Fish species detection

Introduction

The Fish Market dataset is a collection of information about various fish species, along with their physical characteristics. It is intended for classification tasks, where the goal is to predict the species of a fish based on its measurements such as length, height, and width.

Link for the source (Kaggle) Link for my github repository

Data explanation

The dataset comprises **seven columns**, six of which represent the input features, while the seventh serves as the target variable. Each row corresponds to a single fish instance. The attributes are as follows:
Weight: A numerical value representing the fish's weight in grams.
Length1: The vertical length of the fish in centimeters.
Length2: The diagonal length in centimeters.
Length3: The cross length in centimeters.
Height: A numerical value indicating the fish's height in centimeters.
Width: The diagonal width of the fish in centimeters.
The dependent variable is the **Species**, a categorical feature classifying each fish into one of seven species: Perch, Bream, Roach, Pike, Smelt, Parkki, and Whitefish.
The dataset includes a total of 159 samples, evenly distributed across the seven species categories, with no missing values. Since the target variable (Species) is originally represented as categorical text labels, I applied label encoding to convert these labels into numerical format to facilitate model training.
Depending on the algorithm, I applied standardization to the input features particularly for models that assume normally distributed data such as K-Nearest Neighbors (KNN) and Artificial Neural Networks (ANN).
I also applied some statistics to the data on the dataset as you can see in the following table:

Feature	Mean	Std Dev	Min	Max	Count	Range
Weight	398.3	357.98	0	1650	159	1650
Length1	26.25	10	7.5	59	159	51.5
Length2	28.42	10.72	8.4	63.4	159	55
Length3	31.23	11.61	8.8	68	159	59.2
Height	8.97	4.29	1.7	19	159	17.23
Width	4.42	1.69	1.1	8.14	159	7.09

As we can see, the average fish weight is approximately 398 grams, with a wide range from 0 to 1650 grams, indicating a high variance in fish sizes. The standard deviations are also relatively large across all features, such as 357 for weight and 11 for Length3, suggesting significant variability within the data. Length measurements (Length1, Length2, Length3) show means between 26 to 31 cm, with ranges exceeding 50 cm. Height and Width are more compact features, averaging around 9 cm and 4 cm respectively, yet still show variation among species. All features have 159 non-null values,

confirming that there are no missing data points. Overall, the statistics suggest that the dataset includes a diverse set of fish with varying dimensions and body sizes.

Data preprocessing

- Label encoder: Applied to convert categorical class labels into numerical values, enabling the model to process and interpret the target variable effectively.
- Standard scalar: Utilized to standardize the feature values, ensuring they are on a comparable scale, which enhances the performance and stability of models that are sensitive to feature magnitudes.
- Train_test_split: to evaluate the generalization performance of the trained models, this helps in partitioning the dataset into two subsets: a training set and a testing set. Specifically, 70% of the data was allocated for training the machine learning models, while the remaining 30% was reserved for testing. This stratified split ensures that the model learns from a substantial portion of the data while still providing a meaningful, unseen subset for performance evaluation. The 70:30 ratio is a standard practice that balances learning quality with reliable validation.

Machine learning models

Model	Accuracy	Macro Precision	Macro Recall	Macro F1-score
Logistic Regression	96%	97%	96%	96%
ANN	92%	81%	81%	81%
KNN	83%	76%	72%	72%
Decision Tree	79%	71%	72%	71%
SVM	79%	64%	63%	62%
Random Forest	77%	74%	63%	66%
Naive Bayes	60%	59%	62%	59%

