Predicting Abalone Age from Physical Measurements

Ethan Chung

University of Hawaii at Manoa

ICS 435: Machine Learning

Professor Sadowski

May 10, 2024

Predicting Abalone Age from Physical Measurements

Abalones are sea snails that are prized for their unique meat, beautiful shells, and cultural importance, but their price varies significantly with age. Traditionally, methods to determine the age of abalone involve laborious tasks like shell cutting, staining, and ring counting under a microscope. Fortunately, machine learning offers an alternative method to predict the age of abalone using other easier-to-obtain physical measurements, thus saving time and money.

The goal is to develop a regression model capable of predicting the age of abalone through various physical measurements. The dataset utilized, sourced from Kaggle's Playground Series *Regression with Abalone Dataset* and derived from UC Irvine's Machine Learning Repository, comprises of 90,614 labeled samples featuring nine features: sex (male, female, infant), length, diameter, height, whole weight, shucked weight (meat only), viscera weight (post-bleeding), shell weight (post-drying), and the number of rings. The number of rings is equivalent to the abalone's age plus 1.5 years, which serves as the target feature for prediction. To train and validate the model, 80% of the data is allocated for training purposes, while the remaining 20% is reserved for validation. Additionally, stratification was employed to keep the ring distribution approximately the same between the training and validation sets.

Three models, XGBoost, CatBoost, and CatBoost-Categorical, were trained using the training dataset and evaluated using the Root Mean Squared Logarithmic Error (RMSLE) metric with the validation dataset. Both XGBoost and CatBoost models employed one-hot encoding for the 'sex' feature to differentiate between female (F), male (M), and infant (I) categories. In contrast, CatBoost-Categorical utilized native categorical transformation built into CatBoost without employing one-hot encoding.

Hyperparameters Optimization

Hyperparameter optimization was performed on all three models with RandomSearchCV by maximizing the RMSLE score on the validation dataset. GridSearchCV was initially used for the XGBoost model with 3-fold cross-validation, yielding an RMSLE of 0.156 on the validation set. However, this approach was replaced by RandomSearchCV to introduce more randomness in parameter exploration, which evidently improved model performance.

XGBoost Hyperparameters

RandomSearchCV was performed with 100 iterations and 5-fold cross-validation on the validation set, which yielded an RMSLE of 0.199 on the validation set.

The hyperparameter search space was defined as follows:

```
'learning_rate': uniform(0.01, 0.1),
'n_estimators': randint(100, 500),
'min_child_weight': randint(1, 10),
'gamma': uniform(0.5, 2.0),
'subsample': uniform(0, 1),
'colsample_bytree': uniform(0, 1),
'max_depth': randint(3, 6)
```

The best hyperparameters was found to be:

```
{'colsample_bytree': 0.012314958999426362, 'gamma': 2.3119404591579933,
'learning_rate': 0.012610675251619944, 'max_depth': 4, 'min_child_weight': 7,
'n_estimators': 128, 'subsample': 0.7353106396614744}
```

CatBoost Hyperparameters

RandomSearchCV was performed on both models with 10 iterations and 5-fold cross-validation on the validation set. This yielded an RMSLE of 0.156 on the CatBoost model, 0.160 on CatBoost-Categorical model, both evaluated on the validation set.

The hyperparameter search space was defined as follows:

```
'learning_rate': uniform(0.01, 0.5),
'l2_leaf_reg': uniform(1, 10),
'max_depth': randint(4, 16),
'n_estimators': randint(100, 500),
'colsample_bylevel': uniform(0.5, 1.0)
```

On the CatBoost model, the best hyperparameters was found to be:

```
{'colsample_bylevel': 0.8872616831314765, 'l2_leaf_reg': 4.304589215644617,
'learning_rate': 0.029389425030943266, 'max_depth': 7, 'n_estimators': 174}
```

