Statistical learning for big data (MVE441 / MSA220)
Exam May 24 - June 7, 2024 (part 1)

Luca Modica

June 7, 2024

Introduction

In the first part of the exam, we will explore a dataset containing 6 different measurements of 7 fish species, from their weight to their width. The analysis will focus on the 4 following aspects:

- classifiers comparison
- feature importance inspection to find an optimal features subset for classification
- impact of simulated high dimensional features on predictive performance
- analysis of the confidence-based classification and set prediction.

Task 1a: classifiers comparison

In the first task, different classifiers parametric, nonparametric, linear, nonlinear,...) were trained and compared, by assessing and comparing with suitable metrics:

- their overall results (on a macro level)
- prediction performances per class.

Exploratory Data Analysis (EDA)

To set up a proper pipeline for this experiment, I first performed an exploratory data analysis on the training set to understand the dataset characteristics, from the features correlations to the class separation. This step leads to 2 crucial results. The first one is represented by the **critical class imbalance** of the data given (table 1), with the *Bream* species much more represented than most of the classes (especially Whitefish). This will influence the choice of metrics to assess the results of the classifiers, by preferring F1-scores and Cohen Kappa indices to more inaccurate metrics such as accuracy.

Class	Count	Proportion
Bream	433	0.29
Perch	299	0.20
SilverBream	193	0.13
Pike	189	0.13
Smelt	175	0.12
Roach	170	0.11
Whitefish	33	0.02

Table 1: Class distribution, in terms of count and proportion.

The second main finding is related to the **class separation**. To explore this aspect, the dataset was reduced to 2 dimensions by using Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), to look for possible linear separability of the target labels (1).

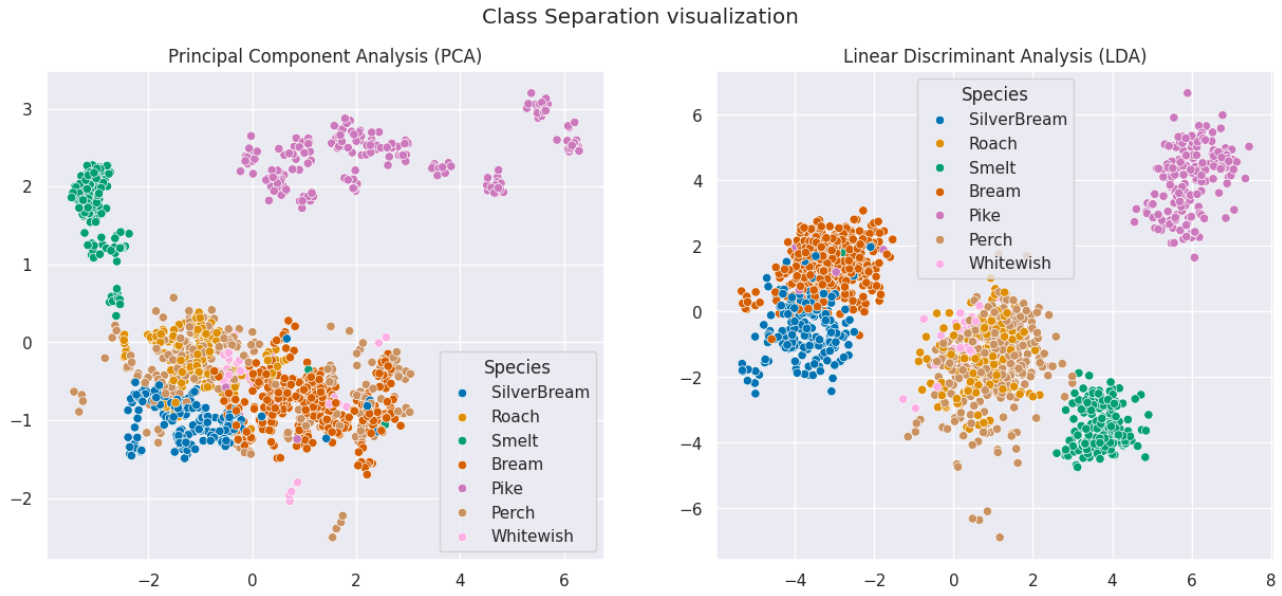

Figure 1: Visualization of the class separation using PCA and LDA ad dimensional reduction methods.

As shown in the plots above, both dimensional reduction techniques show linear separability to a certain extent, especially by looking at LDA results. Labels such as SilverBream and Bream have a minor overlap (which can be reasonable for the similarity of the 2 species) with outliers from other classes; instead, the species Roach, Whitewish, and Perch almost completely lack linear separations.

Before diving into the classifiers comparison, these 2 findings allow us to make first *modeling assumptions*. Despite the clear linear separability of some classes, linear models such as Logistic Regression might struggle with several non-linear relationships, as highlighted by PCA and LDA. On the other hand, models such as Random Forest and eXtreme Gradient Boosting can be promising, not just for their ability to capture non-linear patterns, but also for their robustness to eventual noises in data. Another algorithm worth mentioning K-Nearest Neighbor: It's a type of model that can struggle in the presence of overlap and class imbalance, due to its classification based on data point proximity; despite that, with a correct choice of k (number of neighbors), It can have discrete results thanks to its non-linear decision boundary.

Methods and training pipeline

Considering the findings from the initial data analysis, the following is the structure of the training pipeline used in this experiment.

- To comprehensively assess the classification results, the classifiers trained are Logistic Regression, Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest and eXtreme Gradient Boosting (XGBoost). For all of them but XGboost and KNN, class weights were added to give slightly more importance to the minority class (Whitewish).
- To fine-tune each classifier and train on multiple training-test splits, a grid search with cross-validation (stratified) is used. In particular, the experiment is conducted 5 times, with 5 different fold splits.
- Before training, the data are properly preprocessed. An oversampling method (SMOTE) is first used to take into account the class imbalance in the dataset; this is followed by data normalization (especially for models sensible to different scales, for example, KNN).
- The training and cross-validation metrics chosen for the experiment (on a macro and class level) are F1 score, sensitivity (model's ability to correctly identify positive instances), and specificity (how well the negative cases are correctly predicted as negative). The same metrics were used on the test set by also adding a Cohen Kappa index, for a final agreement measure between predicted labels and actual labels.

Results

After running the experiment, the following are the optimal hyper-parameters found for classification performances, for each classifier.

- *Logistic Regression*. Regularization strength: 10.
- *Decision Tree*. Max depth: 10, min samples split: 5.
- *Support Vector Machine*. Regularization strength: 10, kernel: RBF.
- *K-Nearest Neighbors*: metric: manhattan, number of neighbors: 10, distance weight: uniform.
- *Random forest*: max depth: 20, min samples split: 5, number of estimators: 100.
- *eXtreme Gradient Boosting*: learning rate: 0.1, max depth: 5, number of estimators: 200.

To get a first grasp of the overall performance and assess them, I inspected the cross-validation metrics of the experiment results, on a macro level. As shown in Figure 2, the results were overall reasonable across all models, especially about their ability to identify real negative instances (specificity ≥ 0.97).

