.85
0.80
0.75
Accuracy Score
Random Forest
Perceptron
Decision Tree

Inference & Future work

- **Inference**
  - It is observed that Random Forest and Decision Tree have a high accuracy compared to MLP. This can be due to the lack of enough training samples to train the Neural Network while Random Forest and Decision Tree do not need as many training samples to learn patterns in the dataset.
- **Future work**
  - Since neural network accuracy is lower because training data is lesser, we can use a layered neural network model to train the model better even with the availability of lesser training data.