DD/MM/YYYY

Andy McDill

Fish Size and Species Predictions

A regression and classification project to utilize statistical models to predict weight (g) and species of fish

**Table of Contents**

[1. Project Summary 2](#_Toc120540909)

[1.1 Data Description/Transformation 2](#_Toc120540910)

[1.1.1 Weight 2](#_Toc120540911)

[1.1.2 Length1 2](#_Toc120540912)

[1.1.3 Length2 2](#_Toc120540913)

[1.1.4 Length3 3](#_Toc120540914)

[1.1.5 Height 3](#_Toc120540915)

[1.1.6 Width 3](#_Toc120540916)

[1.1.7 Species 3](#_Toc120540917)

[1.2 Correlations 4](#_Toc120540918)

[1.3 Outliers 5](#_Toc120540919)

[2. Statistical Analysis 6](#_Toc120540920)

[2.1 Linear Regression 6](#_Toc120540921)

[2.1.1 Rationale 6](#_Toc120540922)

[2.1.2 Model Preparation 6](#_Toc120540923)

[2.1.3 Model Results 8](#_Toc120540924)

[2.1.4 Model Assumptions 9](#_Toc120540925)

[2.2 K-Nearest Neighbors 10](#_Toc120540926)

[2.2.1 Rationale 10](#_Toc120540927)

[2.2.2 Model Preparation 11](#_Toc120540928)

[2.2.3 Model Results 11](#_Toc120540929)

[3. Appendix 1 13](#_Toc120540930)

[4. Appendix 2 14](#_Toc120540931)

# Project Summary

A dataset was provided that contained information, 159 records (rows) and 7 variables (columns), on fish sizes and fish species. The variables are described in [section 1.1](#_Data_Description/Transformation). A linear regression model was generated and evaluated to predict the weight of the fish in grams (g) while a K-nearest neighbor (KNN) classification model was generated to predict 7 different possibilities of fish species. The final results of the two generated models were sufficient to prohibit the comparison of additional model testing.

## Data Description/Transformation

The dataset that was retrieved off the data science forum, Kaggle (Kaggle.com), was ultimately very clean with a couple outliers and one clear data error. All variables, except for the categorical fish species, were undeniably related with each other; a [correlation matrix](#_Correlation_Matrix) was provided to show these correlations. The variables and these data anomalies, along with any transformations, are described below.

### Weight

The weight of fish was provided in grams and is the response variable (variable to be predicted) in the linear regression model. A value of 0 was discovered in the original data which was classified a clear data error because all other variables did not support it by being as significantly low, therefore, this record was removed from analysis.

Weight was log-transformed to achieve a better version of normality and was tested against other transformations (exponents, square root, etc.) but the log-transformation appeared to the best. “lWeight” was created as the transformed variable.

### Length1

This is the first of three length variables that were provided and measured in centimeters (cm). The smallest of the length variables that measures the fish from the nose to the beginning of the tail. The data was determined to have a rough normal distribution and no transformations were required.

### Length2

The second of the three length variables that were measured in cm; but this variable measures the fish from the nose to the notch of the tail. The notch of the tail is defined as the beginning of the V shape in a majority of fish tails when the tail splits into two segments. The data also had a rough normal distribution and no transformations were required.

### Length3

The last of the three length variables that were measured in cm and this measurement begins from the nose to the end of the tail; what would standardly be thought of as the full length of a fish. The data also had a rough normal distribution and no transformations were required.

*\*Note: Multiple definitions/descriptions were provided online regarding the 3 length measurements but as the data describes, length1 is the lowest measurement of each record while length3 is the highest measurement of each record; therefore, it is safe to assume the individual descriptions in sections 1.1.2-1.1.4.*

### Height

This variable was also measured in cm and is closely related to width. A description was provided onlinethat height is a percentage of Length3 but this cannot be the case because the [correlation](#_Correlation_Matrix) percentage between Height and Length3 is 70% while in fact it should be close to 100% if it was a percentage of Length3. Therefore, height can be confirmed as the measurement in cm from the bottom to the top of the fish. This variable also had a rough normal distribution and no transformations were required.

