STA2453 Zooplankton Classification Final Report

Hojung Kim (1004222566)

April 2, 2025

1 Introduction

This project aims to automate zooplankton classification using geometric and environmental features. Zooplankton are vital indicators of lake ecosystem health, serving as a crucial link in the food web by feeding on phytoplankton and becoming a food source for fish.

Fishing in Ontario's lakes has a significant economic impact. According to Ontario News (https://news.ontario.ca/en/release/1005496/ontario-boosting-its-fish-populations), more than 1.5 million anglers contribute $1.6 billion annually to the economy.

2 Data

I used data from the Ministry of Natural Resources and Forestry, Ontario. The dataset consists of .tif mosaics containing zooplankton images, along with geometric and environmental features. For example, each plankton image includes attributes such as transparency, symmetry, latitude, and longitude.

The classification model will use the "Class" column, focusing on classifying seven classes: Calanoid_1, Cyclopoid_1, Bosmina_1, Harpacticoida, Chironomid, Chydoridae, and Daphnia.

The data comes from two lakes in Ontario: Lake Huron and Lake Simcoe. However, the dataset is highly imbalanced, and some geometric features and labels may be missing.

3 Methods

3.1 Data processing

Before applying classification models, I merged all available data from Lake Simcoe for training and testing.

As shown in the Distribution of calss histogram below, the `'TooSmall'` class accounts for 51.25% of the dataset. Since this class provides no meaningful information about plankton (as these samples were too small to classify), I decided to exclude them to prevent introducing significant noise into the model.

There were also some outliers that negatively affected model performance. As you can see in the graph, because of few outliers regular sized planktons cannot be shown in histograms. I removed them for two main reasons:

1. They are not plankton but misclassified source data.
   For example, after removing the "TooSmall" class, the image with the largest `Area..ABD.`

value is from the file `20180529_Simcoe_200_2mm_rep2_redo_000002.tif` and is labeled `'CountGT500'`. When cropping other `'CountGT500'` images from the same `.tif` file, one appears extremely large and clearly not a plankton. You can verify this in the GitHub repository: `crop_images/big_particles/20180529_SIMC_200_2mm_rep2_redo_KG_data`

2. Although some large plankton species exist, most of the top 100 largest images by area are clearly not relevant:
    - 54% are labeled as `Floc_1`
    - 25% as `Bubbles`
    - 20% as `CountGT500`

Since Floc refers to clusters of aggregated particles (not actual organisms), removing these samples does not interfere with our goal of classifying the seven target classes.

3.1.1 Area Threshold Justification

I set the exclusion threshold at:

$$\text{Area..ABD.} > 2{,}000{,}000$$

Here's why:

- The largest target class is `Calanoid_1`, which is typically less than 1.5 mm in length (USGS Fact Sheet)
- Assuming a circular or elliptical shape, we can estimate the area as:

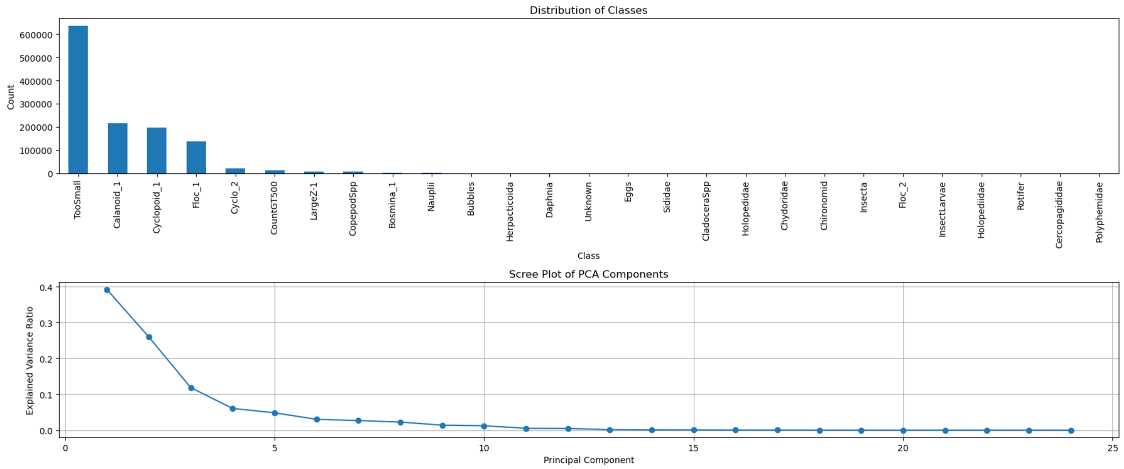
$$\text{Area} = \frac{\pi}{4} \cdot \text{length}^2 \approx \frac{\pi}{4} \cdot (1.5)^2 \approx 1.76 \text{ mm}^2$$

