Zoo Animals Classification

Jackson Cashman, Holly Lee, Ruby Nguyen, Samantha Reyes, Victoria Tran
California State University Long Beach

ABSTRACT

In this research study, we present various machine learning methods for classifying zoo animals based on their physical features. We utilize the Decision Trees and Support Vector Machine (SVM) methods to predict the animal's class type that belongs to one of the seven categories such as Mammal, Bird, Reptile, Fish, Amphibian, Bug, and Invertebrate. The dataset used in this study consist of 101 animals from a zoo including 16 various attributes of the animals.

The result of the SVM model achieved an accuracy score of about 92 percent, however, by applying the one vs all multi-class classification method yielded an accuracy score of 100 percent. Meanwhile, The performance on Decision Tree achieved an accuracy score of about 94 percent.

1. INTRODUCTION

Living organisms are often grouped together and classified based on their unique attributes. Zoo animals, in particular, are an important part of animal conservation and education, providing opportunities for the public to learn about the diversity of animal species. Managing and monitoring a population of zoo animals comes with many challenges as it requires proper knowledge and training. With the proper knowledge, zoo animals can be treated and cared for differently based on their behavior and genetic differences, providing sustainable lifestyle and healthy habits. Traditionally, animal classification has been done by experts to study various species of animals over a great amount of time which are expensive and time-consuming. With the recent growth of machine learning, introducing automated systems allows for efficiency by producing more accurate, less time effort, and objective results in classifying various traits of animals and species.

In this study, we studied different machine learning approaches for the multi-classification of zoo animals based on their unique characteristics. As such, we used 16 different physical traits to train and test our Decision Tree and SVM models, this include identifying a variety of categories of animals such as mammal, bird, reptiles, fish, amphibian, bug, and invertebrate. Our motivation is to develop a robust and accurate prediction rate for animal classification that can be applied in the real-world zoo settings to improve the experience and conservation of zoo animals. Moreover, we can apply the methods for future experiments to classify new and unidentified species of animals. Further sections within this paper will discuss related works, proposed methods, and experience results of achieving the accuracy scores for each specific model and comparing methods with improvements on various dataset.

2. RELATED WORKS

2.1. Support Vector Machines

SVM is a supervised learning algorithm that aims to find an optimal hyperplane from known data, in order to accurately categorize future samples. To find the optimal hyperplane the uses support vectors, or the data points closest to the edge of the hyperplane. These support vectors are the hardest points to distinguish and are heavily relied on for fitting the hyperplane.

Despite its ease of implementation, especially on smaller datasets, SVM is primarily used for binary classification. In this experiment we approach using SVM as a multi class classification method. Typically, data would be modified in order to fit the binary method, however, the data provided is already classified in a binary way allowing us to use SVM without reducing the classifiers. SVM for multi-class classification is generally either in a one-against-all or a one-against-one as observed in a study aimed at categorizing student performance using SVM[3]. Here we apply the one-against-all method[4] where each class of animal is tested.

2.2. Decision Trees

We would be using decision trees as one of the machine learning methods because it is utilized for classification tasks. A decision tree is a non-parametric supervised learning algorithm. [5] It has a hierarchy tree structure with roots, internal nodes, and leaf nodes. The root and each internal node represent a feature and each leaf node represents an outcome. A decision tree starts with the root node at the top, which represents the starting point for the classification process. Branches from the root node represent the possible choices or decisions that can be made. The set is divided into subsets based on the parent node and is sent to the next remaining internal node. The number of internal nodes is based on the maximum depth of the tree.

The maximum depth is a crucial parameter to decision trees as it determines the maximum number of splits that can be made before reaching a leaf node. Overfitting may occur if the maximum depth is set too high [6]. The decision tree may capture noise in the training data or create splits based on irrelevant features. As a result, an overfitted model can perform well on the training sets but inadequately on testing data sets since the model only focuses on the specific training dataset. If the maximum depth is set too low, the model might be underfitted because it does not capture the complexity of the data.

