Callealta Andres | A01763701
August 2024

Getting abalone's age with linear regression

Introduction

In this project, I worked on codding a linear regression model to predict the number of rings in abalones, which are marine mollusks. The goal was to create a model that can make accurate predictions based on various physical measurements of abalones.

Note: the point of getting the numbers of rings is because the age in years of an abalone is equal to the number of rings plus 1.5.

Why this project?

Abalone rings are used to estimate the age of the mollusks, which is important for understanding their growth and managing fisheries. By predicting the number of rings from measurements such as length and weight, we can help in research and management of abalone populations. Otherwise, some people have to count the rings manually, which is a penible job.

Abalone Dataset Variable Descriptions:

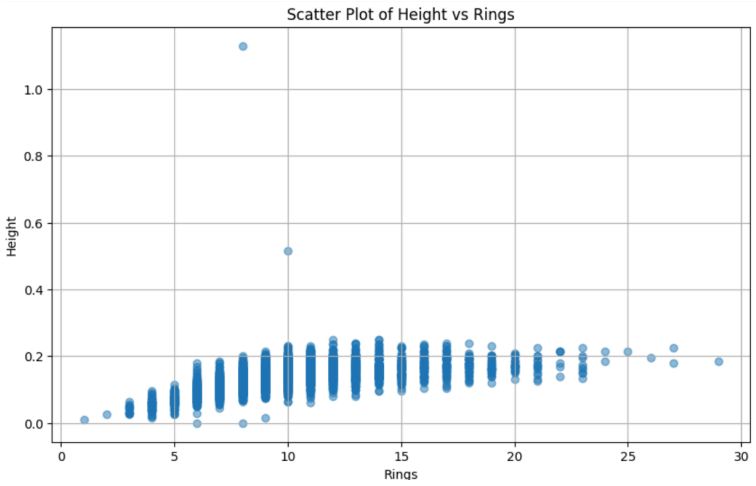
1. Sex:
    - Category: Categorical
    - Description: The gender of the abalone, which can be male (M), female (F), or infant (I). This is crucial for understanding reproductive patterns and growth differences among the genders.
2. Length:
    - Category: Continuous
    - Description: The longest shell measurement (in millimeters). It is a critical indicator of the abalone's size and growth stage.
3. Diameter:
    - Category: Continuous
    - Description: The measurement perpendicular to length (in millimeters). Together with length, it helps estimate the abalone's overall size and health.
4. Height:
    - Category: Continuous

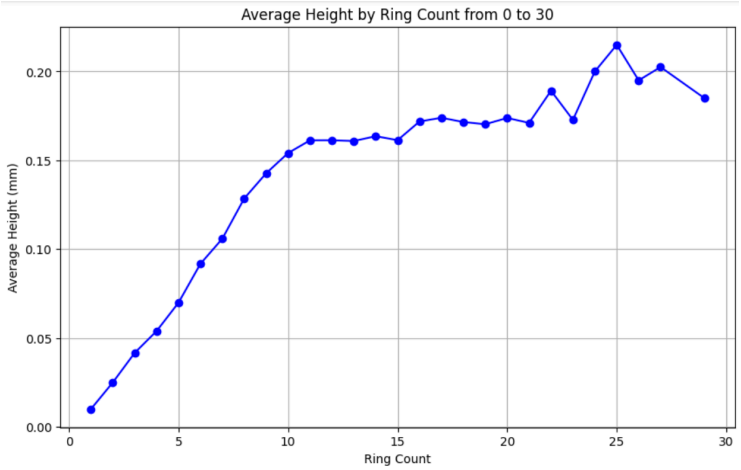
    - Description: The measurement of the abalone with meat in the shell (in millimeters). This measurement is vital for assessing the abalone's overall volume and maturity.
5. Whole Weight:
    - Category: Continuous
    - Description: The total weight of the abalone (in grams). This includes all parts of the abalone and is essential for understanding its overall health and market value.
6. Shucked Weight:
    - Category: Continuous
    - Description: The weight of the abalone's meat only (in grams). This measure is particularly important for the seafood industry to estimate yield from each shell.
7. Viscera Weight:
    - Category: Continuous
    - Description: The gut weight of the abalone (after bleeding, in grams). This is used to assess the abalone's reproductive and digestive health.
8. Shell Weight:
    - Category: Continuous
    - Description: The weight of the dried shell (in grams). This can indicate the age and the environmental conditions the abalone has been exposed to.
9. Rings:
    - Category: Integer
    - Description: The number of rings on the shell, plus 1.5 gives the age in years. This is the most direct measure of age in abalones and is critical for understanding lifecycle and harvesting appropriateness.

