Machine Learning

Course Project Report

(Phase-II)

Title of the project: Abalone

Student : Ekta Sai Meda and medaekta.s-26@scds.saiuniversity.edu.in

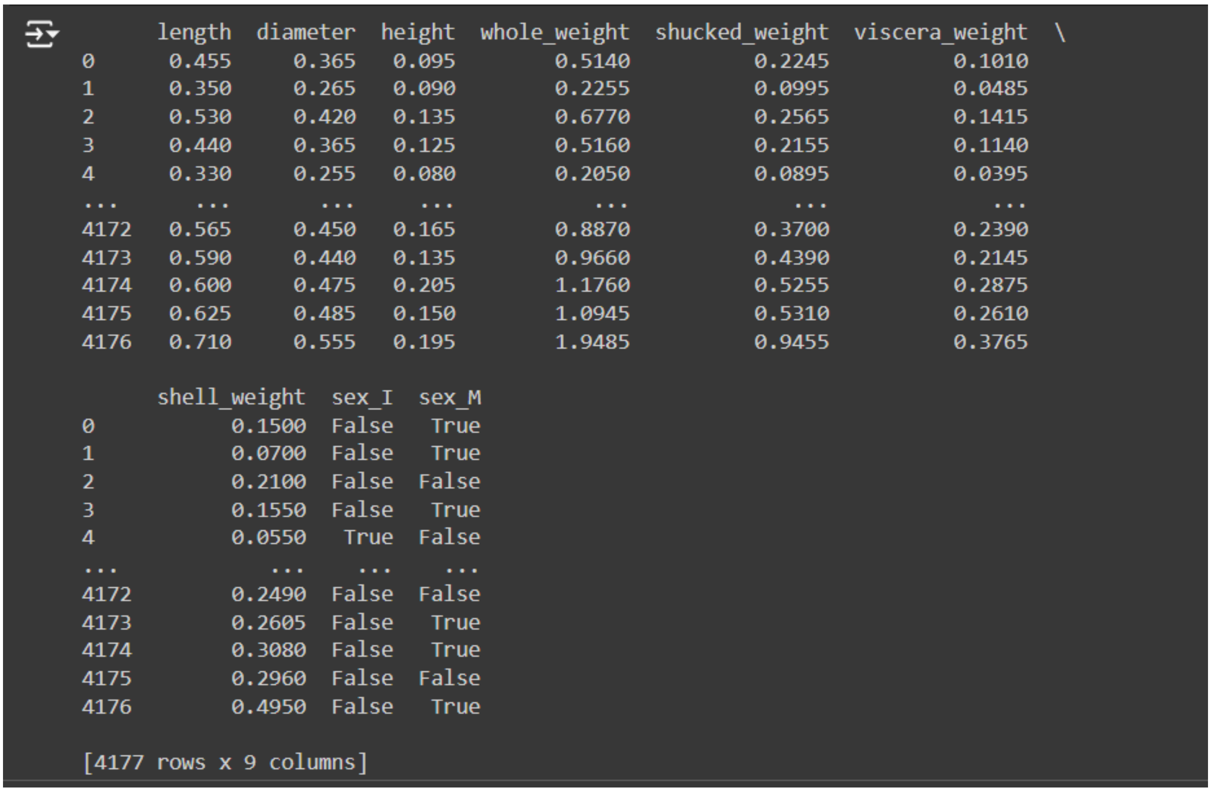
ML Category: Regression

1. Introduction

- Abalone is a marine snail inhabiting cold coastal waters, highly valued based on its age, usually determined through time-consuming laboratory procedures. In this regard, this work aims to construct predictive models for fast and accurate abalone age estimation using advanced machine learning techniques. Algorithms applied to a dataset of physical characteristics included Linear Regression, Decision Tree, Random Forest, and Support Vector Machine. Model performance was measured using R-squared with R^2 and RMSE. The proposed method is a replaceable alternative to the conventionally technique-based methods with data-driven efficient solutions, which contribute to the advancement of studies in abalone ecological research and commercial operations in the abalone industry

