A.I. Angling: Applying Machine Learning to Fish Weight Estimation

Justin Luo
Class of 2027
Flintridge Preparatory School

FLINTRIDGE PREPARATORY SCHOOL

Abstract

This experiment involves a machine learning module that predicts a fish's weight based on its length, width, and height. With a dataset, it takes the inputs (lengths, widths, and heights) and tries to guess the weight. Once it guesses, it checks the actual weight in the dataset and continues with each fish in the dataset.

My hypothesis was that with a limited dataset of 128 fish, it would be able to get a low margin of error around 20%. My independent variable for the experiment is the number of fish, while the dependent is the percent error it outputs.

The results of the training concluded that with 31 training cases, it was able to get an average 14% margin of error, proving and even surpassing my hypothesis of 20%.

Introduction

Since the 1990s, machine learning models have been exploding in popularity amongst researchers, technology enthusiasts, and industry professionals alike. Today, it's used in a variety of fields such as facial recognition, social media advertisements, weather forecasts, and even stock market predictions!

This experiment uses a python machine learning model to estimate fish weight based on their length, width, and height. My hypothesis is that the output weight versus the actual weight of the fish on the dataset will have an average margin of error of lower than 20%.

Linear regression is one of the fundamental training types used in the machine learning experimentation process. It is a basic form of regression where the model is not penalized at all for its choice of weighting, meaning it can freely choose how much importance each variable is to the result.

Lasso regression, a form of linear regression, introduces penalties to the model, which penalizes the model based off the absolute value of the sum of the weights. The penalty term in Lasso encourages the model to have smaller and more evenly distributed weights amongst the variables. In other words, picture the weights as the coefficients in front of the variables. If a coefficient is 9, and the other is 1, that would result in the sum of the weights being 10, which is very high. This would lead to a "punishment", which essentially means it's telling the model that it's doing bad. When it does that, the model pushes for a lower weight sum until it reaches almost 0 to avoid more punishment. The absolute value is extremely crucial, as Lasso focuses on the size of the weights, not whether it's negative or positive.

Finally, ridge regression, another form of linear regression, uses penalties as well. Instead of punishing the model for the sum of the weight, it punishes it on the square of the weight. For example, if the weight is 3, it will look at 3 to the power of 3 which is 9, which will lead to a big punishment. This encourages the model to go for a smaller weight, since when the weight becomes less than one, the squared of it gets even smaller, which leads to less punishment.

All these forms of regression have their own pros and cons, like speed of training and adaptability to outlier datapoints, but later in the methodology, I pick the one that's outputted the most accurate results.

Introduction (Continued)

In the end, the goal of the experiment is to train the model to get the best possible description between the relationship of the input and output variables, while maintaining a weight sum of almost 0.

This enables us to start testing, where we finally see the effects of training in action.

Materials

As this is not a physical experiment, traditional equipment like beakers, eye protection, masks, and whatever else you think of when you hear the word "experiment" isn't needed. Instead, everything is done online. Nevertheless, there still are materials that are required.

First, a computer was required to research, code and complete the experiment. The coding portion was done in Visual Studio Code, a popular integrated development environment. Specifically, I used the Jupyter Notebook extension, a module that allows you to segment your code and run each segment individually and see the individual outputs, which was particularly useful for data and graph collection.

Then, I needed a dataset. I found a set with around 160 fish in various species with all their dimensions and weight on Kaggle.com, a place where people from all around the world post their machine learning notebooks and datasets for other's reference. I then split the set into 80% for training and 20% for testing.

Conducting this experiment was low cost (requiring only a single computer) and concise (contained within a single file).

