Fish and weight predictions

Statistical analysis to predict the weight and species of fish after recording ability was compromised

Andy McDill, MSc

Table of Contents

[1 Introduction 2](#_Toc146025801)

[1.1 Definitions 2](#_Toc146025802)

[1.2 Objective Summary 3](#_Toc146025803)

[1.3 Fish Specifications 3](#_Toc146025804)

[2 Data Preprocessing 4](#_Toc146025805)

[2.1 Missing Values 4](#_Toc146025806)

[2.2 Extreme Values 4](#_Toc146025807)

[2.3 Derivations 4](#_Toc146025808)

[3 Exploratory Data Analysis 5](#_Toc146025809)

[3.1 Distributions 5](#_Toc146025810)

[3.2 Correlations/Relationships 6](#_Toc146025811)

[4 Predictive Analytics 7](#_Toc146025812)

[4.1 Regression 8](#_Toc146025813)

[4.1.1 K-Nearest Neighbors 8](#_Toc146025814)

[4.1.2 Random Forest 8](#_Toc146025815)

[4.1.3 Extreme Gradient Boosting 9](#_Toc146025816)

[4.2 Classification 10](#_Toc146025817)

[4.2.1 Support Vector Machine 10](#_Toc146025818)

[4.2.2 Random Forest 11](#_Toc146025819)

[4.2.3 K-Nearest Neighbors 13](#_Toc146025820)

[5 Conclusion 14](#_Toc146025821)

# Introduction

This report is designed to explain the objectives, conclusions and rationale regarding the statistical analysis that was required by a local fishing business. Preprocessing, exploratory analysis, and machine learning were implemented to understand and project future outcomes from the data that was provided. All statistical analysis was conducted in Python version 3.9.12 and all visualizations were created in Tableau Public 2.1.

## Definitions

|  |  |  |
| --- | --- | --- |
| Label | Abbreviation | Explanation |
| *Response Variable* |  | Variable input into statistical models that the analyst is attempting to predict/classify. Data in response variables is required to help the model learn. |
| *Explanatory Variable* |  | Variables input into statistical models that the analyst uses to predict the response variable. Data in these variables are required to explain the variance in the response variable. |
| *Q3* |  | 75th percentile: 75% of data is below this value |
| *Q1* |  | 25th percentile: 25% of data is below this value |
| *Interquartile Range* | IQR | Q3 – Q1 (used in outlier calculation) |
| *Range* |  | Maximum value – minimum value |
| *Summary Statistics* |  | Minimum, Q1, Median, Mean, Standard Deviation, Q3, and Maximum |
| *Synthetic Minority Oversampling Technique* | SMOTE | An algorithm that balances the dataset by increasing the number of less frequent classes to prevent bias in machine learning; does so by using explanatory values that are very similar to the original values. |
| *Pearson Correlation Coefficient* |  | Metric that calculates how variables are related; -1 = negative relationship/1 = positive relationship/0 = no relationship |
| *Mean Absolute Error* | MAE | Metric that evaluates a regression model’s performance in the response variables unit of measurement. |
| *Overfitting* |  | Occurs when a machine learning algorithm performs well on the training data but performs poorly on new data |
| *Minimum/Maximum Normalization* |  | Formula that transforms all values between 0 and 1 while maintaining the variation. |
| *Residuals* |  | Original value – predicted value |
| *Leaf Node* |  | End point of a decision tree leading to a final prediction (used in Random Forest and XGBoost) |
| *F1-Score* |  | Metric that evaluates a classification model’s performance and considers all misclassifications and correct classifications |

## Objective Summary

A local fishing business catches and sells a variety of freshwater fish and the sale price is dictated by the weight of the fish in grams and the fish species. Recently the scale to weigh the fish has been damaged and is inoperable (and the price to afford a new scale is not in the budget) and the employee who classifies the fish species has retired. Therefore, to determine the correct price to sell the new catches, the specifications of the fish from previous catches was used to predict the weight in grams and correctly classify the fish species.

The heavier the fish, the higher the price will be and the pricing model fluctuates based on how rare the species is (See table below). It is important that the average predictions of weight are within ±50 grams of the true value across all species which will be sufficient to maintain the business’ previous profit margins along with customer satisfaction.

|  |  |
| --- | --- |
| Species | Pricing |
| *Perch* | 5¢/gram + $5 |
| *Bream* | 5¢/gram + $6.50 |
| *Roach/Pike* | 5¢/gram + $8 |
| *Smelt* | 5¢/gram + $8.50 |
| *Parkki* | 5¢/gram + $10 |
| *Whitefish* | 5¢/gram + $12 |

## Fish Specifications

The last month’s catches have been recorded and consists of 159 total fish with 7 identifying measurements/classifications (i.e. variables; described in the table below). These will be used to predict the weight and/or classify the species.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Purpose | Data Type | Description |
| *Weight* | Response/Explanatory | Numeric | Weight of fish measured in grams |
| *Length1* | Explanatory | Numeric | Measurement from the nose to the beginning of the tail in centimeters |
| *Length2* | Explanatory | Numeric | Measurement from the nose to the notch of the tail (when the tail splits into two segments) in centimeters |
| *Length3* | Explanatory | Numeric | Measurement from the nose to the end of the tail (full length of fish) in centimeters |
| *Height* | Explanatory | Numeric | Measurement from the lowest point to the highest point in centimeters |
| *Width* | Explanatory | Numeric | Measurement from the left to the right in centimeters |
| *Species* | Response/Explanatory | Categorical | Classification of 7 different freshwater fish (Bream, Parkki, Perch, Pike, Roach, Smelt, or Whitefish) |

# Data Preprocessing

## Missing Values

There were no true missing values within the dataset however there was one measurement of weight that was recorded as 0. After further exploration, this was a clear data error based on the other variables having values relatively far from the minimums. This data point was removed in all analyses.

## Extreme Values

Three exceptionally heavy fish were observed and classified as outliers by the box and whisker plot (shown in the figure in section 3.1 below) which uses Q1 - 1.5\*IQR to determine low outliers and Q3 + 1.5\*IQR to determine high outliers. These outliers were either used in the analysis or removed from the analysis depending on the statistical model that was used. Certain algorithms are sensitive to outliers and this sensitivity could effect the predictions that are within the normal range of values which is why it is important to handle them appropriately.