2. Dataset and Features

- The Abalone dataset used for this project was sourced directly from the UC Irvine Machine Learning Repository, originally published in 1995. The dataset can be found here. It contains 4,176 samples, each of which shows different physical attributes of abalones.Labels for all other eight numerical features are present in each row of the dataset and correspond to the following physical characteristics of abalone:
  - sex: Categorical feature with three subcategories—male, female, and infant. Notably, "infant" refers to the age of the abalone rather than its sex, indicating a potential limitation in the dataset.
  - length: Measured in millimetres, representing the longest shell measurement.
  - diameter : Measured in millimetres, representing the shell width perpendicular to length.
  -Height : Measured in millimetres, representing the height with meat in the shell.
  - Whole Weight : Measured in grams, representing the weight of the whole abalone.
  - Shucked Weight : Measured in grams, representing the weight of the abalone meat.
  - Viscera Weight : Measured in grams, representing the gut weight after bleeding.
  -Shell Weight : Measured in grams, representing the weight of the shell after drying.



```
     length  diameter  height  whole_weight  shucked_weight  viscera_weight  \
0     0.455     0.365   0.095        0.5140          0.2245          0.1010
1     0.350     0.265   0.090        0.2255          0.0995          0.0485
2     0.530     0.420   0.135        0.6770          0.2565          0.1415
3     0.440     0.365   0.125        0.5160          0.2155          0.1140
4     0.330     0.255   0.080        0.2050          0.0895          0.0395
...     ...       ...     ...           ...             ...             ...
4172  0.565     0.450   0.165        0.8870          0.3700          0.2390
4173  0.590     0.440   0.135        0.9660          0.4390          0.2145
4174  0.600     0.475   0.205        1.1760          0.5255          0.2875
4175  0.625     0.485   0.150        1.0945          0.5310          0.2610
4176  0.710     0.555   0.195        1.9485          0.9455          0.3765

      shell_weight  sex_I  sex_M
0           0.1500  False   True
1           0.0700  False   True
2           0.2100  False  False
3           0.1550  False   True
4           0.0550   True  False
...            ...    ...    ...
4172        0.2490  False  False
4173        0.2605  False   True
4174        0.3080  False   True
4175        0.2960  False  False
4176        0.4950  False   True

[4177 rows x 9 columns]
```

Also, the dataset contains a feature "Rings" that were hand-counted under a microscope by researchers. To determine the age of an abalone, just add 1.5 years to the number of rings. Although it was created in 1995, this dataset is still very relevant today due to the biological characteristics of abalone, as with many animals, that change very slowly over thousands to millions of years due to evolutionary processes. Thus, predictive models trained off of this dataset are still very much applicable to today's abalone populations. The data is preprocessed so that all values are numerical attributes. Categorical feature "Sex" is also encoded. The missing values will be removed and continuous values scaled by dividing by 200. Thus, the dataset is ready to be used with most of the machine learning models. It makes sure that these preprocessing steps ensure the cleanliness and normalisation of the dataset for the effective training of machine learning models used in this project.

3. Methods

3.1 Baseline - Linear Regression:

- Linear Regression is another core machine learning algorithm developed to establish the relationship between a dependent variable (target) and one or more independent variables, referred to as features. The relationship is assumed to be linear, and the model should find that best-fitting line which minimises the sum of the squared residuals of observed and predicted values. Mathematically, it can be expressed in the following form:

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + ... + \beta_p x_{ip} + \epsilon_i \text{ , i=1,2,3,4,.......n}$$

Where, y_i represents the continuous numeric response for the ith observation, β_j is the regression coefficient for the jth variable, $x_{i,j}$ shows the jth variable for the ith observation, and i is called the random error or the noise that is not able to be explained by the linear model.The parameters β are estimated using the Ordinary Least Squares (OLS) method, which minimises the residual sum of squares (RSS).

- **Mean Squared Error(MSE)**: 4.675903975878888
- **R^2 Score(R2):** 0.5454718270324019
- **Mean absolute error(MAE):** 1.5693587109234113

3.2 Support Vector Machines

- Support Vector Machines are non-parametric classifiers useful in both cases: classification and regression. SVM functions like a linear binary classifier designating the class label to test data based upon the maximum span between the two classes. It is possible to apply them on nonlinear data using kernel methods. An SVM creates a hyperplane in infinitely dimensional space. This infinite dimensional space separates classes or fits them into a line of regression. Essentially, the hyperplane will be determined by the closest training data points because it aims to maximise the distance between the two classes. This makes SVMs especially powerful models that could achieve high accuracy with a relatively small number of training examples.
- **Results:**

kernel	**MSE**	**MAE**	**MAPE**	**R2**
linear	4.788017	1.526880	0.147777	0.534574
poly	5.423970	1.547286	0.147386	0.472755
rbf	4.627016	1.480766	0.142497	0.550224

3.3 Decision Tree

- Another easy supervised learning method applicable for both classification and regression problems is the decision tree. It recursively breaks down the data into smaller subsets based on certain criteria until the final decision is made based on a voting mechanism. Decision trees are mainly classified into classification trees and regression trees. In classification trees, the output variable is discrete while, in regression trees, the output variable is continuous. Entropy and information gain are normally used to develop decision trees. At every node, the data is divided iteratively until all the leaves are pure. To avoid overfitting, a depth limit is usually imposed on the decision tree. Information gain for each attribute is calculated by using certain equations.