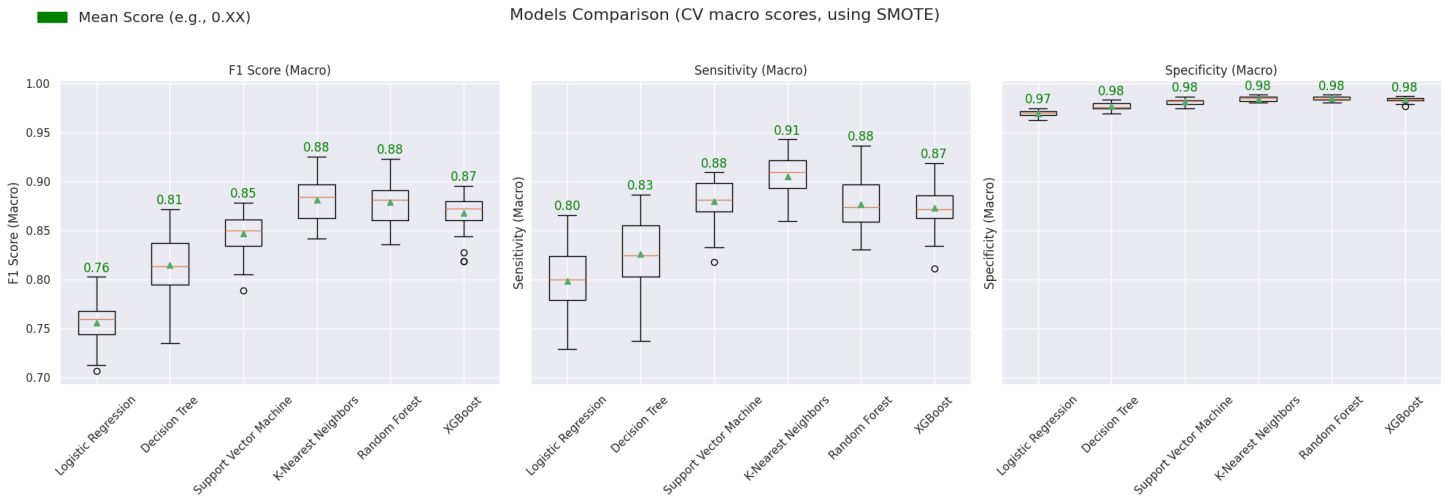

Figure 2: Cross-validation results of the experiment, on a macro level.

As could be expected from the modeling assumption, the class imbalance and especially class overlap posed a tougher challenge to linear models such as the Linear regression, both in terms of F1-score (0.76) and sensitivity (0.80). This is followed by the decision tree model: thanks to the nature of the algorithm, Its non-linear decision boundary can help to have better results (0.81 F1-score and 0.83 sensitivity), but with a larger variability of the score across the different cross-validation splits. RBF SVM seems to slightly improve the situation (F1-score = 0.5), with the applied kernel that seems to capture more on-linear relationships. Finally, the macro cross-validation performances show KNN, random forest, and XGBoost as the best models in this scenario (F1-score ≥ 0.87). The better results on the chosen macro metrics, compared to the previously mentioned models, can be justified by looking to their decision boundaries (Figure 3).

- By averaging multiple decision trees on different feature splitting and data subsamples (bootstrap aggregating), *random forest* can leverage the non-linear decision boundaries of a single decision tree classifier.
- Although the different approach, XGBoost can also handle non-linear relationships (even in overlapping areas) much better, by optimizing the results on errors of a previous weak learner.
- The neighbor proximity strategy implied by KNN also creates a reasonable decision boundary, leading to the macro performance shown above.

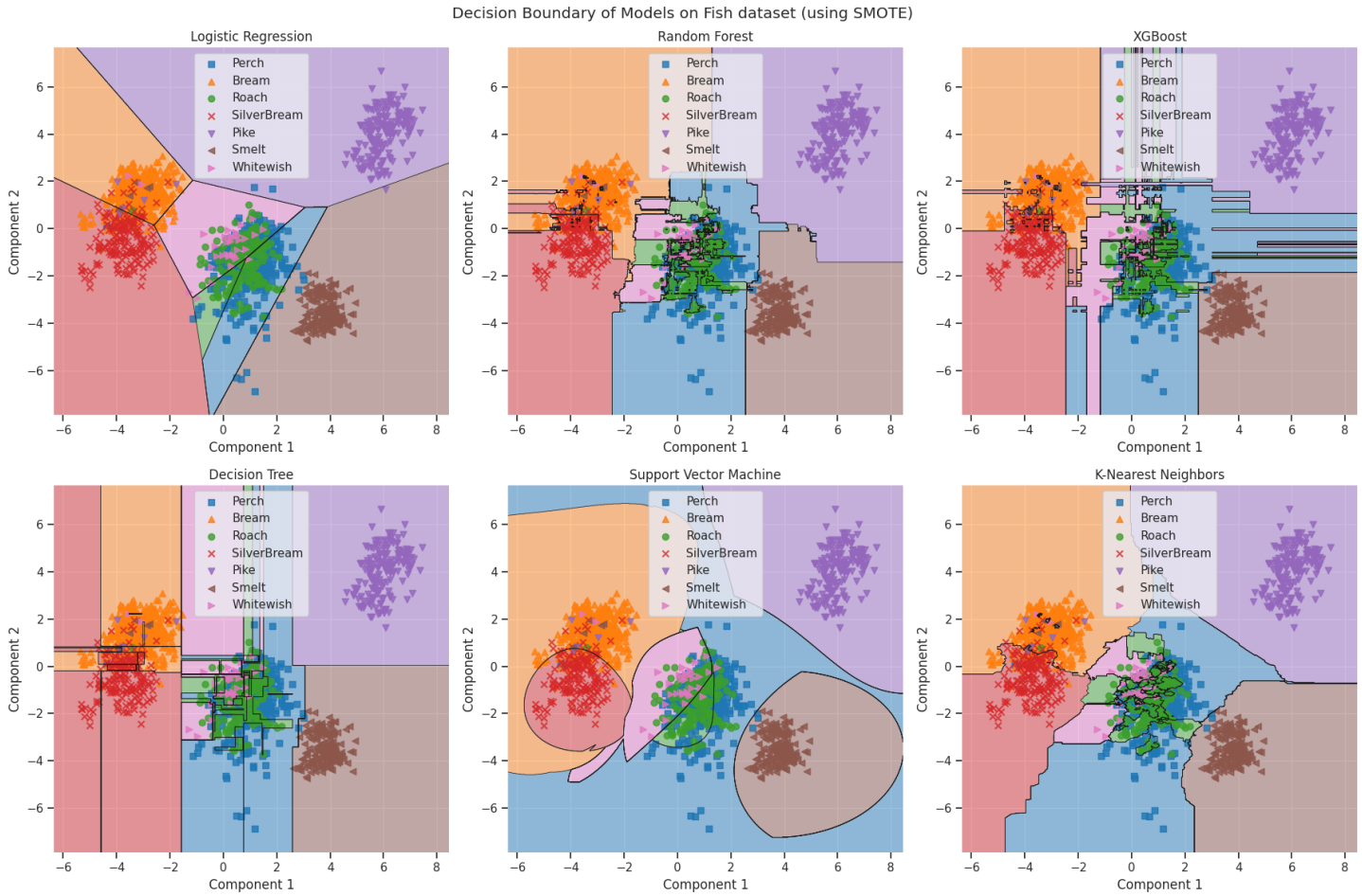

Figure 3: Decision boundary of each model. The data points shown are from the train set, while the decision boundary of the models are created with their optimal hyper-parameters. The visualization is done by projecting the dataset with LDA, to highlight linear and non-linear relationships in the data.