On the CatBoost-Categorical model, the best best hyperparameters was found to be:

```
{'colsample_bylevel': 0.6374640924013616, 'l2_leaf_reg': 8.927250793353691,
'learning_rate': 0.01188270397466217, 'max_depth': 7, 'n_estimators': 247}
```

Unfortunately, the number of iterations for both CatBoost models had to be reduced due to this error occurring with more iterations, thus these hyperparameters may not be optimal.

```
TerminatedWorkerError: A worker process managed by the executor was
unexpectedly terminated. This could be caused by a segmentation fault while calling
the function or by an excessive memory usage causing the Operating System to kill the
worker. The exit codes of the workers are {SIGKILL(-9), SIGKILL(-9)}
```

Additionally, despite reducing the iterations, some fits still failed due to unknown errors. Out of a total of 50 fits attempted on both models, 15 fits failed on each model, resulting in only 35 successful fits for each model. Regardless, there is still an improvement with hyperparameter tuning over the base models without tuning.

Model Evaluation

To visualize the accuracy of the model predictions, Actual vs. Predicted figures were generated for the three models (see Figures 1.1, 2.1, 3.1). A notable difference between the XGBoost model and both CatBoost models is the variance in predicted value spread. The spread width of predicted values in the XGBoost model appears narrower (by ~10 rings) in contrast to the CatBoost models (by ~15 rings). Most likely, the cause for this is due to the CatBoost models receiving 10x less iterations than the XGBoost model for hyperparameter tuning.

Another observation is the capping of predicted values at certain thresholds. In the XGBoost figure, predictions appear to be capped at 12.24 rings, while in the CatBoost and CatBoost-Categorical figures, they are capped at 18.77 and 17.21 rings respectively. Moreover, Figure 4 illustrates a significant concentration of data points, particularly those occurring more than 1000 times, within the range of 4 to 17 rings. On the other hand, the data between 1 to 3 rings and 18 to 29 rings are comparatively underrepresented. This imbalance in the distribution of data could hinder the models' ability to generalize. This could potentially be remedied by introducing more samples from the UCI dataset into the training process.

The difference in feature importance between the XGBoost and CatBoost models would likely need further investigation (see Figures 1.2, 2.2, 3.2). Shell weight had a larger impact on both CatBoost models, but strangely, sex, and specifically infant, had the largest impact on the XGBoost model. This difference, again, could hinder the models' ability to generalize.

Nonetheless, when evaluated on clean test sets, the models appear to perform moderately well as indicated in Table 1. On average, there is a marginal decrease of 0.00167 in RMSLE across the three models when comparing their private scores (using 80% of the test set) to the validation score (using 20% of the train set), potentially showing that the models did not overfit.

Tables

Table 1

Model scores between Validation, Private, and Public sets.