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2$$

$p_{i,k}$ is the ratio of class k instances among training instances in i th node

- **Mean Squared Error (MSE):** 8.807655502392345
- **R-squared (R2) Score:** 0.14383879902539387
- **Mean absolute error(MAE):** 2.04

3.4 Random Forest

- The Random Forest algorithm is one of the robust tree-based learning algorithms in machine learning. It works by creating many Decision Trees during the training process. Each tree is created on a random subset of the dataset, and at each split, features are chosen randomly. All this randomness will bring variety to the trees, hence reducing overfitting risks and increasing the accuracy of predictions. The results from all trees are combined during prediction, either by voting in the case of classification tasks or by averaging in the case of regression tasks. Once more, it is the ensemble approach that is behind the magic, this time supported by multiple trees, for achieving stable results with accuracy. Random Forests are used for classification as well as regression and, therefore, can be regarded as an efficient method factory since it is able to cope with complex data, avoid overfitting, and make reliable predictions in many applications
- **Mean Squared Error (MSE):** 4.805265550239234
- **R-squared (R2) Score:** 0.5328970435574765
- **Mean absolute error(MAE):** 1.56

3.5 AdaBoost

- AdaBoost, also called Adaptive Boosting, is one of a kind method of ensemble learning that falls under the broader family of machine learning for both classification and regression tasks. The basic idea motivating AdaBoost is iteratively training weak classifiers on the training dataset, where each subsequent classifier increases the weight of the data points that were misclassified in the previous run. The final model of AdaBoost is a combination of all the individually used weak classifiers during the training process. In such a way, weights are assigned to the models under consideration based on their accuracy. In other words, the greatest weight will be applied to the weak model with the highest accuracy, while the model with the lowest accuracy is given less weight.
- **Mean Squared Error (MSE):** 9.01497758062926
- **R-squared (R2) Score:** 0.12368574927865306
- **Mean absolute error(MAE) :** 2.60

3.6 Gradient Boosting

- Thus, gradient boosting is a form of ensemble technique in which models are built sequentially, with each new model making corrections on residual errors on the previous one. Here, gradient descent has to be used to minimise a loss function.
- **Mean Squared Error (MSE):** 4.7367260255976165
- **R-squared (R2) Score:** 0.5395595295863018
- **Mean absolute error(MAE) :** 1.54

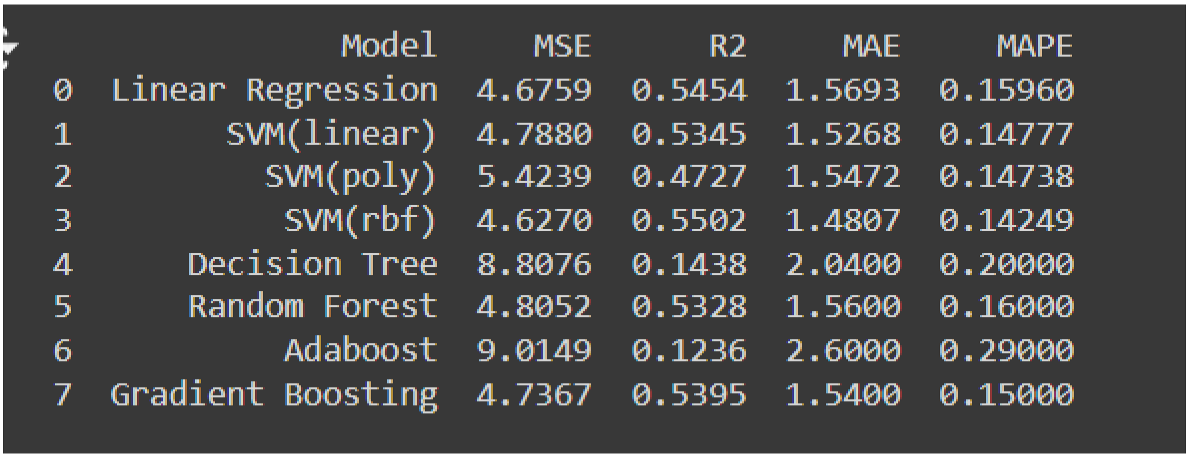
4. Results

- Among all of the methods, that of the Support Vector Machine with a Radial Basis Function kernel (SVM(rbf)) is the best for predicting the age of abalone.

