Crab Age Prediction using PCA, Regression, Random Forests, and Neural Nets

Graham Branscom, Andrew Raine, and Mahika Calyanakoti
May 5, 2024

Abstract: **This paper investigates the optimal age for harvesting crabs in commercial crab farming, addressing the need for efficient management to maximize profitability. Utilizing data analytics techniques, including statistical modeling and machine learning algorithms, the study analyzes crab growth patterns to determine the ideal timing for harvesting based on a variety of features, such as height, weight, and shell metrics. By offering insights into the relationship between crab age and input features, the project aims to accurately model crab age, which could be useful for the crab farming industry.**

EXECUTIVE SUMMARY

Study goal: We aim to build accurate models to predict crab age and classify crabs as old or young based on the indicator features. We also want to identify which features are the most important in determining age.
Data description: We utilize a Kaggle dataset of 3891 observations and 9 features, including physical features such as height, weight, shell weight, diameter, age, and more. The data includes information about crabs from the Boston Area from 2018-2019.
Methods Used: Our methods include EDA, PCA for clustering, linear regression, regularization (LASSO, Ridge, and Elastic Net), logistic regression, Random Forests, XGBoost, Feedforward Neural networks, and PCA on latent vectors, along with various data visualizations.
Findings: The study found that PCA clustering effectively distinguishes crab age in three dimensions, with vanilla regression outperforming regularized models (MSE 5.24 vs. 5.39). Logistic regression achieved 78% accuracy but with a low recall rate (55%). Random Forest models surpassed XGBoost, identifying key predictors (shell weight, shucked weight). The feedforward neural network achieved 80% classification accuracy, successfully predicting age and classifying crabs, with PCA visualizations revealing distinct age clusters.

I. INTRODUCTION

"As the sea-crab swimmeth always against the stream, so doth wit always against wisdom."
- Linda Zhao

Crabs, those enigmatic crustaceans of the sea, have long tantalized the taste buds of foodies worldwide, prompting a global quest for the perfect pinch of delectability. Beyond their culinary allure, crabs hold a pivotal role in coastal economies, with their claws intricately woven into the fabric of aquaculture. The rise of commercial crab farming, touted for its low labor costs and swift growth rates, has sparked a shell game of sorts, as farmers endeavor to crack the code of optimal crab harvesting. However, navigating the murky waters of crab age determination presents a challenge similar to solving a shell puzzle—a delicate balance between reaping maximum profit and avoiding the pinch of diminished returns.
Inspired by the whims of crab connoisseurs and the plight of crab farmers alike, this paper attempts to unravel the mysteries of crab age and its implications for the lucrative crab farming

industry. As we delve into the depths of data science, we seek to discern the optimal moment for harvesting based on our models' crab age prediction. Through a data analytics lens, we endeavor to bring clarity to the murky depths of crab farming.

II. DESCRIPTION OF DATA

Our dataset, sourced from Kaggle, includes information regarding various physical attributes of crabs in the Boston area between 2018 and 2019. These include:

1. **Sex**: Gender of the Crab - Male, Female and Indeterminate (categorical variable)
2. **Length**: Length of the Crab (in feet)
3. **Diameter**: Diameter of the Crab (in feet) measuring the longest possible length of the crab
4. **Height**: Height of the Crab (in feet)
5. **Weight**: Weight of the Crab (in ounces)
6. **Shucked Weight**: Weight of the Crab without its shell (in ounces)
7. **Viscera Weight**: the weight that wraps around the crab's abdominal organs deep inside body (in ounces)
8. **Shell Weight**: Weight of the Shell (in ounce)
9. **Age**: Age of the Crab in months

Our dataset has 3891 observations, each with 9 features. Our target response variable is age, which we aim to predict given the other input features. This type of continuous response variable lends itself well to regression-type analysis. However, we also want to incorporate a more classification-based analysis. As part of our feature engineering, we create a new column, called **Young|Old**, which provides a value of 0 if young (age was less than the median age), or 1 if old (age was greater than the median age). By picking the median, we automatically avoid issues with class imbalance, such that our final dataset has an even split of "old" and "young" crabs.

III. EXPLORATORY DATA ANALYSIS

Next, we aim to get a better understanding of the dataset and its features.