What We Did

1. **Data Preparation:**
    - We started with a dataset containing various measurements of abalones, such as length, diameter, and weight.
    - We removed columns that were not useful for our model, specifically "Sex" and "Height", to focus on the features most relevant for predicting the number of rings.
    - The dataset was then shuffled to ensure random distribution of data points before splitting it into training and validation sets. This helps in getting a fair evaluation of the model's performance.

      Note 1: Here is the plotting scatter of the "Height" variable. I decided to not use it because we do not see a strong correlation with the target variable.





Note 2: I also decided to delete the “Sex” variable because it is the only one that is not a continuous/ quantitative variable but a categorical one.

There are two variables that we are not using, we will check our model’s performance for this scenarios:

Including all the variables:

```
Coefficients: [ 9.91762452  0.42069191 -0.27717925  0.78129561  1.18763003  1.80504363
 -3.03063986 -0.81322021  1.97337634]
Pourcentage de prédictions correctes : 83.56%
R-SQUARED-TRAINING : 0.5423
R-SQUARED-VALIDATION : 0.4811
```

Without ‘Sex’, with ‘height’:

```
Coefficients: [ 9.91762452 -0.3561836   0.9530654   1.27333147  1.86243241 -3.1284493
 -0.71446661  1.98593365]
Pourcentage de prédictions correctes : 83.31%
R-SQUARED-TRAINING : 0.5314
R-SQUARED-VALIDATION : 0.4678
```

Without ‘height’, with ‘Sex’ (normalized):

```
Coefficients: [ 9.91762452  0.47027204 -0.04788068  1.14566052  2.00294249 -3.2433408
 -0.71480297  2.39093753]
Pourcentage de prédictions correctes : 83.73%
R-SQUARED-TRAINING : 0.5244
R-SQUARED-VALIDATION : 0.5268
```

Without ‘height’ and without ‘Sex’:

```
Coefficients: [ 9.91762452 -0.11895728  1.36966073  2.08409966 -3.37115497 -0.59546423
  2.43891169]
Pourcentage de prédictions correctes : 83.42%
R-SQUARED-TRAINING : 0.5107
R-SQUARED-VALIDATION : 0.5180
```

In conclusion, even if the differences are not blatant, our data without the ‘Height’ variable is the more performant, with a percentage of 83.73% (second one being 83.56) of good predicted values, and a validation R-squared of 0.5268 (second one being 0.5180). We also see that the difference between The R-squared training and validation is very low, this indicates low level of variance and no overfitting. Here we have 70% of the data for the training set, and 15% for validation and training, ~600 instances each one.

2. **Model development:**
    - We implemented a linear regression algorithm from scratch. Linear regression is a method used to model the relationship between a dependent variable (number of rings) and one or more independent variables (physical measurements).
    - We trained the model using gradient descent, which iteratively adjusts the model’s parameters to minimize the error in predictions.

3. **Evaluation:**

- After training, we evaluated the model's performance by measuring how well it predicted the number of rings on both the training and validation datasets.
- We calculated the R-squared value, which indicates how well the model explains the variation in the number of rings. A higher R-squared value means better performance.

4. **Visualization:**
    - We created a graph to visualize how the model's cost (a measure of prediction error) decreased over time as the model was trained. This helps us understand if the model was learning effectively

Results

- **Final Model Coefficients:** The values of the model parameters that were learned during training. Useful to understand how much a variable weighs, how much it influences the model.

Coefficients Explained:

Bias Term (9.918544775128105):

This term, also called the intercept, represents the mean value of the response variable ('Rings' in this case) when all other explanatory variables are zero. This provides a starting point for predictions.