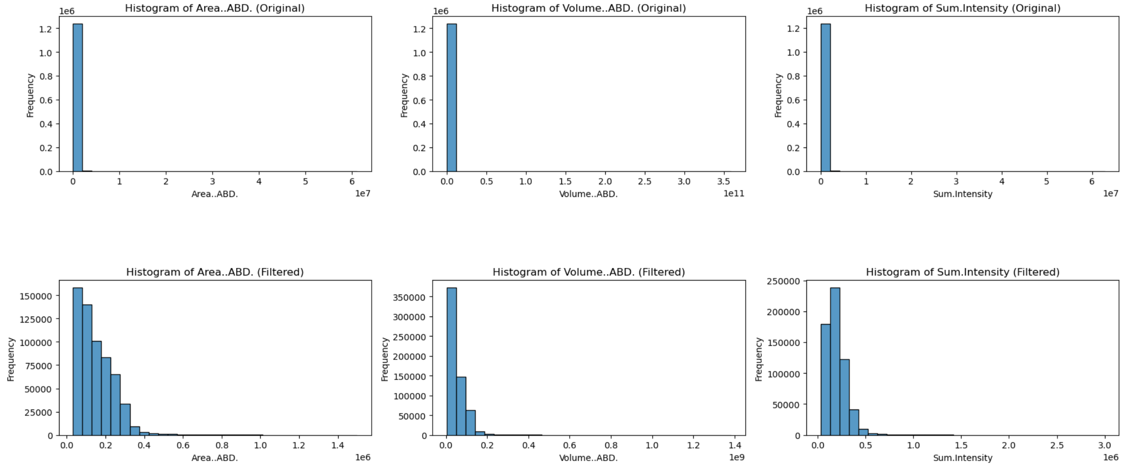
- Using a slightly more generous threshold of 1.5 mm^2, this translates to an `Area..ABD.` of about 1,500,000 in our data's scale.
- This threshold corresponds to the 99.53rd percentile, so removing these outliers excludes only the top 0.47% of the data.

3.1.2 Feature Scaling and Dimensionality Reduction

Since many features are heavily right-skewed, I applied a log transformation to positively skewed data. Additionally, I performed standardization since most features have a large range.

The covariance matrix revealed some highly correlated features, which could lead to multicollinearity. To address this, I used Principal Component Analysis (PCA) with 10 principal components. The number of components was determined using the elbow point from the scree plot, as shown in the graph below.





3.2 Models

I will test various statistical and machine learning models, including logistic regression and XGBoost.

The goal is to classify the seven plankton classes of interest while also identifying “Others” for plankton outside this list. One approach to achieving this is hierarchical classification. This involves training a binary classifier to differentiate Known vs. Others, and if a sample is classified as Known, a second classifier predicts the specific plankton type. Therefore there will be three steps for models, which are train model for binary classification, train model for multiclass classification, and combine them.

Eventually we split dataset into two. First we need to split the dataset into two, one will be used to train binary and multiclass model and the other will be used to test with combined model of binary and multiclass. To do this, we can avoid data double dipping. However, I decided to use same data for binary and multiclass training since we have fully independent test set for combined model testing.

Given the class imbalance in the data, especially among the seven focal plankton classes, I applied SMOTE (Synthetic Minority Oversampling Technique) to augment underrepresented classes including `'Bosmina_1'`, `'Herpacticoida'`, `'Chironomid'`, `'Chydoridae'`, `'Daphnia'`. I manually adjusted the target sizes for oversampling to avoid introducing too much duplication and noise, which would degrade performance.

```
SMOTE Results Table:
           Class  Before SMOTE  After SMOTE
0     Calanoid_1        137869       137869
1    Cyclopoid_1        126557       126557
2      Bosmina_1          1859        18000
3  Herpacticoida           371         3700
4        Daphnia           368         3680
5     Chydoridae            33          330
6     Chironomid            23          230
```

Now I will train logistic regression and XGBoost for binary classification and multiclass classification. After we finish training, we will try some combinations of the models in hierarchical classification.