The criterion is also an important parameter for decision trees. We would be using Entropy and Gini index for the research. Entropy measures the impurity or uncertainty of the data. The value typically ranges from 0 to 1 but depending on the number of classes, it can be greater than 1. 0 indicate the perfect purity or 100% certainty, where 1 or higher indicate it reaches the maximum of impurity and uncertainty.
Entropy is defined as:

$$Entropy(S) = -\sum_{c \in C} P(c) log_2 P(c)$$

- S is the data set that entropy is calculated
- c represents the classes in set 'S'
- P(c) is the proportion of data points that belong to class c to the number of total data points in set 'S'

To determine the best feature for splitting and create the most effective decision tree, the algorithm uses information gain to measures the change in entropy based on a particular attribute. The attribute with the highest information gain will be the most accurate classification, making it the best choice for the split. Information gain is defined as:

$$Information\ Gain(S) = Entropy\left(\frac{p}{p+n}\right) - \sum_{k=1}^{d} \frac{p_k + n_k}{p + n} Entropy\left(\frac{p_k}{p_k + n_k}\right)$$

- p is the number of positive examples
- n is the number of negative examples
- p_k and n_k are the number of positive and negative examples in each subset respectively

The second criterion, Gini index, is similar to Entropy but it measures the likelihood of classifying a randomly selected data point from a dataset incorrectly. If all training data belongs to the same class, the Gini index will be 0 as it represents the dataset is completely pure. Gini is defined as:

$$Gini(S) = 1 - \Sigma_i (p_i)^2$$

3. PROPOSED METHODS

3.1. Data Preprocessing

We used two Machine Learning models to perform classifications on our zoo dataset and compare their accuracy.

Data preprocessing is first performed to define the input features and target values for both SVM and Decision Tree models. A total of 16 values including hair, feathers, eggs, milk, airborne, aquatic, predator, toothed, backbone, breathes, venomous, fins, set of legs numbers, tail, domestic, cat size were used as predictors. We excluded the animal's name to remove irrelevant value from our testing set. Most of the predictors are assigned as 0 and 1 to identify whether the animal possesses a certain feature, in which 1 indicates yes and 0 as no. The number of legs are measured in a set of values of {0, 2, 4, 6, 8}. The target value is identified as the class type which maps it into 7 different values, we label the range from 1 through 7. For instance, the target value of 1 falls under the category of Mammal class type.

Since the features vary on different measurement scales, feature scaling is applied to standardize all the input data in order to have a similar range of values. We applied the StandardScaler() function to transform the input feature so they have a mean of 0 and standard deviation of 1. The use of StandardScaler() enables the dataset to reduce the impact of outliers and improve the performance of the model by ensuring that there is no bias that prevents certain features from affecting the overall performance of the dataset. Fit_transform method is used during the feature scaling step to ensure the scaling is consistent in both training and testing set.

The dataset is split into two sets with the ratio of 7.5 for training and 2.5 for testing. In this step, we use the splitting method to train the model on the training set and evaluate the performance on the testing set. This estimates how well the model will perform on new testing data.

In order to identify which set of animals were selected in the testing dataset for SVM model, we used the inverse_transform and assigned it to a new X test value to convert the values back to before they were standardized. We will use these values later to match with the predicted value in the Confusion Matrix to identify correct and incorrect labels.

3.2. Building Model

The SVM model is generated on the training set with the random state parameter of 0, meaning a fixed sequence of the random numbers is used each time the algorithm runs to obtain the same results and using the kernel value of RBF to measure the similarity between the class data. Additionally, using sklearn allows us to utilize the decision function shape method for the one vs all multi-class classification. In this experiment we also tried using the sklearn multi-class library enabling a OneVsRest wrapper. Both are applied in separate cases allowing us to compare results. Consequently, the fit method is applied to the SVM classifier with input features and the target value as arguments to find the decision boundary that separates the classes by label. After using the fit method, we used the SVM classifier to make predictions on the testing dataset. The predict function takes a set of predictors as its arguments and returns a vector of predicted values. In particular, for each set of input features, the SVM classifier predicts the class label that the animal falls under.

For the Decision Tree, we've exploited several features including the criterion, max depth, and randomness when building a model on our zoo dataset. We wanted to observe how accurate a Decision Tree model could get with prediction upon testing with certain parameters. The way the tests were going to be carried out was that we were interested in whether switching the value of some of the parameters of DecisionTreeClassifier() would produce any changes in accuracy upon prediction, and if it did, how. As a result of considerations, we decided that the criterion, maximum depth, and random state were our variables of interest. We would generate our Decision Tree model allowing for different values of maximum depth of the tree. Specifically, we set the range of values from 1 to 10 for this particular variable and had it processed as a for loop. Then, at each value of maximum depth, we wanted to further compare the performance of the two criterions that were often used to calculate impurity in the model: Entropy and Gini. Additionally, we set the random state to 0 in both Decision Tree models deploying different criterions while at the same maximum depth.