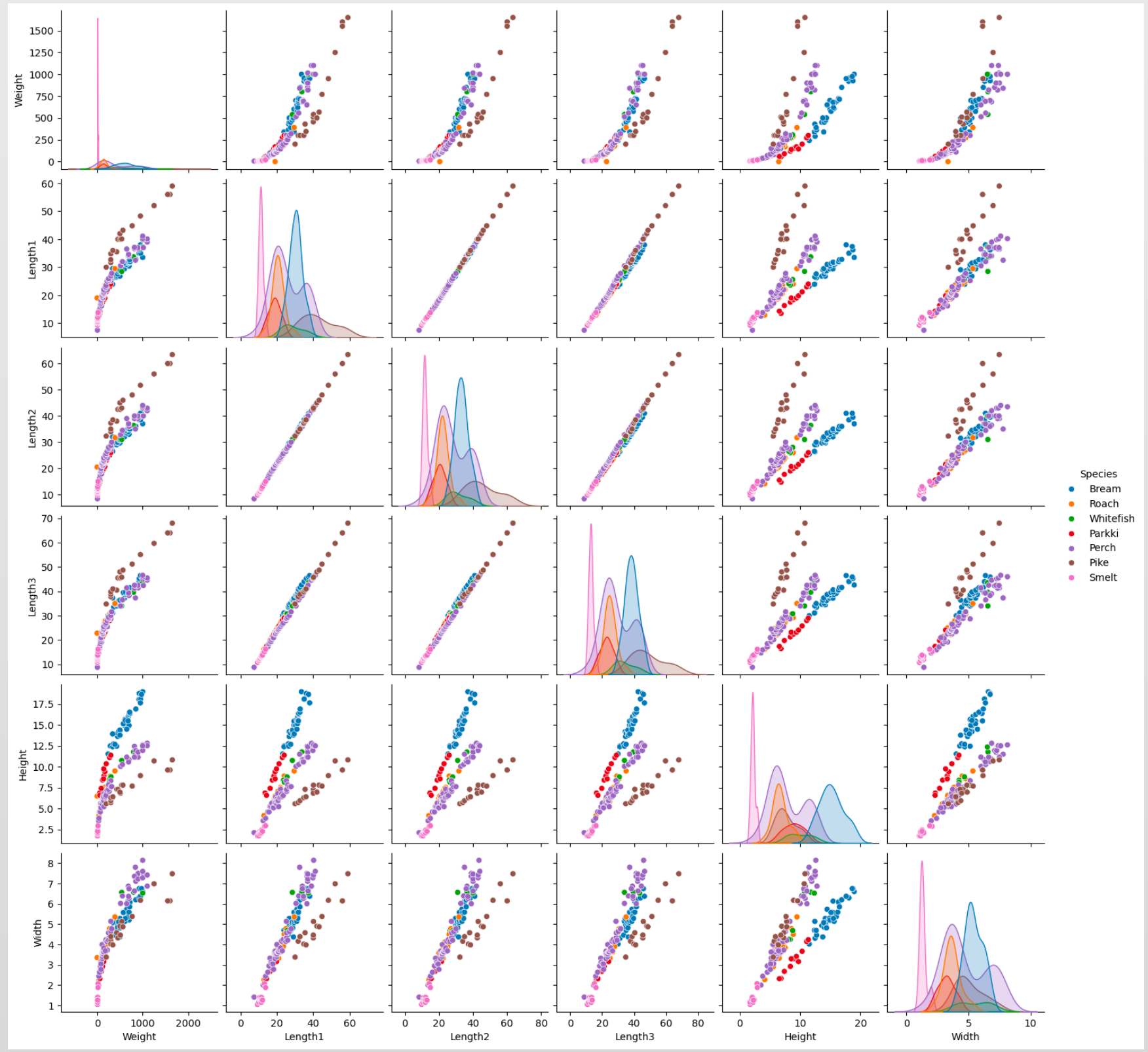


Figure 1: The plotted points of data from the set showing the correlation between all the length, width, height, and weight.