A. Data Summary

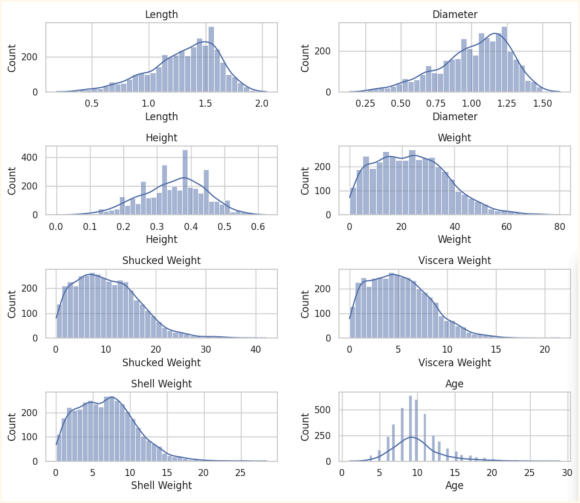
We start by getting a summary of the numerical features, which helps us identify any potential outliers. Through our analysis, we find no null values as well as no extreme outliers.

	Length	Diameter	Height	Weight
Mean	1.31	1.02	0.35	23.56
Std	0.30	0.25	0.10	13.88
Min	0.19	0.14	0.00	0.06
Max	2.04	1.63	0.63	80.10

	Shucked weight	Viscera weight	Shell weight	Age
Mean	10.20	5.13	6.79	9.96
Std	6.27	3.10	3.94	3.22
Min	0.03	0.01	0.04	1.00
Max	42.18	21.55	28.49	29.00

B. Variable Distributions

We utilize data visualization tools to plot the distribution of the different variables through histograms. All of the variables follow a general curve pattern:



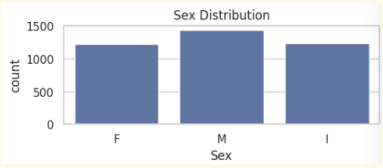


Figure 1. Variable Distributions

Based on the Sex Distribution histogram, we can see that our data does not have any drastic class imbalance issue.

"There is a saying in Baltimore that crabs may be prepared in fifty ways and that all of them are good."

\- Neil Fasching

C. Data Split

For the purposes of our supervised machine learning models, we split the data into training and testing datasets. We use an 80/20 split, using 80% of our dataset for training, and the remaining 20% for testing.

IV. PRINCIPAL COMPONENT ANALYSIS

The next step of our analysis is to conduct PCA for dimensionality reduction. We define new features using a combination of the original seven indicators to potentially improve model accuracy. We exclude Sex since it is a categorical variable.

Before applying PCA, we use StandardScaler() to standardize our data, and then reduce our dataset to two dimensions. PC1 is a weighted sum of all seven indicators, whereas PC2 is a weighted difference of length, diameter, and height from weight, shucked weight, viscera weight, and shell weight. We then plot a scatterplot of the PCA of Crab Variables, with PC1 on the x-axis and PC2 on the y-axis. We color the points blue for young crabs and orange for old crabs to see if we can see some sort of clustering effect:

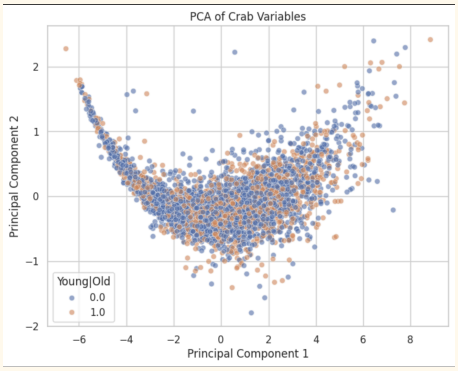


Figure 2. PC2 vs PC1

As we can see from the scatterplot above, we see a lot of overlap between the blue and orange points, which tells us that there is not much of a clear clustering effect in two dimensions. This might be because the variance in the features in such a low dimension does not accurately distinguish between the old and young crabs.