After assessing the performance of the models on a macro level, I inspected the cross-validation metrics per fish species for a comprehensive evaluation of the classifications. An overview of the results are given in Figure 4.

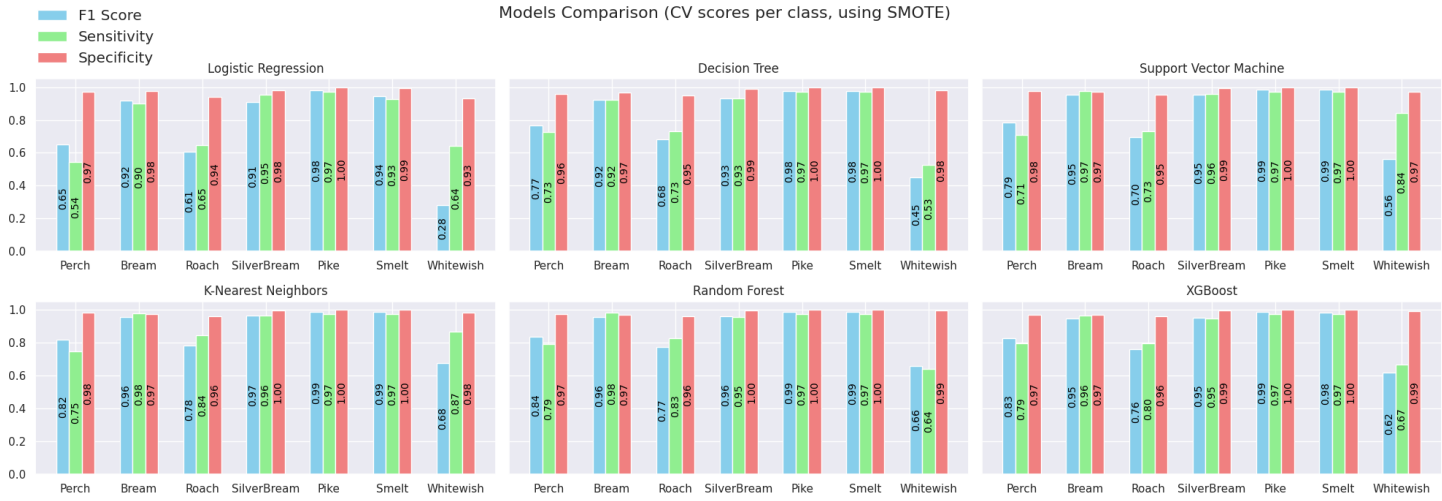

Figure 4: Cross-validation scores per class, for each trained model. The performances mostly deteriorated for the under-represented species (Whitewish) and the ones that mostly overlaps with other classes (Perch and Roach).

As can be noticed from the bar bar plots above, the performances drastically deteriorated on the most underrepresented class (Whitewish), which is also overlapping with most of the other species (from Figure 1); a similar pattern can be seen on the Roach and Perch, which are also classes that are overlapping in the dataset. By investigating the performances class-wise: logistic regression and decision tree seems to have more difficulties in classifying the 3 problematic classes mentioned (low F1-score and sensitivity); this is caused by the logistic regression not being able to capture non-linear pattern, and the lack of enough dataset generalization of the decision tree. A last important observation related to the discrete performances of KNN class-wise, despite the challenging scenario for a neighbour proximity algorithm.

As a last model performance validation, down below I reported the scores obtained from the test set predictions (Table 2).

		Logistic Regression	Decision Tree	SVM	KNN	Random Forest	XGBoost
Macro scores	F1 score	0.75	0.89	0.86	0.89	0.91	0.91
	Sensitivity	0.81	0.90	0.89	0.92	0.91	0.91
	Specificity	0.97	0.98	0.980	0.98	0.99	0.99
	Cohen Kappa	0.77	0.88	0.88	0.89	0.90	0.90
"Perch"	F1 score	0.62	0.82	0.83	0.81	0.86	0.82
	Sensitivity	0.48	0.77	0.76	0.72	0.79	0.75
	Specificity	0.98	0.97	0.98	0.99	0.99	0.98
"Bream"	F1 score	0.92	0.93	0.94	0.94	0.94	0.95
	Sensitivity	0.94	0.95	0.99	0.99	0.99	1.00
	Specificity	0.96	0.96	0.95	0.95	0.95	0.95
"Roach"	F1 score	0.57	0.77	0.74	0.75	0.78	0.79
	Sensitivity	0.60	0.79	0.71	0.79	0.81	0.83
	Specificity	0.94	0.97	0.97	0.96	0.97	0.96
"SilverBream"	F1 score	0.92	0.97	0.99	0.98	0.98	0.99
	Sensitivity	0.98	0.96	0.98	0.98	0.98	0.98
	Specificity	0.98	1.00	1.00	1.00	1.00	1.00
"Pike"	F1 score	0.98	0.96	0.98	0.98	0.98	0.98
	Sensitivity	0.96	0.96	0.96	0.96	0.96	0.96
	Specificity	1.00	0.99	1.00	1.00	1.00	1.00
"Smelt"	F1 score	0.93	0.98	0.99	0.99	0.99	0.99
	Sensitivity	0.95	0.98	0.98	0.98	0.98	0.98
	Specificity	0.99	1.00	1.00	1.00	1.00	1.00
"Whitewish"	F1 score	0.32	0.82	0.56	0.76	0.82	0.82
	Sensitivity	0.75	0.88	0.88	1.00	0.88	0.88
	Specificity	0.93	0.99	0.97	0.99	0.99	0.99

Table 2: Test results summary for the Fish dataset.

Other than confirming the cross-validation results, the Cohen Kappa index shows a reasonable agreement between predicted labels in test set and ground truth, in line with the other results already discussed.

Task 1b: feature importance for optimal features subset

After comparing several different classifiers, my data analysis turns to the importance of the features in the dataset: the goal is to explain why those are important and to find the set of features that can result in optimal classification performance. This investigation will be done with the following methodologies.

- To assess the importance of the features, 3 different feature selection methods will be compared: *Random Forest feature importance*, *ANOVA F-values*, and *Lasso Coefficients*.
- The performance of a selected set of features will tested on the same models of the previous task, by using the optimal hyper-parameters found.