Coefficient for feature Sex (0.49620489962872455):

This coefficient indicates that for each unit increase in the coding of the variable 'Sex', the predicted number of rings increases by 0.496. The variable 'Sex' must be numerically coded for this interpretation to be valid (e.g., using ordinal or one-hot encoding).

Coefficient for feature Length (0.3752889553640898):

For each additional millimeter in abalone length, the predicted ring count increases by 0.375. This suggests that length is a positive predictor of age.

Coefficient for feature Diameter (0.7753734969756032):

A higher coefficient for diameter relative to length suggests that diameter is a stronger predictor of age. For each additional millimeter of diameter, the predicted ring count increases by approximately 0.775.

Coefficient for feature Whole weight (0.39304623165576846):

Each additional gram of total weight increases the predicted ring count by 0.393. This indicates a positive relationship between total weight and age.

Coefficient for feature Shucked weight (-1.8509088143185144):

This coefficient is negative, meaning that the higher the flesh weight, the lower the predicted ring count. Each gram of flesh weight reduces the predicted ring count by 1.850, which could reflect a trade-off between shell growth and body mass accumulation.

Coefficient for feature Viscera weight (-0.3134693320155136):

Each gram increase in viscera weight is associated with a decrease in ring count by 0.313, potentially indicating that heavier viscera could be related to faster growth and earlier maturity.

Coefficient for feature Shell weight (2.1720635728217554):

This coefficient is the largest, showing that shell weight has the strongest positive influence on ring count. For each additional gram of shell weight, the predicted ring count increases by 2.172. This highlights the importance of the shell as an indicator of abalone age.

Cross validation:

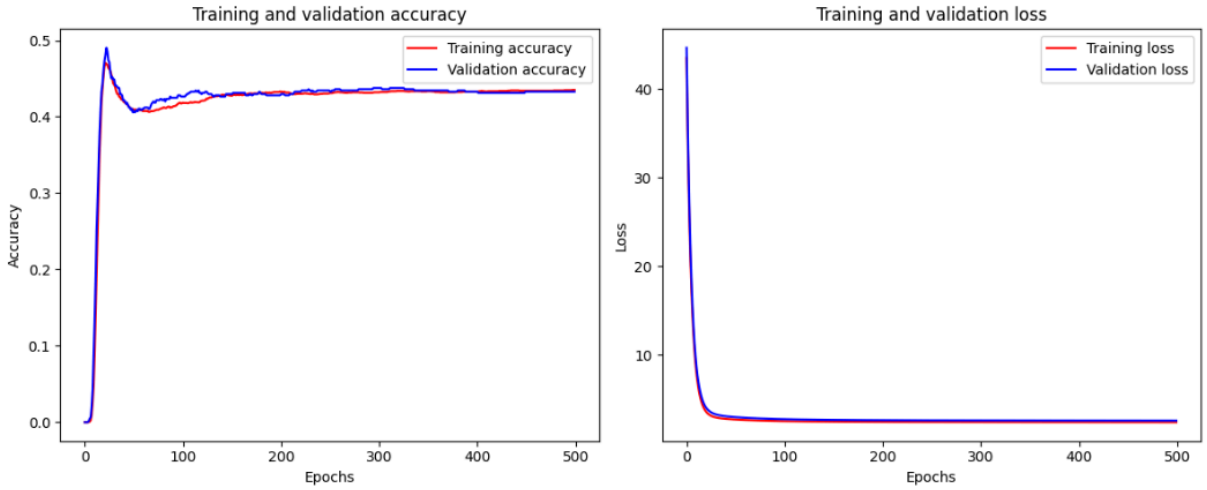
To get an idea of how well the model is going to perform with real and unseen (not in the dataset) data, I decided to use the cross validation, this method is only for evaluation, not for building. For the 4177 rows in abalone.data, we have ‘k’ equals to five, so we have 5 folds. In other words, I build 5 different models, and for each one I will check the R-squared, to see the performance of each one. This is important to first: calculate the mean of the measures to get an idea of how well the model will perform on average. And second: to get the standard deviation of the measures, and get an idea of how much the model is expected to vary in practice, with real and unknown data. Here are the results:

```
R-squared for fold 1: 0.5008
R-squared for fold 2: 0.5442
R-squared for fold 3: 0.5260
R-squared for fold 4: 0.4957
R-squared for fold 5: 0.5504

Average R-squared over 5 folds: 0.5234
Variance of R-squared over 5 folds: 0.0005
```