We then use t-SNE (t-distributed Stochastic Neighbor Embedding) to visualize the data

points using another method and see if we get any different results:

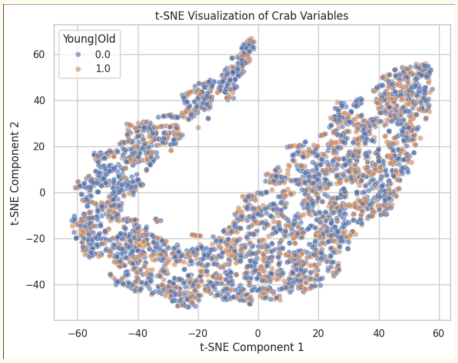


Figure 3. t-SNE Visualization of Crab Variables

We still do not see a clear clustering in this 2D visualization. However, we notice an interesting pattern here: the points seem to form a horseshoe shape, reminding us of a specific species of crab: horseshoe crabs!

The parabolic shape in the scatterplots, especially Figure 2, suggests a relationship between the two principal components where the variation first decreases and then increases. This could indicate that the variables contributing to these components have a quadratic relationship, or it may point to the presence of two or more distinct groups in the data. It's also possible that the first principal component captures most of the variance, while the second one captures a non-linear relationship.

Thus, to investigate further we utilize a three-dimensional PCA visualization instead, using the first three PCs:

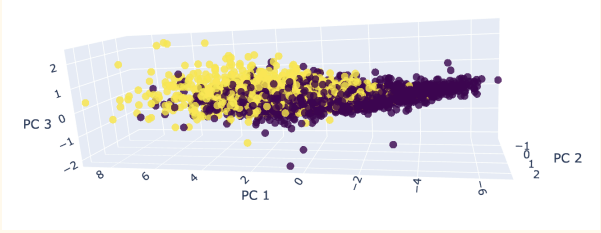


Figure 4. 3D PCA Scatterplot of Young vs Old

We see from this scatterplot that there is a much clearer clustering effect, with the yellow and purple dots representing the young and old crabs.

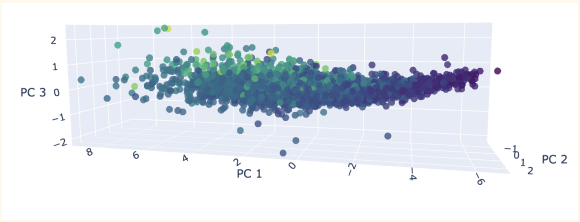


Figure 5. 3D PCA Scatterplot by Age

When we plot a more continuous coloring of the points by age, we see a clear trend with the older (more purple) points towards the right of the plot, and the younger (green and blue) points towards the left. This aligns with what we see in the classification 3D PCA plot.

V. LINEAR MODELS

A. Simple Linear Regression

We start with a vanilla multiple linear regression for Age. In order to check that the linear model assumptions are met, we plot the residual analysis charts, including the Distribution of Residuals and Q-Q Plot of Residuals. We see the residuals centered around 0 with an equal distribution of positive and negative residuals, as well as a fairly linear relationship in the Q-Q plot, meaning that our linear model assumptions were met:

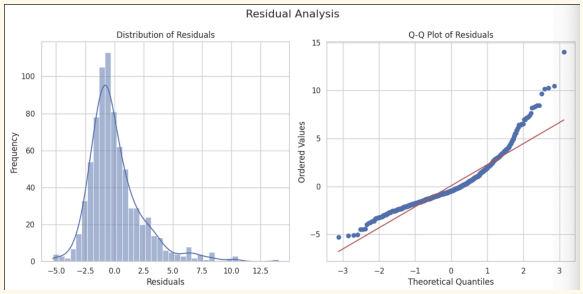


Figure 6. Residual Analysis for Linear Regression

Feature	β-value	p-value
Height	10.61	<1.0E-4
Diameter	5.34	<1.0E-4
Weight	0.31	<1.0E-4
Shell Weight	0.25	<1.0E-4
Viscera Weight	-0.36	<1.0E-4
Shucked Weight	-0.68	<1.0E-4
Length	-1.62	0.047

This suggests that there are positive correlations between height, diameter, weight, and shell weight with age, but negative correlations with viscera weight, shucked weight, and length. This makes sense since younger crabs tend to have smaller dimensions and less developed shells. However, very old crabs lose viscera weight and shucked weight as they age and their inner body deteriorates.

After calculating errors for the test data, we see a model Mean Squared Error (MSE) of 5.39, which is fairly low considering the average age is around 10 years.