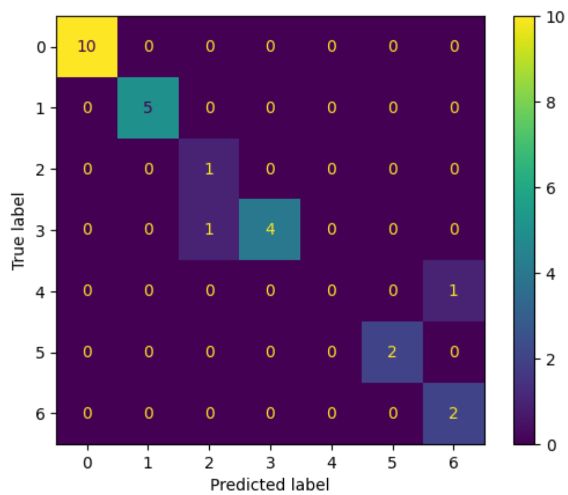
3.3. Computing methods

We computed the Confusion Matrix on the SVM model to evaluate the predicted values in producing the number of true labels in relation to the predicted labels. The Confusion Matrix consists of a 2x2 table that assesses the performance of true positives, true negative, false positives, and false negatives values. Then, we plot and display the Confusion Matrix to visualize the resulting labels. Using the output values from the Confusion Matrix, we then calculated the score of accuracy, precision, recall and F1 to evaluate how well the testing set performs.

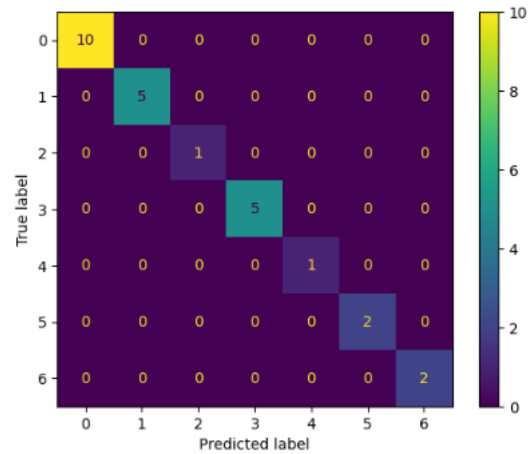
For the Decision Tree model, we wished to achieve comprehensive results of the accuracy of prediction, and we believed that a single generation of a particular model with tuned parameters would not be sufficient enough to reflect the true accuracy of that model. Therefore, we have devised a proper measure to our test by running 1000 iterations for generating each Decision Tree model characterized by a particular criterion at a particular maximum depth. That way, we were able to record the range of accuracy of prediction, including the minimum and maximum, of a particular model. Correspondingly, we could compute the average accuracy, which we believed to be the correct indication of the performance of our Decision Tree models.

4. EXPERIMENTAL RESULTS

4.1. Support Vector Machines



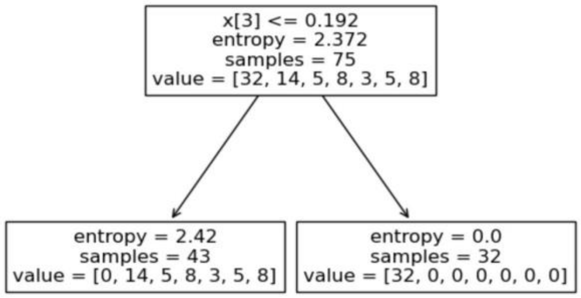
When using the decision function shape parameter set to 'ovr', the zoo classification predictions from the SVM model consistently provided correct output, with 24 out of the 26 test-set animals yielding correct predictions. This performance yields an accuracy of 92.3% and an F1 score of 91.2%. The model predictably struggled with classifying an amphibian, as that was the classification category that featured the fewest rows. The stingray was the other animal that was misclassified, which stands apart from most other fish by being larger in size as well as being the only fish from the dataset to be venomous. A larger dataset would have likely boosted the accuracy of the model, which could have introduced it to more of the less populated categories of animal, such as amphibians, and could have introduced it to other venomous fish such as the lionfish.



When using the OneVsRest wrapper class the zoo classification predictions were significantly more accurate, where each animal was correctly placed. This performance yielded an accuracy of 100% and an F1 score of 100%. This is likely due to the nature of the wrapper, which takes each class independently and runs the SVM to determine whether it is part of that class or not, rather than trying to determine which class it is in.