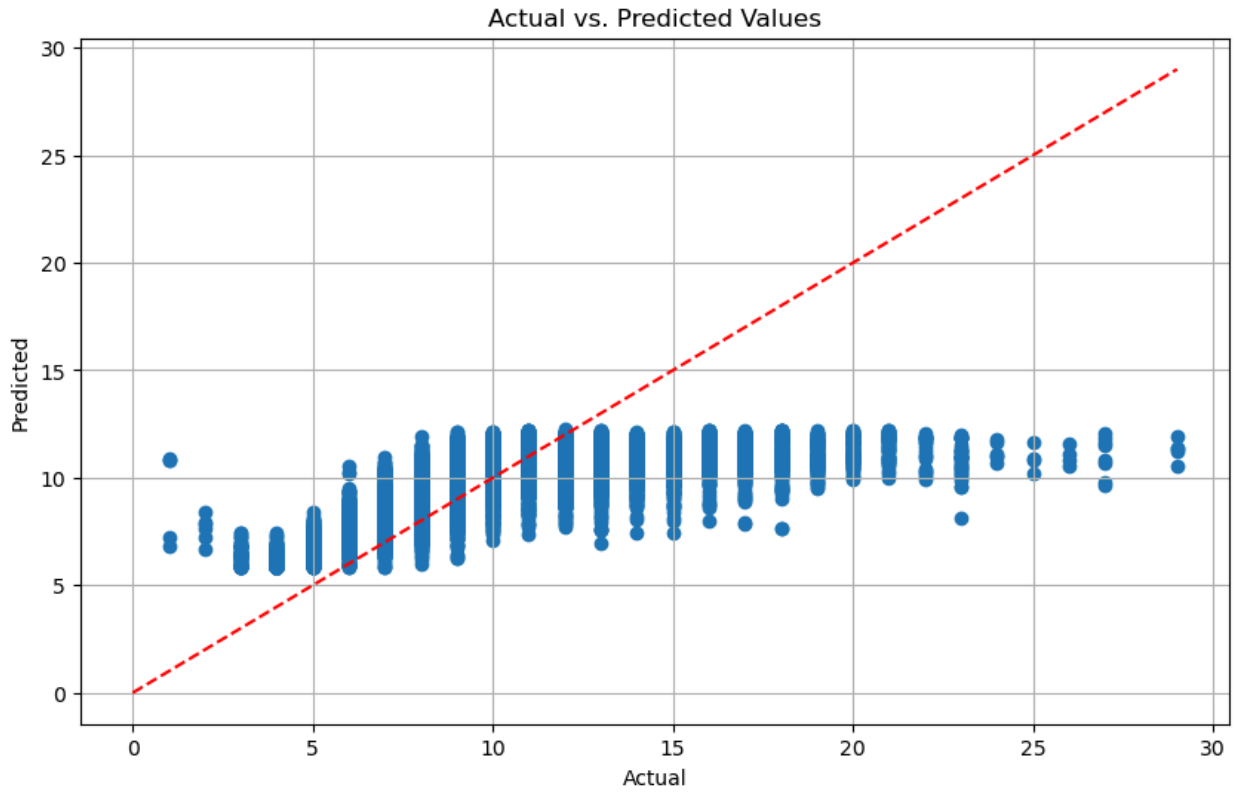
Model	Validation Score (20% of Train Set)	Public Score (20% of Test Set)	Private Score (80% of Test Set)
XGBoost	0.19991	0.19895 (-0.00096)	0.19806 (-0.00185)
CatBoost	0.15503	0.15411 (-0.00092)	0.15350 (-0.00153)
CatBoost-Categorical	0.16010	0.15906 (-0.00104)	0.15845 (-0.00165)

Note. All scores shown on this table are calculated through RMSLE.