## Derivations

The categorical variable *species* was one hot encoded in order to be used in the statistical model because Python requires categorical data to be converted into numerical data. One hot encoded variables are numeric binary variables with values of 0 or 1 (1 if the category is true/0 if the category is false). Therefore, C-1 new binary variables were created in place of species (where C is the number of categories) and are as follows: Species\_Parkki, Species\_Perch, Species\_Pike, Species\_Roach, Species\_Smelt, and Species\_Whitefish. Noticeably, Species\_Bream is not present because if all the aforementioned variables contain a value of 0, there is only 1 other option, Bream; and this is done to avoid redundant information which ultimately allows the models to better predict new, unseen data.

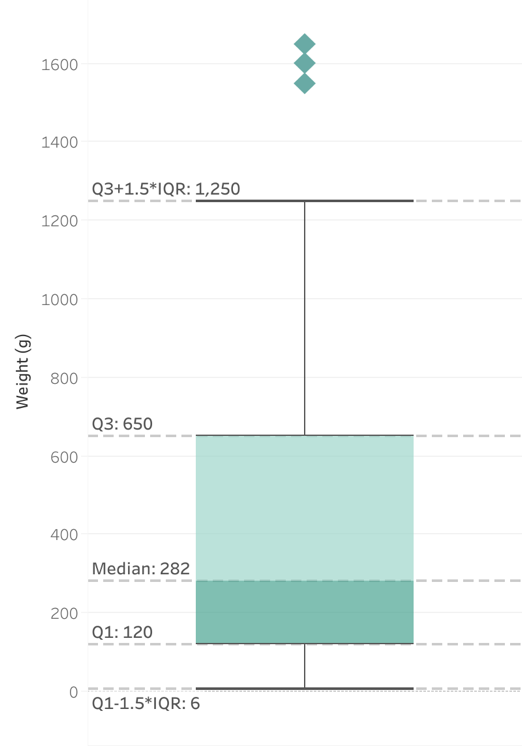
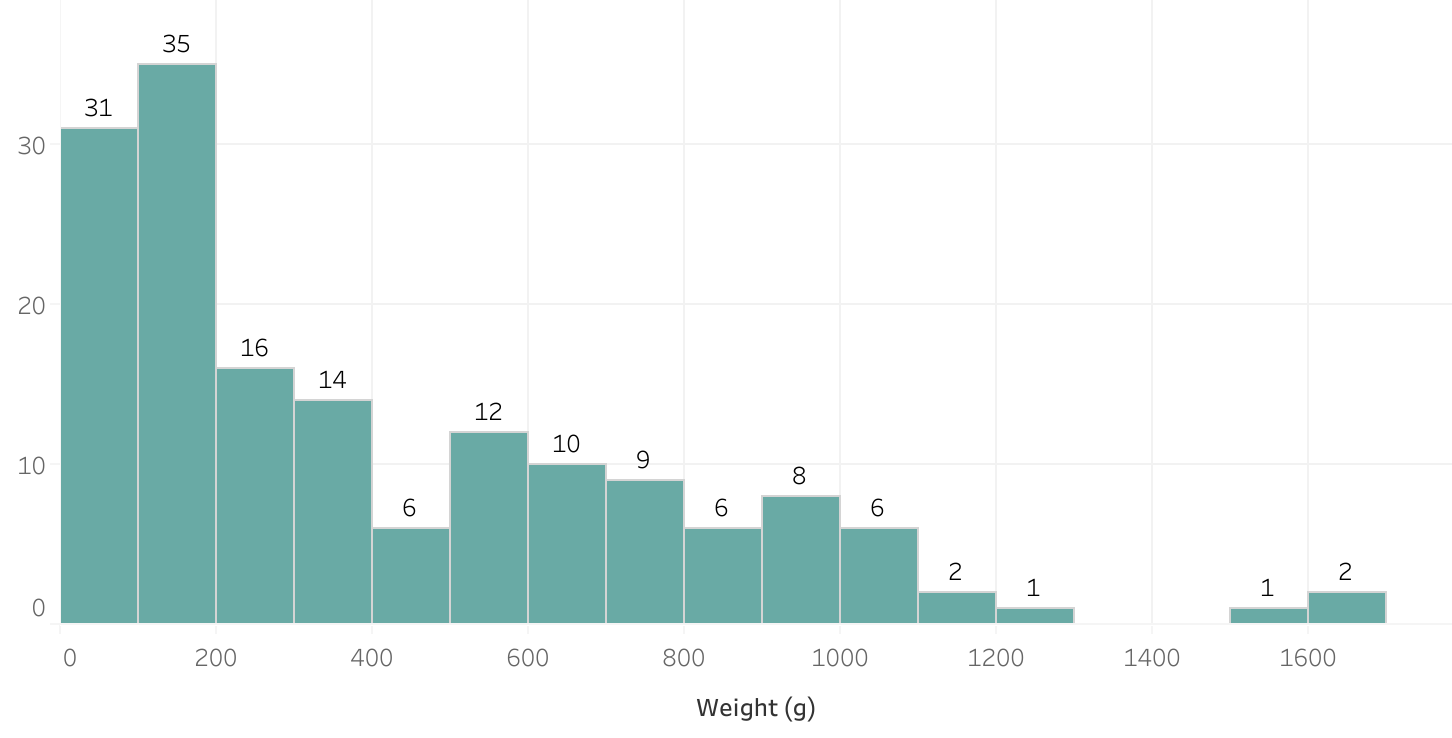
A new variable, *Length*, was created as the average of *Length1, Length2,* and *Length3* because of their obvious correlation and was tested in all models to evaluate if the predictive performance was more successful by including one variable as opposed to all three variables. It was determined that it did not improve performance in any manner, therefore, the singular *Length* variable was not used and the original three variables were used.