### Width

Similar to Height, this variable was measured in cm and was described as a percentage of Length3 but once again, according to the [correlation matrix](#_Correlation_Matrix), this could not be the case and it width is determined to be the measurement in cm from left to right of the fish. Width had a rough normal distribution and no transformations were required.

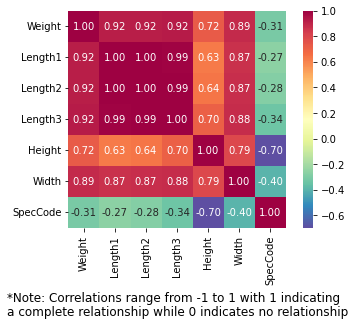
### Species

This variable is the only categorical variable in the dataset and was used as the response variable for the KNN classification algorithm. Each category was feature encoded (coded with a numerical value into a new variable “SpecCode”) in order to be input into the linear regression model. The seven fish species that were captured in the data and the value that was encoded are as such: 1=Bream, 2=Parkki, 3=Perch, 4=Pike, 5=Roach, 6=Smelt, and 7=Whitefish. The categories were misbalanced with a majority of records classified as Perch or Bream so oversampling was required prior to the classification model.

## Correlations

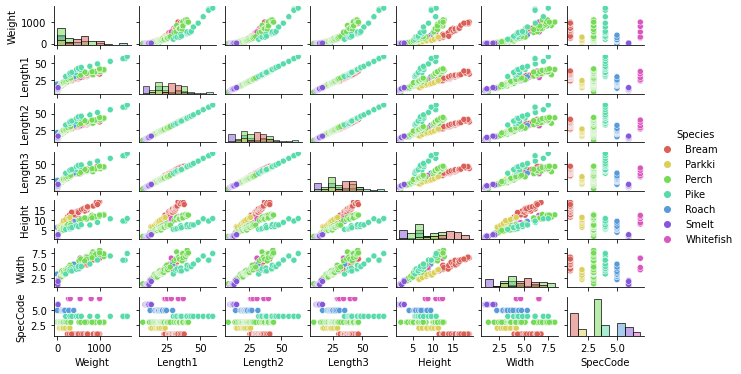
Most variables appear to be correlated and related with each other which is displayed in the correlation matrix below.

#### Correlation Matrix

**

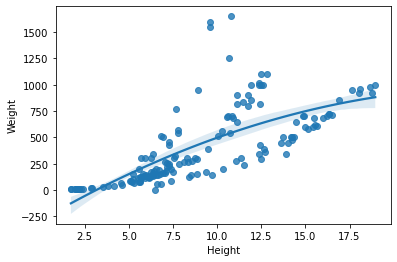
To remain in line with the correlation matrix, all relationships except for the feature encoded variable, *SpecCode*, have a pretty clear view of a linear relationship as shown in the scatter plot matrix. However, the weight and height scatter plot indicate a different type of relationship which is explored in more detail below.

#### Scatterplot Matrix



At first glance, an exponential relationship may be in play, so to verify, a scatterplot of height and weight with a regression line is displayed below.

#### Scatterplot (Weight vs Height)

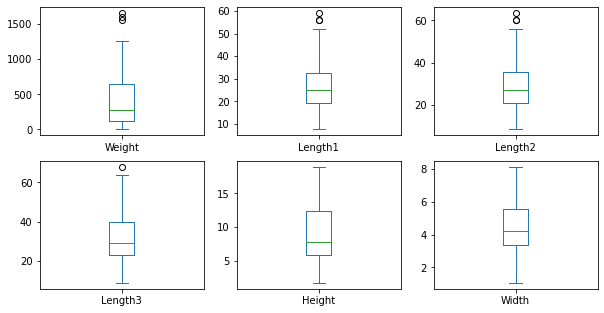


The regression line visibly displays that there are more data points in a linear fashion as opposed to an exponential relationship, therefore, we can assume a linear relationship between weight and height.

