# Binary Classification of Abalone Age

Aladdin Alakhras
University of Missouri-St. Louis

December 9, 2024

## Contents

**1 Phase One: Preparation and Exploratory Data Analysis 3**
1.1 Data Cleaning . . . 4
1.2 Feature Selection and Normalized Dataset . . . 4
1.3 Min-Max Normalization . . . 5
1.4 Correlation heat map . . . 6
1.5 Visualization Setup . . . 6
1.5.1 Target variable distribution . . . 6
1.5.2 Distribution of continuous variables . . . 7
1.5.3 Correlation Analysis with Target Variable . . . 8
1.5.4 Scatter Plots Showing the Relationship Between Continuous Features . . . 8
1.6 Distribution of Output Labels in Training and Validation Sets . . . 9

**2 Phase 3: Model selection evaluation 9**
2.1 This code implements a machine learning workflow . . . 9
2.2 Plot Learning Curves . . . 11
2.3 ROC AUC Evaluation . . . 11
2.4 Custom Predictions Using My Prediction Function . . . 12
2.5 Discussion on Architecture Size for Overfitting with Output as an Additional Input Feature . . . 13
2.6 Conclusion . . . 14

**3 Phase 4: Feature importance and reduction 14**
3.1 Identify and Remove Least Important Features . . . 15
3.2 Validation Accuracy with Feature Reduction . . . 15
3.3 Comparison of Model Accuracies . . . 16
3.4 Use model-agnostic methods such as LIME or Shapley values to derive feature importance. . . . 16

**4 Technical Enhancements 18**
4.1 Increase Epochs . . . 18
4.2 Add Additional Layers . . . 18

5 Balance the Dataset 18

6 Adjust Training and Validation Set Size 18

7 Additional Features for Analysis 18

7.1 Geographical Location . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

7.2 Abalone Species . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

7.3 Color . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

7.4 Number of Predators . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

7.5 Living Environment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

8 References 19

**Abstract**

This report explores a machine learning approach to efficiently classify abalone snails into "young" or "old" categories. Using TensorFlow and Keras, this model examines features such as gender, height, and weight to accelerate age determination, improving research and commercial methodologies.

# Introduction

Abalones are endangered marine snails that inhabit cold coastal waters worldwide. Their price generally increases with age; however, determining an abalone's age can be complicated. To address this challenge, we will develop a machine learning model to classify abalones as either "young" or "old" based on physical and biological attributes.

# About the Dataset

In this project, we will use features such as gender, height (including the meat in the shell), and shell weight to classify abalones as either "young" or "old." The dataset contains 4,177 observations and 8 features. The categorical feature, sex, indicates whether an abalone is male, female, or infant, while the other 7 features are numeric and describe the abalones' size and weight.

Sourced from the UCI Machine Learning Repository and published in 1995, each entry provides essential details such as the number of rings, sex, length, diameter, height, and weight. Researchers counted the rings using a microscope, and the age of an abalone is calculated as the number of rings plus 1.5 years.

Different sexes of abalones also have varying body compositions and economic values. This dataset has been cleaned to remove missing values and scaled for analysis. While previous research treated it as a three-category classification problem, we will simplify it to a two-category classification, grouping ring counts into those with 10 rings or fewer and those with more than 10.

# 1 Phase One: Preparation and Exploratory Data Analysis

In this section, we outline the preparation steps undertaken prior to conducting Exploratory Data Analysis (EDA) on the Abalone dataset. This preparation is crucial for ensuring the data is clean, well-structured, and ready for insightful analysis.

## 1.1 Data Cleaning

In data preparation, we identified and fixed errors by removing rows with missing or invalid values to ensure accurate representation of the abalones. A sample of the dataset, which includes features like sex and size, is shown below. We found two rows with zero height values, which were removed for analysis.

Abalones are classified into ”old” = 9 rings) and ”young” ¡ 9 rings. Although this threshold is somewhat arbitrary, it results in an imbalanced target variable, which could impact our predictive model. Addressing this imbalance is crucial for improving model performance and achieving reliable classifications.

To evaluate the balance of our dataset, we calculated the distribution of the output labels. The results are as follows:

| **Index** | **Sex** | **Length** | **Diameter** | **Height** | **Whole Weight** | **Shucked Weight** | **Viscera Weight** | **Shell Weight** | **Rings** | **Is Old** |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | M | 0.455 | 0.365 | 0.095 | 0.5140 | 0.2245 | 0.1010 | 0.150 | 15 | Old |
| 1 | M | 0.350 | 0.265 | 0.090 | 0.2255 | 0.0995 | 0.0485 | 0.070 | 7 | Young |
| 2 | F | 0.530 | 0.420 | 0.135 | 0.6770 | 0.2565 | 0.1415 | 0.210 | 9 | Young |
| 3 | M | 0.440 | 0.365 | 0.125 | 0.5160 | 0.2155 | 0.1140 | 0.155 | 10 | Young |
| 4 | I | 0.330 | 0.255 | 0.080 | 0.2050 | 0.0895 | 0.0395 | 0.055 | 7 | Young |

Table 1: Sample Data from the Abalone Dataset after cleaning and adding age ’Is Old’

| Target | Old | Young |
|---|---|---|
| Observations | 2081 | 2094 |
| Percentage | 49.8% | 50.2% |

Table 2: Distribution of Target Variable

The dataset almost balanced ; however, it is sufficiently balanced for our analysis, allowing us to proceed with our modeling efforts.