# Exploratory Data Analysis

## Distributions

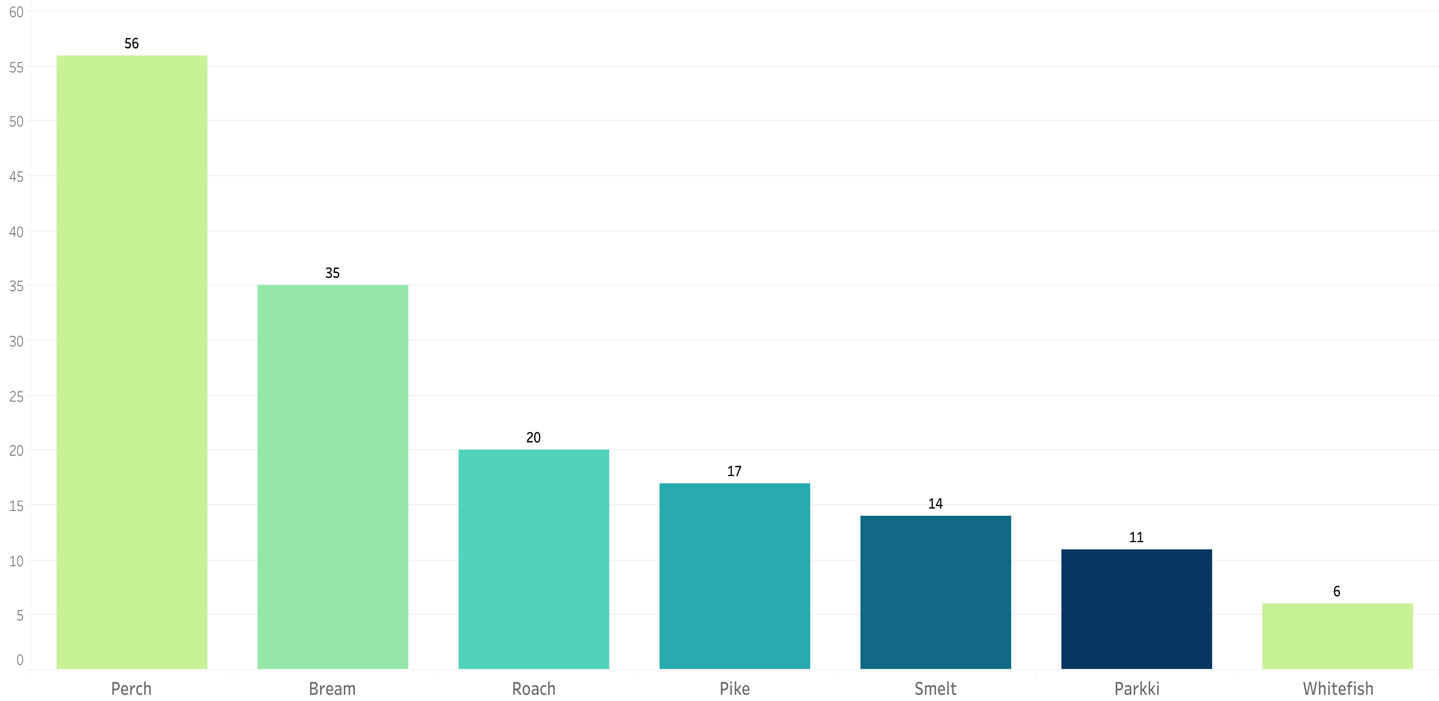
The distributions of the response variables have a higher degree of importance than the explanatory variables because the distribution of the values of Weight and/or Species could bias the model to automatically gravitate the predictions to more frequent values.

The *weight* measurements are skewed right with a range of 1,644.1, including outliers, and 1,244.1, excluding outliers. The distribution of the weight is shown below as a histogram (frequency counts of values) and a box and whisker plot (visualization of summary statistics – with data points for outliers). Transformations of *weight* were tested to give it a normal distribution, however, no transformation was successful; therefore, the skewed variable remained untouched.



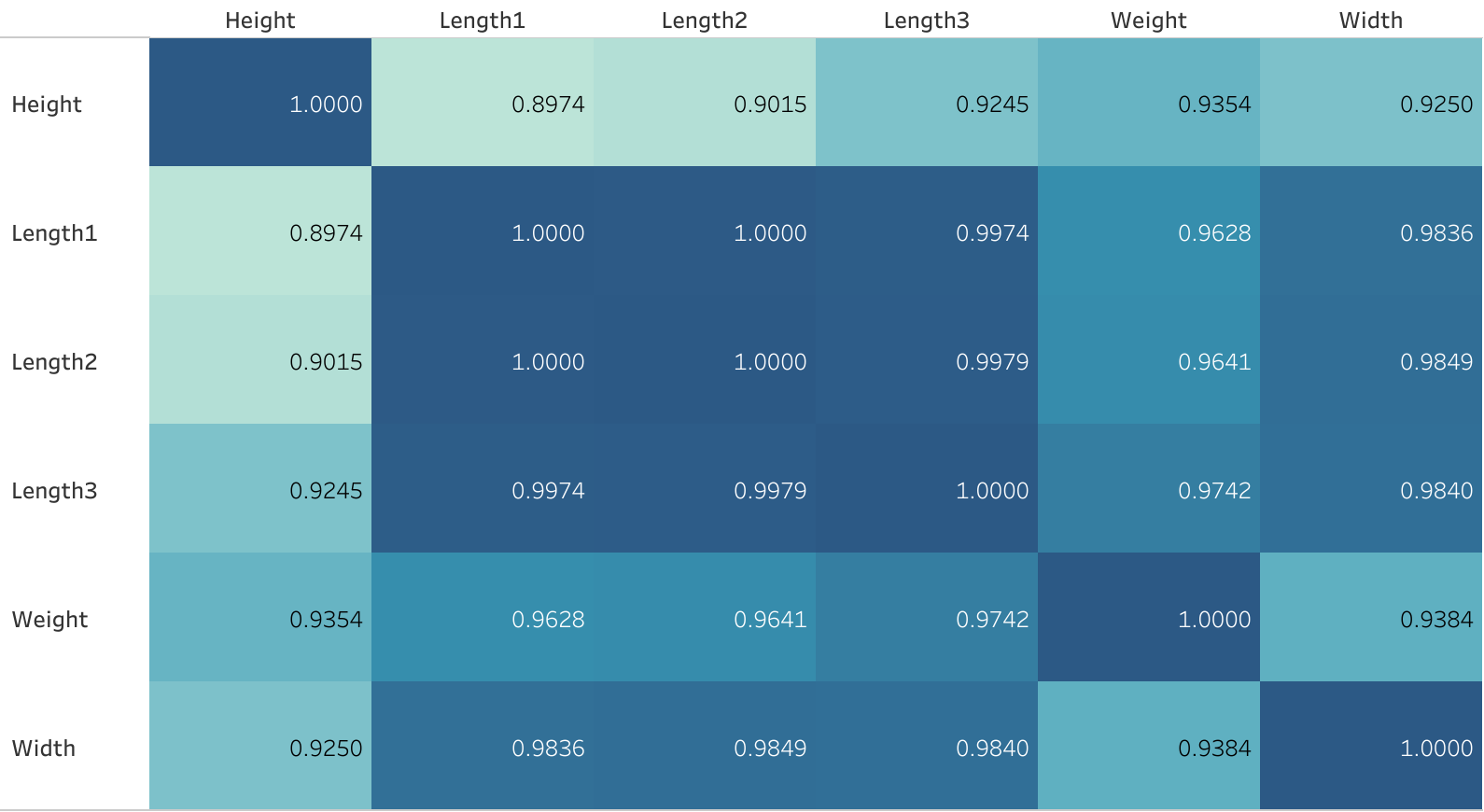
From the objective summary, the sufficient average predictions of weight are ±50; and after seeing the distribution of values and knowing the range, 50 grams is less than 5% of the range of values (excluding outliers) and less than 10% of the IQR. Therefore, it was concluded that the average difference of 50 grams between the original and predicted values would be sufficient.