Figures

Figure 1.1

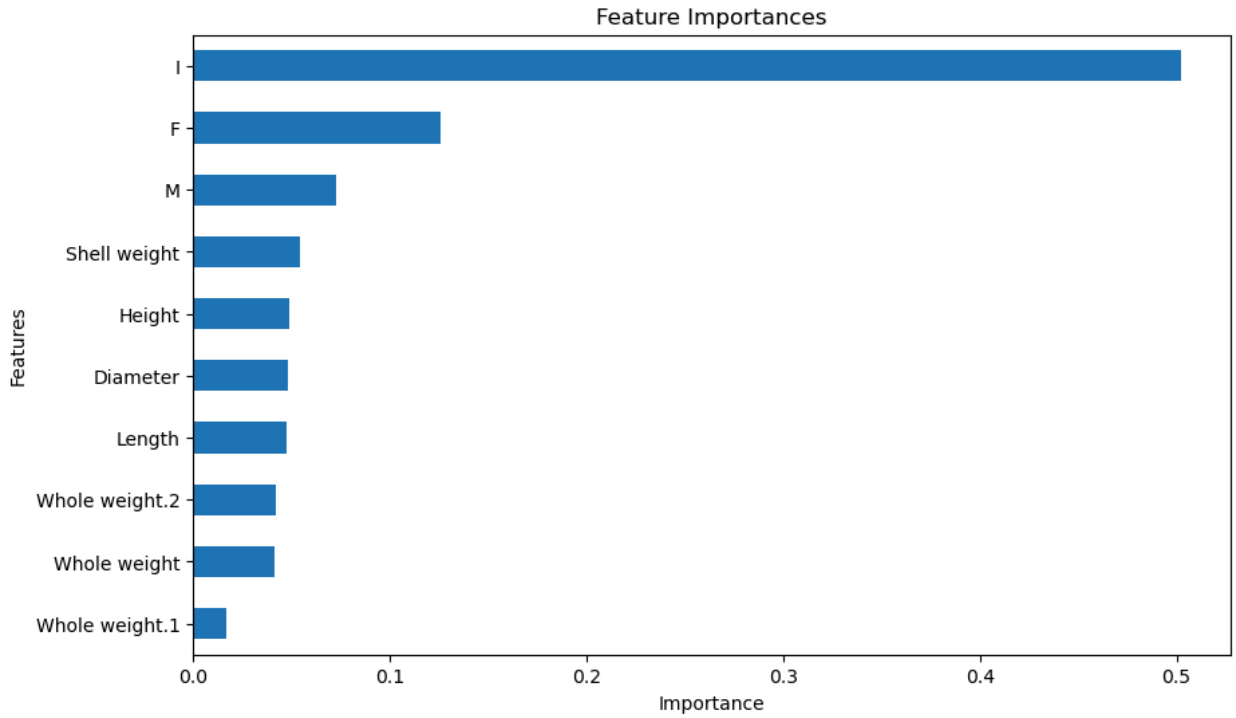
Actual vs. Predicted Values (number of rings) on XGBoost Model



Note. The actual value (red) represents the number of rings obtained from the validation set, and the predicted value (blue) represents the number of rings that the model predicted.