We observe an average R-squared value of 0.5243, indicating that 52.43% of the variation in the number of rings is explained by the variables in our model. Additionally, we notice a very low variance across the five folds, which suggests that the model's performance is consistent. The downside of this analysis is that, while we can reliably expect an R-squared around 0.5243, this value is relatively low. Therefore, we know the model is unlikely to perform significantly worse or better than this.

Visualization of model behavior:



Above are the training and validation accuracy graph (left) and the training and validation loss graph (right). For each one, the red line is for the training set, and the blue line for the validation set. What conclusions can we make with this information ?

Bias: In our case, we have accuracy peaks between 0.4 and 0.5, and the two curves (validation/test) are very close one to the other. This indicates a medium bias, meaning that the model does not really capture the patterns of the data.

Variance: The two sets curves are very close, suggesting that the model generalizes quite well. There are no significant differences between training and validation performance, indicating low variance. The R-squared test of 0.5286 is very close to the validation R-squared, confirming that the model behaves similarly on both sets: low variance.

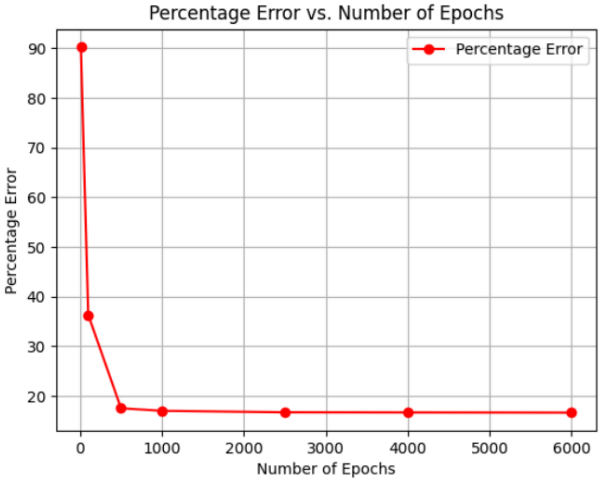
Underfitting: The model has moderate accuracy (~83%) and validation and test R-squared around 0.51, which shows that the model is in a state of slight underfitting. The model could capture more complexity in the data to improve its predictions. In this particular case, we could have one principal option to improve the model: putting more data in the training set. Solution: have a distribution of 80|10|10 instead of the actual 70|15|15.

- **Prediction Accuracy:** We calculated the percentage of correctly predicted rings and compared the model's performance on training and validation data.
    - Here with the parameters of 1000 epochs and a learning rate of 0.01 we have a percentage of correct predictions of 83.01%.
- **R-squared Values:** This metric showed how well our model's predictions matched the actual values.

  We have R-SQUARED training of 0.4809 and a R-SQUARED validation of 0.4896.

  A good model should have a R-Squared value of 1, between zero and 1 is a normal value.

- Finally, here is a graph of the progression of the errors in the predictions, related to the number of epochs. The code used for this one in *#comments*, in the same python file.