B. Linear Regression with Interactions

The next step of our analysis is to investigate any possible interactions between the variables in our linear model. More specifically, we study the interactions between length, diameter, height, and weight. After running our model we get the following beta-values:

Feature	β-value	p-value
Diameter x Height	58.53	0.056
Length	9.16	0.166
Height	4.67	0.679
Height x Weight	2.17	0.156
Length x Diameter x Height x Weight	0.94	0.113
Length x Weight	0.78	0.081
Shell Weight	0.36	<1.0E-4
Diameter x Weight	-0.114	0.843
Length x Diameter x Weight	-0.19	0.480
Weight	-0.27	0.640
Viscera Weight	-0.27	<1.0E-4
Shucked Weight	-0.62	<1.0E-4
Diameter x Height x Weight	-1.10	0.398
Diameter	-1.67	0.853
Length x Height x Weight	-1.85	0.070
Length x Diameter x Height	-4.05	0.737
Length x Diameter	-5.76	0.238
Length x Height	-35.31	0.142

Based on the results, we can see that certain interactions are quite significant in our model, due to the fact that they have small p-values using alpha = 0.1. More specifically, the interaction between Diameter and Height seems to be the most significant for our model.

We see an improved MSE of 5.24 with this model, and the linear model assumptions are met based on the plots below:

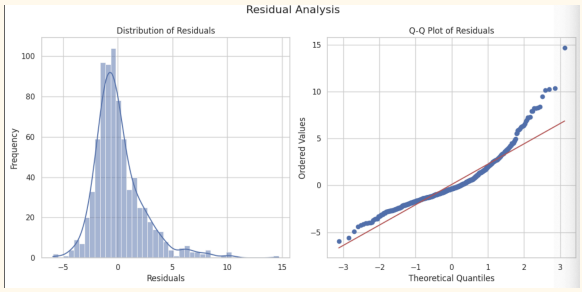


Figure 7. Residual Plots for Linear Regression with Interactions

C. Regularized Regression Model Comparison

To take our analysis a step further, we utilize regularized regression using Ridge (L2), LASSO (L1), and Elastic Net (L1 and L2) penalties.

More specifically, we utilized GridSearchCV() to explore various parameters in our hyperparameter tuning process, which utilized 5-fold cross-validation to optimize the parameters based on the MSE.

```
# Defining the range of alpha values to explore
  alpha_range = np.logspace(-4, 2, 50)
# 50 values from 10^-4 to 10^2

# Set up the GridSearchCV object
  lasso_grid = GridSearchCV(
      estimator=model(random_state=42,
      max_iter=10000),
      param_grid={'alpha': alpha_range},
      cv=5,  # 5-fold cross-validation
      scoring='neg_mean_squared_error'
  )
```

Ridge included all seven variables, whereas LASSO and ElasticNet both included length, weight, shucked weight, viscera weight, and shell weight (they both excluded diameter and height).

The final MSE values are as follows:

Model	MSE
Ridge	5.3961
LASSO	5.3948
Elastic Net	5.3962

We see that the LASSO model performed marginally better than the other two regularized models, suggesting that using just an L1 penalty worked the best. However, the vanilla regression model and linear regression with interaction effects outperform all of the regularized ones.

D. Logistic Regression on Young or Old

Finally, we use a logistic regression model to classify crabs as either old or young. We use evaluation metrics to see how well our model performs:

Metric	Value
Accuracy:	0.775
Precision:	0.764
Recall:	0.549
ROC AUC:	0.836

Based on the 78% accuracy, we can say this model performs quite well. The precision of 0.764 suggests that when the logistic model predicts a positive outcome (old), it is correct about 76.4% of the time. However, the recall of 0.549 indicates that the model only identifies 54.9% of all actual positive (old) cases. Therefore, while the model is relatively reliable when it predicts a positive result, it misses a significant number of cases when the crab is old. This could mean the model is more conservative in predicting positives, possibly prioritizing

minimizing false positives over capturing all true positives.

VI. TREE MODELS

For the next part of our analysis, we explore tree-based models.