Figure 1.2

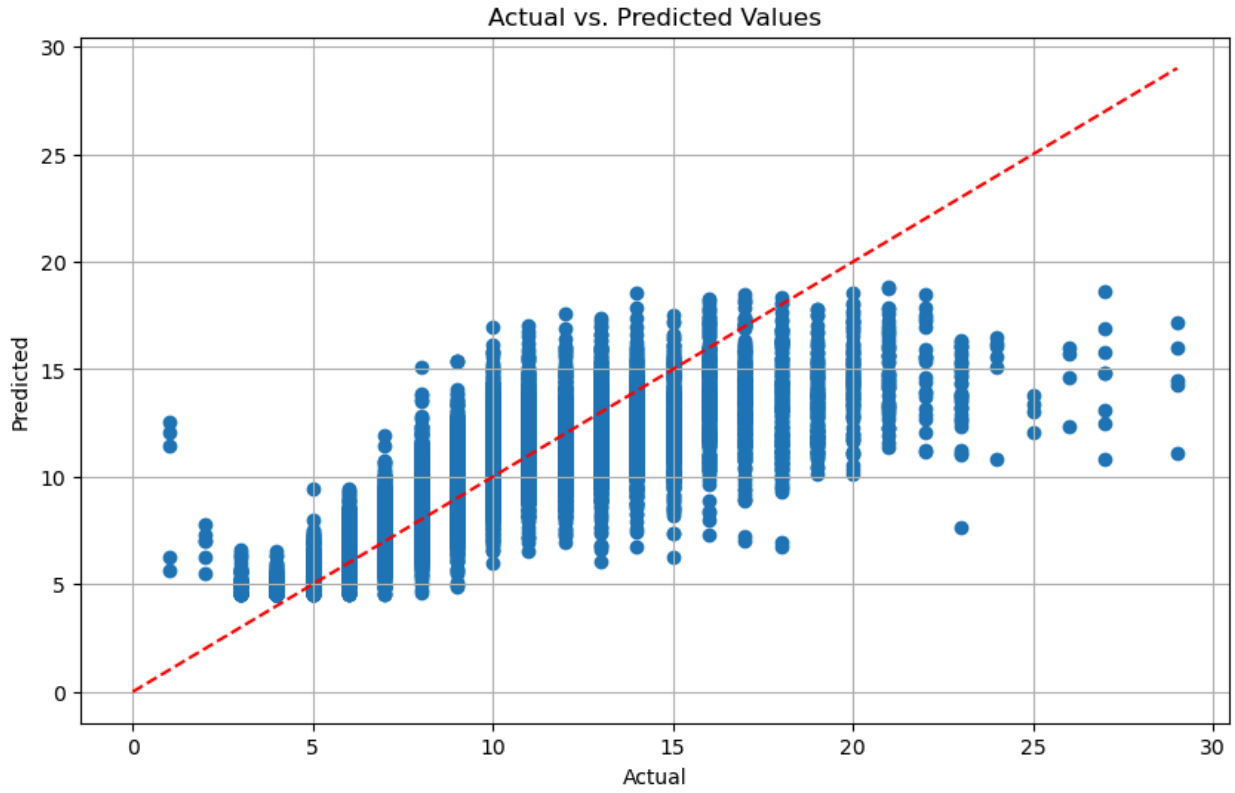
Feature Importances on XGBoost Model



Note. 'Whole weight.1' refers to 'Shucked weight', and 'Whole weight.2' refers to 'Viscera weight'.

Figure 2.1

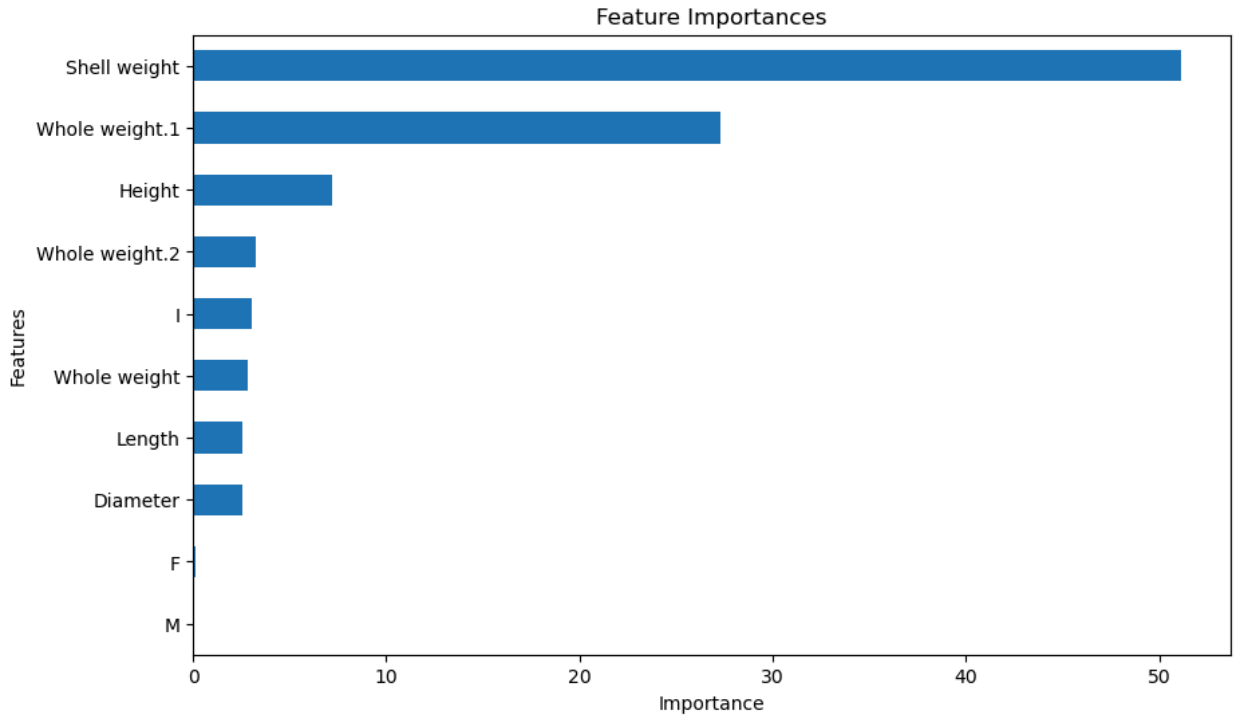
Actual vs. Predicted Values (number of rings) on CatBoost Model