3.2.1 Logistic Regression

Logistic regression requires several assumptions:

1. Linearly Separable Features: Zooplankton features are not linearly separable, but I will use logistic regression as a baseline due to its simplicity, interpretability, and ability to handle imbalanced classes.

2. No Multicollinearity: Since some features are highly correlated, I applied PCA as part of exploratory data analysis (EDA). The scree plot analysis suggested using seven principal components.

3. Standardized Features: Since feature ranges vary significantly, I standardized all features.

3.2.2 XGBoost

As you can see in Result section, since logistic regression shows bad performance in multiclass classification and has potential risk in binary classification, I moved on tree based model, specifically XGBoost.

There are some reasons I choose XGBoost. 1. XGBoost does not require strict distributional assumptions. 2. XGBoost handles nonlinear data well and interact between features automatically. 3. XGBoost handles imbalanced calsses better than other tree based models. 4. XGBoost is also efficient and scales well with large datasets. Since our data has wide range of features as well as highly imbalanced classes, XGBoost can help increase model performance.

4 Results

4.1 Logistic Regression

4.1.1 Binary Classification

Logistic regression performed reasonably well in distinguishing between Known and Other plankton classes, achieving an accuracy of 88%. It had a high recall (0.99) for the `'Known'` class but much lower recall (0.64) for the `'Other'` class. This means many samples from outside the target classes were incorrectly classified as `'Known'`. While the model is interpretable and fast, its linear decision boundary likely limited its ability to handle complex feature relationships.

4.1.2 Multicalss Classification

When applied to the seven `'Known'` plankton classes, logistic regression performed poorly. It achieved an overall accuracy of 55%, with especially low F1-scores for rare classes such as Chironomid, Chydoridae, Daphnia, and Herpacticoida. These results indicate that logistic regression is not suitable for heavily imbalanced or nonlinearly separable data in multiclass settings.

```
Logistic Regression Binary Classification Report:
              precision    recall  f1-score       support
0              0.957513  0.641950  0.768602  29454.000000
1              0.862104  0.987435  0.920523  66771.000000
accuracy       0.881684  0.881684  0.881684      0.881684
macro avg      0.909808  0.814692  0.844563  96225.000000
weighted avg   0.891308  0.881684  0.874021  96225.000000

Logistic Regression Multiclass Classification Report:
               precision    recall  f1-score       support
Bosmina_1       0.406402  0.899142  0.559786    466.000000
Calanoid_1      0.876004  0.543240  0.670611  34320.000000
Chironomid      0.000957  0.571429  0.001912      7.000000
Chydoridae      0.000194  0.200000  0.000389      5.000000
Cyclopoid_1     0.776055  0.562221  0.652054  31774.000000
Daphnia         0.011489  0.606742  0.022552     89.000000
Herpacticoida   0.009033  0.609091  0.017803    110.000000
accuracy        0.554927  0.554927  0.554927      0.554927
macro avg       0.297162  0.570266  0.275015  66771.000000
weighted avg    0.822426  0.554927  0.658948  66771.000000
```

4.2 XGBoost

4.2.1 Binary Classification

XGBoost substantially outperformed logistic regression in binary classification, reaching an accuracy of 91%. It maintained high precision and recall across both `'Known'` and `'Other'` classes. The model was better able to identify and separate `'Known'` samples, reducing false positives and minimizing downstream misclassification in the hierarchical pipeline.

4.2.2 Multiclass Classification

XGBoost achieved excellent performance when classifying the seven `'Known'` plankton classes directly. The overall accuracy was 91%, and class-level precision and recall were high even for minor classes such as Bosmina_1, Daphnia, and Herpacticoida. These results highlight XGBoost's capacity to model nonlinear relationships and effectively handle class imbalance.

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits

XGBoost Ninary Classification Report
              precision    recall  f1-score       support
0              0.940938  0.764820  0.843787  29454.000000
1              0.904170  0.978823  0.940017  66771.000000
accuracy       0.913318  0.913318  0.913318      0.913318
macro avg      0.922554  0.871821  0.891902  96225.000000
weighted avg   0.915424  0.913318  0.910561  96225.000000
```

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
XGBoost Multiclass Classification Report:
               precision    recall  f1-score       support
Bosmina_1       0.818363  0.879828  0.847983    466.000000
Calanoid_1      0.918771  0.921474  0.920120  34320.000000
Chironomid      0.250000  0.142857  0.181818      7.000000
Chydoridae      0.000000  0.000000  0.000000      5.000000
Cyclopoid_1     0.914815  0.912224  0.913518  31774.000000
Daphnia         0.473684  0.404494  0.436364     89.000000
Herpacticoida   0.380952  0.290909  0.329897    110.000000
accuracy        0.914903  0.914903  0.914903      0.914903
macro avg       0.536655  0.507398  0.518529  66771.000000
weighted avg    0.914569  0.914903  0.914711  66771.000000
```

4.3 Hierarchical Classification

To combine the benefits of the binary/multiclass separation, I implemented a hierarchical classification structure. This reduced the confusion between target and non-target classes by filtering out irrelevant samples early.