The categorical variable *species* has an imbalance of classes, as seen below, with the majority falling under perch and the less frequent catches were smelt’s and whitefish. An oversampling method called SMOTE was implemented to balance the classes for all classification algorithms in order for the model to properly learn attributes of all the classes and refrain from assuming the class falls under the majority (i.e. Perch).



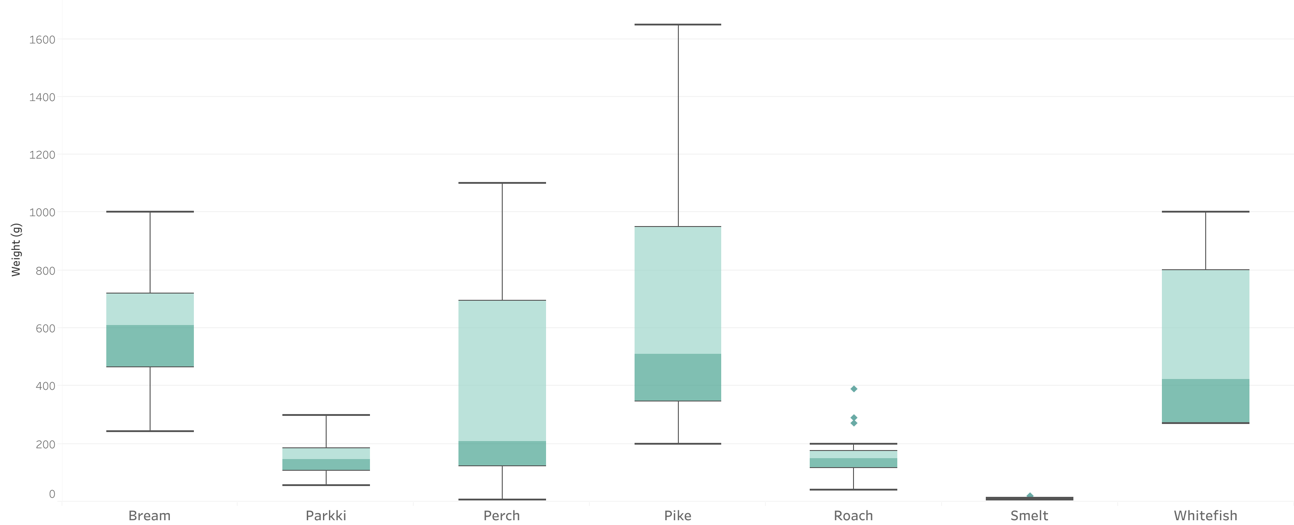
## Correlations/Relationships

All numeric variables maintain a practical and calculated relationship with *weight* (i.e. the longer, higher and wider a fish is, the heavier the fish will be) and the Pearson Correlation Coefficient matrix confirms this, as shown below.



All variables are highly correlated with *weight* which indicates a higher likelihood of successful predictive performance; however, all variables are also highly correlated with each other. This is called multicollinearity and is only problematic when interpreting and understanding relationships between variables, such as in a linear regression setting, but the current objective is solely for future predictions, so this is not a concern.

The relationship between *weight* and *species* was visualized to clarify a distinction between the several types of fish and this would determine if *species* would be a valuable explanatory variable. As seen below, we can conclude a clear distinction.



All variables should be included in the statistical models when predicting *weight* and *species* because of the clear relationships they possess. And because of the clear distinction of different fish and their sizes (as shown in the boxplot above), the numeric variables maintain valuable explanatory power for classifying *species* because of their high correlation with *weight*.

# Predictive Analytics

K-Nearest Neighbors, Random Forest, Extreme Gradient Boosting, and Support Vector Machines are the machine learning algorithms employed for the regression and classification objectives. These models were chosen because of their robust attitude to handle a variety of data and maintain accurate predictions. Each regression model’s performance was evaluated by the mean absolute error (MAE) and the classification performance was evaluated by F1-Score. All explanatory variables were used in each model and the model with the lowest MAE/Highest F1-Score was chosen for implementation.

Prior to machine learning conduction for all algorithms, the data was split into a training (70%) and testing (30%) dataset. The model learns the algorithm on the training data, the performance metric is recorded, and is then implemented on brand new data. The metric on the test data is the value used to evaluate the model’s performance and the difference of the training metric and testing metric should be minimized to prevent overfitting. (Note: Minor overfitting is expected due to the introduction of brand new data to the model.)

