Semester Project Final Report

CX 4240: Introduction to Computational Data Analysis

**Age of Abalone**

Group Members:

Yves Esslinger

Amanda Gregory

Zhengyang Qi

Claire Xiao

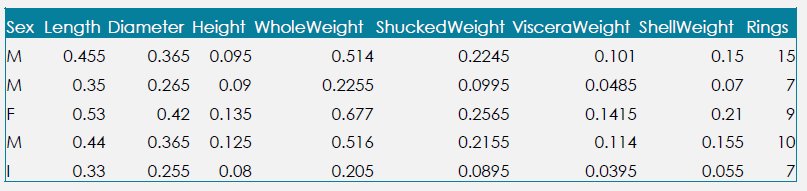
**Introduction**

Abalone is a common name for shellfish such as molluscs or marine snails, that can range from 20 millimeters to up to 200 millimeters. Marine biologists that study the age of these sea creatures must go through an extensive process of cutting the shell, staining it, and counting the large number of rings on the inside using a microscope. This process is very tedious, time consuming, and frankly “boring” according to researchers at the Marine Resources Division in Australia. There is so much more that these biologists could be doing with that time, and we are aiming to provide an efficient method of determining the age of abalone using data analytics.

Our main goal is to create a model with improved accuracy that predicts the age of abalone based on several physical variables and predictors. The overarching hope is to simplify the arduous process for determining the ages of these molluscs. There are some positive impacts that our efforts could have on the fishing industry: since there is a positive correlation between the age of an abalone and its monetary value, this information could prove useful to researchers and competitors in the fishing industry. It could also relieve some of the tedious work for these marine researchers. We chose this problem as it poses a challenge on creating a model with high accuracy. Furthermore, the dataset has been used in the past in two research papers in the 1990s with published accuracies, giving us a benchmark. Thus, we wished to apply our own skills and methods to attempt to surpass the previous models while also using newer methods that may more appropriately fit the data. We will delve into analyzing the data, summarizing previous research, and finally applying different machine learning models for comparison and analysis.

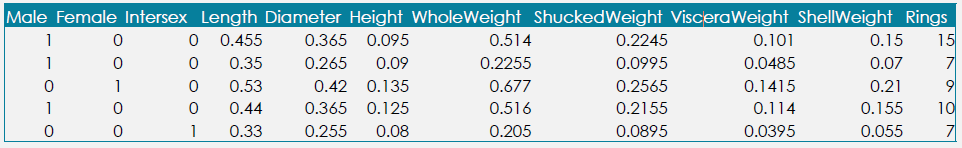
**Data**

We began our research into this problem by thoroughly examining the data, which involved some preprocessing. The dataset in question was conducted by a team of scientists in Australia in 1994 studying the population biology of abalone from the North Coast and islands of Bass Strait. It consists of 4177 unique rows with 8 features. There are 7 continuous variables and 1 categorical variable.



*First 5 rows of data from raw dataset*

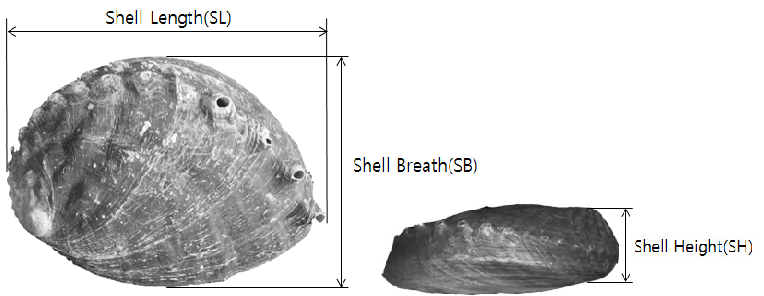
Above, the right-most column titled “Rings” is the output variable. For model purposes, we transformed the categorical variable into a binary variable using one-hot encoding.