The first step of the investigation was to explore the correlations between the features, to have a first grasp of their meanings, and to check for possible redundancy (Figure 5).

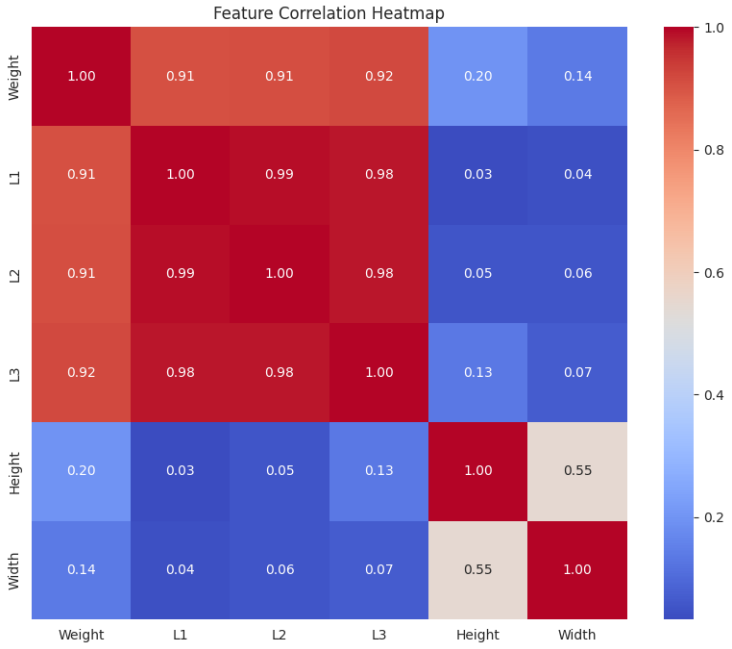

Figure 5: Feature correlation heat map.

As we can notice, one of the first insights from the heat map above is the strong correlation between the 3 length-related features (L1, L2, L3): this implies that a subset of them may be sufficient to retain optimal classification performances. Another interesting observation is the similar high correlation between the weight of the fish and the 3 features previously mentioned. From a biological point of view, a greater length of the fish might be connected to its weight; but this can also mean that its weight might also be a piece of redundant information.

After initial correlation analysis, the 3 feature selections will be now put into comparison. The consistency of their selections is assessed by performing the methods 5 times, with 5 different fold splits on the train set. The result of the experiment is shown with bar plots below, which display the 3 different feature rankings and the importance attached to the features by each method (Figure 6). For each feature selection method, the final confidence is asses by averaging the given score across the 25 different folds, for more confidence attached to the estimates.

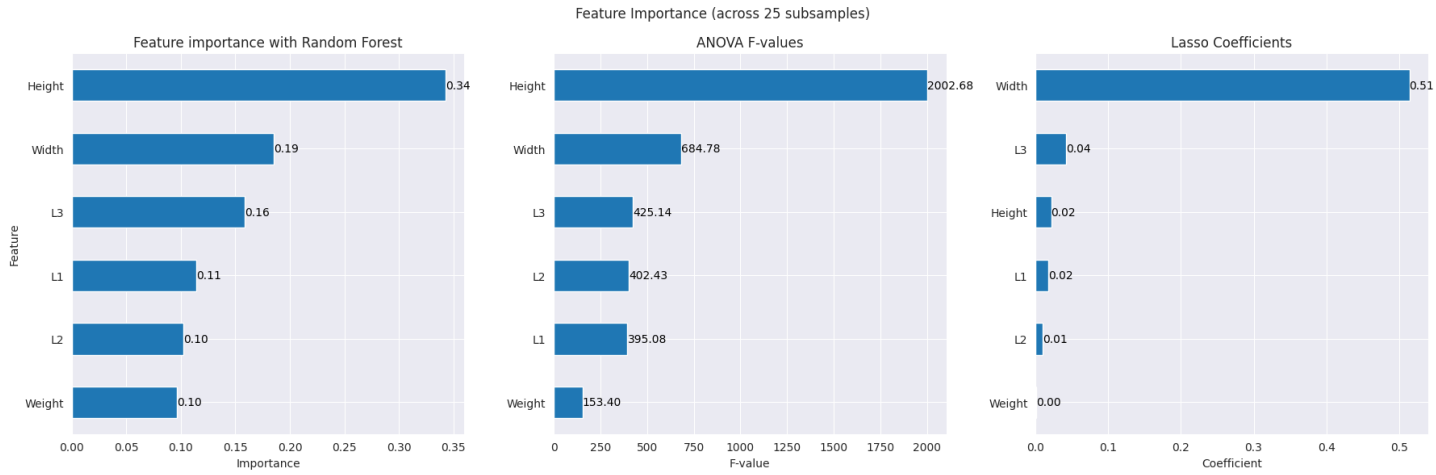

Figure 6: Feature importance ranking, per feature selection method.

A significant result from the rankings above is the low importance attached to the weight of the fish by all 3 methods, despite their different approach to computing feature importance: this reinforces, even more, my previous hypothesis about its redundancy in an eventual optimal set of features. A second remarkable insight is represented by the Height and the Width being consistently in the top-ranked features of the 3 methods. The dominance of those 2 features can have intuition in their biological meaning: in most cases, those measurements can capture important morphological differences between the fishes. Although the rankings, a last observation is the related to the low score attached by Lasso to the "Height" feature, in comparison to the other 2 feature selectors: this might be due to the low linear correlation of the height to the other information, while instead It had a high F-value (high variance related to the target variable) and a high importance attached by a random forest.

To conclude the feature importance analysis, the last step of this task was to experiment to investigate the classification performance degradation in function of the number of the features removed: this in the order of the rankings shown above. For each classifier, a rank of a feature selection method was attached to determine the order of the feature removal. The results of the experiment, resulting in an elbow plot, are shown below (Figure 7).