The following table summarizes key statistics for the features in the Abalone dataset:

| **Statistic** | **Length** | **Diameter** | **Height** | **Whole Weight** | **Shucked Weight** | **Viscera Weight** | **Shell Weight** | **Rings** |
|---|---|---|---|---|---|---|---|---|
| Count | 4175.0 | 4175.0 | 4175.0 | 4175.0 | 4175.0 | 4175.0 | 4175.0 | 4175.0 |
| Mean | 0.5241 | 0.4079 | 0.1396 | 0.8290 | 0.3595 | 0.1807 | 0.2388 | 9.9351 |
| Standard Deviation | 0.1201 | 0.0992 | 0.0417 | 0.4903 | 0.2220 | 0.1096 | 0.1392 | 3.2242 |
| Min | 0.0750 | 0.0550 | 0.0100 | 0.0020 | 0.0010 | 0.0005 | 0.0015 | 1.0 |
| 25% | 0.4500 | 0.3500 | 0.1150 | 0.4423 | 0.1863 | 0.0935 | 0.1300 | 8.0 |
| 50% (Median) | 0.5450 | 0.4250 | 0.1400 | 0.8000 | 0.3360 | 0.1710 | 0.2340 | 9.0 |
| 75% | 0.6150 | 0.4800 | 0.1650 | 1.1535 | 0.5020 | 0.2530 | 0.3288 | 11.0 |
| Max | 0.8150 | 0.6500 | 1.1300 | 2.8255 | 1.4880 | 0.7600 | 1.0050 | 29.0 |

Table 3: Summary Statistics of the Abalone Dataset

## 1.2 Feature Selection and Normalized Dataset

We focused on selecting the most relevant features for our analysis. Key attributes such as gender, height, weight, and the number of rings were prioritized to explore their relationships and impacts on age classification.

| Index | Sex | Length | Diameter | Height | Whole Weight | Shucked Weight | Viscera Weight | Shell Weight | Rings | Is Old |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.0 | 0.455 | 0.365 | 0.095 | 0.5140 | 0.2245 | 0.1010 | 0.150 | 15 | 1.0 |
| 1 | 1.0 | 0.350 | 0.265 | 0.090 | 0.2255 | 0.0995 | 0.0485 | 0.070 | 7 | 0.0 |
| 2 | 0.0 | 0.530 | 0.420 | 0.135 | 0.6770 | 0.2565 | 0.1415 | 0.210 | 9 | 0.0 |
| 3 | 1.0 | 0.440 | 0.365 | 0.125 | 0.5160 | 0.2155 | 0.1140 | 0.155 | 10 | 0.0 |
| 4 | 2.0 | 0.330 | 0.255 | 0.080 | 0.2050 | 0.0895 | 0.0395 | 0.055 | 7 | 0.0 |

Table 4: Convert ’Sex’ and ’Is old’ columns to numeric representation

## 1.3 Min-Max Normalization

We used Min-max normalization is a technique used to scale data to a fixed range, typically [0, 1]. The formula for min-max normalization of a feature $x$ is given by:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}$$

| Index | Sex | Length | Diameter | Height | Whole Weight | Shucked Weight | Viscera Weight | Shell Weight | Rings |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.5 | 0.539007 | 0.517857 | 0.112108 | 0.267081 | 0.163245 | 0.219223 | 0.181637 | 0.269231 |
| 1 | 1.0 | 0.312057 | 0.285714 | 0.071749 | 0.071162 | 0.066981 | 0.051350 | 0.048902 | 0.153846 |
| 2 | 0.0 | 0.581560 | 0.553571 | 0.103139 | 0.218811 | 0.143049 | 0.271231 | 0.166667 | 0.461538 |
| 3 | 1.0 | 0.255319 | 0.214286 | 0.040359 | 0.039574 | 0.036015 | 0.030283 | 0.026946 | 0.115385 |
| 4 | 1.0 | 0.368794 | 0.330357 | 0.112108 | 0.075776 | 0.063615 | 0.059250 | 0.058383 | 0.115385 |

Table 5: Normalized Sample Data from the Abalone Dataset

## 1.4 Correlation heat map

A correlation heat map provides us with more visually intuitive relationship between all variables. From the heat map we can tell that feature variables are highly correlated.

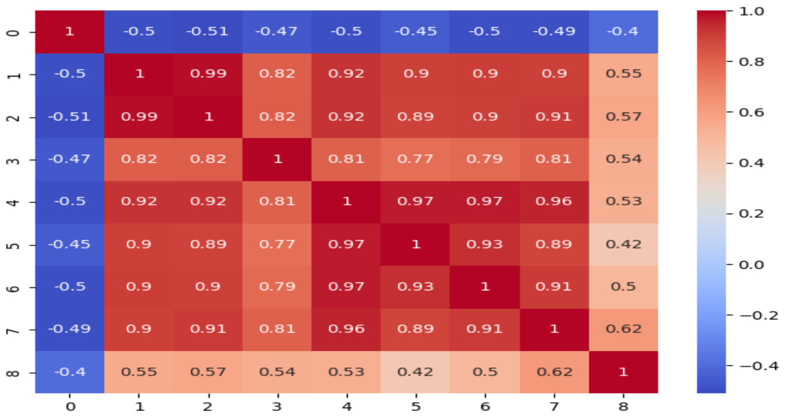


Figure 1: A correlation heat map provides us relationship between all variables.