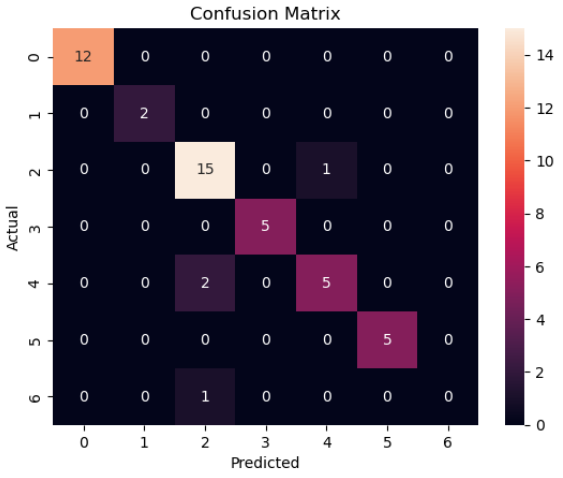
Performance

Among the evaluated models, Logistic Regression achieved the highest accuracy, attaining a score of 96%, indicating its strong performance on this dataset. In contrast, the Naive Bayes classifier yielded the lowest accuracy, with a score of 60%, reflecting its limitations in handling the distributional characteristics of the data.

Data visualization

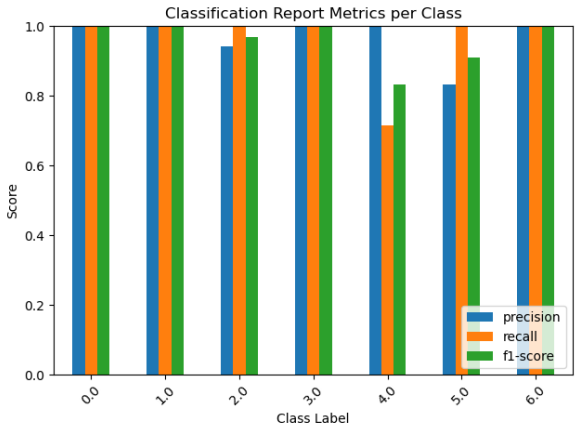
To better understand the performance of our classification models beyond numeric metrics, we used **confusion matrix** visualization. A confusion matrix is a powerful and intuitive graphical representation used to evaluate the performance of a classification model. It shows the relationship between actual labels and predicted labels across all classes in the dataset.

Each row of the matrix represents the actual class, while each column represents the predicted class. The



diagonal cells represent the number of correctly classified instances for each class, while the off-diagonal cells show misclassifications where the model predicted the wrong class.
The above plot is an example of a **confusion matrix generated for the Artificial Neural Network (ANN)**.

Another visualization used in this analysis is the **bar plot,** which clearly displays the performance metrics precision, recall, and F1-score for each class individually**.** This plot provides a straightforward comparison between how well the model performs across different classes, helping identify any imbalances or weaknesses in classification. By visualizing these metrics side by side, it becomes easier to interpret model behavior and spot any class that may require further improvement or tuning.
Here an example for the **logistic regression**.



Conclusion

I picked the Fish Market dataset because it is simple yet diverse, making it ideal for classification. It contains physical measurements of various fish species, something relatable and easy to visualize. This dataset helps in practicing how real-world data can be used to predict categories. In real-life, this kind of data can assist fish sellers, market researchers, or food inspectors. It can help automate fish classification to improve sorting and reduce manual errors. The measurements are continuous, which makes the dataset suitable for many algorithms. Also, there are no missing values, which saved time in cleaning and preprocessing. I used label encoding to handle the categorical target (fish species).
Standardization was important for models like ANN and KNN.

The train-test split (70:30) helped in checking how well the model performs on unseen data. Out of all models tested, Logistic Regression performed the best with 96% accuracy. It also had strong precision, recall, and F1-score, which means it made few mistakes. This shows that sometimes simple models can outperform more complex ones. ANN came second with 92% accuracy, but had lower precision and recall. Naive Bayes was the weakest, with only 60% accuracy due to its assumptions. From the confusion matrix, I could see which species were most often confused. For example, Roach and Parkki were often misclassified, likely due to similar sizes. The dataset taught me how feature scaling affects model outcomes. It also showed how important it is to understand model limitations.
Overall, this project helped me build confidence in both modeling and interpretation.