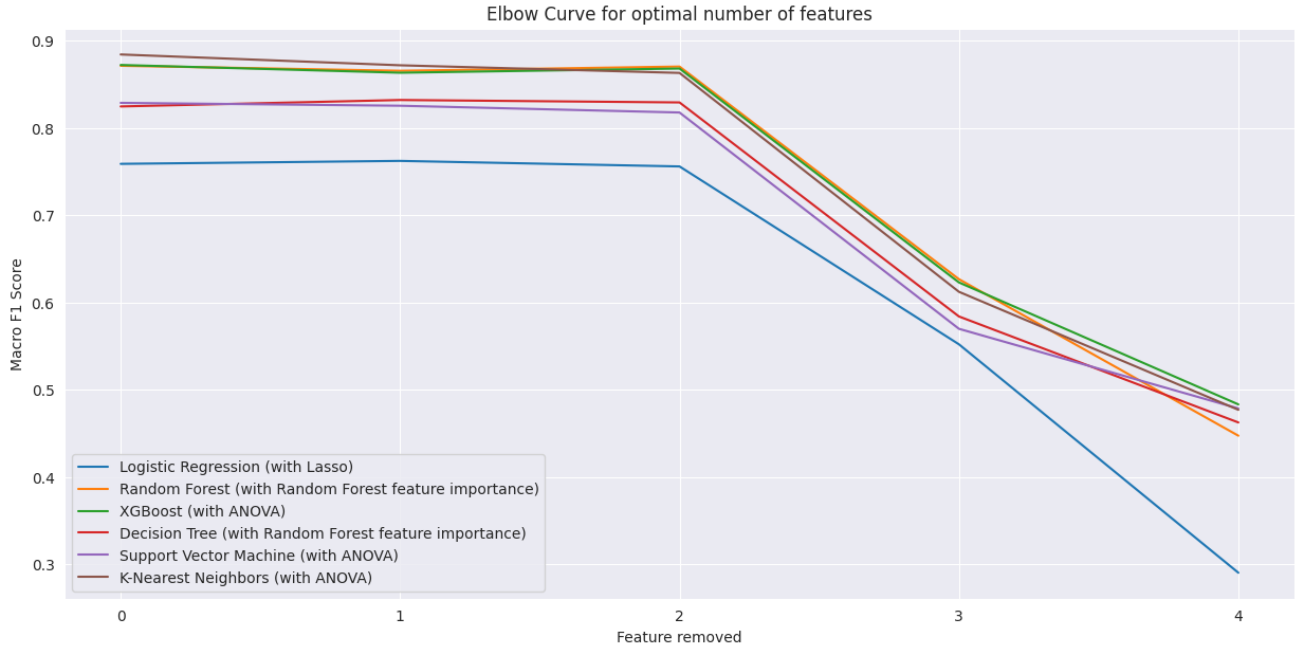

Figure 7: Elbow plot showing the model performances (Cross-Validation F1-score) in function of the number of features removed. The features are removed from the least to the most important ones, based on the feature selection method attached to each model.

Considering that the Weight species is the least ranked feature across all the feature selectors, the elbow plot clearly shows that It can be removed from an eventual optimal set of features since its absence leads to almost no drop in performance. A similar result is given by removing one of the length-based features between L2 and L3 (always based on the different feature selection methods) as the second feature removed: this confirms the hypothesis is based on the feature correlation heat map, where a subset of those length-based columns is enough to keep reasonable classification performance. Finally, we can notice across all classifiers and different feature selectors a remarkable drop in the F1-score performance (more significant by using Lasso as selectors), meaning that 4 features might be a good number of the optimal set.

In conclusion, based on the result obtained, a set of features that results in optimal classification performance can be obtained by selecting a subset of 2 length-based features, and by removing the measurement related to the Weight for its high correlation with many features, other than its irrelevancy to all the features selector methods. This conclusion, also based on Figure 7, can be drawn on average for all different classifiers.

Task 1c: performance with simulated high dimensional features

In the third task related to the Fish Dataset, I analyzed the robustness of a subset of classifiers from the first task, alongside the reliability of a subset on feature selectors from the second task. In particular, I turned to low dimensional dataset into a high dimensional one, by incrementally adding simulated features. The investigation takes into consideration the following classifiers and feature selection methods:

- KNN and random forest as classifiers
- ANOVA F-Test as feature selectors.

Similarly to the tasks above, for the reliability of the results, the experiments will be conducted with the following settings.

- To observe the classification performance trend, each experiment will be performed with 10-fold splits by measuring the following cross-validation scores: F1-Score, sensitivity, specificity, and Cohen Kappa Index.
- The feature selection with ANOVA will be conducted in 5 different 5-fold splits (in other words, a different split of 5-folds 5 times): It will be counted, across all folds, how many times a specific feature is selected.

The experiment is divided into 2 parts: adding unrelated and correlated features. For both of them, the following number of features will be progressively added: [50, 100, 150, 200, 300, 500, 1000, 2000].

0.1 Adding unrelated features

In the first part of the task, the dataset is turned into a high-dimensional one by adding unrelated features: that is, information that has less than 0.1 linear correlation to the target variable. The plot below shows the results of the first investigation (Figure 8).

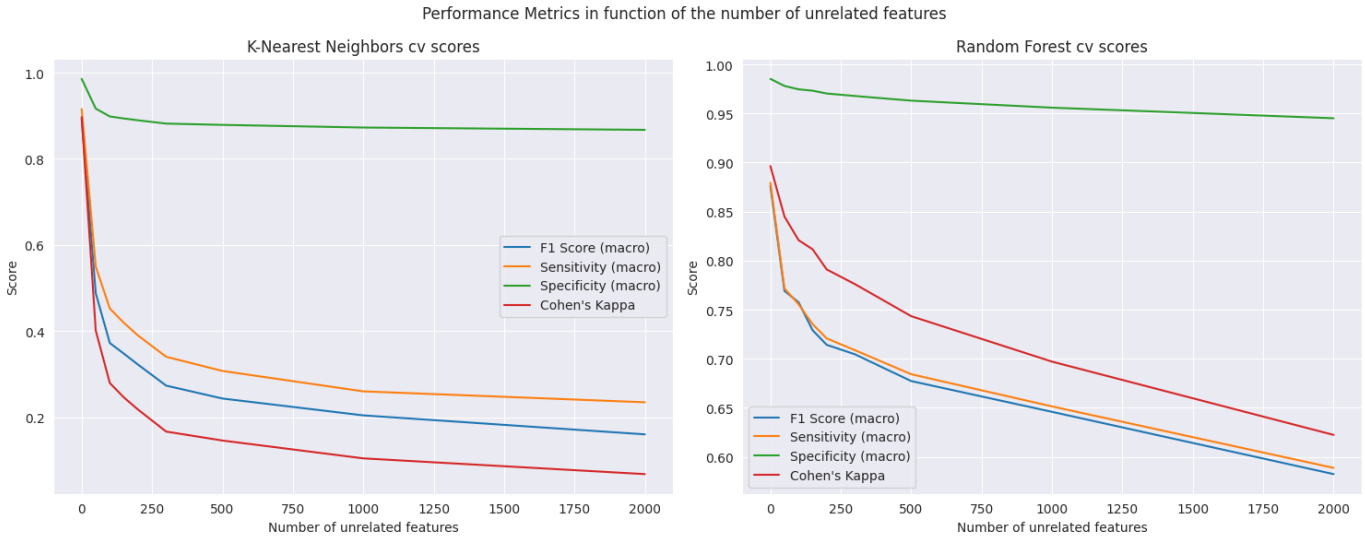

Figure 8: Performance deterioration of KNN and random forest on a macro level, in function of the number of unrelated added features.

Even maintaining a reasonable specificity score by adding more and more unrelated features, the first crucial insight in the plot above is the difference in terms of performance deterioration between the 2 models. Although really good results in task 1a), KNN seemed to be sensible even to a relatively small (50) number of features added, with an immediate drop in all CV metrics (especially on the agreement between the predicted labels and actual ones). The reason behind this can lie in the distance-based strategy to classify data points: by increasing the dimensionality with irrelevant features in the feature space, the distance between the observation becomes less and less meaningful.