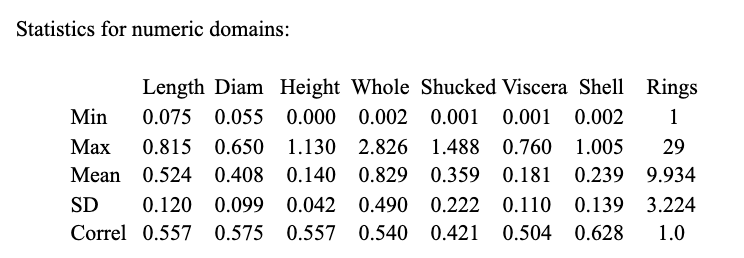
*First 5 rows of data after applying one-hot encoding*

We then performed data analysis to improve our knowledge on the dataset. We found only two significant outliers and removed these two rows. With these two points removed, the data appeared to be clean and had no other issues that could affect our model and results.

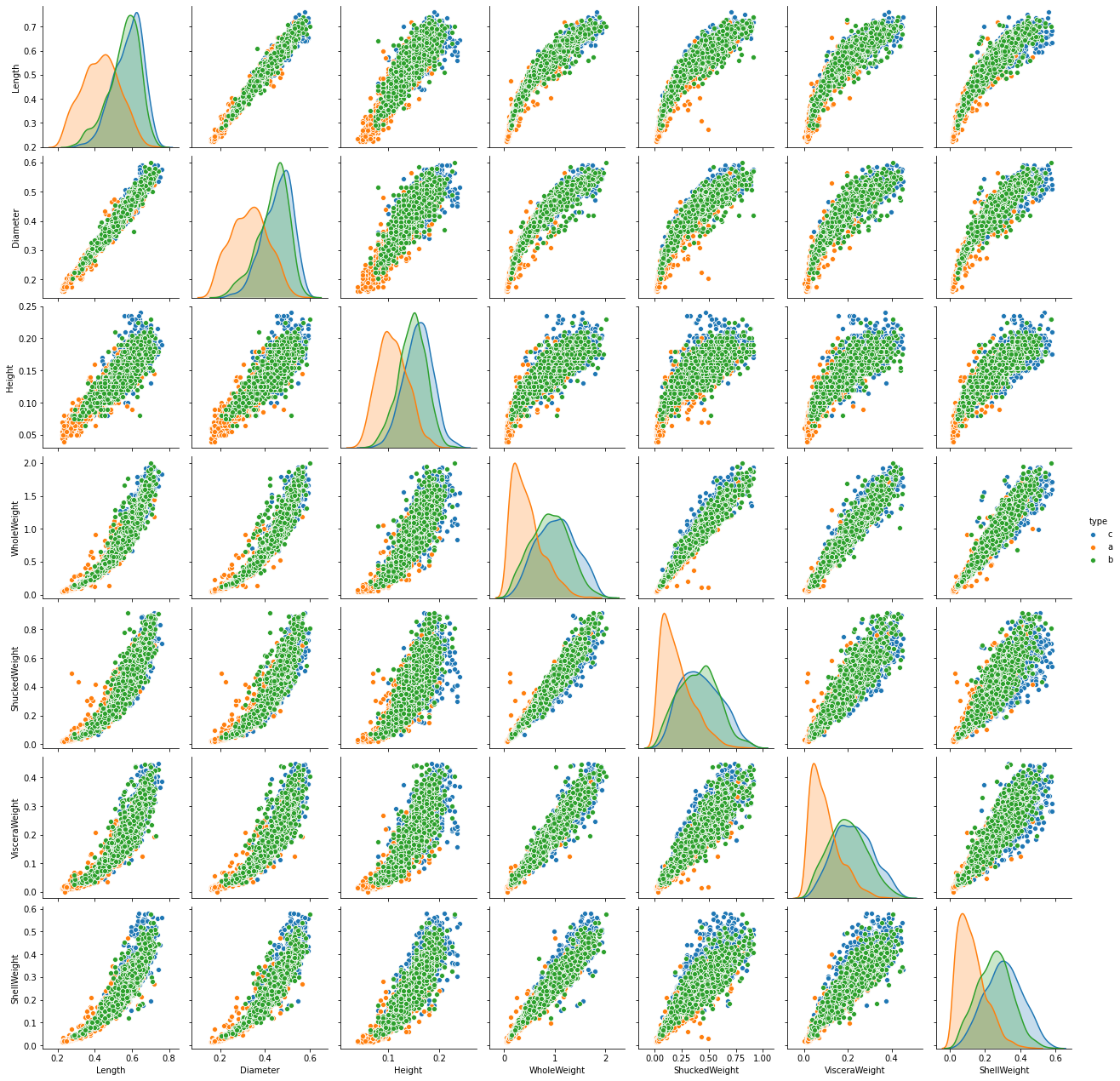
Analyzing the nature of the variables in our dataset is crucial to understanding how our output variable is predicted using the several predictor variables. Our first variable is Sex, a categorical variable defined by the abalone being either Male, Female, or Infantile. As mentioned previously, the Sex variable was converted into 3 different binary variables. The second variable Length, a continuous variable, is defined as the longest shell measurement on an abalone in millimeters. The third variable, Diameter, also a continuous variable, is defined as the longest shell measurement perpendicular to length in millimeters. The fourth variable Height, another continuous variable, is the measurement of the height of an abalone when the meat is still in the shell. Below is a figure that displays an example of abalone Length, Diameter, and Height.