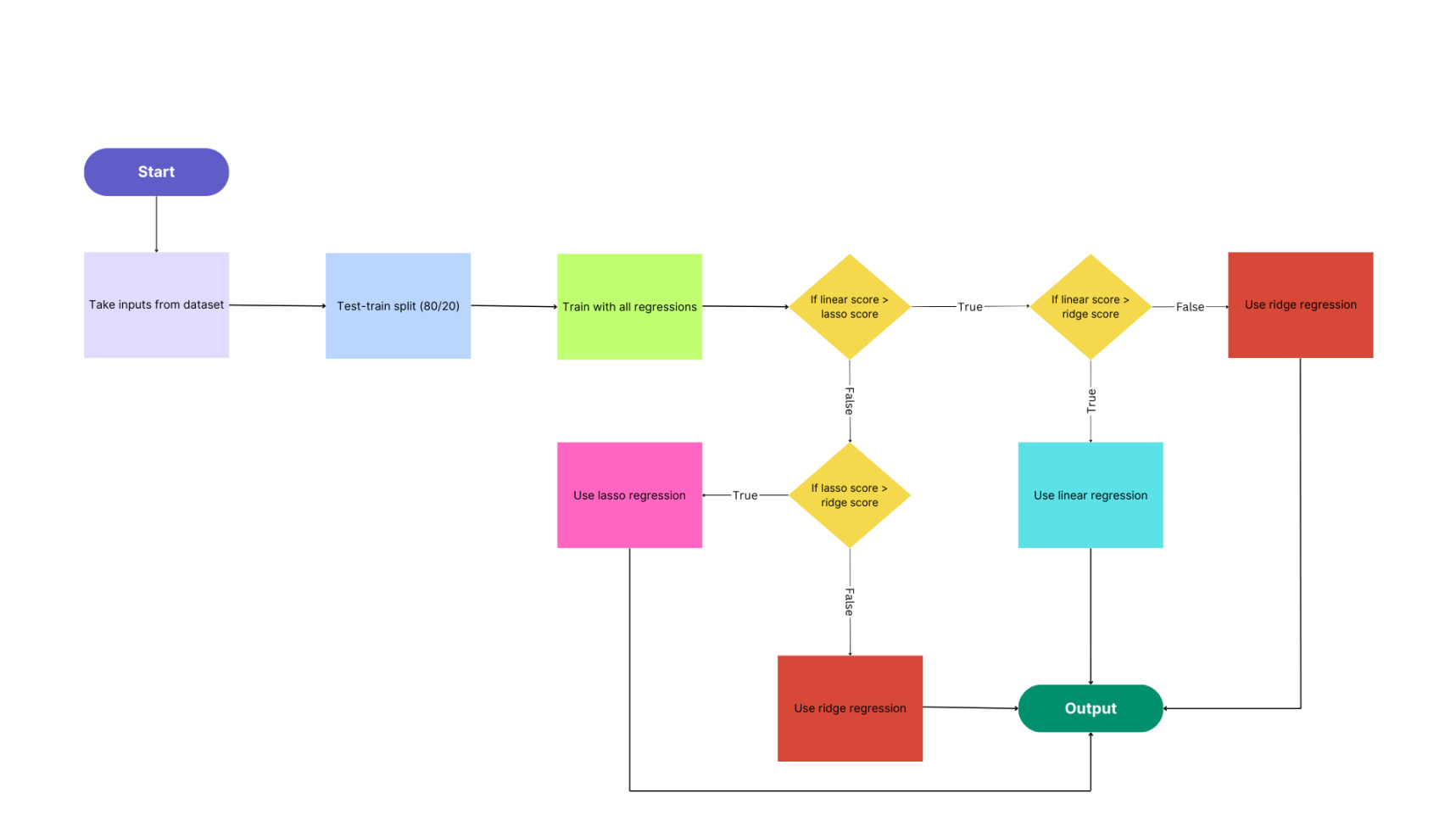


Figure 2: The overall flowchart of assessing which regression is best with the provided dataset, also known as performance measuring

Methodology

I first downloaded the dataset from Kaggle, and checked if one, it had at least 100 species, and two, if it had close to all 3 lengths needed for accurate predictions (width, height, length).

Then, I imported all the modules I need to organize and code the experiment. The modules I used were Pandas (organizing dataset), Matplotlib (plotting data), Seaborn (addition to matplotlib), Numpy (adds complex math to arrays), and Itertools (adding to iterators).

Once I finished importing and downloading the modules, I organized the dataset with the Pandas module. Specifically, I used the data.isna() function to check if there was missing data and counted the number of fish species in the set with the value_counts() function. Then, I graphed the correlation between the data with Matplotlib (Figure 1). Finally, I coded the model using another module called sklearn.

Using the flow chart (Figure 2), I found the best regression, which in this case, it was linear regression. The formula for linear regression that I utilized was the simple y = ax + b, where a and b are the inputs (lengths of fish) and y is the output (weight of fish). After that, I split the dataset into two groups. Splitting the data into 80% and 20% is called a test-train split, where 20% is used for testing, and 80% is used for training the model. This often provides the best results, as getting new data can be difficult and could possibly slow down the experiment.

With the provided dataset, the model takes the length of the fish as x, and takes the total weight of the fish as y. This prompts it to continually guess and check for a and b until it finally gets the best answer possible. This is repeatedly done with 80% of all the fish in the dataset, until it is finished training and ready for the testing phase of the experiment.