## Regression

### K-Nearest Neighbors

K-Nearest Neighbors (KNN) is an algorithm that uses the Euclidean distance formula to calculate how far each data point is from K data points, where K is an integer defined by the analyst and is chosen based on the performance metric.

[Click here for further KNN Explanation and Theory](https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point.)

Prior to model generation, outliers were removed because of KNN’s sensitivity to the extreme values and all explanatory variables were normalized using the minimum/maximum normalization formula. Because KNN calculates the distance between data points, normalization is important so higher values do not carry more weight than lower values.

K=2 was determined to provide the best performance and the predicted values were plotted against the original values, as seen below, to understand how well this algorithm performs.



The test MAE is within the confines of the objective, however some data points have higher residuals and are far from the reference line which is skewing the average. The model predicts lower values more accurately, which is expected with our skewed data, but has difficulty predicting larger values.

*Note: KNeighborsRegressor from the Scikit-Learn Python package was used to perform the algorithm.*

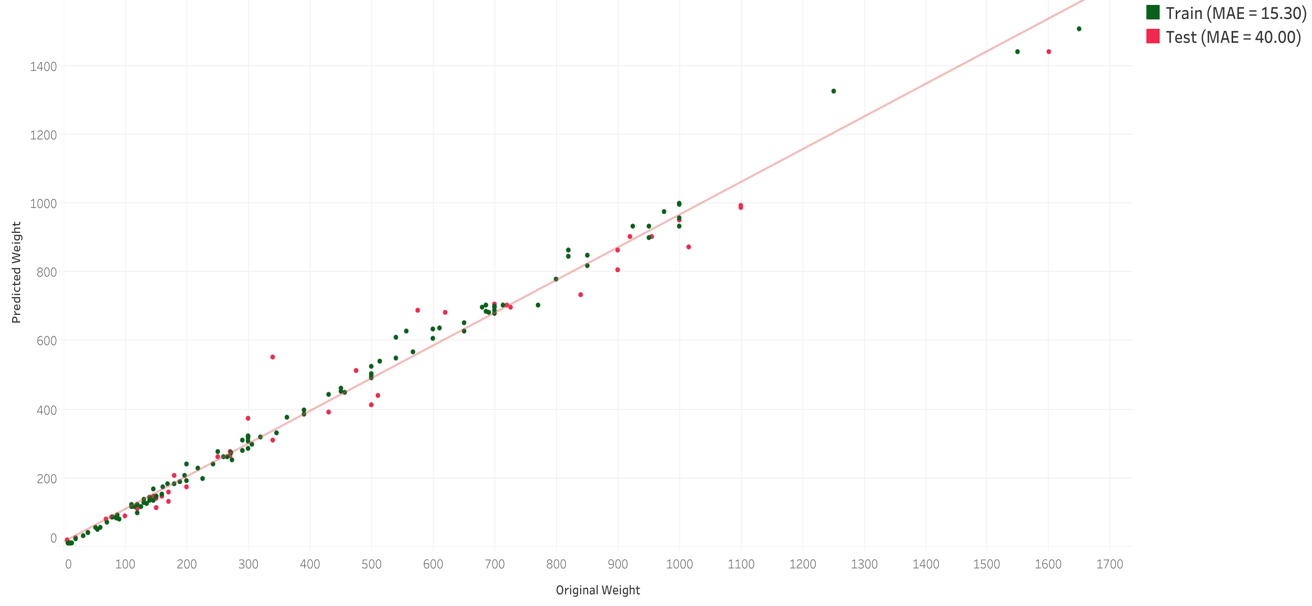
### Random Forest

Random Forest is an algorithm that combines a user defined number of decision trees and within each tree, a random sample of the input data is utilized. The collection of these trees are full of conditional statements and are used for high quality predictions.

[Click here for further Random Forest Explanation and Theory](https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.)

Outliers were not removed and no other transformations were required prior to model activation.

The parameters of the random forest model that provided the best performance were as follows: n\_estimators (number of trees) = 17/max\_depth (maximum number of levels in each tree) = 9/max\_features (maximum number of variables to use in each tree) = 8/max\_leaf\_nodes (maximum number of leaf nodes in each tree) = 45. See the predicted values plotted against the original values below to evaluate the model’s performance.



The models performance continues to remain with the objective and it is clear that the performance is more successful than KNN. The larger values continue to evade perfection but they are not significantly incorrect.

*Note: RandomForestRegressor from the Scikit-Learn Python package was used to perform the algorithm.*

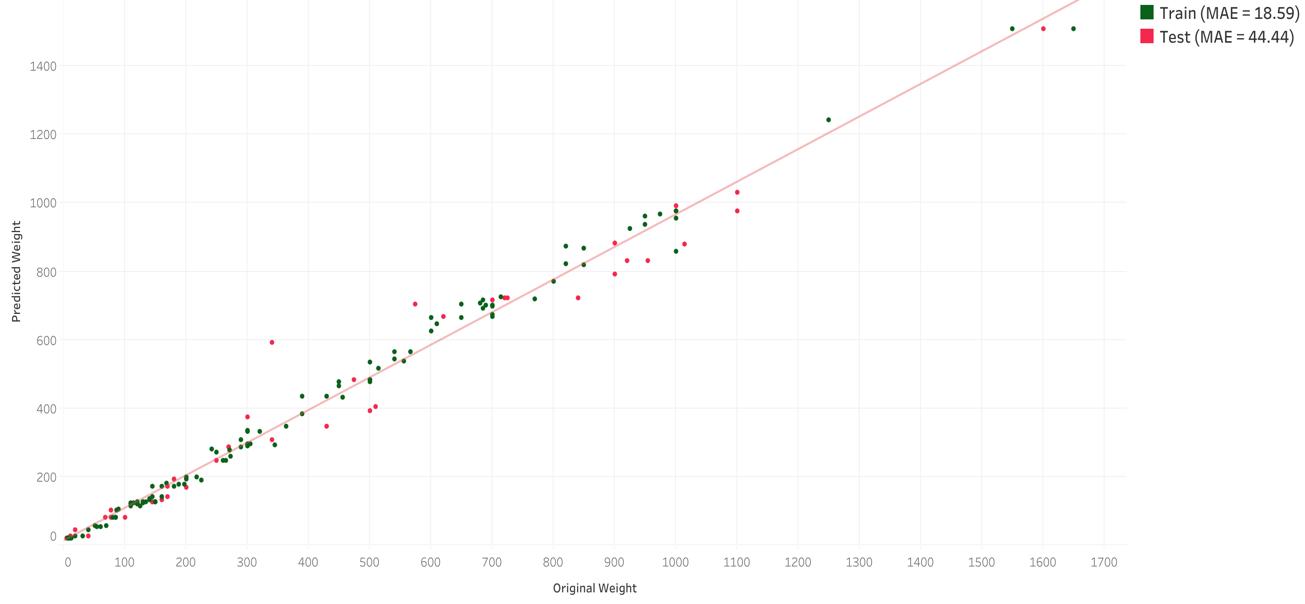
### Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is a similarly structured algorithm to random forest, however it combines a number of weaker models to obtain an overall stronger model and is one the leading machine learning models within data analysis.