*Diagram Displaying Length (SL), Diameter (SB) and Height (SH) of an Abalone*

Continuing to examine the variables included in our model dataset, the fifth variable is Whole Weight. Whole weight is a continuous variable that is defined as the entire weight of the abalone in grams. Viscera Weight, the sixth variable, is determined by the gut weight of the abalone after bleeding measured in grams. Our seventh variable used in the model is shell weight, another continuous variable that is determined by the weight of the abalone shell after being dried, measured in grams. Finally, the variable that our model is trying to predict is Rings. Rings is defined as the number of rings in an abalone shell after being stained. Rings is the output variable that our model is trying to predict due to its direct correlation with age. The age of an abalone in years is determined by adding 1.5 to the number of rings. The figure below displays the numerical statistics for the 8 continuous variables previously discussed (gender was omitted).

*Numerical statistics for the 8 continuous variables utilized in the model* 

We created a pair plot of the independent variables (“sex” omitted) to analyze feature correlation. The pair plot (pictured below) displays histograms of each variable along the diagonals, and all of the other plots display interaction between each variable with one another. As seen along the diagonals, there are high levels of overlap between all 3 classes of abalone age. The different classes were defined as 1-8 rings (Orange), 1-9 rings (Green) , and 11+ rings (Blue). We chose these three classes as this 3-category classification is roughly equal in size and was also the grouping used in previous studies. This significant level of overlap shows that obtaining a high level of accuracy in our model could be initially difficult due to the similarity in the values of the predictor variables. As shown in the plot, all interactions between variables indicate that the variables are positively correlated with one another. A notable mention on the pair plot is that Rings, the output variable that the model utilizes to predict age, is positively correlated with each of the indication variables. This shows that abalone are likely to be older as the value of each of the predictor variables increases.

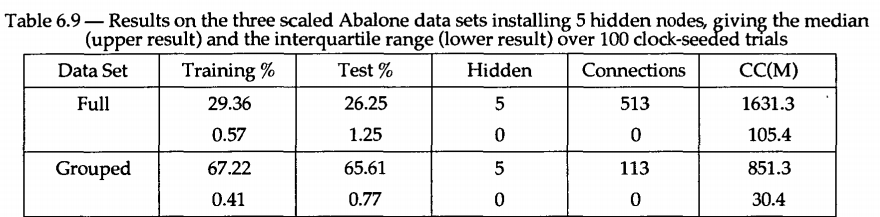


*Pair Plot of data separated by class: 1-9 rings (Orange), 9-10 rings (Green), 11+ rings (Blue)*

**Previous Work**

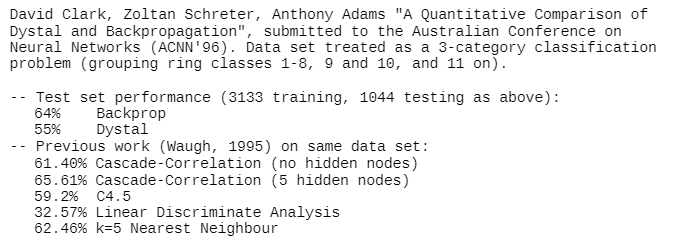
The two previous works on this data set used the data to demonstrate new methods developed in the 1990s and their effectiveness on multivariate datasets.