On the other hand, Random Forest seems to better react to the added noise, by showing a more graceful performance decline. The robustness of this kind of model can be attributed to its ensemble nature and feature selection mechanism for creating multiple decision trees: selecting features that reduce the variance the most to different the different classes

(decision tree mechanism), alongside the multiple tree averaging, helps this ensemble model mitigate the impact of the added unrelated features.

The difference between KNN and Random Forest is highlighted even more by showing the performances class-wise (Figure 9).

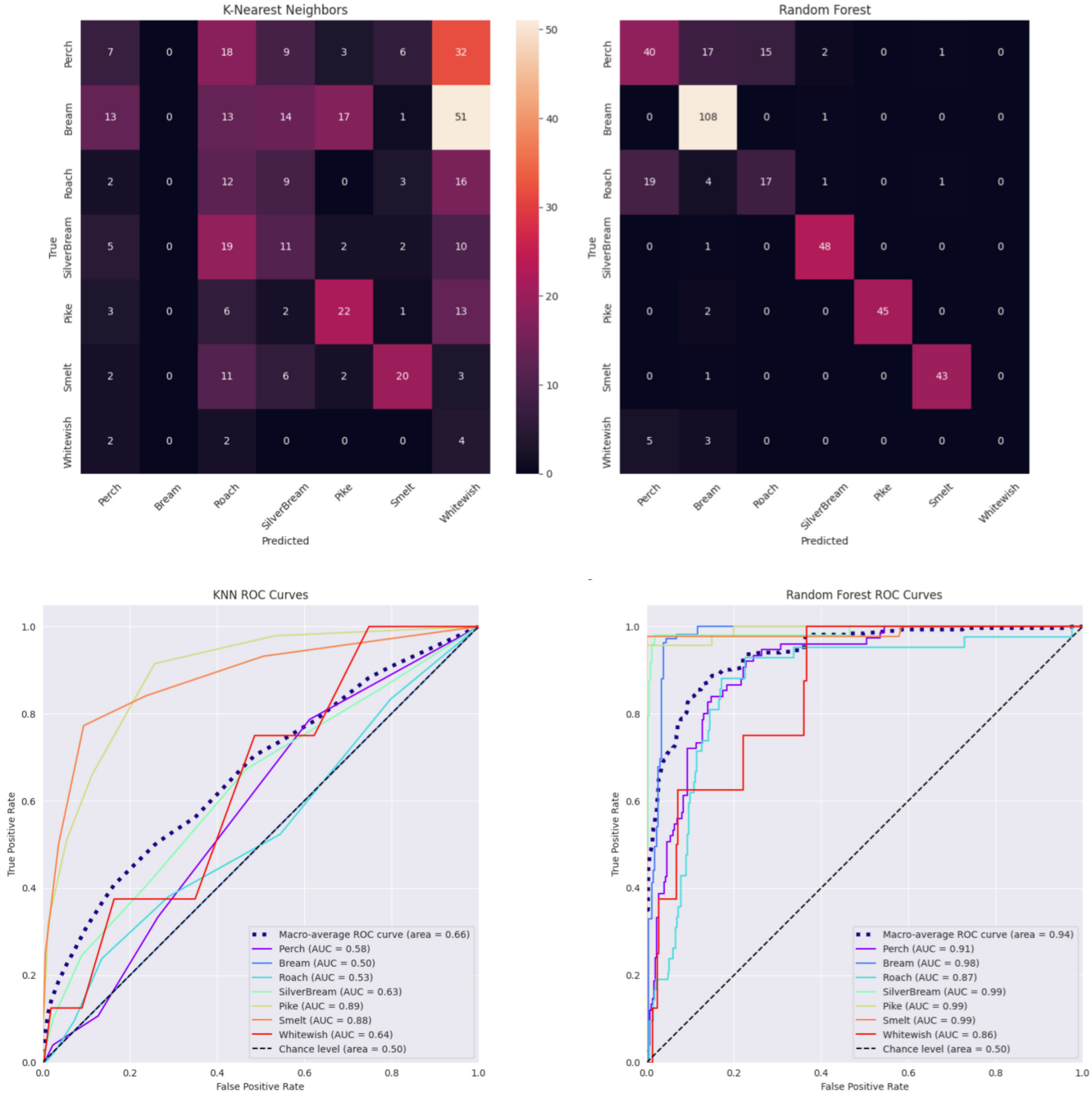

Figure 9: Performance deterioration of KNN and random forest class-wise, for 500 unrelated added features.

With a moderate noise added (500 unrelated features), for KNN model we can observe a large number of misclassifications (especially for the overlapped and underrepresented classes), alongside most of the ROC curves very close to the chance level area. On the other hand, Random Forest seems to have almost no perturbations, with a slight exception for the the underrepresented fish species (Whitewish).

The second conclusion drawn in the first part of the task is instead related to the impact on the feature selection, which was demonstrated to be irrelevant by using the ANOVA F-test as a feature selector. The selection count results

with 2000 unrelated features, across the 25 different fold splits, are shown below (Table 0.1).

Feature	Selection Count	Feature Type
L2	25	Original
L3	25	Original
Width	25	Original
Height	25	Original
L1	25	Original
Weight	25	Original
unrelated_788	8	Added
unrelated_1136	8	Added
unrelated_759	7	Added
unrelated_1836	7	Added

Table 3: Top 10 feature selection, with 2000 unrelated features added.

Despite the high number of irrelevant features added, the feature selection method appears to be robust to such noise. This behavior by the focus of ANOVA on the variance between and within the group of features, which leads to select features that show a high difference in the target variable (across all the possible labels); since an unrelated feature doesn't show a relevant difference, the selector will give low F-value and high p-value, making it less likely to be selected.

0.2 Adding correlated features

In a similar fashion to the first part, in the second section of the task, the dataset will be turned high dimensional by adding features that are correlated to the original ones. The plot below shows the performance deterioration obtained by adding such noise (Figure 10).

Figure 10: Performance deterioration of KNN and random forest on a macro level, in function of the number of correlated added features.

In this case, the impact seems to be irrelevant for both models and in all CV metrics considered. A brief consideration can be made related to mostly imperceptible performance improvement for KNN by adding correlated features, thanks to more features supporting the distance-based classification. For Random Forest, instead, we can observe a very slight decrease in performance: this is most likely due to the less capability to reduce the overall variance in averaging the trees.

As in the first part of the task, down below the predictive performance per class is shown (for 2000 correlated features), illustrating also in this case the high robustness to the added noise for both models.