  MSE: This is the average of squared differences between observed actual outcomes and predicted outcomes.

  R-squared (R2): It tells the proportion of variance for the dependent variable explained by the independent variables in a model.

  One of the metrics is the **Mean Absolute Error (MAE)** ,which takes a mean of the absolute values of all errors in the predictions without considering their directions.



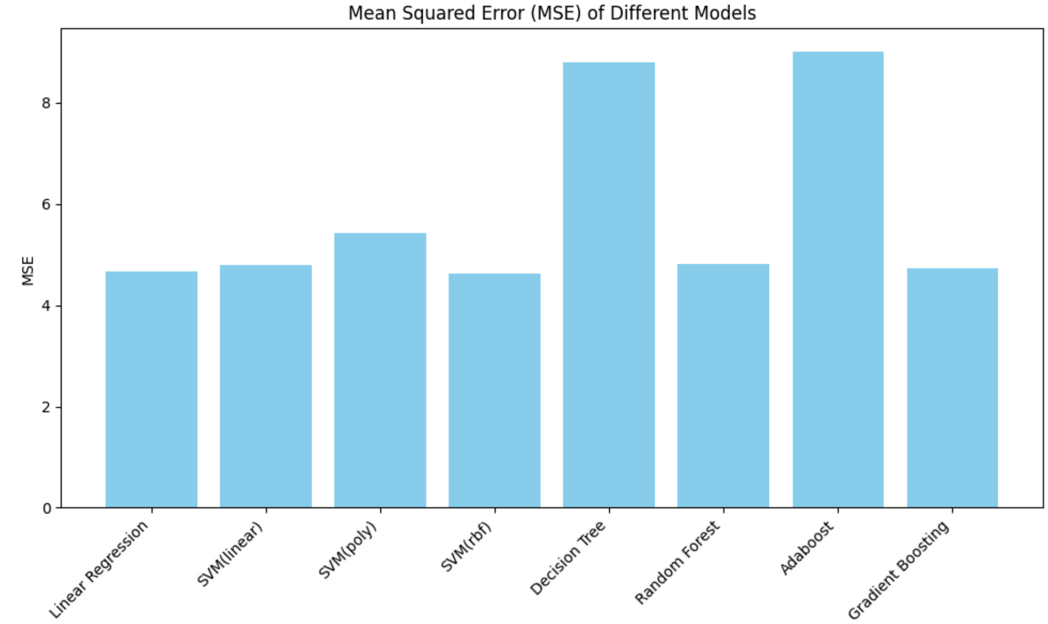
```
              Model     MSE      R2     MAE     MAPE
0  Linear Regression  4.6759  0.5454  1.5693  0.15960
1        SVM(linear)  4.7880  0.5345  1.5268  0.14777
2          SVM(poly)  5.4239  0.4727  1.5472  0.14738
3           SVM(rbf)  4.6270  0.5502  1.4807  0.14249
4      Decision Tree  8.8076  0.1438  2.0400  0.20000
5      Random Forest  4.8052  0.5328  1.5600  0.16000
6           Adaboost  9.0149  0.1236  2.6000  0.29000
7  Gradient Boosting  4.7367  0.5395  1.5400  0.15000
```