## 1.5 Visualization Setup

### 1.5.1 Target variable distribution

Exploratory data analysis (EDA) helps us understand the training and validation data. By plotting the target variable, we can see how ”young” and ”old” abalones are distributed based on their ring count. The distribution is slightly skewed to the right, showing more young than old abalones, as our threshold for ”old” is set at more than 1 rings.

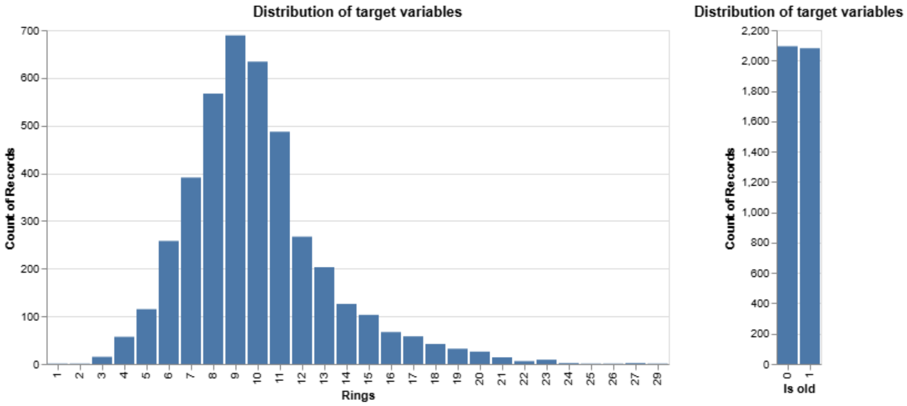


Figure 2: Length and Rings

### 1.5.2 Distribution of continuous variables

We plot the distribution of all numeric features within two targeted classes.From the plot we can group the numeric variables into three groups: (length, diameter), (height), and (whole weight, shucked weight, viscera weight, shell weight). The first group is left skewed. The means of two classes are similar and the old abalones have less deviation from mean.The second group has some outliers and the third group is right skewed. In the third group, we can observe a difference in mean weights and the distribution of old abalones are more bell-shaped. float