Three combinations were tested:

- Logistic → Logistic: Accuracy = 58% Moderate improvement over flat logistic regression, but still limited by both models' weaknesses.
- Logistic → XGBoost: Accuracy = 82% Significant improvement, as the more capable XGBoost model handled multiclass classification after a basic filtering step by logistic regression.
- XGBoost → XGBoost: Accuracy = 85% This was the best-performing configuration. With accurate filtering and robust multiclass prediction, this model achieved the highest precision and recall across the board.

```
Final Hierarchical Classification Performance for Logistic Regression:
              precision  recall f1-score support
Bosmina_1         0.241  0.9055   0.3807     550
```

```
Calanoid_1       0.7863  0.5405   0.6406   43025
Chironomid       0.0008  0.7143   0.0016       7
Chydoridae          0.0     0.0      0.0       5
Cyclopoid_1      0.6792  0.5619    0.615   39651
Daphnia          0.0076  0.5051    0.015      99
Herpacticoida    0.0081  0.6825   0.0161     126
Other            0.9587  0.6419    0.769   36818


accuracy                          0.5803
macro avg        0.3352  0.5689   0.3047  120281
weighted avg     0.7998  0.5803    0.669  120281

Final Hierarchical Classification Performance for Logistic Regression + XGBoost:
              precision  recall f1-score support
Bosmina_1        0.4227  0.8855   0.5723     550
Calanoid_1       0.8195  0.9083   0.8616   43025
Chironomid        0.037  0.1429   0.0588       7
Chydoridae          0.0     0.0      0.0       5
Cyclopoid_1      0.7719  0.9025   0.8321   39651
Daphnia          0.1542  0.3131   0.2067      99
Herpacticoida    0.2245  0.3492   0.2733     126
Other            0.9587  0.6419    0.769   36818


accuracy                          0.8236
macro avg        0.4236  0.5179   0.4467  120281
weighted avg     0.8434  0.8236    0.821  120281

Final Hierarchical Classification Performance for XGBoost:
              precision  recall f1-score support
Bosmina_1        0.6088  0.7018    0.652     550
Calanoid_1       0.8514  0.9055   0.8776   43025
Chironomid          0.0     0.0      0.0       7
Chydoridae          0.0     0.0      0.0       5
Cyclopoid_1      0.8091  0.8953     0.85   39651
Daphnia          0.1869   0.202   0.1942      99
Herpacticoida    0.3462  0.2857    0.313     126
Other            0.9407  0.7615   0.8417   36818


accuracy                          0.8558
macro avg        0.4679   0.469   0.4661  120281
weighted avg     0.8625  0.8558   0.8553  120281
```

5 Conclusion

This project explored the use of machine learning to automate the classification of zooplankton based on geometric and environmental features. Careful data preprocessing was essential, including the removal of noisy classes, filtering out outliers, applying log transformation and standardization, and addressing class imbalance through targeted oversampling. These steps helped improve the reliability of the models.

Logistic regression served as a useful baseline model due to its simplicity and interpretability. However, it showed clear limitations when dealing with nonlinear feature relationships and severely imbalanced class distributions. In contrast, XGBoost performed significantly better in both binary and multiclass classification tasks. Its ability to model complex interactions and its robustness to data imbalance made it a strong candidate for this classification problem.

To improve classification performance further, a hierarchical classification framework was implemented. This structure first identified whether a sample belonged to one of the target plankton classes before applying a second model to predict the specific class. The hierarchical approach helped reduce confusion between relevant and irrelevant samples and led to notable performance improvements. The best results were achieved when XGBoost was used for both stages of the hierarchical pipeline, resulting in an overall accuracy of 85 percent.

In summary, the combination of structured preprocessing, hierarchical classification, and XGBoost produced the most effective model for zooplankton classification. Future extensions could explore the use of raw image data through deep learning, incorporate spatial and seasonal features, or apply cost-sensitive learning to further improve classification of rare plankton classes.