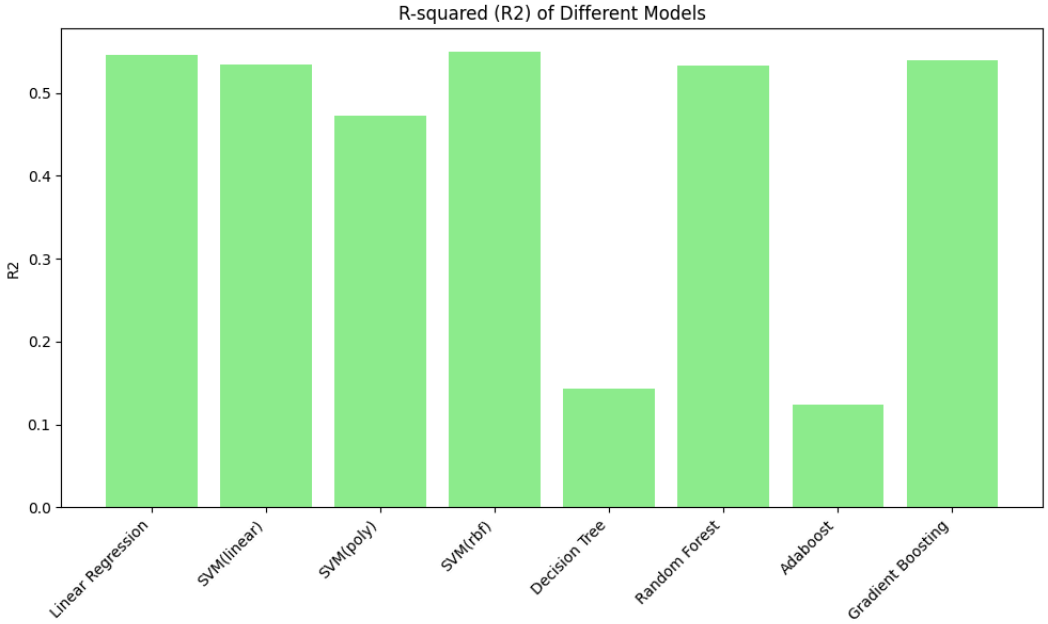
MSE: The SVM with RBF kernel has the lowest **MSE (4.6270)**, indicating it has the smallest average squared errors among the models.

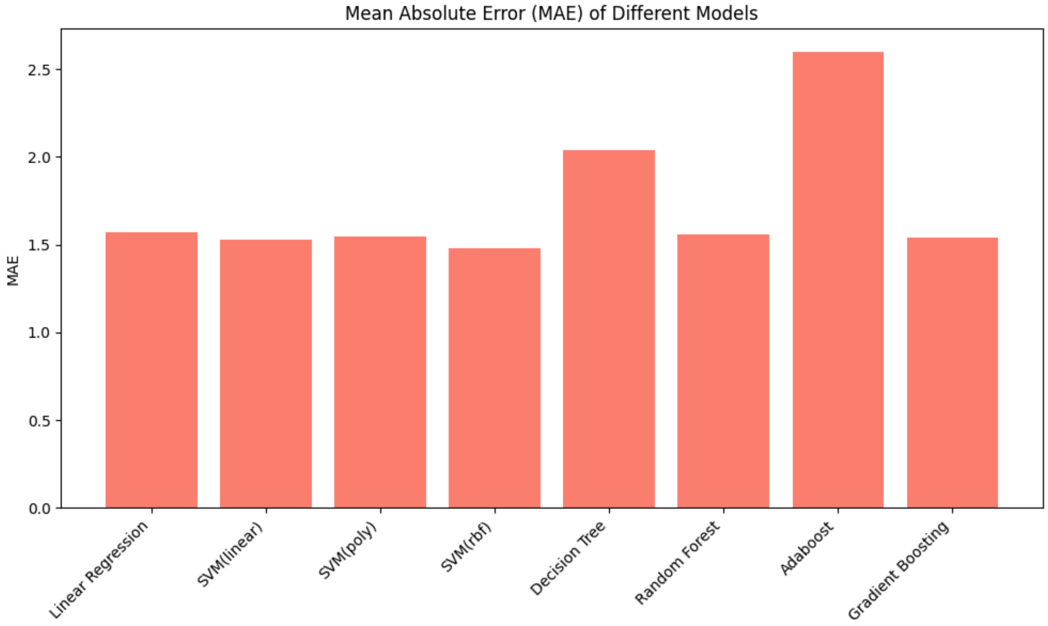
R2: The SVM with RBF kernel also has the highest **R2 (0.5502)**, indicating it explains the highest proportion of variance in the age of abalone.

MAE: The SVM with RBF kernel has the lowest **MAE (1.4807)**, indicating the smallest average absolute errors.