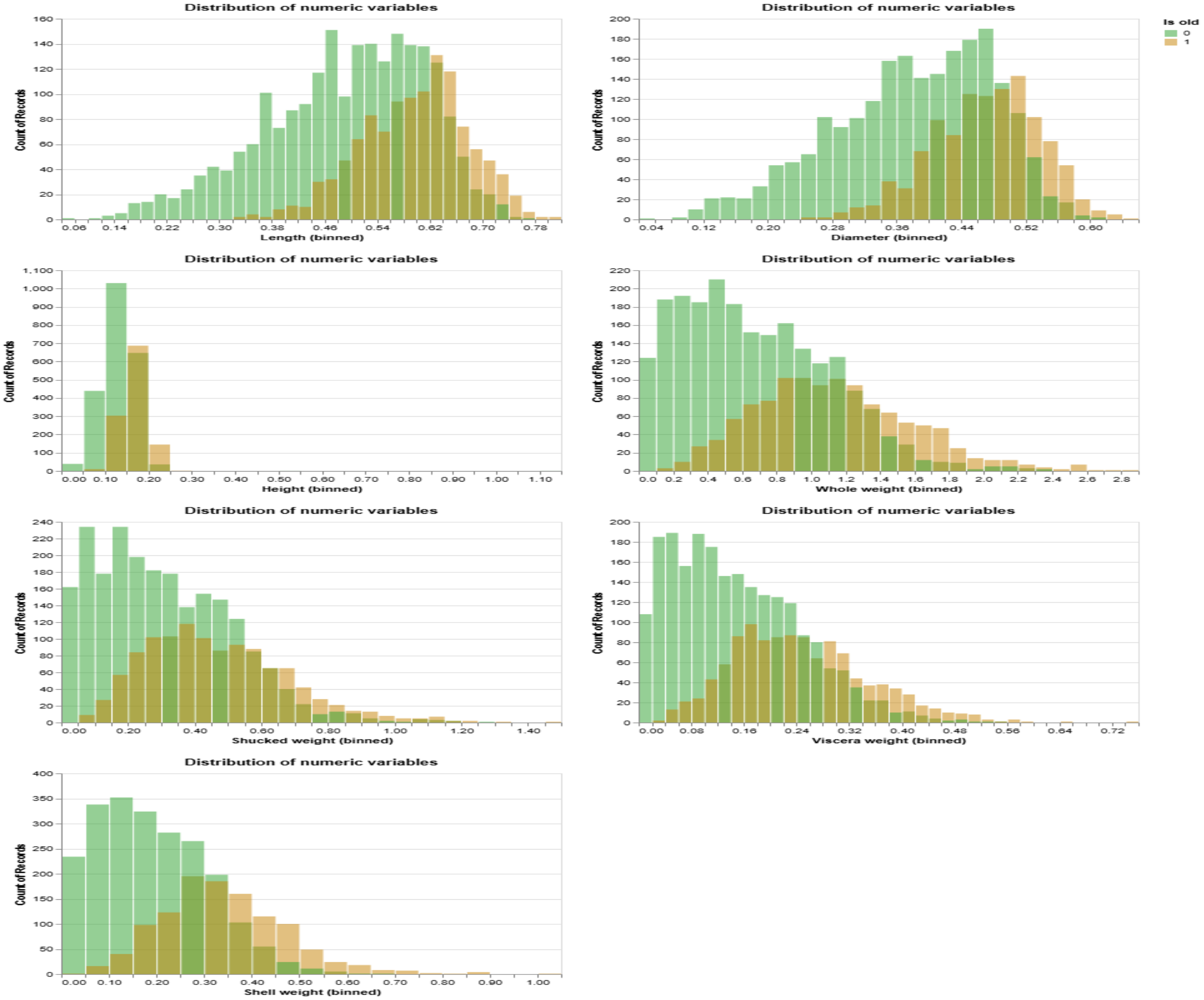


Figure 3: Distribution of continuous variables.

### 1.5.3 Correlation Analysis with Target Variable

In our analysis, we aim to explore the correlation between the selected predictor variables—length, height, and whole weight—and the target variable, rings. Our focus is also on identifying potential differences between young and old abalones.

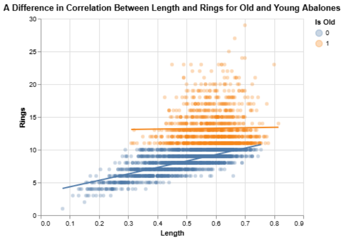


Figure 4: Length and Rings

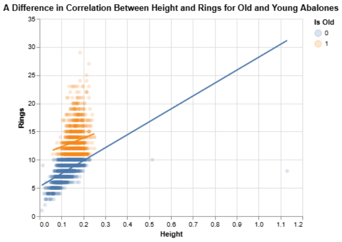


Figure 5: Height and Rings

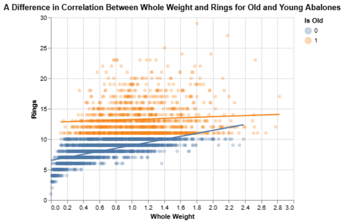


Figure 6: Whole weight and Rings

### 1.5.4 Scatter Plots Showing the Relationship Between Continuous Features

Strong correlations exist between abalone size (length, diameter, height) and weight, potentially causing multicollinearity issues in some models. Size shows linear correlation, but the size-weight relationship is non-linear. Weight features have weaker correlations than size features. Older abalones show a slightly higher weight.

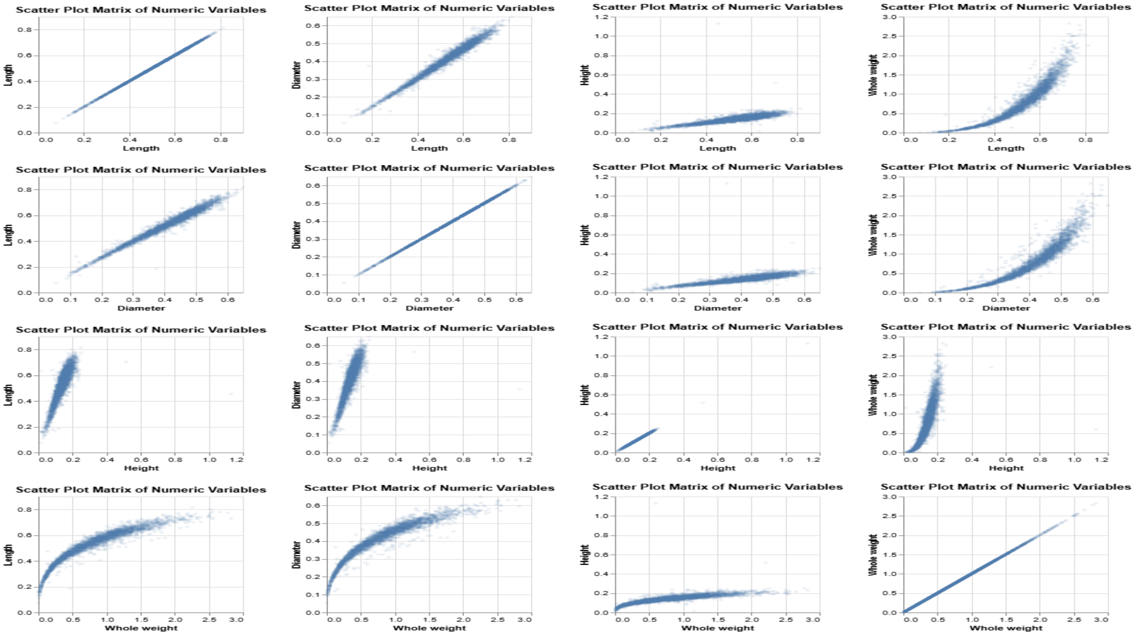


Figure 7: Distribution of continuous variables.

### 1.6 Distribution of Output Labels in Training and Validation Sets

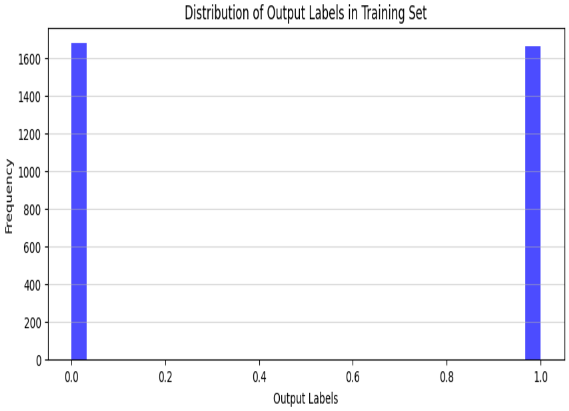


Figure 8: Description of what the image represents.