Note. The actual value (red) represents the number of rings obtained from the validation set, and the predicted value (blue) represents the number of rings that the model predicted.

Figure 2.2

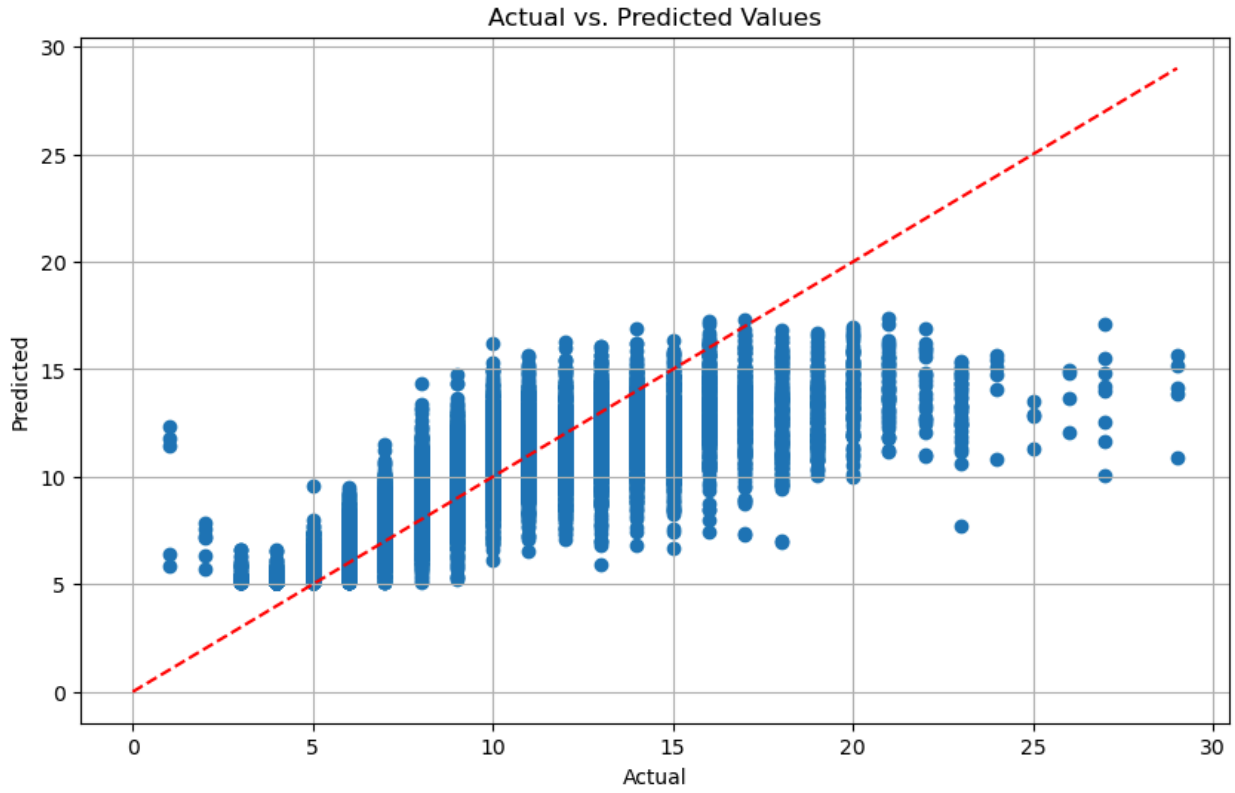
Feature Importances on CatBoost Model



Note. 'Whole weight.1' refers to 'Shucked weight', and 'Whole weight.2' refers to 'Viscera weight'.