Considering the **MSE, R2,** and **MAE** metrics, the **SVM** with **RBF** kernel emerges as the best-performing model for predicting the age of abalone. It achieves the best balance between accuracy and error minimization, making it the most reliable method among the ones tested


Mean Squared Error (MSE) of Different Models
MSE
0
2
4
6
8
Linear Regression
SVM(linear)
SVM(poly)
SVM(rbf)
Decision Tree
Random Forest
Adaboost
Gradient Boosting


R-squared (R2) of Different Models
R2
0.0
0.1
0.2
0.3
0.4
0.5
Linear Regression
SVM(linear)
SVM(poly)
SVM(rbf)
Decision Tree
Random Forest
Adaboost
Gradient Boosting


Mean Absolute Error (MAE) of Different Models
MAE
0.0
0.5
1.0
1.5
2.0
2.5
Linear Regression
SVM(linear)
SVM(poly)
SVM(rbf)
Decision Tree
Random Forest
Adaboost
Gradient Boosting

5. Hyperparameter Tuning

Hyperparameter tuning is the process of optimising the settings that control the behaviour of a machine learning algorithm. These settings, unlike model parameters, are not learned from the data. Efficient tuning improves model performance, often requiring expert knowledge or systematic search methods like grid search, random search, or advanced optimization techniques.

5.1 SVM with RBF kernel

Explanation of the hyperparameters tuned:

C: Controls the trade-off between achieving a low error on training data and minimising the margin. Higher C values focus on classifying training data correctly.

Gamma: Defines how far the influence of a single training example reaches. Low values mean 'far,' and high values mean 'close.'

Parameter grid:

Grid search:

- **C:** A range from 1 to 9 (inclusive).
- **Gamma:** Values ranging from 0 to 0.1 (inclusive), using np.linspace(0, 0.1).

Results Obtained for the Best Configuration:

- **Best Parameters:** 'C': 9
  'gamma': 0.055102040816326547
- **Best Grid Search Score:** 0.561711353072861

Result for the 25% testing dataset:

Grid Search :

- **SVR RBF Score:** 0.5759845489697898

5.2 Decision Trees

Hyper-parameters of a Decision Tree:

- **max_depth:** This constrains the depth of the tree from becoming too deep, which could cause overfitting.
- **max_features:** Number of features to consider when looking for the best split.
- **max_leaf_nodes:** The maximum number of leaf nodes in the tree.

Parameter grid:

Grid search:

- **max_depth:** A range from 1 to 20
- **max_features**: range from 1 to 8
- **max_leaf_nodes:**None, 10, 20, 30, 40, 50

Results Obtained for the Best Configuration:

- **Best Parameters:** max_depth=7, max_features= 3, max_leaf_nodes= 40
- **Best Grid Search Score:** 0.48016460355736346

Result for the 25% testing dataset:

Grid Search :

- **Decision Trees Score:** 0.45304304343514323

5.3 Random Forest

Hyper-parameters of a Random Forest:

- **n_estimators:** Number of trees in the forest.
- **max_depth:** Maximum depth of each tree to prevent overfitting.

Parameter grid:

Grid search:

- **max_depth:** A range from 0 to 10
- **n_estimators:** A range from 10 to 200 in steps of 10

Results Obtained for the Best Configuration:

- **Best Parameters:** max_depth=9
- **Best Grid Search Score:** 0.5496856459913332

Result for the 25% testing dataset:

Grid Search :

- **Random Forest Score:** 0.543162589973887

5.4 AdaBoost

Hyper-parameters of a AdaBoost:

- **learning_rate:** Weight applied to each classifier at each boosting iteration.
- **n_estimators:** Number of boosting stages to be run

Parameter grid:

Grid search:

- **learning_rate:** A range from 0.1 to 1 in steps of 10. (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.)
- **n_estimators:** A range from 10 to 101 in steps of 10 (10, 20, 30, 40, 50, 60, 70, 80, 90, 100)

Results Obtained for the Best Configuration:

- **Best Parameters:** learning_rate=0.30000000000000004, n_estimators=10
- **Best Grid Search Score:** 0.4445099866824263

Result for the 25% testing dataset:

Grid Search :

- **AdaBoost Score:** 0.46456015751092794

5.5 Gradient Boosting

Hyper-parameters of a Gradient Boosting:

- **max_depth:** Maximum depth of each tree to prevent overfitting.
- **n_estimators:** Number of boosting stages (trees) to be run.