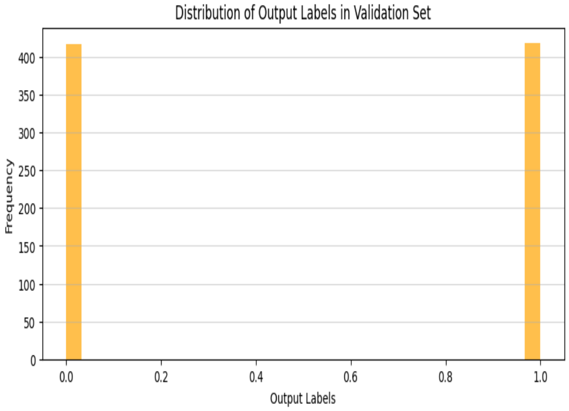


Figure 9: Description of what the image represents.

## 2 Phase 3: Model selection evaluation

### 2.1 This code implements a machine learning workflow

that starts by defining and training a logistic regression model as a baseline for binary classification. It then evaluates multiple neural network architectures to improve performance, using ReLU activation in hidden layers and a sigmoid activation for the output layer. The models are trained and evaluated based on key metrics, including accuracy, precision, recall, and 1 score. Learning curves are plotted to visualize performance over epochs, and the best-performing model is saved using model checkpointing and early stop. Finally, a classification report summarizes the model's predictive capabilities, identifying the architecture that achieved the highest validation accuracy. **Classification Report Metrics:**

| Model Architecture | Acc. on Training Set | Acc. on Validation Set | Training Loss | Validation Loss | Total Parameters |
|---|---|---|---|---|---|
| (2, 1) | 0.662 | 0.668 | 0.640 | 0.635 | 35 |
| (4, 1) | 0.837 | 0.838 | 0.369 | 0.369 | 63 |
| (8, 1) | 0.844 | 0.848 | 0.362 | 0.358 | 119 |
| (16, 8, 1) | 0.796 | 0.794 | 0.553 | 0.553 | 391 |
| (32, 16, 8, 1) | 0.792 | 0.794 | 0.508 | 0.505 | 1191 |
| (64, 32, 16, 8, 1) | 0.833 | 0.832 | 0.543 | 0.542 | 3815 |

Table 6: Summary of model performances

**Accuracy:** 85.00%

**Precision:** 85.00%

**Recall:** 94.00%

**F1 Score:** 89.00%

**Random Baseline Classifier Accuracy:** 53.4%

**Logistic Regression - Training Accuracy:** 73.7%

**Logistic Regression - Validation Accuracy:** 73.8%

**The best performing model architecture is:** (8, 1) with validation accuracy: 84.8

**Number of weights for each layer:**
(8, 64), (64,), (64, 32), (32,), (32, 16), (16,), (16, 8), (8,), (8, 1), (1,), (1, 1)

**Number of biases for each layer:**
(64,), (64,), (32,), (32,), (16,), (16,), (8,), (8,), (1,), (1,), (1)

## 2.2 Plot Learning Curves

Learning curves for each architecture are plotted to visualize how both training and validation accuracies evolve over epochs. This helps identify how well each model learns and if any of them are overfitting. Load the Best Model: The model with the best performance (lowest validation loss) is loaded for final evaluation.

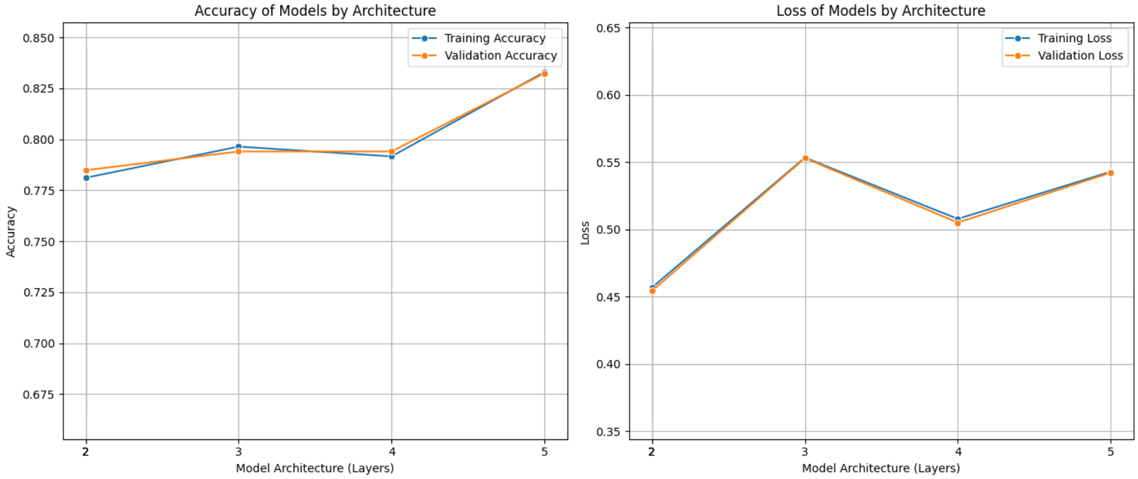


Figure 10: Learning Curves for Various models.