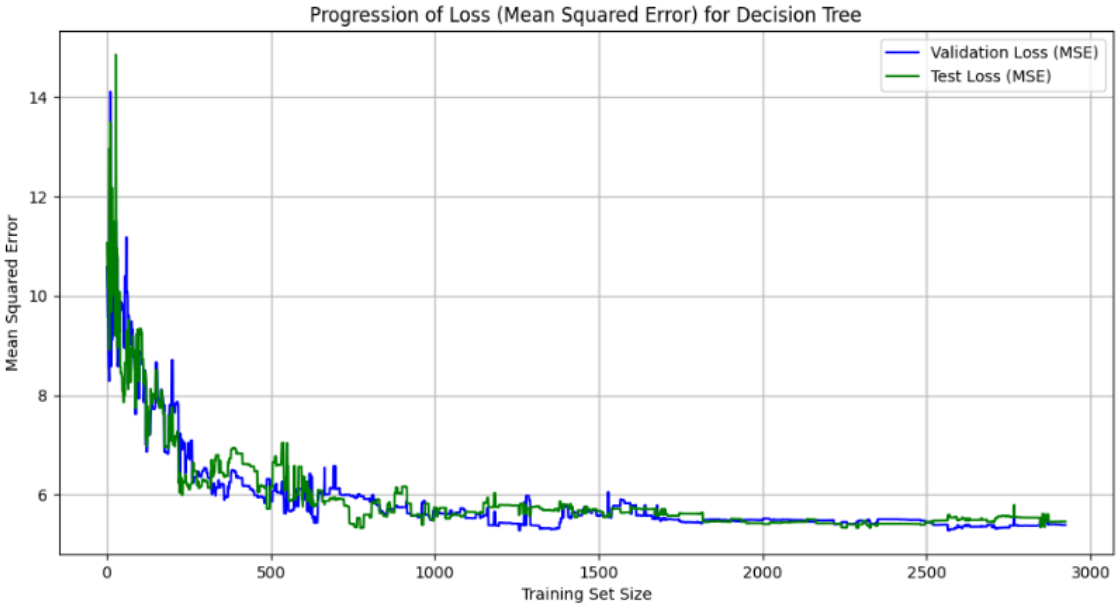


In summary, this project aimed to build a basic linear regression model to predict the number of rings in abalones, evaluate its performance, and understand its learning process through visualizations and metrics.

Decision Tree to predict the number of abalone's rings:

Why a decision tree ? I chose to implement a decision tree, to try to predict the number of rings because there was a non-linearity relation between some of the features and the target variable. And with decision trees it is easier to integrate categorical variables like 'Sex' in the dataset. Then we have also a relatively small dataset (4177 rows) and that is an important reason. Then, this type of algorithm tends to be less sensitive to outlier values, like the two ones we have.

Let's see the results of the first decision tree:




```
R-squared (validation): 0.4632
R-squared (test): 0.4688
```

On the graph we can see the progression of loss of our basic tree, with the parameters:

```
DecisionTreeRegressor(max_depth=4, random_state=42)
```

Where the other parameters are put on 'default', we are going to see that more in detail after.

We notice that our two curves, validation and test loss are pretty similar, and that the R-squared validation, and test are pretty close to each other, this indicates a low level of overfitting (or no overfitting). But the R-squared is relatively low, this can be an indicator of underfitting. For this first tree, I used a set distribution of 70|15|15, for train validation and test. Seeing that we have an indicator of underfitting we can imagine that if we have to do it better, one solution can be to add more data in our training set. We could imagine this new distribution: 80|10|10.

Now I will improve the decision tree

In the decision trees from the library sklearn.tree, that I use, we can adjust a lot of hyperparameters:

```
class sklearn.tree.DecisionTreeRegressor(*, criterion='squared_error',
splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1,
min_weight_fraction_leaf=0.0, max_features=None, random_state=None,
max_leaf_nodes=None, min_impurity_decrease=0.0, ccp_alpha=0.0,
monotonic_cst=None)                                                [source]
```

A decision tree regressor.