[Click here for further XGBoost Explanation and Theory](https://www.nvidia.com/en-us/glossary/data-science/xgboost/)

Outliers were not removed and no other transformations were required prior to model activation.

The parameters of the XGBoost model that provided the best performance were as follows: n\_estimators (number of trees) = 20/max\_depth (maximum number of levels in each tree) = 3/learning\_rate (rate at which the algorithm descends to minimize the loss function) = 0.3/subsample (percentage of data sampled in each tree) = 0.5/colsample\_bytree (percentage of variables sampled in each tree) = 0.6. The predicted values were plotted against the original values to evaluate the model’s performance.



As before, the models performance satisfies the objective but the performance is similar to KNN and not as successful as the random forest.

*Note: XGBRegressor from the xgboost Python package was used to perform the algorithm.*

## Classification

As mentioned in Section 3.1, SMOTE was implemented for all classification algorithms to balance the dataset which will provide an opportunity for all classes to be properly represented. The parameter, n\_neighbors, was passed with a value of 2 to synthetically generate new data and works the same way as KNN. SMOTE was only applied to the training data in order to evaluate the performance on untouched, fresh data.

*Note: SMOTE from the imblearn Python package was used to perform the upsampling.*

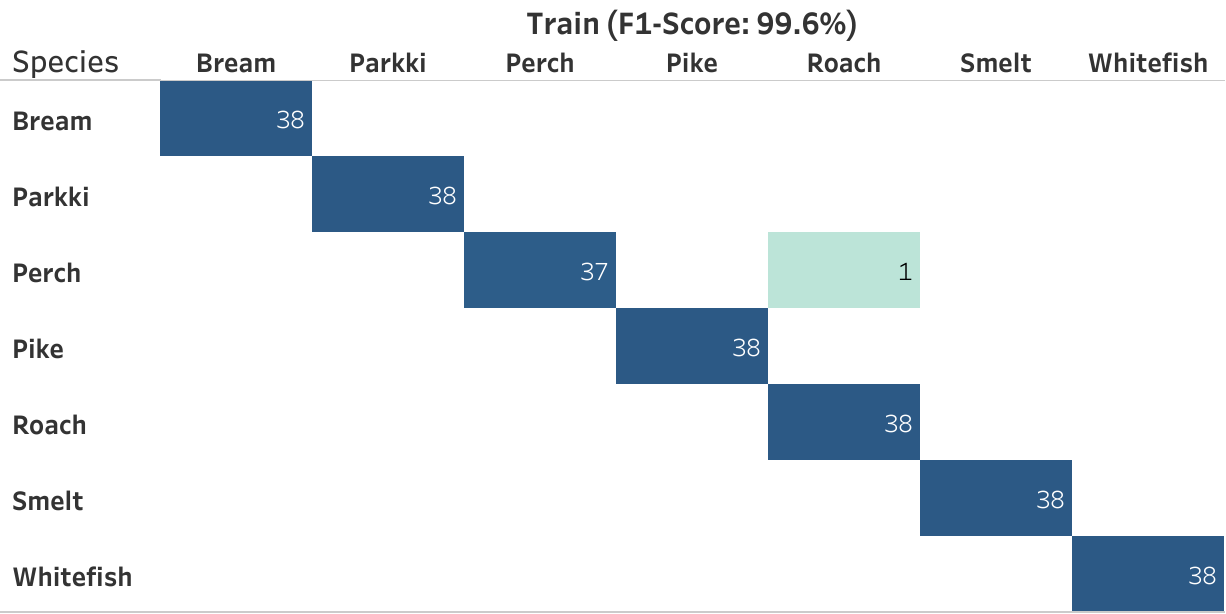
### Support Vector Machine

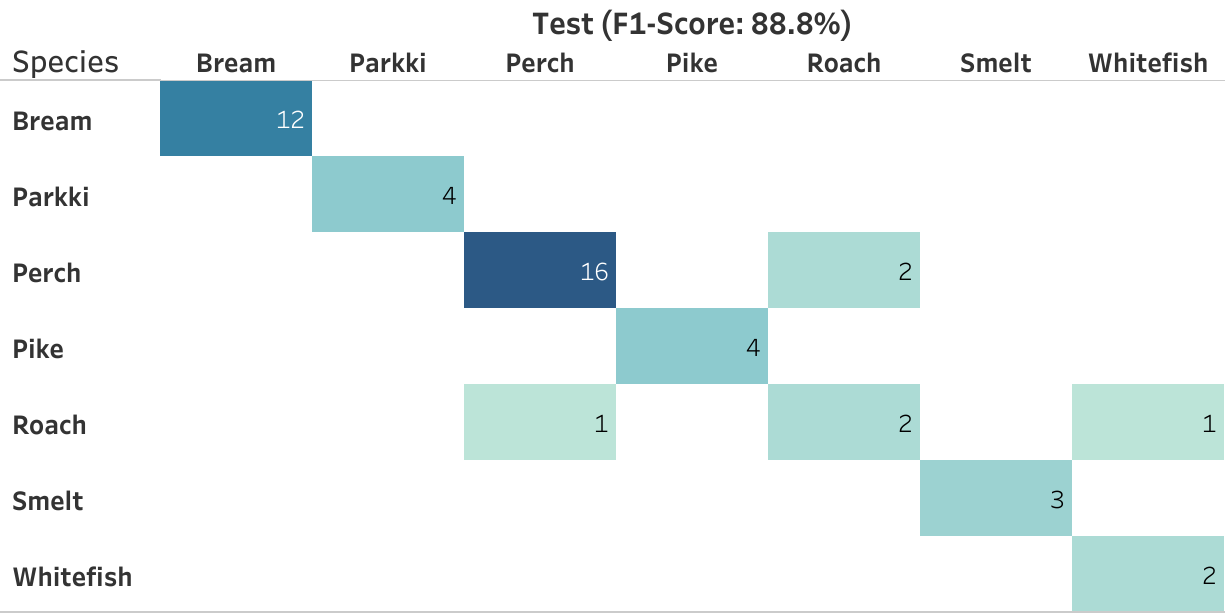
Support vector machines (SVM) use distances between data points, similar to KNN, to separate classifications by a hyperplane. The hyperplane can be manipulated by user defined parameters to improve the classifications by fitting the data.