In 1995, Sam Waugh published his PhD thesis titled “Extending and benchmarking Cascade-Correlation” and applied a cascade-correlation model to the data. This is a seldom-used learning architecture that begins with a minimal network and then automatically trains and adds new hidden units one by one to form a multi-layer structure. Waugh was able to reach a maximum of 65.61% accuracy using a grouped classification of 3 groups of even size: class 1-8, class 9 and 10, and class 11+ rings. While his full dataset model only reached 26.25% test accuracy, from both these results Waugh concluded that the dataset may have overlapping features and that it may be possible to get accurate estimates of the number of rings from the given attributes but exact matching is not possible.



*Table of results from Sam Waugh’s PhD Thesis demonstrating the model results*

The second past published use of the dataset was by David Clark, Zoltan Schreter, and Anthony Adams titled "A Quantitative Comparison of Dystal and Backpropagation''. These researchers used two models: dystal and backpropagation. Similar to Waugh, they grouped the output variable into the same 3 groups of even size. These researchers were able to hit a maximum of 64% accuracy. Only a small excerpt of the research paper is available online and no online copy of the report appears to be accessible, so we were not able to delve into further analysis on the models used and the conclusions from these researchers.



*Results from second study demonstrating the model(s) results*

Both of these previous published works are over 20 years old. Many advancements have been made in machine learning and we believe these past works serve as a benchmark for our project and allow us to competitively compare our results. From these works, we can infer that grouping models will have much higher accuracy than keeping the output as a discrete value. It is important to note that with our goal to accurately predict the age of abalone, using a 3-category classification model will sacrifice specific age prediction for improved model accuracy. With that in mind, we will aim to create a model that can accurately classify Abalone ring/age while also outperforming the two previous models above.

**General Approach**

In addition to their approaches, we are also interested in framing the task as a regression, where we treat the number of rings as a continuous variable. Although the number of rings are discrete, most of the other variables are continuous, so we think it may result in a better performance than a classification task if we use regression models to fit the dataset. Furthermore, since the number of rings is equal to the age of an abalone plus 1.5, it is okay for our “ring” output variable to be a decimal. Thus, we have framed 3 tasks:

1. Regression: We will use different regression methods to predict the number of rings given all other information, while treating the number of rings as a continuous variable, and the predicted result will be rounded in practice.
2. One-to-One Classification: We will use Waugh’s approaches to treat each number of rings as a distinct class and learn different classification models and predict the exact number of rings on an abalone shell.
3. Grouped Classification: We will use Clark’s method to group the number of rings into 3 categories and train classification models that predict which of the 3 categories an abalone belongs to. The categories will be the same as Clark’s grouping method: class 1 will be number of rings ranging from 1 to 8, class 2 will be number of rings ranging from 9 to 10, and class 3 will be number of rings ranging from 11 to 29, which is the maximum number of rings in the dataset.

**Models Used**

We decided to use different models for each task. For the regression task, we used different linear regression techniques, including Lasso regression and Ridge Regression, and we also used different regressors based on other machine learning models, including Random Forest Regressor, Multi-Layer Perceptron regressor, and Support Vector Regressor. For both of the classification tasks, we used Random Forest Classifier, Multi-layer Perceptron Classifier, and Support Vector Classifier accordingly with a purpose of offering a comparison for the performance of each model on the regression task and the classification task, while we did not use linear regression because it is not applicable to classification tasks.