## 2.3 ROC AUC Evaluation

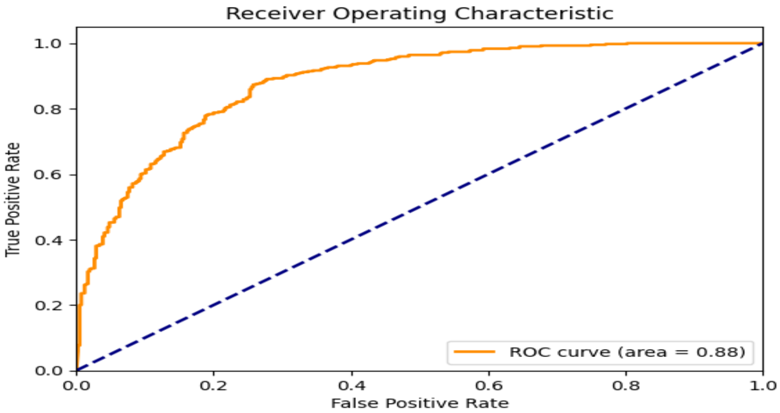


Figure 11: ROC AUC Evaluation.

## 2.4 Custom Predictions Using My Prediction Function

```
def my_prediction_function(model, data):
    output = data
    for layer in model.layers:
        if layer.get_weights():  # Check if the layer has weights
            if len(layer.get_weights()) == 2:
            # Extract weights and bias
                weights, bias = layer.get_weights()
                # Apply linear transformation
                output = np.dot(output, weights) + bias
            else:
                # Apply the layer’s call method to the output
                output = layer.call(output)
        # Apply activation function (if any)
        if hasattr(layer, ’activation’):
            if layer.activation.__name__ == ’relu’:
                output = np.maximum(0, output)
            elif layer.activation.__name__ == ’sigmoid’:
                output = 1 / (1 + np.exp(-output))
    return output
```

**Predictions:**

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

**Binary Predictions:**

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

**Note:** Predictions from both methods are the same.

**Status:** Process completed successfully!

### 2.5 Discussion on Architecture Size for Overfitting with Output as an Additional Input Feature

Single Neuron with Sigmoid Activation:

Using a simple architecture with a single neuron and sigmoid activation can help understand the baseline capabilities of the model. However, since it is relatively simplistic, it might lack situational awareness to capture complexities in the dataset. While this may prevent overfitting due to limited capacity, it might also underfit if the relationships in the data are complicated.

```
model = Sequential()
model.add(Dense(1, input_dim=XTRAIN_np.shape[1], activation='sigmoid'))
print(model.summary())
model.compile(loss='binary_crossentropy', optimizer='sgd', metrics=['accuracy'])
optimization
early_stop = EarlyStopping(monitor='val_loss', patience=10, verbose=1, mode='min',
restore_best_weights=True)
history = model.fit(
    XTRAIN_np, YTRAIN_np,
    validation_data=(XVALIDATION_np, YVALIDATION_np),
    epochs=40, batch_size=32, verbose=1, callbacks=[early_stop]
)
```

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense_28 (Dense) | (None, 1) | 10 |

Table 7: Summary of the model architecture.

Total params: 10 (40.00 B)
Trainable params: 10 (40.00 B)
Non-trainable params: 0 (0.00 B)

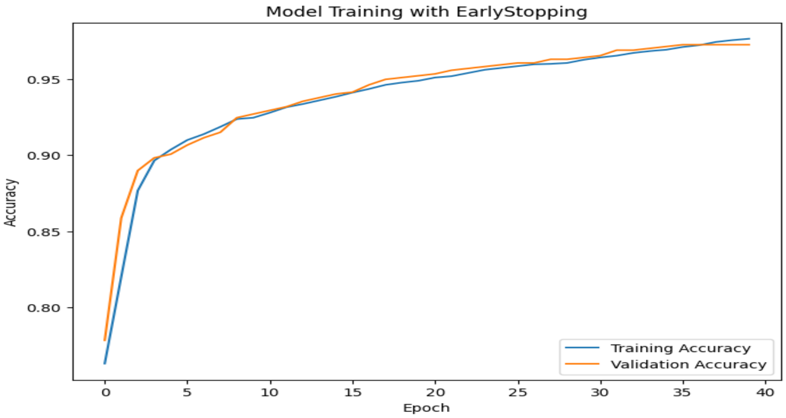


Figure 12: Model Training with EarlyStopping.

## 2.6 Conclusion

Model Selection and Evaluation: The core of the report is dedicated to model selection and evaluation. It begins by establishing a logistic regression model as a baseline. The primary focus is on training and evaluating various neural network architectures (ranging from simple (2,1) to more complex architectures like (64,32,16,8,1)), using binary cross-entropy as the loss function and SGD as the optimizer. Model performance is assessed using key metrics (accuracy, precision, recall, F1-score), and learning curves are presented to visualize the training process and identify potential overfitting. Early stopping is employed to prevent overfitting. A classification report provides a summary of the best model's performance. ROC AUC is also calculated.