[Click here for further SVM Explanation and Theory](https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496#:~:text=Support%20Vector%20Machine%20(the,dots%20closest%20to%20the%20line.)

Prior to model generation, outliers were removed because of the SVM’s sensitivity to the extreme values and all explanatory variables were normalized using the minimum/maximum normalization formula. Because SVM calculates the distance between data points, normalization is important so higher values do not carry more weight than lower values.

The parameters of the SVM model that provided the best performance were as follows: kernel (different shape and structure of hyperplane) = poly/degree (tunes the shape of the polynomial kernel) = 3/coef0 (integer that further tunes the shape of the polynomial kernel) = 5. The original classes were tabulated against the predicted classes to evaluate the model’s performance.





SVM misclassified 1 observation in the balanced training data and misclassified 4 observations in the test data with an F1-Score of 88.8%. Perches and Roaches are being mistaken for each other, however, the model is valuable in reaching the business goal.

*Note: SVC from the Sci-Kit Learn Python package was used to perform the algorithm.*

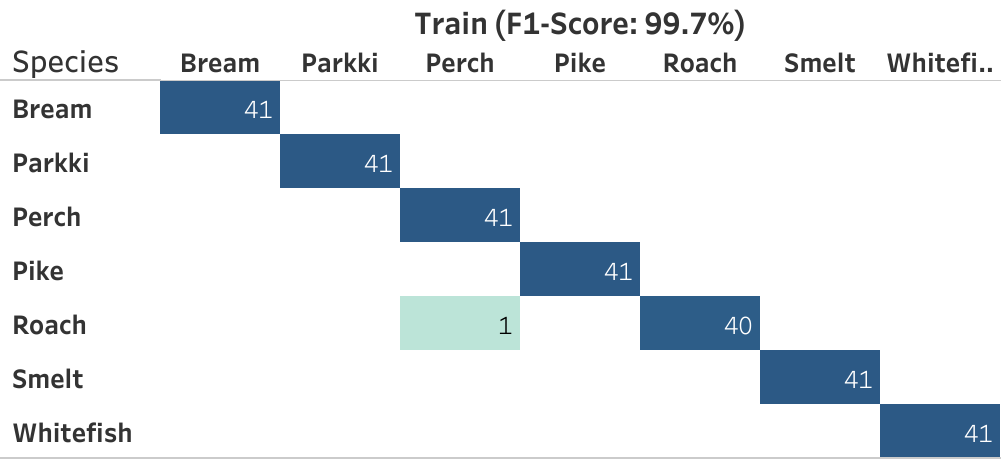
### Random Forest

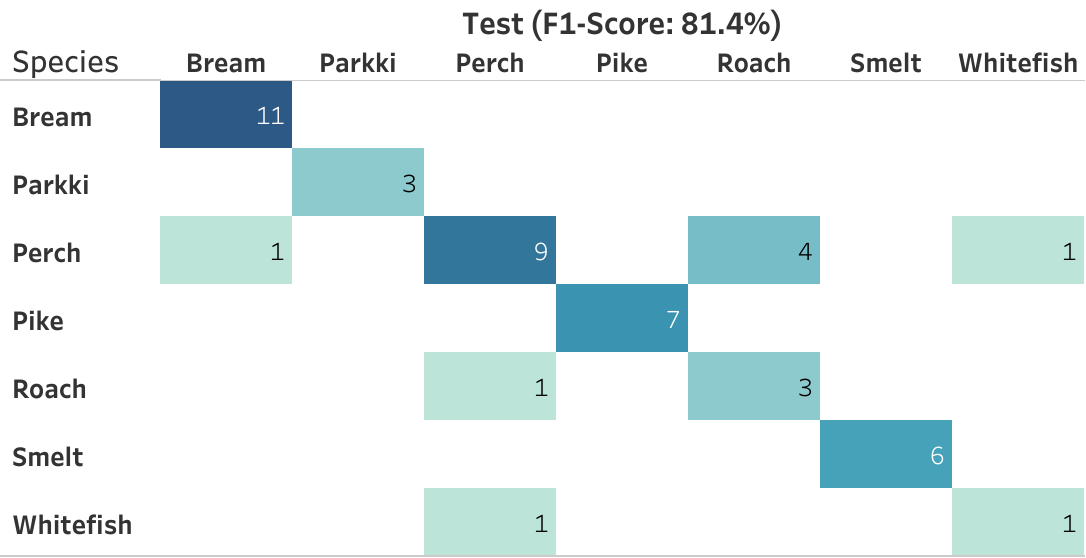
Random Forest was also used to classify the species and the theory remains the same but the output will be a class instead of a value.

[Click here for further Random Forest Explanation and Theory](https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.)

Outliers were not removed and no other transformations were required prior to model activation.

The parameters of the random forest model that provided the best performance were as follows: n\_estimators (number of trees) = 8/max\_depth (maximum number of levels in each tree) = 12/max\_features (maximum number of variables to use in each tree) = 1. See below the original classes tabulated against the predicted classes.





The Random Forest classification performance was not as successful as SVM with 9 misclassifications and an F1-Score of 81.4% in the test data.