## Outliers

Z-scores were calculated for each variable to numerically determine which values are outliers in addition to the boxplot below. Three records in Weight were identified to be outliers which will be noted for the analysis. The other variables support these three records as outliers.

#### Boxplot of all variables



# Statistical Analysis

A regression and classification model were performed to predict the weight and species, respectively. Linear regression was performed to predict weight values due to the high level of relationship between weight and other variables. KNN was performed to classify the fish species. Both models performed at a high level of success, therefore, no additional models were generated for comparison.

## Linear Regression

Linear regression was decided because of the linear relationships that each variable (feature) displayed against the response variable, weight.

### Rationale

A variety of linear regression calculations exist within the predictive analytics realm; however, a standard algorithm called ordinary least squares (OLS) are used in this instance.

Since we have multiple features explaining the response variable, we will use multiple linear regression with the model shell as below:

In the equation above, *y* references the response variable, in this case, Weight. is the intercept which is not significantly relevant; to are the quantified estimations that explain the relationship between the corresponding feature and the response variable; and the features that pertain to these estimators are denoted by the to . Lastly, denotes the error term (i.e. how much variance exists in our response *y*).

As previously mentioned, OLS are used to calculate the estimators to achieve a quantified relationship between each term and the response variable, which is done by minimizing the sum of square differences between the observed and predicted values.

### Model Preparation

Standardization is typically required of the variables prior to input into the model, however, since all explanatory variables were in the same unit, centimeters, (excluding the species categorical variable), standardization was not necessary.

Due to the significant correlations between explanatory variables, interaction terms, (multiplication of two or more features), were included to account for these relationships. This is because if one variable in the interaction term changes, it would have an effect on the other. These terms are included as such:

* *Length1 \* Length2*
* *Length1 \* Length3*
* *Length2 \* Length3*
* *Length1 \* Length2 \* Length3*
* *Height \* Length1*
* *Height \* Length2*
* *Height \* Length3*
* *Width \* Length1*
* *Width \* Length2*
* *Width \* Length3*
* *Height \* Width*

Forward feature selection was processed by starting with the 6 original variables and progressively adding each interaction term to analyze the improvement of the model. Once all terms were added, backward selection took place which removed the term that contributed the least to the model until the best evaluation metrics were found.

The data was split into a 70-30 percent partition where the 70% of data was used to train the model while the 30% was used to test the model to evaluate how well it functioned. A handful of metrics were used for evaluation which consisted of:

* Coefficient of determination (): Determines the proportion of variance in the model that is explained by the explanatory variables (i.e. how well the model fits the data, 1 being perfect).
* Adjusted coefficient of determination: A more conservative calculation of , also a goodness-of-fit metric.
* Mean absolute error (MAE): Average of the absolute value of the errors (predicted values – true values). The lower the value, the better the model.
* Root mean squared error (RMSE): The standard deviation of the errors and how concentrated the predicted values are around the regression line. Low values indicate a better model as well.
* F-statistic and Prob(<F): A statistical test that indicates if at least one explanatory feature is related to the response variable, Weight. If the corresponding p-value of the F-statistic is ≤ 0.05, then at least one variable is related to the response.
* Akaike Information Criterion (AIC): A metric that evaluates how well the model fits the data that it was generated from – the lower the better.

After the best evaluation metrics were found based on the features that were included in the model, the 3 outliers were compared by inclusion and exclusion into the model. Excluding the outliers noticeably improved the model because of OLS’s sensitivity to outliers – therefore 3 records were excluded from the analysis.