The reason for our choice of models is that we want to see how the vastly different genres of machine learning models perform on each of the 3 tasks. Linear regression is probably one of the oldest and most interpretable machine learning models with a strong assumption of the linear relationship between the dependent variable and the independent variables, while the different modification to linear regression, such as Lasso and Ridge regression, helps with regularizing the possible overfitted model and facilitate the performance on the test set. Support vector machine is also a vastly used and highly interpretable model that assumes linear separability of the data points in a certain dimensional space, which makes the choice of parameters important in terms of finding the most fitting model.

Random forest is an ensemble model that creates multiple decision trees on randomly chosen features and combines the output of each tree as the final output. Compared to the previous two models, random forest is a strong model that may have better performances yet is also more subjective to overfitting, while it is not as interpretable as the previous two models as well because of its randomness. We also employed a multilayer perceptron model in all three tasks, which represents deep learning models implemented on neural networks. Neural networks usually have better performance with a large amount of training data, while they are much more subjective to overfitting and are much less interpretable than the previous models. By including these different models in our project, we want to examine which model performs the best on the dataset and see if there is a rather significant trade-off between interpretability and performance. We are also interested in whether our models are able to perform better than the previous works in the 1990's.

**Fine-Tuning**

In terms of implementation, We used the Scikit Learn library and Google Colab as a platform for training, testing, and fine-tuning each model. Our training and testing split is the same as the two previous works: we take the first 3313 data points for training and the rest of 1044 data points for testing. Our independent variables are all continuous except for the variable “sex”, which is a catagorical variable with 3 possible categories: “Male”, “Female”, and “Intersex”. Thus, we created 3 dummy variables for each of the 3 categories to replace the original “sex” variable while keeping other variables as its original form, and for each of the 3 tasks we used this set of independent variables.

For the dependent variable, which is the number of rings, we prepared it differently for the 3 tasks. For task 1, we converted it to continuous; for task 2, we created 29 classes corresponding to the number of rings; for task 3, we created our dependent variable that is 0 for the number of rings ranging from 1 to 8, 1 for the number of rings ranging from 9 to 10, and 2 for the number of rings ranging from 11 and on.

Besides choosing the models and preparing the data, we also used random search as our fine-tuning method to better improve the performance of each model. For almost each parameter of each model, we selected a range that we believed the optimal parameter should occur, and tried a random combination of each option for the parameters to train a model and test its performance.

We did this random search for 200 iterations for almost each model. To account for the over-fitting problem, we are using k-fold cross validation with a k of 5 to measure the model performance during fine-tuning. However, due to our limited computational resources, we were not able to perform such random search on multilayer perceptrons, so instead we only fine-tuned the activation function and the number of layers and perceptrons in a small range while keeping the other parameters to be default values.

Similarly, we did wish to do a grid search after the random search, which is to permute all combinations of possible values of the parameters in the vicinity of the best parameters of random search, but we were not able to perform it because fine-tuning each model takes a rather long time on our hardware, which could be improved upon in future researches. Further details of the range of parameters and our best models can be found in the appendix.

To further improve our accuracy, we also attempted using feature scaling. Different units for the independent variables may be a cause for the low accuracy. However, we discovered that many of our models performed worse when we used scaled features. We decided then to stick to unscaled features as we obtained better results.

**Test Results**

|  | **Regression**  **(metrics = R squared)** | **One-to-One Classification (metrics = accuracy)** | **Grouped Classification (metrics = accuracy)** |
| --- | --- | --- | --- |
| **Random Forest** | 0.567 vs 0.502 | 0.278 vs 0.244 | 0.641 vs 0.482 |
| **Linear - Lasso** | 0.519 vs 0.301 | - | - |
| **Linear - Ridge** | 0.518 vs 0.514 | - | - |
| **MLP** | 0.575 vs 0.560 | 0.284 vs 0.265 | 0.662 vs 0.655 |
| **SVM** | 0.551 vs 0.488 | 0.266 vs 0.256 | 0.626 vs 0.617 |
| **Best Previous** | - | 0.263 | 0.656 |

*Comparisons are between a fine-tuned model and a baseline model in a form of fine-tuned vs baseline*

|  | RMSE |
| --- | --- |
| **Random Forest** | 2.070839575693145 |
| **Linear - Lasso** | 2.1286071892578247 |
| **Linear - Ridge** | 2.13580844772018 |
| **MLP** | 1.9921314240901977 |
| **SVM** | 2.054125281762722 |

*Residual Mean Squared Error values from the regression models*

For the regression task, we use R-squared and residual mean squared error as the evaluation metrics, and for the classification task we used accuracy as the evaluation metric. For each of our models, we provide a comparison between our fine tuning and the default parameter setting of sklearn.