# 3 Phase 4: Feature importance and reduction

The following table lists the accuracy associated with each feature used in the model for the Abalone dataset:

| **Feature** | **Importance Score** |
|---|---|
| Sex | 80.60% |
| Length | 81.32% |
| Diameter | 81.44% |
| Height | 79.40% |
| Whole weight | 76.17% |
| Shucked weight | 80.12% |
| Viscera weight | 81.68% |
| Shell weight | 83.23% |

Table 8: Feature Importance Scores for the Abalone Classification Task

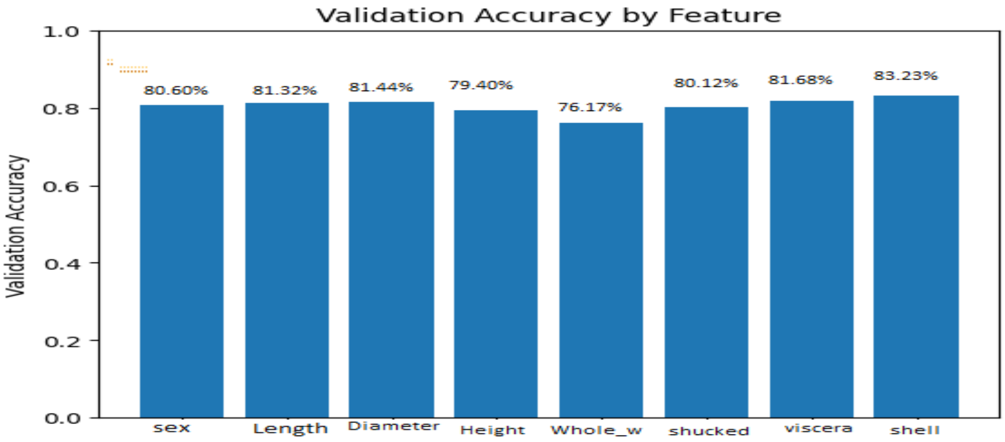


Figure 13: Valiation Accuracy by Feature.

## 3.1 Identify and Remove Least Important Features

The table below lists the accuracies achieved with various feature combinations:

| Features | Accuracy |
|---|---|
| ['Sex', 'Length', 'Diameter', 'Height', 'Shucked weight', 'Viscera weight', 'Shell weight'] | 85.03% |
| ['Sex', 'Length', 'Diameter', 'Shucked weight', 'Viscera weight', 'Shell weight'] | 84.67% |
| ['Sex', 'Length', 'Diameter', 'Viscera weight', 'Shell weight'] | 84.79% |
| ['Length', 'Diameter', 'Viscera weight', 'Shell weight'] | 82.51% |
| ['Diameter', 'Viscera weight', 'Shell weight'] | 82.75% |
| ['Viscera weight', 'Shell weight'] | 82.51% |
| ['Shell weight'] | 83.23% |

Table 9: Accuracies achieved with various feature combinations.

## 3.2 Validation Accuracy with Feature Reduction

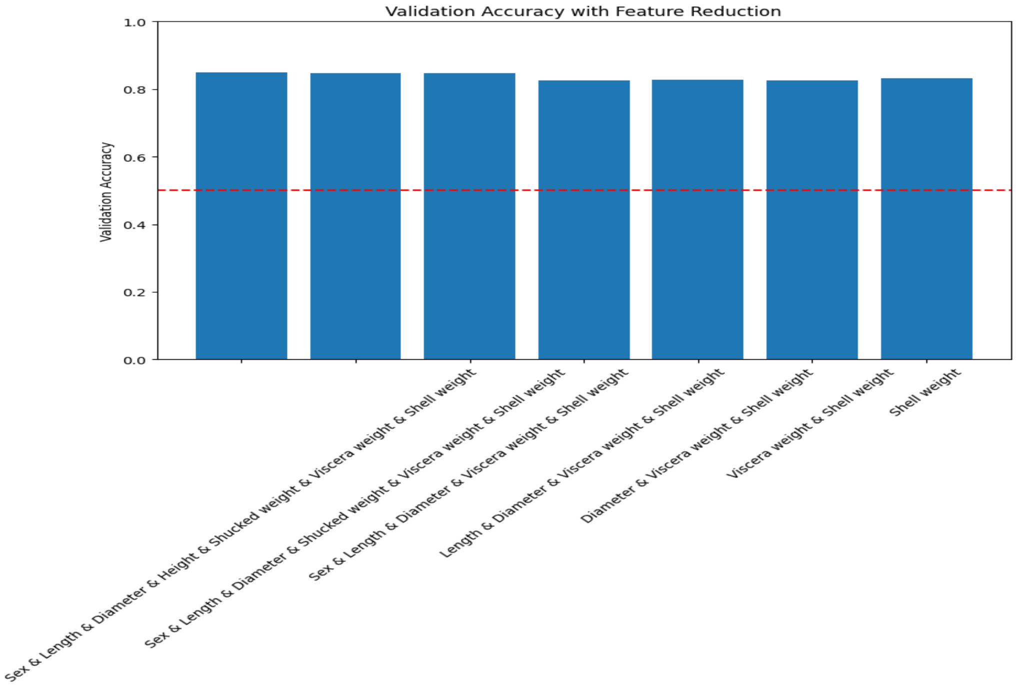


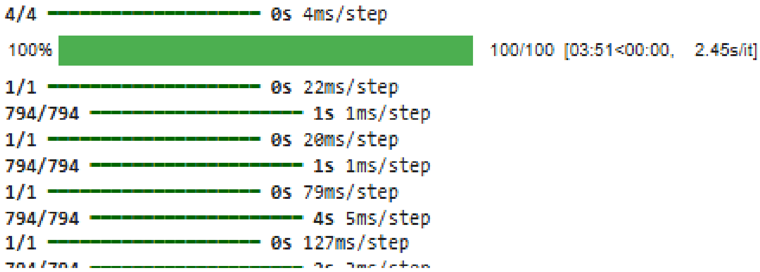
Figure 14: Valiation Accuracy by Feature.