### Model Results

The coefficients that were calculated from the OLS method were exponentiated because weight was log-transformed as the response variable. Since the estimators were predicting the log-transformed weight values, they need to be transformed back into their original scale. After this transformation, the final model was output to be as such:

* *Y = Weight*
* *L1 = Length1*
* *L2 = Length2*
* *L3 = Length3*
* *W = Width*
* *H = Height*
* *= Length1 \* Length3*
* *= Length2 \* Length3*
* *= Length1 \* Length2 \* Length3*
* *= Height \* Length2*
* *= Height \* Length3*
* *= Width \* Length2*

The above equation is the linear regression model, and each coefficient can be interpreted by each unit increase/decrease, the response variable (weight) will increase/decrease by X amount. For example, if Length1 increases from 20 to 21, then weight will change from 12.928 g (0.6464\*20) to 13.574 g (0.6464\*21); however, weight will also be affected by all other terms in the model. This model can be used to predict weights if we receive all 3 length variables, height, and width. The species variable was considered to be insignificant in predicting weight values.

There are two sets of evaluation metrics, one for the training data and another for the testing data. If the model worked well against the training data and did not work well on the testing data, overfitting of the model would be in play. Underfitting takes place when the model does not fit the training or testing data well. The model has been determined to be strong and effective based on the following evaluation metrics:

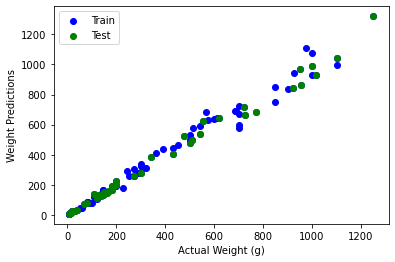
|  |  |  |
| --- | --- | --- |
| Linear Regression Strength and Effectiveness | | |
| *Set* | ***Evaluation Metric*** | ***Value*** |
| Train |  | 0.9946 |
| Adjusted | 0.9939 |
| MAE | 26.23 |
| RMSE | 40.65 |
| F-Statistic | 1592.64 |
| p-value (Prob (<F)) | 1.96e-103 |
| AIC | -166.18 |
| Test |  | 0.9956 |
| Adjusted | 0.9940 |
| MAE | 22.24 |
| RMSE | 34.86 |

### Model Assumptions

Although the model appears to be effective, there are five assumptions that must be met before we can confidently draw inferences and assume our predictions to be accurate.

The first assumption is that the response variable and explanatory variables must have a linear relationship which can be tested by plotting the actual weight values against the predicted weight values. According to the scatter plot below, we can see that the first assumption has been met decisively.

#### Predicted vs True Values



The second assumption is that the residuals (predicted values-actual values) should have a normal distribution. This assumption can be tested by plotting the residuals on a histogram and Q-Q plot; in which case it has been met and can be observed in [Appendix 1](#_Appendix_1_–).

Thirdly, little to no multicollinearity, or correlation between explanatory variables, should exist within the model. We have already determined that most of the variables are highly correlated, however, multicollinearity does not affect the accuracy and effectiveness of predictions which is the objective of this linear regression. Therefore, we can use this model with high multicollinearity for predicting new data points but we should not confidently draw inferences about the data and relationships between the response and explanatory variables.

The fourth assumption is that there should be no autocorrelation in the data which exists when the residuals are dependent on one another. This can be tested by performing a hypothesis test with the Durbin-Watson value. The null hypothesis is first determined to assume that there is auto-correlation within our data and identify our alpha value to be 0.05. The Durbin-Watson value is calculated within the OLS model summary in python and comes out to be 2.038; therefore, since it is above our alpha value, we can reject the null hypothesis and conclude that there is no autocorrelation in the data.

The last assumption is the residuals should be homoscedastic which means that the residuals should be equal across the line of regression as opposed to graphically showing a specific relationship or cluster. This can be shown by the figure in [Appendix 2.](#_Appendix_2)

## K-Nearest Neighbors

The K-Nearest Neighbors (KNN) algorithm is an unsupervised, non-parametric algorithm that is primarily used for classifying categorical data based on corresponding numerical data; however, it can also be used for regression. In this case, it is being used to classify 7 different fish species based on Length, Width, Height and Weight.