Parameter grid:

Grid search:

- **max_depth:** A range from 1 to 10. (1,2,3,4,5,6,7,8,9)
- **n_estimators:** A range from 1 to 100 in steps of 10 (1, 11, 21, 31, 41, 51, 61, 71, 81, 91)

Results Obtained for the Best Configuration:

- **Best Parameters:** max_depth=4, n_estimators=61
- **Best Grid Search Score:** 0.5584925376653727

Result for the 25% testing dataset:

Grid Search :

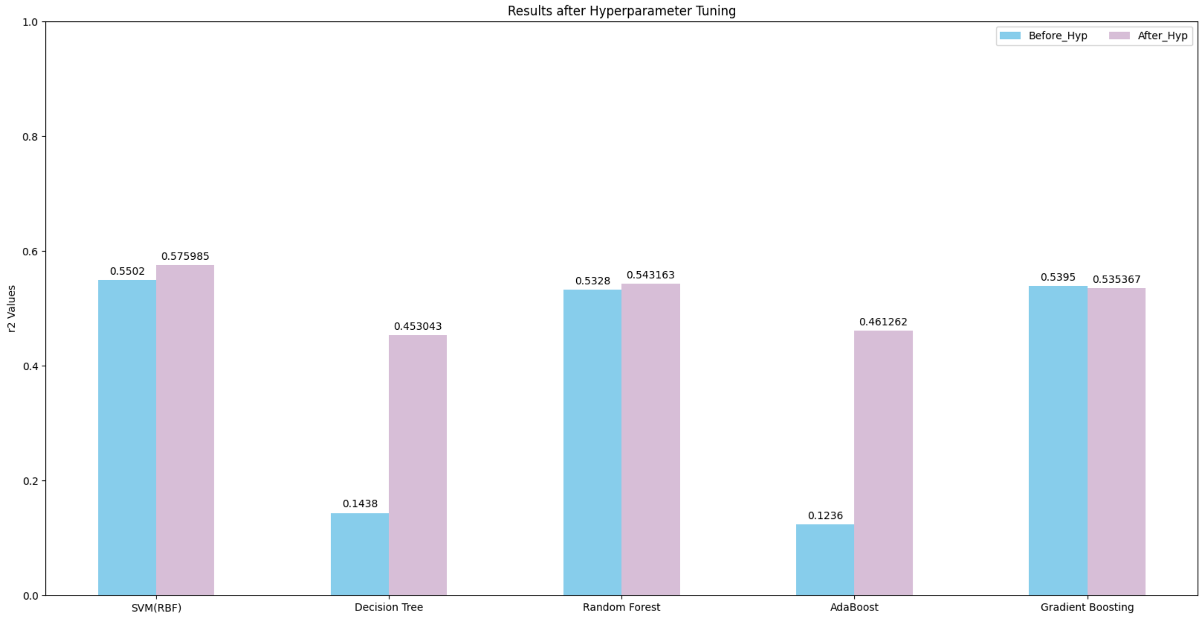
- **Gradient Boosting Score:** 0.5358646814487781

6. Results after hyperparameter tuning

Table:

	Methods	**Before_hyp**	**After_hyp**
0	SVM(RBF)	0.5502	0.57595
1	Decision Tree	0.1438	0.453043
2	Random Forest	0.5328	0.543163
3	AdaBoost	0.1236	0.461262
4	Gradient Boosting	0.5395	0.535367

Graph:



7. Feature Reduction

- Principal Component Analysis is the process of dimensionality reduction: it projects data onto a space with lower dimensions such that maximum variance is retained. If PCA is applied to the standardised training and test data, principal components are obtained from the data. One can transform the feature sets into three principal components which help to enhance the model—say, SVM with the RBF kernel—by focusing on the important features to avoid the burden of computational power. The R^2 score for this result shows the correctness of the predictions on unseen data.