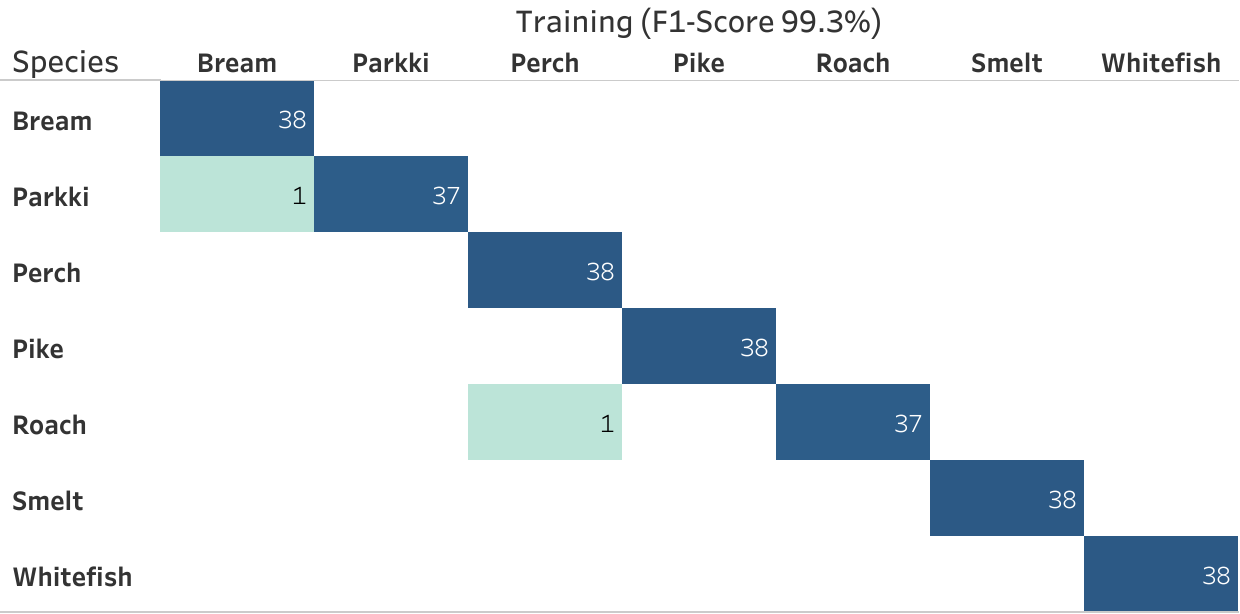
*Note: RandomForestClassifier from the Sci-Kit Learn Python package was used to perform the algorithm.*

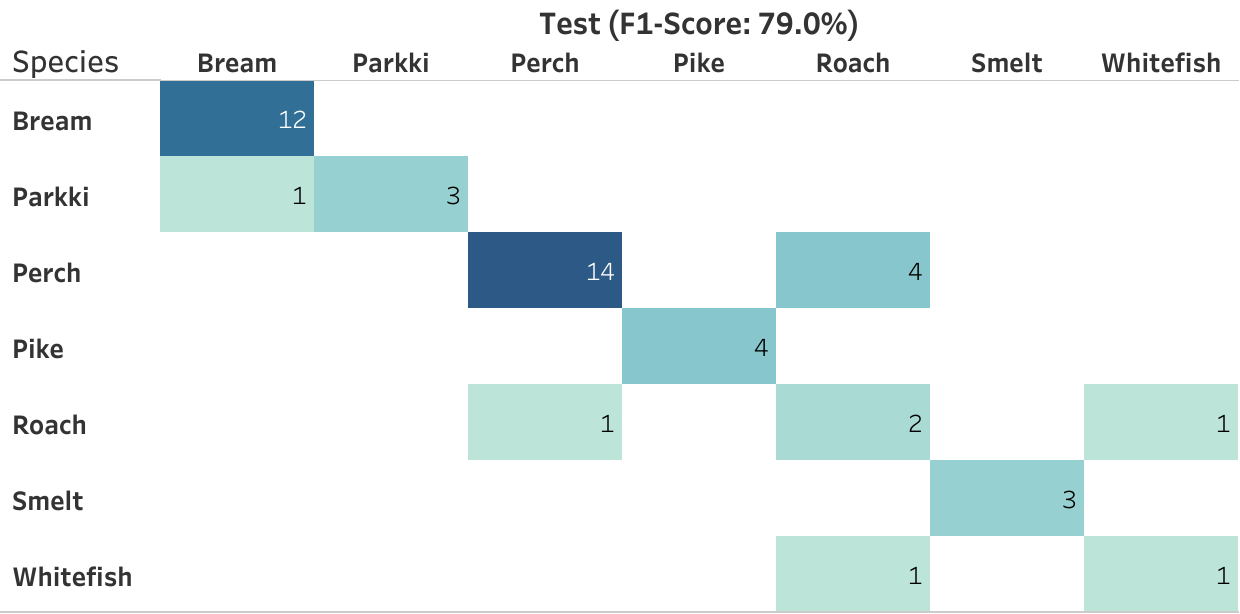
### K-Nearest Neighbors

KNN was utilized again to classify the species and the same data transformations were conducted (i.e. outlier removal and minimum/maximum normalization).

[Click here for further KNN Explanation and Theory](https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point.)

K=2 was determined to provide the best performance on both the train and test data and once again the original values were tabulated against the predicted values.





Similar to Random Forest, there were 9 misclassifications with an F1-Score of 79%.

*Note: KNeighborsClassifier from the Sci-Kit Learn Python package was used to perform the algorithm.*

# Conclusion

The regression algorithm that provided the best result was Random Forest with a MAE of 40, which is within the confines of the objective. This algorithm can handle outliers and performed better than KNN and XGBoost, therefore it would be the sought after algorithm to predict fish weights.

The classification algorithm that provided that best result was Support Vector Machines with 4 misclassifications and an F1-Score of 88.8%. It performed better than Random Forest and KNN and is the obvious choice to implement for classification of fish species.

By these implementations, business can be conducted as usual for the local fishing business until a new scale can be purchased and a fishing expert can be employed. However, in the mean time, the handler of the fish can overwrite the predictions and classifications if he/she believes they are significantly incorrect.