### 3.3 Comparison of Model Accuracies

| Metric | Value |
|---|---|
| Accuracy (All Features) | 84.67% |
| Accuracy (Reduced Features) | 83.23% |
| Removed Features | Whole weight, Height, Shucked weight, Sex, Length, Diameter, Viscera weight, Shell weight |

Table 10: Comparison of Model Accuracies Before and After Feature Reduction

### 3.4 Use model-agnostic methods such as LIME or Shapley values to derive feature importance.



```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
100% ████████████████████ 100/100 [03:51<00:00,  2.45s/it]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
794/794 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step
794/794 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step
794/794 ━━━━━━━━━━━━━━━━━━━━ 4s 5ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 127ms/step
794/794 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step
```

Figure 15: Valiation Accuracy by Feature.

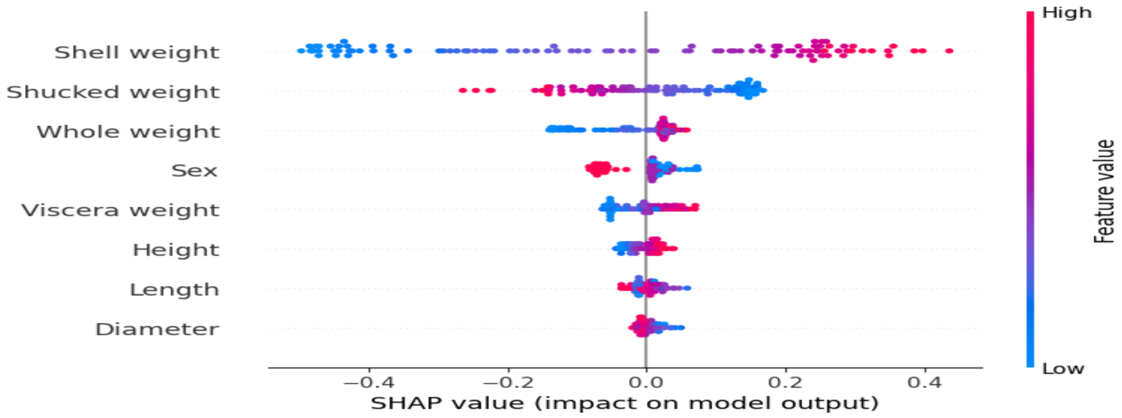


Figure 16: Valiation Accuracy by Feature.

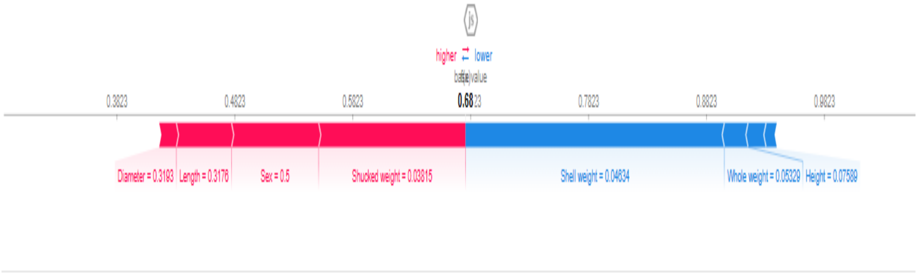


Figure 17: Valiation Accuracy by Feature.

# 4 Technical Enhancements

## 4.1 Increase Epochs

This will provide the model with additional time to learn from the data.

## 4.2 Add Additional Layers

allowing it to capture more intricate patterns in the data.

# 5 Balance the Dataset

It is crucial to ensure that the dataset contains an equal number of samples for each class. You can achieve this through:

# 6 Adjust Training and Validation Set Size

Modify the number of records in your training and validation sets based on your needs:

[noitemsep]**Increase Rows:** If your dataset is limited, consider gathering more data to augment it. **Decrease Rows:** If the majority class has a disproportionately large number of records, reducing its size may help in achieving better balance and model performance.

# 7 Additional Features for Analysis

## 7.1 Geographical Location

Including the locations where abalones are collected can help us understand how different environments affect their characteristics.

## 7.2 Abalone Species

Adding information about the species will allow us to account for biological differences that may impact classification.

## 7.3 Color

The color of abalones can vary, and using this feature can help the model better distinguish between different types.

## 7.4 Number of Predators

Knowing how many natural predators are in the area could provide insights into the abalones' health and behaviors, potentially influencing classification.

## 7.5 Living Environment

Features such as the type of habitat (e.g., rocky, sandy) and environmental conditions (like pollution and temperature) are crucial for understanding their characteristics.

# 8 References

1. kaggle
2. youtube
3. geeksforgeek
4. IBM