Figure 3.1

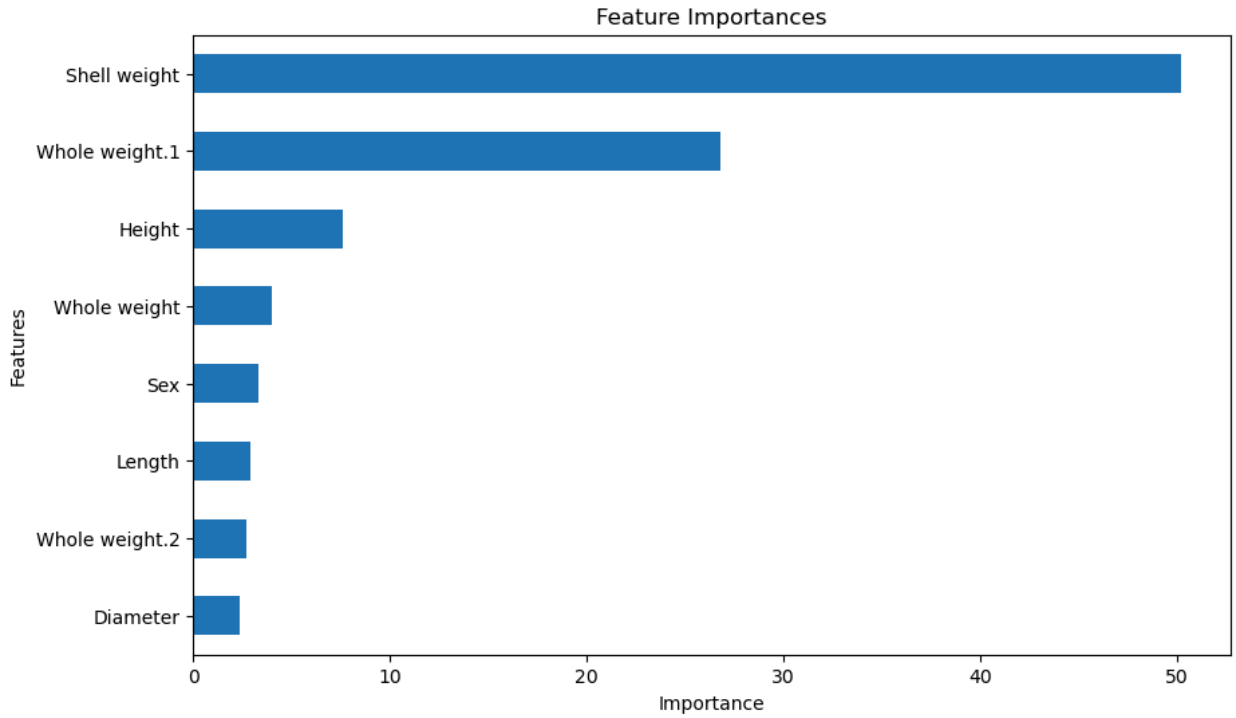
Actual vs. Predicted Values (number of rings) on CatBoost-Categorical Model



Note. The actual value (red) represents the number of rings obtained from the validation set, and the predicted value (blue) represents the number of rings that the model predicted.

Figure 3.2

Feature Importances on CatBoost-Categorical Model



Note. 'Whole weight.1' refers to 'Shucked weight', and 'Whole weight.2' refers to 'Viscera weight'.

Figure 4

Rings Distribution within the full training dataset