Introduction

For this project I used the Abalone dataset available from UCI's machine learning repository. This dataset can be found here: https://archive.ics.uci.edu/ml/datasets/abalone. The data consisted of seven continuous features and one categorical feature. My objective with this dataset was to predict the number of 'Rings' an abalone would have. The number of rings for each instance was an integer in range 1-30.

Overview

I leveraged three distinct classifications models, ZeroR, Gaussian Naive Bayes, and a Deep Neural Network. For each model I tried a variety of discretization methods for both features and targets. In addition to this I also explored the results of each model on subsets of the data, to do this I either dropped or combined features. Lastly I employed different evaluation techniques such as cross validation and test/train splits to obtain statistics about each model. To make all this possible I leveraged the Python packages Keras, Pandas, Numpy, and Sklearn.

Conclusion

In experiments 1, 2, 3, 4, 5 I implemented ZeroR models. Unsurprisingly all these models performed almost identically, discretizing features and/or removing data instances had no significant effect. All models had less than 20% accuracy. In experiment 3 I was able to artificially inflate model accuracy, precision and recall. I could accomplish this by dropping classes that did not occur frequently. Again, this result is unsurprising due to the nature of the ZeroR algorithm.

For experiment 6 I implemented a forward feed neural network. I also combined target labels into four different bins. The categorical accuracy of the model was 56%. Unfortunately

this model performed poorly. Nonetheless, I believe if given additional time to test different configurations of nodes, hyperparameters, etc. this model would be able to perform quite well. I am curious to how this model would perform if I encoded the labels similar to experiment 7.

For experiment 7 I used a Gaussian Naive Bayes Classifier and dropped categorical features. In addition to this I reframed the problem as a binary classification problem by combining target classes. This model had the best performance by far with accuracy of 76%. It is noteworthy that I achieved these results without the categorical feature.

Project Strength and Weaknesses

Something very important to highlight is that the goal of this project was not to create the best classification model for this dataset. The goal of this project was to explore the methods available to solve classification problems. I believe this project did showcase quite a few methods of tackling classification problems such as discretizing features, combining features, utilizing different models, etc. This is also what I believe the project's greatest weakness is. In an effort to showcase so many different methods available the project feels inorganic. If I were to do a similar project in the future I would certainly use less clean data and focus on making the best classification model instead of showing off the ones available. (Maximum 1 ZeroR experiment)

MODEL STATISTIC AND DETAILS BELOW

EXPERIMENT 1	
Code: https://colab.research.google.com/drive/1gfBIpmaHG0rYyvAmdufhM59_ghDYr3Kk?usp=sharing	
Model: ZeroR	
Data: Continuous, Categorical	
Preprocessing: 1. Converted categorical feature 'Sex' to one-hot encoding. 2. Split the entire dataset into a training and test set (80/20 split).	
Details: For this experiment I used a train and test set (80/20 split) to evaluate the ZeroR model on all the data.	
Training Set Metrics: Accuracy: 0.16 Precision: - Macro Average: 0.01 - Weighted Average: 0.03 Recall: - Macro Average: 0.04 - Weighted Average: 0.16	**Test Set Metrics:** Accuracy: 0.19 Precision: - Macro Average: 0.01 - Weighted Average: 0.03 Recall: - Macro Average: 0.05 - Weighted Average: 0.19

EXPERIMENT 2		
Code: https://colab.research.google.com/drive/1gfBIpmaHG0rYyvAmdufhM59_ghDYr3Kk?usp=sharing		
Model: ZeroR		
Data: Continuous, Categorical		
Preprocessing: 1. Converted categorical feature 'Sex' to one-hot encoding. 2. Dropped all samples belonging to classes with less than 20 instances.		
Details: For this experiment I used 3-fold cross validation to test the ZeroR model on all the data.		
Fold 1 Metrics: Accuracy: 0.17 Precision: - Macro: 0.01 - Weighted: 0.03 Recall: - Macro: 0.06 - Weighted: 0.17	**Fold 2 Metrics:** Accuracy: 0.17 Precision: - Macro: 0.01 - Weighted: 0.03 Recall: - Macro: 0.06 - Weighted: 0.17	**Fold 3 Metrics:** Accuracy: 0.17 Precision: - Macro: 0.01 - Weighted: 0.03 Recall: - Macro: 0.06 - Weighted: 0.17

EXPERIMENT 3	
Code: https://colab.research.google.com/drive/1OPSkWSXedpQxHCzOGuK96cyrqVf2Su-f?usp=sharing	
Model: ZeroR	
Data: Categorical	
Preprocessing: 1. Dropped categorical feature 'Sex'. 2. Dropped all samples belonging to classes with less than 200 instances. 3. Discretized all continuous features using 3 quantiles. 4. Split dataset into training and test sets (80/20 split).	
Details: For this experiment I evaluate a ZeroR model on a subset of the original dataset with discretized features. I did not include the feature 'Sex' nor did I include any classes that appeared infrequently, less than 200 instances. All features were discretized using a quantile strategy with 3 bins. I obtained metrics for both a training and test set.	
Training Set Metrics: Accuracy: 0.20 Precision: - Macro Average: 0.03 - Weighted Average: 0.04 Recall: - Macro Average: 0.12 - Weighted Average: 0.20	**Test Set Metrics:** Accuracy: 0.18 Precision: - Macro Average: 0.02 - Weighted Average: 0.03 Recall: - Macro Average: 0.12 - Weighted Average: 0.18

