**Fish Data Analysis and Modeling Notes**

*Variables*

* Weight (dependent) – weight of fish (g)
  + Skewed
  + One value as 0
  + Correlated with all variables
* Length1 (explanatory) – Length of fish from nose to the beginning of the tail (cm)
  + Normally distributed
  + Highly correlated with Length2 and Length3 and still correlated with Height and Width
* Length2 (explanatory) – Length of fish from nose to notch of the tail (cm)
  + Normally distributed
  + Highly correlated with Length1 and Length3 and still correlated with Height and Width
* Length3 (explanatory) – Length of fish from nose to the end of the tail (cm)
  + Normally distributed
  + High correlated with Length1 and Length2 and still correlated with Height and Width
* Height (explanatory) – height in cm
  + Normally distributed
  + Mostly correlated with Width, still correlated with Length1, Length2 and Length3
* Width (explanatory) – width in cm
  + Normally distributed
  + Highly correlated with Height, Length1, Length2, and Length3
* Species (dependent/explanatory) – 7 different species of fish
  + Misbalanced – majority are one or two species
  + Mostly correlated with height, seldomly correlated with any other variable

*Outliers*

* 3 outliers found in weight exceeding 1.5\*IQR + Q3 (Other variables are consistent with the these outliers indicating they are correct)
* 1 outlier found in weight that is 0. Clear data error as Length1, Length2, and Length3 do not support it

*Data Manipulations/Transformations*

* Outlier flag created to test the sensitivity of the outliers in the model
* 0 value of weight was dropped because of data error
* Species was one hot encoded creating 6 new binary variables (species\_parkki, species\_perch, species\_pike, species\_roach, species\_smelt, species\_whitefish). Note: No species\_bream to avoid multicollinearity.
* Weight will be log transformed and left as is to test on different machine learning models. The transformation normally distributes the values a little bit better than how the original data is but is not significantly better.
* Length variable was created as average of Length1, Length2 and Length3 to test the effects of dimensionality reduction in various models
* Variables are NOT linearly related to Weight, therefore Linear Regression is not applicable. KNN, XGBoost, and Random Forest will be tested on Weight.
* SVM, Random Forest, and KNN were tested on Species.

*KNeighborsRegressor*

* All outliers and data error were removed for KNN due to the distance calculations being sensitive to outliers
* Predictor: Weight/Explanatory: Length1, Length2, Length3, Height, Width, Species\_XXX (one hot encoded species)
* Weight is log transformed which allows the test data to improve by 1 point
* Continuous variables normalized to be within the same range for distance calculations and applied to PCA to prevent multicollinearity.
  + 1st component explained 88% of variance and 2nd and 3rd components did not provide additional explanatory power
  + 1st Component and one hot encoded species variable input into KNN model
* GridSearchCV used to find the best K which is K=2 based on mean absolute error
* Training data = MSE: 1470/MAE: 21.23. Testing Data = MSE: 3936/MAE: 40.50
* Range of Weight is 1245 so being 21/40 grams of weight off on average is a valuable model and the minor overfitting is not problematic

*XGBRegressor*

* Data error was removed but outliers kept in due to lack of sensitivity with model
* Predictor: Weight/Explanatory: Length1, Length2, Length3, Height, Width, Species\_XXX (one hot encoded species)
* Log transformation of weight did not improve model performance so left as is
* GridSearchCV was used to find the best number of trees, maximum depth and learning rate. N\_estimators: 20, max\_depth=3, learning\_rate=.3, subsample=0.5 to alleviate some overfitting
* Training data = MSE: 818/MAE: 18.59. Testing Data = MSE: 4649/MAE: 44.43
* KNN appears to be less overfit with better accuracy on new data but XGBoost could still be a valuable model

*RandomForestRegressor*

* Data error was removed but outliers kept due to lack of sensitivity with model
* Predictor: Weight/Explanatory: Length1, Length2, Length3, Height, Width, Species\_XXX (one hot encoded species)
* Log Transformation of weight did not improve model performance so left as is
* GridSearchCV was used to find the best number of trees and maximum depth. N\_estimators: 17, max\_depth=9
* Training data = MSE: 733/MAE: 15.30. Testing data = MSE: 3898/MAE: 40.0
* KNN appears to be less overfit and performs about the same on new data. Either model is valuable but random forest is more robust and can handle a variety of data variability.

*SVC*

* Outliers were removed because support vector machines are sensitive to outliers
* Predictor: Species/Explanatory: Weight, Length1, Length2, Length3, Height, and Width
* Data was normalized because of the distance based measures used in support vector machines and all continuous data needed to have the same range
* SMOTE algorithm with n\_neighbors=2 was used to upsample the training data to balance the classes to eliminate bias on the less frequent classes
* Best parameters were kernel: ‘poly’, degree: 3, coef0: 5
* Training data = Macro F1: 99% (1 misclassification). Testing data = Micro F1: 89% (4 misclassifications)
* Valuable model with minor overfitting as expected with new data.

*RandomForestClassifier*

* Data error was removed but outliers were kept because Random Forest is not sensitive to outliers
* Predictor: Species/Explanatory: Weight, Length1, Length2, Length3, Height, and Width
* SMOTE algorithm with n\_neighbors=2 was used to upsample the training data to balance the classes to eliminate bias on the less frequent classes
* Best parameters were: n\_estimators: 8, max\_depth=12, max\_features=1
* Training data = Micro F1 score: 99% (1 misclassification). Testing data = Macro F1 score: 81% (9 misclassifications)
* Not as good of a performance as SVM with more overfitting and more misclassifications

*KNeighborsClassifiers*

* Outliers were removed because of the sensitivity of this algorithm
* Predictor: Species/Explanatory: Weight, Length1, Length2, Length3, Height, and Width
* Data was normalized because of the distance based measures used in KNN and all continuous data needed to have the same range
* SMOTE algorithm with n\_neighbors=2 was used to upsample the training data to balance the classes to eliminate bias on the less frequent classes
* Best parameters: n\_neighbors=2
* Training data = Micro F1 Score: 99% (2 misclassifications). Testing data = Macro F1 Score 79% (9 misclassifications).
* Does not perform as well as SVM.