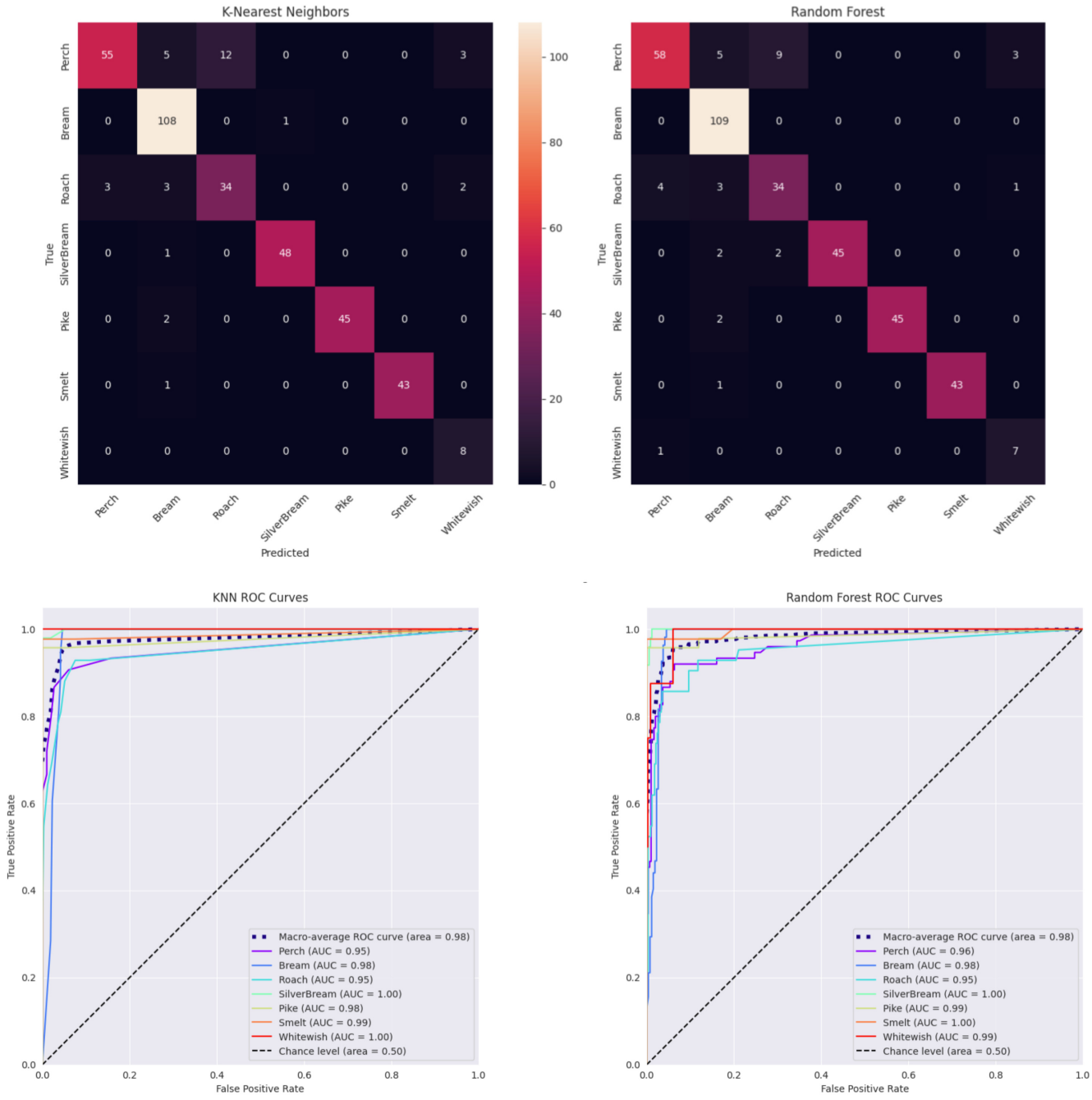

Figure 11: Performance deterioration of KNN and random forest class-wise, for 2000 correlated added features.

To conclude the experiment, I checked also in this scenario the impact of the added features on the feature selection. As can be observed in the table below (Table 0.2, top 10 features), ANOVA F-Test seems to almost not select the original feature of the dataset, even with a relatively small (100) correlated features added.

Feature	Selection Count	Feature Type
correlated_53	25	Added
correlated_47	25	Added
correlated_95	25	Added
correlated_59	25	Added
correlated_83	24	Added
Height	24	Original
correlated_5	24	Added
correlated_77	23	Added
correlated_17	16	Added
correlated_65	15	Added

Table 4: Top 10 feature Selection, with 100 correlated features added.

The reason behind this behavior always relies on how this feature selector works, which also highlights a limitation of such a method. With the correlated features added to the dataset, those can have a stronger statistical signal: this leads ANOVA to prioritize such features, leading to selecting potentially redundant information and overshadowing the original features.

Task 1d: confidence-based classification and set predictions

To conclude the data analysis of this dataset, I lastly revisited task 1a to delve deeper into the confidence with which each classifier assign an observation to a specific class. In particular, we will explore for each model already used how many observation are classified with confidence as specific target and which one are instead assigned to a set prediction (that is, where some observations may be more appropriately allocated to a set of possible classes rather than a single definitive class). The goal of this task is to show the confidence of each model and, by analysing their confidence, if it's possible to find potentially mislabeled observation in the original training data.

In order to show such results each model will be retrained with 5 different 5-fold splits, and setting a threshold of 0.9 to determine if an observation is labelled with high confidence. If data point has a low confidence level, It will be assigned to a set prediction of the top 2 classes; other-wise, other than being a confident prediction, if the label assigned by the model is different from the actual one, It will be labelled as potential mislabelled observation. The mislabelling will be also assessed with the confusion matrices, with the goal to find potential misclassification patterns.

After running the experiment, the following plots in Figure 12 shows the first results in terms of overall confidence distribution per model.