Random forest and multilayer perceptrons perform generally better than other models in each task, but as a trade-off, they are not as interpretable as linear regression and support vector machines, while the best model in each task is a multilayer perceptron model. It is worth to mention that although support vector machines did not perform as well as random forest and multilayer perceptrons, their performances are close, so it is a rather good model to choose when we require a high level of interpretability. Linear regression, either with Lasso regularization or Ridge regularization, is not performing closely well as other models, which is probably because certain assumptions of it are violated, such as the linear relationship assumption or the multivariate normality assumption. Nevertheless, their RMSE values are ~2 for all models and the MLP regression model has the lowest at a value of ~1.99. Even though the RMSE values are not very high, the regression models do have difficulty determining the exact number of rings in an abalone.

**Conclusion:**

Our project exhibited multiple models for predicting abalone age that had better performance than previous researches. We found that multi-layer perceptron with proper fine tuning methods performed better than the other models that we created in regards to metrics. In terms of model choices, there is a trade-off between interpretability with performance, indicated by the generally better performance of multi-layer perceptrons and random forest than other more interpretable models. However, we also discovered that support vector machines may perform as well as random forest and multi-layer perceptrons after properly fine-tuned, which makes it a good choice in the occasion when we need both interpretability and performance. We also found that linear regression is not the most feasible method for our abalone data, particularly because several assumptions for linear regression are not satisfied. Although random search fine-tuning improves the model’s performance, grid search may improve the model’s performance even further if the hardware resource permits. As stated in the previous works, we also believe that this dataset exhibits a large degree of overlapping in its features, so more features may potentially increase the performance. The fact that Grouped Classification was only able to achieve a maximum of 0.662 accuracy and the pairplots do not form distinct clusters is indicative of a dataset with many overlapping classifications. Many of the features are also redundant, with 4 of the 8 attributes measuring the weight of the abalone. While our goal was to accurately predict the age of an abalone, regression models varied between 0.488 and 0.575 in their R-squared value. When analyzing our regression models, we also took the RMSE into consideration. The best regression model has a RMSE value of ~1.99, indicating that the MLP regression model could be used for age estimation. However, from the given data, we believe it is unfeasible to accurately predict the exact age of an abalone. From these results, we can confidently say that exact prediction is extremely difficult and there exists high overlap between the output variables. This can be addressed in future studies by requesting more features to be measured that could help improve model predictions through stronger correlations with the output variable.

**References:**

<https://eprints.utas.edu.au/21965/1/whole_WaughSamuelGeorge1997_thesis.pdf>

<https://archive.ics.uci.edu/ml/datasets/Abalone>

**Appendix:**

**Fine-tuning grids:**

Random Forest Regressor/Classifier fine-tuning grid:

{'bootstrap': [True, False],

'max\_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None],

'max\_features': ['auto', 'sqrt', 'log2'],

'min\_samples\_leaf': [1, 2, 4],

'min\_samples\_split': [2, 5, 10, 15, 20],

'n\_estimators': [200, 244, 288, 333, 377, 422, 466, 511, 555, 600],

'oob\_score': [True, False]}

Multilayer Perceptron Regressor/Classifier fine-tuning grid:

{'activation': ['identity', 'logistic', 'tanh', 'relu'],

'alpha': [0.0001],

'early\_stopping': [False],

'hidden\_layer\_sizes': [(100,), (200, 100), (300, 200,) ,(300, 200, 100)],

'learning\_rate': ['constant'],

'max\_iter': [5000, 10000],

'solver': ['adam'],

'verbose': [2],

'warm\_start': [True]}

Support Vector Regressor Fine-tuning grid:

{'C': [0.5, 1, 4, 7, 10, 20, 30],

'gamma': ['auto', 'scale'],

'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],

'max\_iter': [5000, 6250, 7500, 8750, 10000],

'shrinking': [True, False]}

Support Vector Classifier Fine-tuning grid:

{'C': [1, 4, 7, 10, 20, 30],

'decision\_function\_shape': ['ovo', 'ovr'],

'degree': [2, 3, 4, 5],

'gamma': ['auto', 'scale'],

'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],

'max\_iter': [5000, 6250, 7500, 8750, 10000],

'probability': [True, False],

'shrinking': [True, False]}

Lasso Regression Fine-tuning grid:

{'alpha': [1e-05, 0.0001, 0.001, 0.01, 0.1, 1],

'copy\_X': [True, False],

'fit\_intercept': [True, False],

'max\_iter': [5000, 6000, 7000, 8000, 9000, 10000],

'warm\_start': [True, False]}

Ridge Regression Fine-tuning grid:

{'alpha': [1e-05, 0.0001, 0.001, 0.01, 0.1, 1],

'copy\_X': [True, False],

'fit\_intercept': [True, False],

'max\_iter': [5000, 6000, 7000, 8000, 9000, 10000]}

**Best Models for Task1:**

Random Forest Regressor:

{'bootstrap': True,

'max\_depth': 60,

'max\_features': 'log2',

'min\_samples\_leaf': 4,

'min\_samples\_split': 2,

'n\_estimators': 422,

'oob\_score': False}

Lasso Regression:

{'alpha': 0.001,

'copy\_X': True,

'fit\_intercept': True,

'max\_iter': 7000,

'warm\_start': True}

Ridge Regression:

{'alpha': 1,

'copy\_X': True,

'fit\_intercept': True,

'max\_iter': 5000}

Multilayer Perceptron Regressor:

{'activation': 'tanh',

'alpha': 0.0001,

'early\_stopping': False,

'hidden\_layer\_sizes': (300, 200, 100),

'learning\_rate': 'constant',

'max\_iter': 5000,

'solver': 'adam',

'verbose': True,

'warm\_start': True}

Support Vector Regressor:

{'C': 30,

'gamma': 'scale',

'kernel': 'rbf',

'max\_iter': 5000,

'shrinking': True}

**Best Models for Task2:**

Random Forest Classifier:

{'bootstrap': True,

'max\_depth': 10,

'max\_features': 'sqrt',

'min\_samples\_leaf': 1,

'min\_samples\_split': 20,

'n\_estimators': 422,

'oob\_score': False}

Multilayer Perceptron Classifier:

{'activation': 'tanh',

'alpha': 1e-05,

'early\_stopping': False,

'hidden\_layer\_sizes': (100,),

'learning\_rate': 'constant',

'max\_iter': 10000,

'solver': 'adam',

'warm\_start': True}

Support Vector Classifier:

{'C': 20,

'decision\_function\_shape': 'ovr',

'degree': 4,

'gamma': 'scale',

'kernel': 'rbf',

'max\_iter': 5000,

'probability': False,

'shrinking': True}

**Best Models for Task3:**

Random Forest Classifier:

{'bootstrap': True,

'max\_depth': 100,

'max\_features': 'auto',

'min\_samples\_leaf': 4,

'min\_samples\_split': 5,

'n\_estimators': 244,

'oob\_score': True}

Multilayer Perceptron Classifier:

{'activation': 'relu',

'alpha': 1e-05,

'early\_stopping': False,

'hidden\_layer\_sizes': (300,200,),

'learning\_rate': 'constant',

'max\_iter': 10000,

'solver': 'adam',

'warm\_start': True}

Support Vector Classifier:

{'C': 30,

'decision\_function\_shape': 'ovo',

'degree': 2,

'gamma': 'auto',

'kernel': 'linear',

'max\_iter': 10000,

'probability': False,

'shrinking': True}