### Rationale

KNN is built on a user determined distance formula by counting the majority of categories that is closest to the specified data point. The standard distance formula is the Euclidean distance and is displayed below:

*p* and *q* are two data points in the Euclidean space that returns a numerical value indicating how far *p* is from *q*. This formula currently finds the distance in a two-dimensional space, however, it can be extrapolated into three or more dimensions which is more often the case for the KNN algorithm.

The algorithm will iterate through each individual data point, calculate the distance between all other data points, and determine the category of the individual data point of the K-nearest neighbors based on the distances (where K is user defined based on which K classifies more accurately). A confusion matrix is formulated to compare the correct predictions against the incorrect predictions, which will give us the evaluation metrics of accuracy, precision, recall, and F1-Score.

### Model Preparation

Prior to training the model, first we must normalize all numerical data values. Since KNN calculates the distance between all data points, it needs to assume that all the data is on the same scale, which is achieved through normalization.

As mentioned in section [1.1.7](#_Species), the species variable was misbalanced, meaning the majority of records were classified as Perch and Bream. KNN cannot handle misbalanced data in this way because the algorithm will be biased towards the most prominent categories. Therefore, to give the minor categories a chance, an oversampling method called SMOTE (Synthetic Minor Oversampling Technique) was performed. SMOTE inputs new records from all categories to match the maximum number of the prominent category and these records contain data that is *similar* to the original records.

### Model Results

Besides testing K values from 2 to 50, model results were evaluated with and without outliers. Regardless of the inclusion of the outliers, the best value of K is 2 based on accuracy, precision, recall, and F1-Score. See the results below:

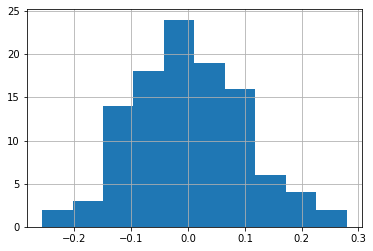
|  |  |  |  |
| --- | --- | --- | --- |
| KNN Evaluation Metrics | | | |
| *Outliers Included?* | ***Set*** | ***Evaluation Metric*** | ***Value*** |
| Yes | Train | Accuracy | 1.000 |
| Precision | 1.000 |
| Recall | 1.000 |
| F1-Score | 1.000 |
| Test | Accuracy | 0.9576 |
| Precision | 0.9651 |
| Recall | 0.9554 |
| F1-Score | 0.9547 |
| No | Train | Accuracy | 0.9964 |
| Precision | 0.9965 |
| Recall | 0.9962 |
| F1-Score | 0.9963 |
| Test | Accuracy | 0.9661 |
| Precision | 0.9701 |
| Recall | 0.9643 |
| F1-Score | 0.9637 |

Accuracy, precision, and recall are good measurements for classification evaluation however F1-Score is the best measurement because it is a harmonic mean of the precision and recall. F1-Score can be low but the accuracy could be high, but if the F1-Score is high then the accuracy will be high as well, so we know the model is working well when the F1-Score is high.

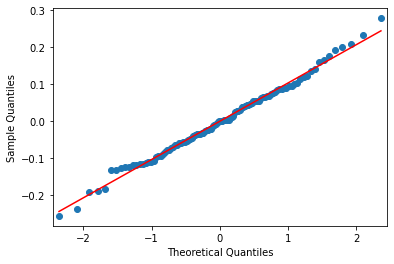
Since KNN is a formula based machine learning algorithm, it is typically sensitive to outliers. With the outliers included, we see our model predicted every class perfectly in the training data, however, when new data was applied to the model, the F1-Score fell by 5%. When we exclude the outliers, the model worked very well on our training data and still dropped by a few percentage points when applied to new data but not as much. Therefore, excluding the outliers with K=2, the best version of our KNN model will be used to avoid overfitting as much as possible.

# Appendix 1

#### Residual Normality - Histogram



#### Residual Normality - Q-Q Plot



# Appendix 2

#### Residual Homoscedasticity