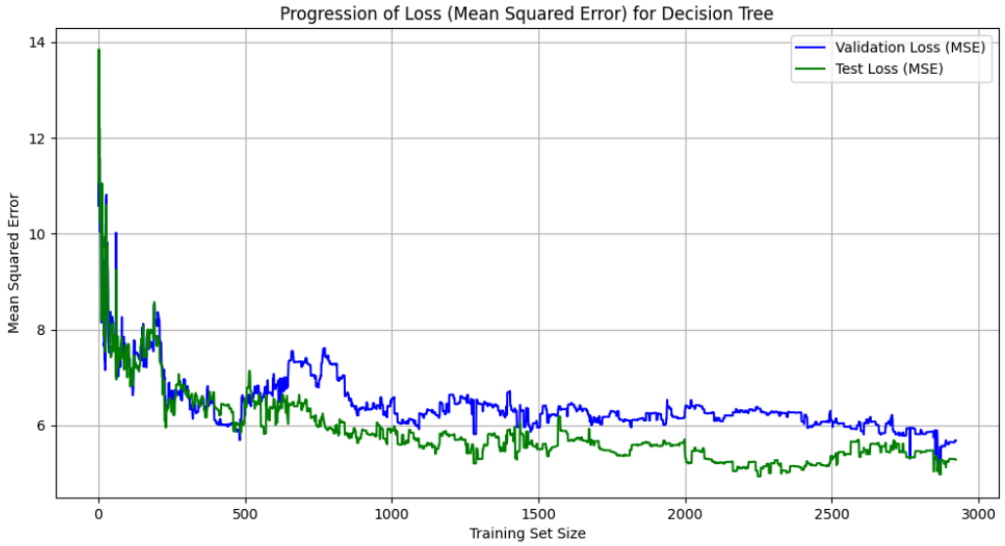
As I did not knew the meaning of all the parameters, I decided to try all of them, with the 3-4 most usual values for each one of them:

```python
parameters_to_test = {
    'criterion': ['squared_error', 'friedman_mse', 'absolute_error'],
    'splitter': ['best', 'random'],
    'max_depth': [None, 6, 7, 8],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 5],
    'max_features': [None, 'sqrt', 'log2'],
    'max_leaf_nodes': [None, 10, 20, 50],
    'min_impurity_decrease': [0.0, 0.01, 0.1]
}
```

After the 7’776 possible combination of the hyperparameters, that take approximately one hour if we generate two trees per second, the result was:

```python
best_params = {
    'criterion': 'squared_error',
    'splitter': 'best',
    'max_depth': 7,
    'min_samples_split': 2,
    'min_samples_leaf': 5,
    'max_features': None,
    'max_leaf_nodes': None,
    'min_impurity_decrease': 0.0
}
```

Let’s analyze the performance of this new tree.



```
R-squared (validation): 0.4056
R-squared (test): 0.4884
```

Our new tree is slightly better than the first one, we came from a R-squared (test) of 0.4688 to a R-squared of 0.4854, it is a significant difference, but I think we can do better. The problem here is that we have a huge gap between the validation and test r-squared. This huge difference means a high variance. It is strongly probable that we have overfitted, our model is not getting the patterns but memorizing.

What are those hyperparameters ?

- The **splitter** controls how the splits at each node are chosen. "best" means that at each node, the algorithm evaluates all possible splits and picks the one that gives the most significant reduction in error.And the other, "random", splits based on a random selection of features or potential splits.
- The maximum **depth** limits how deep the tree can grow. A deeper tree can fit more complex patterns but can also overfit the data.
- The **criterion** defines the measure used to determine the quality of a split. The "squared_error" is effective in “minimizing variance and ensuring smoother decision boundaries in regression tasks”.

Cross validation on our regression tree:

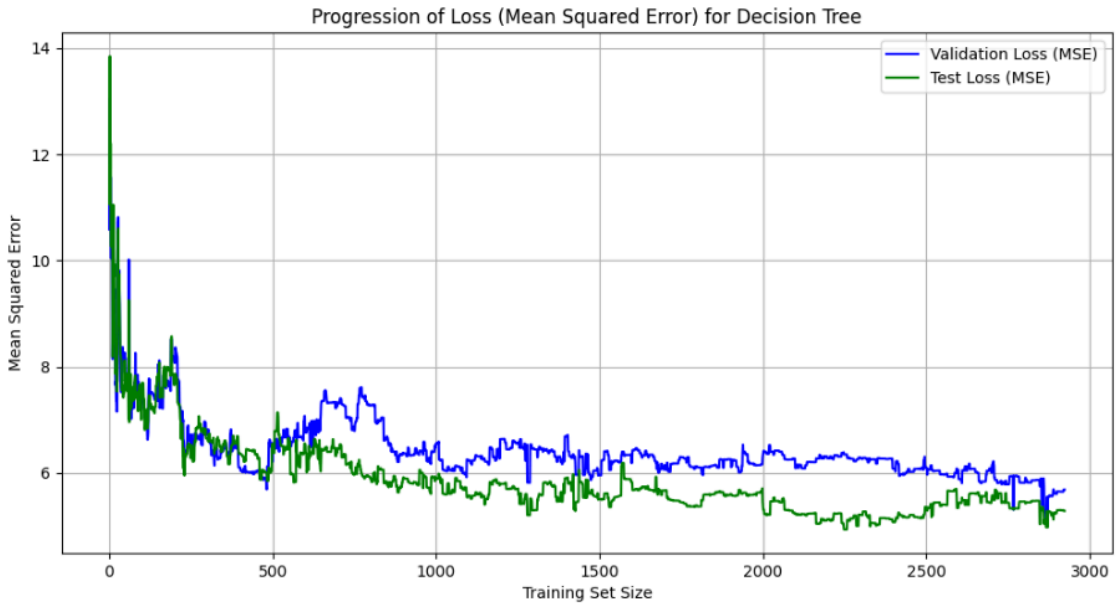
Bad news, we had a high R-squared value compared to the other folds:
(we had 0.4884)

```
MSE scores for each fold: [5.42168895 4.99516834 6.39617677 6.52017678 5.06771791]
Mean MSE: 5.6802
Variance of MSE: 0.4259
R-squared scores for each fold: [0.49916085 0.44319174 0.45023995 0.45207466 0.40737614]
Mean R-squared: 0.4504
Variance of R-squared: 0.0009
```