A. Random Forest Regressor

We begin by creating a random forest to try to predict age, fitting the model on the training data set, and then predicting on the testing data set. We calculate the residuals in order to compute the MSE value, which we found to be 5.234, which is relatively on the lower end compared to our regression models. As we can see from the residual plots of the distribution of residuals and the Q-Q plot, we see that our model assumptions are met:

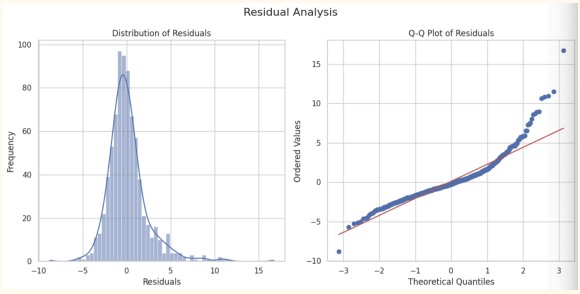


Figure 8. Residual Plots for Random Forest Regressor

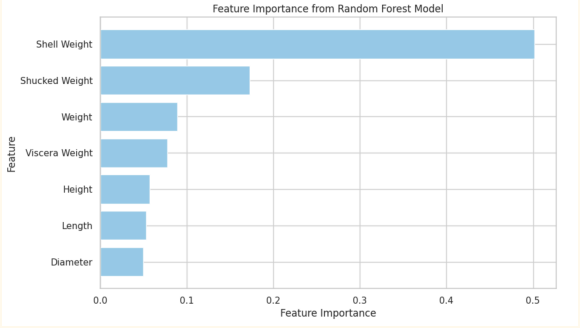


Figure 9. Feature Important for Random Forest Regressor Model

As we can see from the feature importance bar chart, we can see that the shell weight plays the most important role in the random forest model by far, followed by shucked weight. These are the features that truly distinguish between the different crab ages.

B. XGBoost Regressor

When we use the XGBoost method instead, we set the objective to be minimizing the squared error. After evaluating the error, we see an MSE of 5.816, which is worse than our random forest. This tells us that the Random Forest has improved performance when predicting the continuous variable of Age.

C. Random Forest and XGBoost Classification

Now, we switch gears to use the Random Forest and XGBoost models for classification purposes instead. The goal of these models is to predict if

a crab is old or young. After computing the evaluation metrics for both models, we see the following comparison:

Metric	Random Forest	XGBoost Model
Accuracy	0.775	0.757
Precision	0.736	0.688
Recall	0.592	0.603
ROC AUC	0.845	0.834

Comparing the two models, the Random Forest outperforms the XGBoost Model in terms of accuracy and precision, with scores of 0.775 and 0.736 against 0.757 and 0.688, respectively. This indicates that the Random Forest is generally more accurate in its overall predictions and when it predicts a positive class, it is correct more often than the XGBoost Model. However, when it comes to recall, which measures the ability to find all the relevant cases within a dataset, the XGBoost Model slightly surpasses the Random Forest, with scores of 0.603 to 0.592. This suggests that the XGBoost Model is slightly better at detecting positive instances, although it sacrifices some precision and overall accuracy. Additionally, the Random Forest has a higher ROC AUC score of 0.845 compared to 0.834 for the XGBoost Model, indicating that, overall, the Random Forest is better at distinguishing between the classes at various threshold settings.

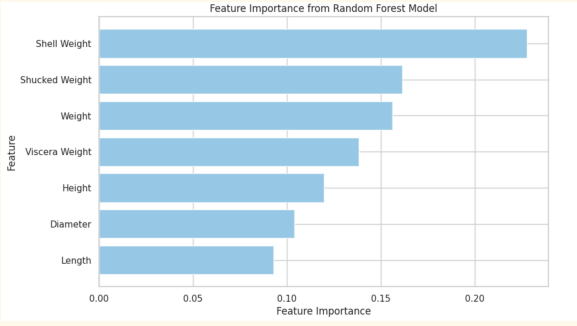


Figure 10. Feature Important for Random Forest Classification Model

We can see from the figure importance plot above that shell weight, shucked weight, and weight are the most important features for predicting age as a category. Notably, the orders of the feature importances for the regression and classification are largely identical.

VII. NEURAL NETWORK FOR REGRESSION AND CLASSIFICATION

A. Neural network architecture

First, we prepare the data by changing the categorical Sex variable to numerical values. We then use the torch library, along with TensorDataset, and DataLoader to apply the neural net. After splitting the data 80/20 into training and testing dataset, we convert the datasets into TensorDatasets and create DataLoaders for the train and test sets.