Results

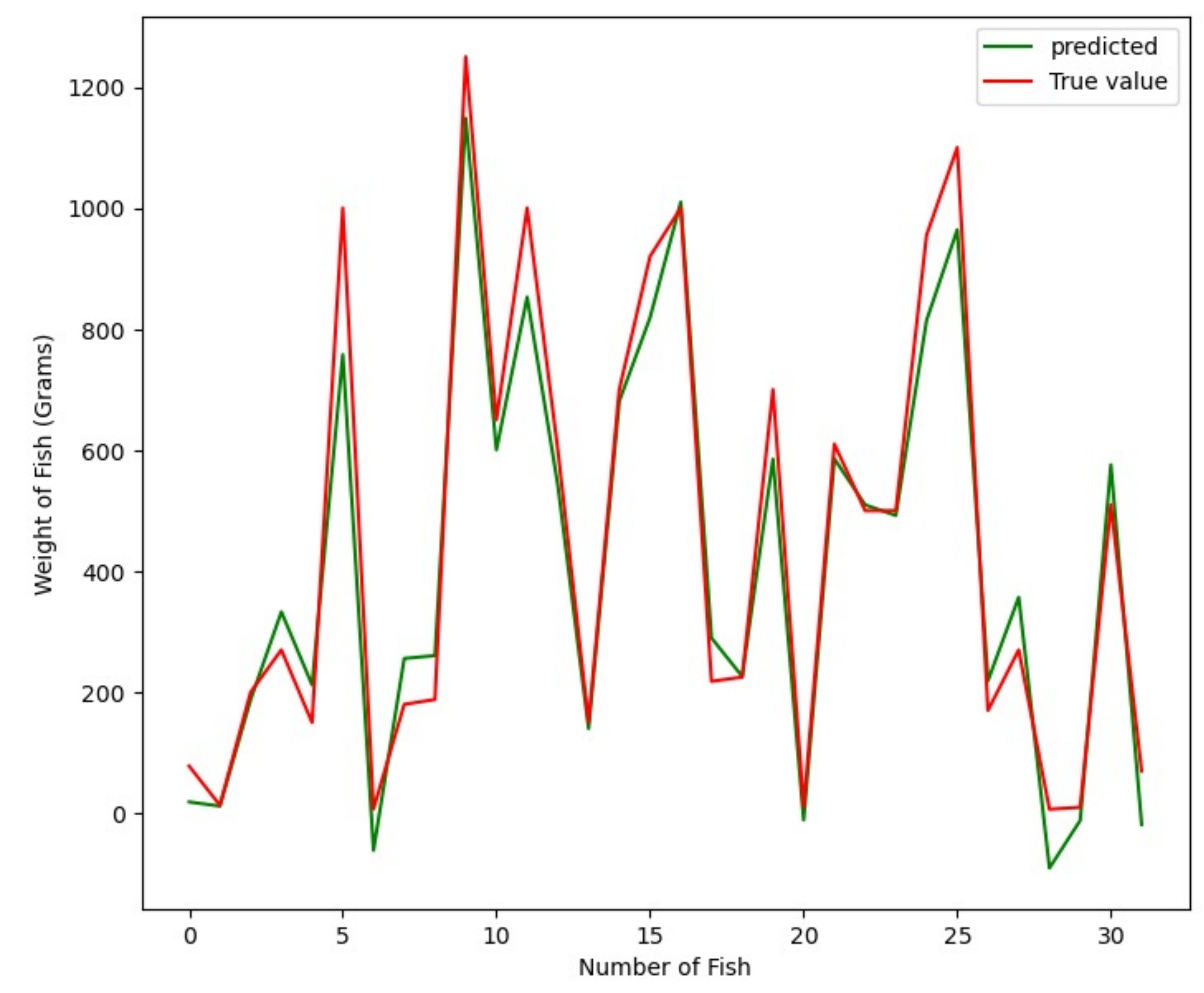
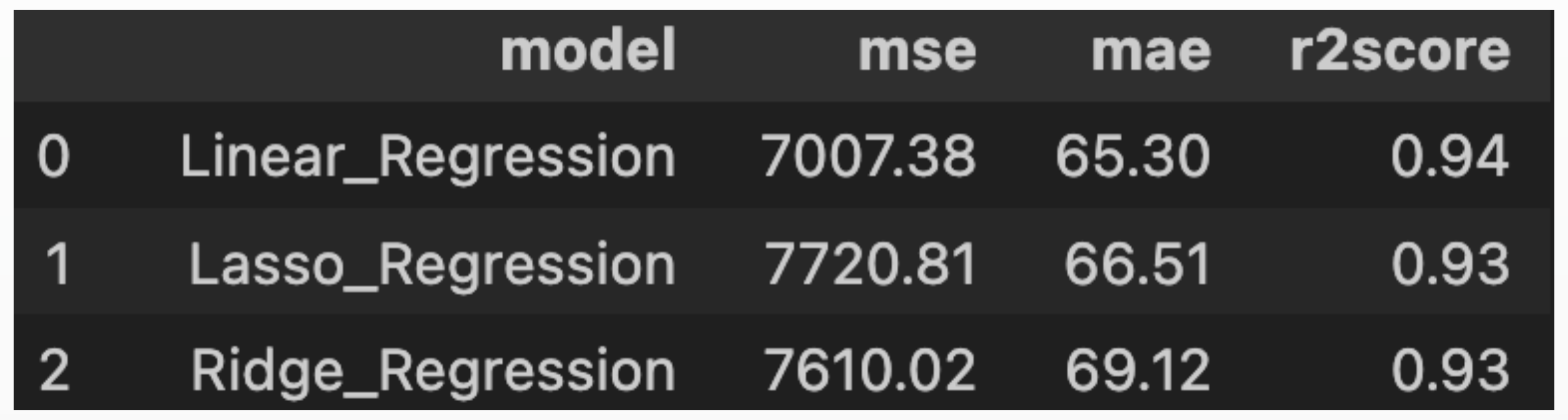