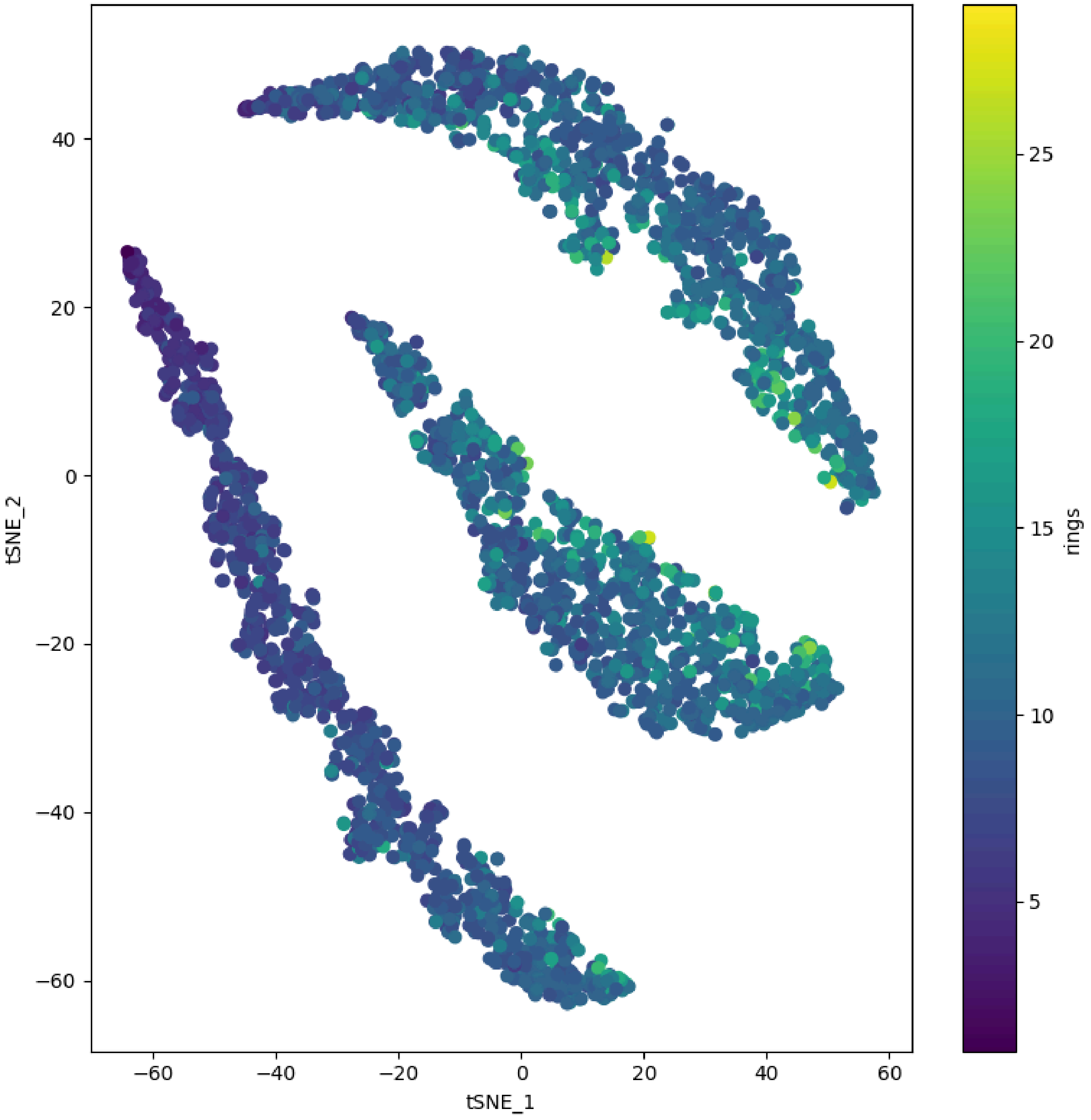
Score after feature reduction: 0.5698455708703736

8. Feature Selection

- First, feature selection will be done using the ranking of features by their f_regression score, keeping the top 50 percent of features most related to y_train. The selected features then transform the training dataset, X_train_fs and the test dataset. After that, it trains a Support Vector Regressor with an RBF kernel, which was found as being the best model after hyperparameter tuning, on the transformed training data for the prediction of y_train. Measured on the transformed test data, its performance gives an R^2 score, which is a measure for the predictive accuracy on unseen data. This approach will simplify this model by looking at only the most important features, therefore enhancing predictive accuracy and computational efficiency.

Score after feature selection:0.4589022715343374

9. Data Visualization



10. Conclusion

Finally, after extended hyperparameter tuning, an SVM model with an RBF kernel turned out to be the most efficient way to predict abalone age based on the dataset. This conclusion is further justified by the fact that careful tuning of parameters such as gamma and C has accounted for a good improvement in model performance, as it was through these parameters that the complex relationship that characterises the data was effectively dealt with to improve predictive accuracy. It is not only the feature selection with SelectPercentile and the dimensionality reduction with PCA, but also the tuning of hyperparameters that made the models computationally efficient and more interpretable by focusing only on the most relevant features.

The t-SNE visualisation helped to provide a deeper understanding of the structure of the data and validated the features used and the models applied for the task. The comprehensive analysis proves that proper tuning of the hyperparameters and better feature engineering methods are extremely necessary for building robust and efficient predictive models. Finally, from the analysis, it has been evident that the SVM with RBF kernel is the best model to predict abalone age, since this predictor is best performed in terms of reliability, efficiency, and adaptability to characteristics of the dataset.