EXPERIMENT 4	
Code: https://colab.research.google.com/drive/1Idrw3hRQCXDrwAejT68zshRdhA94Nuwp?usp=sharing	
Model: ZeroR	
Data: Continuous, Categorical	
Preprocessing: 5. Converted categorical feature 'Sex' to one-hot encoding. 6. Converted discretized all continuous features using an equal bins strategy, n_bins=5 7. Split the entire dataset into a training and test set (65/35 split).	
Details: For this experiment I used a train and test set to evaluate the ZeroR model on all the data.	
Training Set Metrics: Accuracy: 0.16 Precision: - Macro Average: 0.01 - Weighted Average: 0.03 Recall: - Macro Average: 0.04 - Weighted Average: 0.16	**Test Set Metrics:** Accuracy: 0.17 Precision: - Macro Average: 0.01 - Weighted Average: 0.03 Recall: - Macro Average: 0.04 - Weighted Average: 0.17

EXPERIMENT 5				
Code: https://colab.research.google.com/drive/1Idrw3hRQCXDrwAejT68zshRdhA94Nuwp?usp=sharing				
Model: ZeroR				
Data: Categorical				
Preprocessing: 1. Converted categorical feature 'Sex' to one-hot encoding. 2. Converted discretized all continuous features using an equal bins strategy, n_bins=5				
Details: For this experiment I used 5-fold cross validation to test the ZeroR model on all the data.				
Fold 1 Metrics: Accuracy: 0.16 Precision: - Macro: 0.01 - Weighted: 0.03 Recall: - Macro: 0.04 - Weighted: 0.16	**Fold 2 Metrics:** Accuracy: 0.17 Precision: - Macro: 0.01 - Weighted: 0.03 Recall: - Macro: 0.05 - Weighted: 0.17	**Fold 3 Metrics:** Accuracy: 0.17 Precision: - Macro: 0.01 - Weighted: 0.03 Recall: - Macro: 0.05 - Weighted: 0.17	**Fold 4 Metrics:** Accuracy: 0.17 Precision: - Macro: 0.01 - Weighted: 0.03 Recall: - Macro: 0.04 - Weighted: 0.17	**Fold 5 Metrics:** Accuracy: 0.17 Precision: - Macro: 0.01 - Weighted: 0.03 Recall: - Macro: 0.04 - Weighted: 0.17

EXPERIMENT 6	
Code: https://colab.research.google.com/drive/18vJS17nDwF_6bZ0LI59W491VI65hEt_b?usp=sharing	
Model: Naive Bayes Classifier	
Data: Continuous	
Preprocessing: 1. Created a new binary feature, '>median', based on if n_rings > median_n_rings 2. Dropped categorical feature 'Sex'. 3. Split dataset into training and test sets (70/30 split).	
Details: For this experiment I removed categorical features. I also created a new target label, the scheme for classifying the dataset can be seen above. I implemented a Gaussian NBC to solve this binary classification problem.	
Training Set Metrics: Accuracy: 0.76 Precision: - Macro Average: 0.76 - Weighted Average: 0.76 Recall: - Macro Average: 0.76 - Weighted Average: 0.76	**Test Set Metrics:** Accuracy: 0.73 Precision: - Macro Average: 0.73 - Weighted Average: 0.73 Recall: - Macro Average: 0.73 - Weighted Average: 0.73

EXPERIMENT 7	
Code: https://colab.research.google.com/drive/1j7oTZpy1WAaH8qc-hpGdAb06_KT0VMY6?usp=sharing	
Model: Deep Forward Feed NN	
Data: Categorical, Continuous	
Preprocessing: 1. Converted categorical features to one-hot 2. Created a label based on 'Rings' value using the following bins a. n_rings <= 0.25 quantile b. 0.25 quantile < n_rings <= 0.5 quantile c. 0.5 quantile < n_rings <= 0.75 quantile d. 0.75 quantile < n_rings 3. Split dataset into training and test sets (80/20 split).	
Details: For this experiment I created a new target label. This target label was categorical and the strategy for partitioning the original dataset into this class can be seen above. In addition to this I used a deep forward feed neural network to classify the data. The neural network consisted of two hidden layers each with 128 nodes. I used the Adam optimizer and categorical cross entropy as my loss function.	
Training Set Metrics: Categorical Accuracy: 0.56 Precision: 0.78 Recall: 0.30 AUC: 0.82	**Test Set Metrics:** Categorical Accuracy: 0.54 Precision: 0.76 Recall: 0.30 AUC: 0.82