Figure 3: This graph shows the predicted versus the true value (in other words the output fish weight vs the actual fish weight) of our model. It's fairly accurate, as most of the red and green are very close to each other, meaning a low margin of error.

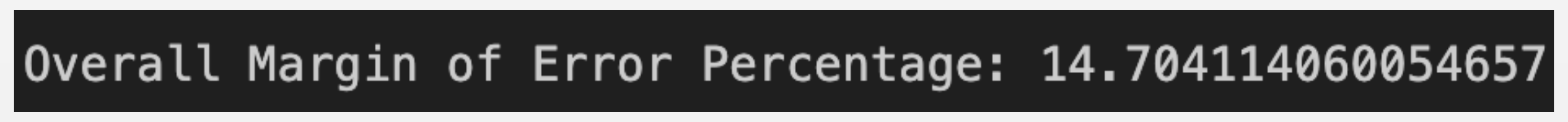


	model	mse	mae	r2score
0	Linear_Regression	7007.38	65.30	0.94
1	Lasso_Regression	7720.81	66.51	0.93
2	Ridge_Regression	7610.02	69.12	0.93

Figure 4: The output score (r2score) of the testing stated that linear regression was performing better (just slightly) than lasso and ridge, which is why I went with linear.

The results of the experiment was able to accurately predict most inputted fish's weights with just their lengths and supported my hypothesis of less than 20% margin of error (Figure 5).

Additionally, it also proved that training a machine learning model to be accurate is very simple and quick. Even with a small dataset, it still managed a very low margin of error, and it will only get smaller the more data there is available.



```
Overall Margin of Error Percentage: 14.704114060054657
```

Figure 5: The outputted margin of error percentage.

Conclusion

In conclusion, machine learning is an amazing tool that is simple, effective, and fast-growing. This model only took a couple seconds to finish training, and already has a relatively small margin of error (considering the tiny dataset used.) Machine learning is already being applied across every discipline imaginable, whether it's farming crops, analytics, or robotics. Just picture what the fast-developing future has in store for it!

This project also has potential for real world applications, as it may help both commercial and recreational fishermen alike with their catches. For commercial fishermen, averaging the length of the same sized fish would save them a tremendous amount of time in reporting their total catches of the day. Recreational fishermen (like me) often forget their weighing scales and unfortunately catch their personal best right after. Using this program, just measuring the fish will be enough to figure out the weight!

Acknowledgements

I'd like to thank my advisor, Dr. Yutao He, for teaching me everything about machine learning and helping me through this experiment.
I'd also like to thank my sister, Nina, for proofreading as well as emotional support.
Finally, I'd like to thank Dr. Frewen for organizing the steam and service fair and allowing me to share my experiment.

Works Cited / Appendix

A. (2022, October 18). *The Motivation for Train-Test Split*. Medium. https://medium.com/@nahmed3536/the-motivation-for-train-test-split-2b1837f596c3#:~:text=In%20supervised%20ML%2C%20we%20utilize,dataset%2C%20and%20slow%20down%20experimentation.
Tutorial:
https://utsavdesai26.medium.com/linear-regression-made-simple-a-step-by-step-tutorial-fb8e737ea2d9
Dataset:
https://www.kaggle.com/code/ybifoundation/fish-weight-prediction/notebook
Types of regression:
https://towardsdatascience.com/whats-the-difference-between-linear-regression-lasso-ridge-and-elasticnet-8f997c60cf29