It is normal to sometimes have a high value because we have a variance relatively high too. Usually we do not use cross validation for modifying the model, but we can try to change the max_depth to 6 instead of 7:

```
MSE scores for each fold: [5.17662376 4.76514523 6.33253573 5.74675137 4.8949142 ]
Mean MSE: 5.3832
Variance of MSE: 0.3393
R-squared scores for each fold: [0.52179923 0.46883227 0.45570998 0.51706973 0.42758397]
Mean R-squared: 0.4782
Variance of R-squared: 0.0013
```

We see that we have a higher mean squared error, but a higher variance. This means that our model will tend to perform better with this max_depth but will be more unpredictable.




```
R-squared (validation): 0.4336
R-squared (test): 0.4854
```

As we have this big difference between the R-squared of the sets, we are going to add a new parameter that will reduce the variance.

```
Validation Mean Squared Error: 5.2291
Validation R-squared: 0.4720
Test Mean Squared Error: 5.1667
Test R-squared: 0.4797
```

Parameter added: `'ccp_alpha': 0.1`

What is it for ? -prunning

Pruning in decision trees is for removing branches that add little value or complexity to the model, to prevent overfitting. It simplifies the tree by cutting off sections that are based on noise or small variations in the training data, helping the model generalize better to new data.

Conclusion, made on the graph above:

Based on the results for R-squared (validation: 0.4720 and test: 0.4797) and the analysis of the Mean Square Error progression curves, here's a short conclusion:

1. **Bias detection**: The bias level appears to be medium. The model reaches a decent performance level, but there's room for improvement. This is evident in the gap between the validation and test curves, indicating that the model isn't fully capturing all underlying patterns in the data, maybe a tree was not the best choice.
2. **Variance detection**:The variance level is now well-balanced. While there are minor fluctuations in the validation curve, they are not significant. There are no clear signs of overfitting, as the model is consistently performing on both the validation and test sets. The nearly equal R-squared values suggest that the model is generalizing well, with only a minimal risk of overfitting.
3. Model fitt: The model is approaching a good fit but still slightly underfits due to the relatively low R-squared value. While it isn't fully capturing the data's complexity, there's no significant risk of overfitting. The minimal gap between the test and validation losses indicates that overfitting is not a concern, and the model generalizes reasonably well to unseen data.

Compliance and Ethical Considerations:

- Environmental Impact: Given that abalone data may relate to marine biology and ecology, our analysis is conducted in a manner that promotes environmental sustainability and respects biodiversity.
- Additionally, our solution significantly enhances efficiency by automating the process of counting rings in abalone shells. Traditionally, this task requires manual effort, which is not only time-consuming but also prone to human error. By implementing our automated system, we reduce the need for manual counting, thereby freeing up valuable resources and minimizing the likelihood of errors. This advancement ensures that workers can focus on more strategic tasks, enhancing productivity and job satisfaction within the marine biology field.