4.2. Decision Trees

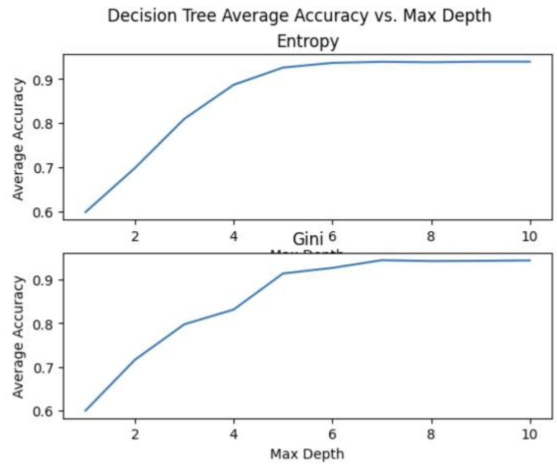
To make sure that our Decision Tree models were behaving correctly, we have compared their generated entropy value to our own calculated entropy value. For instance, the entropy value found at the root node of the Decision Tree model with set parameters (criterion = 'entropy', max_depth = 1, random_state = 0) was 2.372 as shown in the figure below. For our own calculation, we used the Entropy formula and input the sample size along with the distributed value for each animal type that was used in our model. We achieved an Entropy value of 2.37179 which was very close to the generated value.



$$Entropy(S) = -\sum_{c \in C} P(c) log_2 P(c)$$

$$Entropy(S) = -[\frac{32}{75} log_2 \frac{32}{75} + \frac{14}{75} log_2 \frac{14}{75} + \frac{5}{75} log_2 \frac{5}{75} + \frac{8}{75} log_2 \frac{8}{75} + \frac{3}{75} log_2 \frac{3}{75} + \frac{5}{75} log_2 \frac{5}{75} + \frac{8}{75} log_2 \frac{8}{75}] = 2.37179$$

Regarding the performance, the zoo classification predictions were shown to achieve at best an average accuracy of 94% regardless of either Entropy or Gini criterion the Decision Tree models deployed. Additionally, the rate of improved average accuracy corresponding to increased max depth is nearly identical between models using entropy criterion and those that used Gini criterion as shown in the figure below. The only notable difference was presented at max depth of 4 where models with Gini criterion underperformed by 5% compared to those with Entropy criterion. We observed that, by increasing the value of max depth parameter when generating Decision Tree models, the predictions capability improved at a nearly linear rate, particularly between a max depth of 1 to 5. At a max depth of 6, our Decision Trees models have hit an equilibrium point with respect to their accuracy performance, where the rate remained constant from this point on.



5. CONCLUSION

The experimental results concluded that the SVM performed with greater accuracy than the decision tree, with an accuracy score of 100% compared to 94%. Despite score comparisons it is important to consider the nature of the models themselves, especially for future works. Given the nature of SVM not every animal carries the same weight when deciding classifications. The dataset used primarily consists of mammals as shown in the resulting confusion matrix. If the dataset's size was increased, with animals that are harder to categorize it may affect the resulting comparison. Decision trees, on the other hand, allow us to better analyze the categorizations at any given step, allowing insights that are not indicative when comparing accuracy. Additionally, for future work it would be beneficial to observe how decision trees and SVM work in tandem as described in [7].

REFERENCE

[1] J. I. Cashman, H. Lee, R. Nguyen, S. Reyes, V. Tran. (2023, Apr. 2). *CECS456-Project* [Online]. Available: https://github.com/rubynguyen2505/CECS456-Project

[2] R. S. Forsyth, “Zoo Animal Classification,” 15 May 1990.

[3] S. S. Athani, S. A. Kodli, M. N. Banavasi and P. G. S. Hiremath, "Student performance predictor using multiclass support vector classification algorithm," 2017 International Conference on Signal Processing and Communication (ICSPC), Coimbatore, India, 2017, pp. 341-346, doi: 10.1109/CSPC.2017.830586

[4] Chih-Wei Hsu and Chih-Jen Lin, "A comparison of methods for multiclass support vector machines," in IEEE Transactions on Neural Networks, vol. 13, no. 2, pp. 415-425, March 2002, doi: 10.1109/72.991427.

[5] IBM. *What is a Decision Tree?* [Online]. Available: https://www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes

[6] O. Yenigün, “Decision tree parameter explanations,” *plainenglish.io/blog/decision-tree-parameters-explanations-tuning-a2b0749976e5*, 15-Sep-2021. [Online]. Available: https://plainenglish.io/blog/decision-tree-parameters-explanations-tuning-a2b0749976e5#overfitting-in-decision-trees.

[7] J. Chen, C. Wang and R. Wang, "Combining Support Vector Machines With a Pairwise Decision Tree," in IEEE Geoscience and Remote Sensing Letters, vol. 5, no. 3, pp. 409-413, July 2008, doi: 10.1109/LGRS.2008.916834