For the model, we utilize torch.nn for our Feedforward neural network. For our architecture, we use the Tanh() activation function along with nn.Linear(8,8) layers. At the end, we use a softmax of dimension 1 for the classification task. See code below:

```
import torch.nn as nn

class FFN(nn.Module):
  def __init__(self, task):
    super(FFN, self).__init__()

    assert task in {"regression",
"classification"}

    self.task = task

    self.backbone =
nn.ModuleList([nn.BatchNorm1d(8, affine=False)])

for i in range(5):
      self.backbone.append(nn.Linear(8, 8))
      self.backbone.append(nn.Tanh())
      self.backbone.append(nn.Linear(8, 4))

    if task == "regression":
      self.head = nn.Linear(4, 1)
    else:
      self.head = nn.Linear(4, 2)
```

```
def forward(self, x, raw=False):
    for layer in self.backbone:
      x = layer(x)
    x_logits = x
    x = self.head(x_logits)

    if self.task == "classification":
      x = nn.Softmax(dim=1)(x)

    if raw:
      return x, x_logits
    return x
```

For our loss function and optimizer section, we use MSE loss for regression, whereas we use Cross Entropy Loss for the classification analysis. We utilize torch.optim.Adam() optimizer for the neural network. As we run the neural network, we utilize zero_grad(), loss.backward(), and optimizer.step() for each iteration, printing the Loss and Accuracy for each batch in each Epoch, of which we had 30. Further, we utilize the GPU to run our computationally heavy task.

B. Neural network performance

After our final Epoch concludes, we see a test Loss of 0.5164 and an Accuracy of 79.72% for our classification model. On the other hand, for our regression model, after our 30 Epochs complete we see a Test Loss of 5.206, which essentially outcompetes our previous models.

Thus, from these metrics, we can see that the Feed Forward neural network performs quite well in predicting or classifying crabs by age, especially when compared to our other models.

C. PCA on the Neural Network Latent Vectors

We obtain the logit vectors from the neural network, which is the output of the last layer. We obtain these embedding vectors for the test data, and then run a PCA on those vectors. Here, we color the known age label in the test data on the graph. We can see that the age is captured among the first two PCs since as the graph goes from left to right, age generally increases.

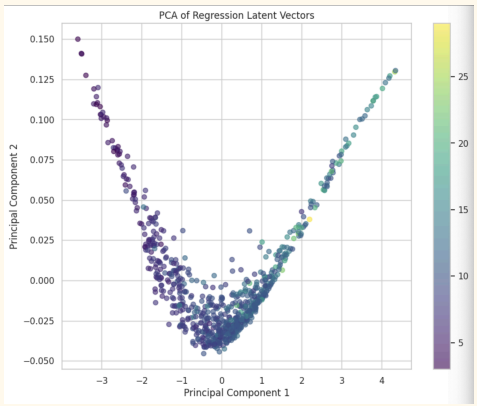


Figure 11. PC2 vs PC1 of Regression Latent Vectors

We run the same analysis as above, but now we include the first three PCs. We can still see that there is a trend where age tends to increase as you move along the graph.

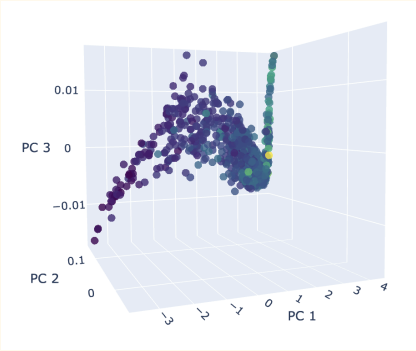


Figure 12. 3D PCA Visualization of Regression Latent Vectors

We now run the same analysis on the test data and use the latent vectors from the classification neural network. These are the vectors right before the softmax is applied. The embeddings are from the test data and the labels are from the known age categories in the test data. We can see that there is a cluster between young and old.

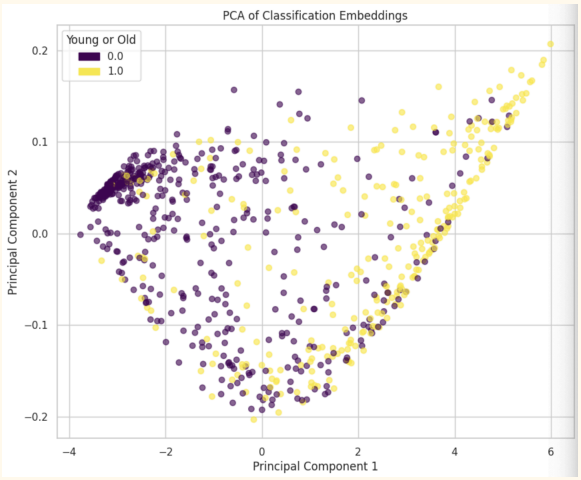


Figure 13. PC2 vs PC1 of Classification Latent Vectors

We now add the third PC and can again see two clusters between young and old.

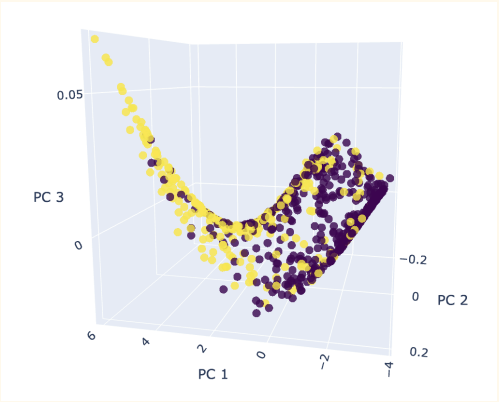


Figure 14. 3D PCA Visualization of Classification Latent Vectors

We now run the same analysis, but for two components from t-SNE. Here, we can see that t-SNE does not perform as well as the PC2 vs PC1 plot (Fig. 13) since there is no clear clustering in the t-SNE plot.

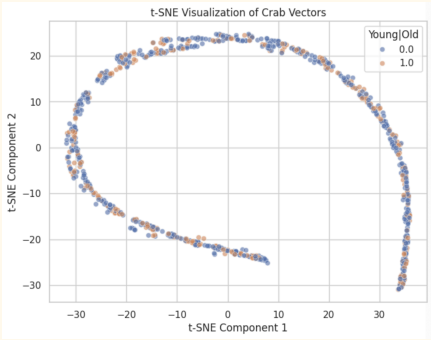


Figure 15. t-SNE Visualization of Classification Latent Vectors

VIII. CONCLUSIONS

In the pursuit of optimizing harvest times in commercial crab farming, this paper provided a comprehensive data-driven approach to predict crab age, which had a mean of 10 years in our dataset. The variety of analytical techniques used, from basic regression to ML algorithms like Random Forests and Neural Networks, showed the complexity of predicting or classifying crab age. Through exploratory data analysis and the application of PCA, we showed various patterns and relationships within the data along with visualizations.

Our findings were that the PCA clustering analysis works quite well at distinguishing crabs by age when graphed in three dimensions. Further, the vanilla regression and with and without interactions performed better than the regularized regression models, with MSE values of 5.24 versus 5.39, respectively. Among the regularized models, the LASSO model performed marginally better than the Ridge and Elastic Net models. Our logistic regression model to predict young or old was also quite successful, with 78% accuracy, though it had a low recall rate of just 55%.

In terms of the tree models, the Random Forest models outperformed the XGBoost models, and the feature important bar charts we derived from the Random Forest model showed that shell weight and shucked weight were the most important variables for the model to predict crab age for both the regression and classification Random Forest models.

Lastly, in our feedforward neural network model, we saw a classification accuracy of nearly 80%, and a test loss of 5.206. These metrics suggest that the FFN was quite successful in predicting crab age and classifying crabs as old or young. We further use various visualizations to analyze the PCA on the neural network latent vectors, which revealed several interesting patterns, namely that clear clusters or patterns among age are seen across the first two and three PCs.

This deep dive into the data shows how various models perform for age prediction, as well as emphasizes the potential of these methods in revolutionizing the crab farming industry.

IX. ACKNOWLEDGEMENTS

We would like to thank Professor Linda Zhao and the Spring 2024 STAT 4710 TAs for their continued help throughout this course.

ChatGPT was used for assistance in making some of the visualization plots, such as the three-dimensional ones. We also used it to help in some of the models, such as for tasks like hyperparameter tuning and calculating error metrics.

"We have the ability to approach our race like ants, or we have the ability to approach our race like crabs."
- Kanye West