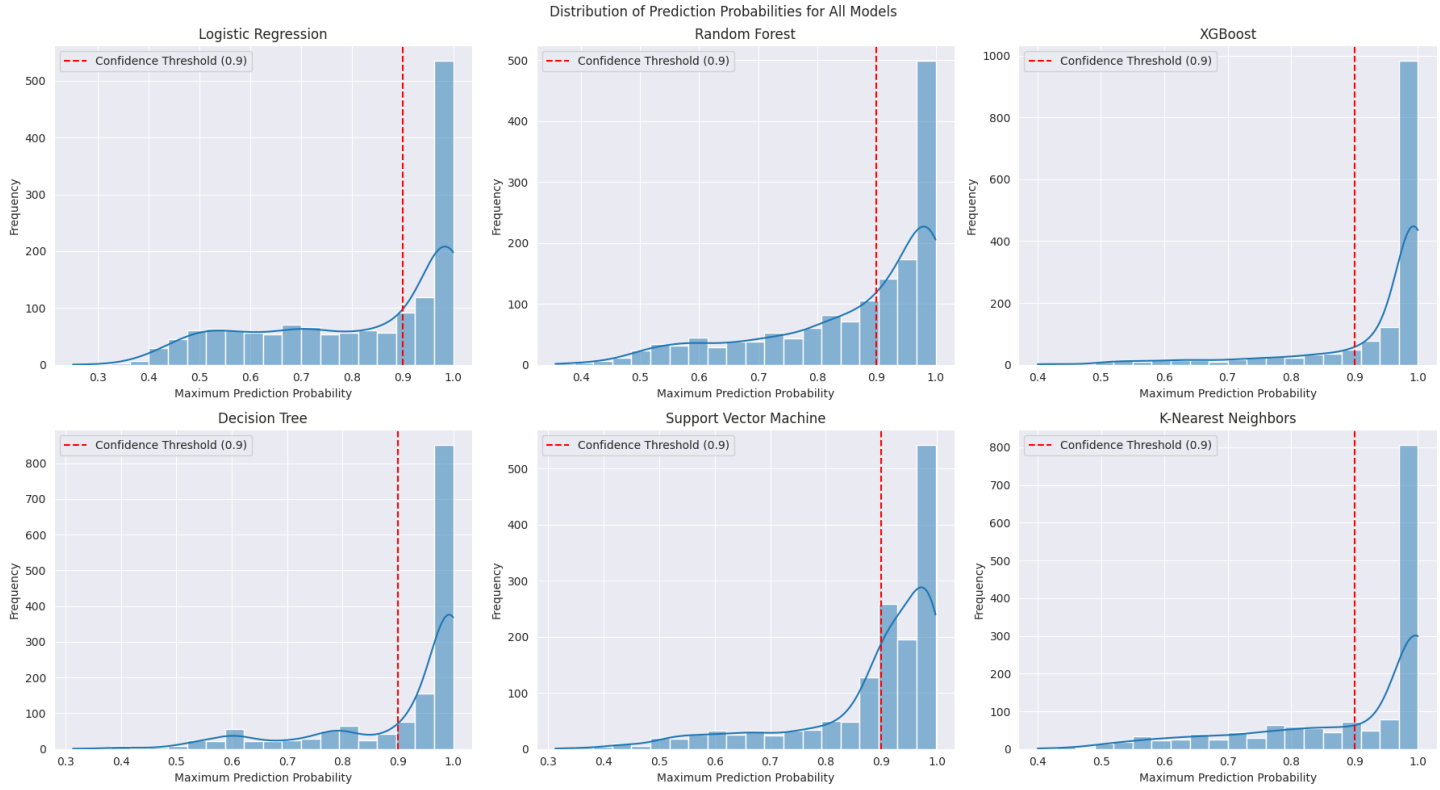

Figure 12: Confidence distribution per model.

The first very notable insight from the distribution is the general tendency of all models to have high confidence, especially for XGBoost and KNN. Overall, only Logistic Regression and Random Forest have a highly higher distribution over a lower confidence. To better inspect this scenario, a useful plot with the confidence probabilities by observations is shown below (Figure 13).

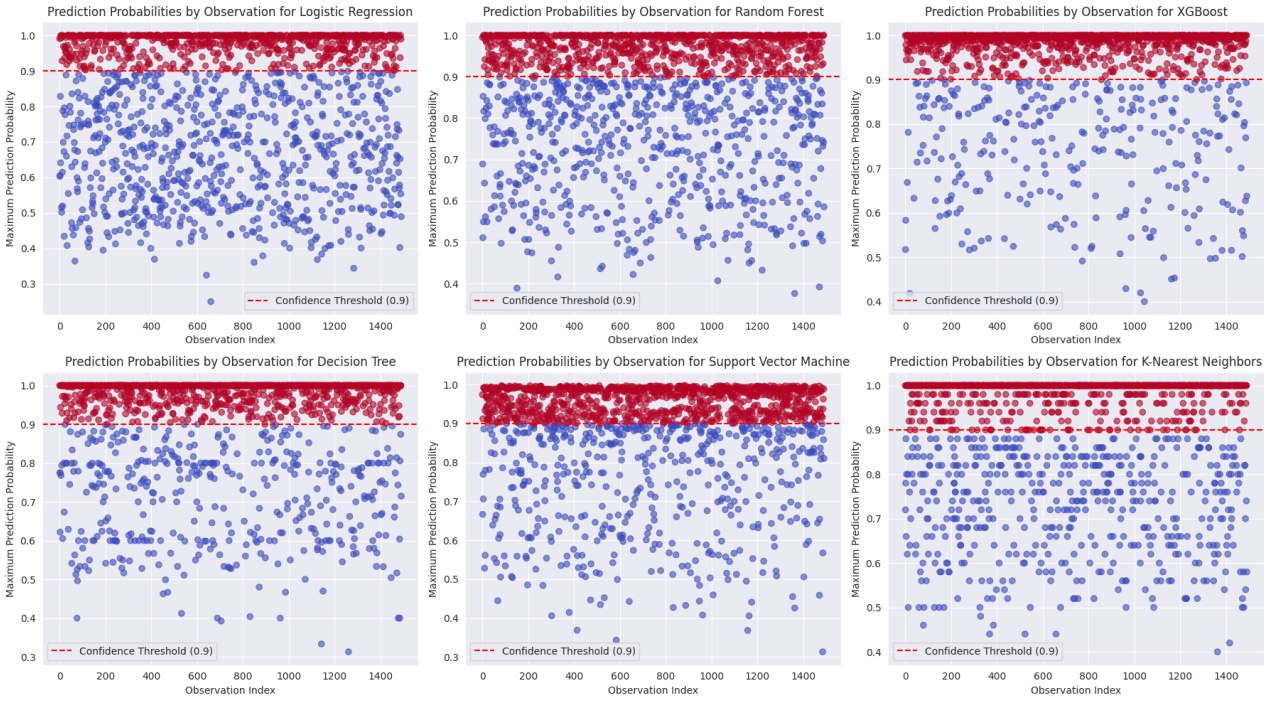

Figure 13: Confidence probabilities by observation, per model.

The plot highlights even more how most of the observations are classified with confidence on a single class, alongside other observations with similar probabilities per class that are more suitable to a set prediction.

With the overall situation concerning the confidence of each classifier, I could start investigating potential mislabelling. The first step was to inspect the confusion matrices, which are shown below.

Figure 14: Confusion matrices for train labels, for each model.

The confusion matrices first reveal one common pattern, which mislabeling the Perch species and the Roach species, across all models, suggesting potential mislabelling between those 2 classes.

To further inspect, in the following Table 0.2 I first summarized the models' confidence information, including a percentage of potentially mislabelled classes.

Model	Confident Predictions	Set Predictions for non-confident prediction (top 2 classes)	Potentially Mislabeled
Logistic Regression	712 (47.72%)	780 (52.28%)	15 (1.01%)
Random Forest	826 (55.36%)	666 (44.64%)	16 (1.07%)
XGBoost	1195 (80.09%)	297 (19.91%)	42 (2.82%)
Decision Tree	1079 (72.32%)	413 (27.68%)	48 (3.22%)
SVM	964 (64.61%)	528 (35.39%)	29 (1.94%)
KNN	981 (65.75%)	511 (34.25%)	22 (1.47%)

Table 5: Summary of Model Predictions Confidence.

We can notice a slight percentage of potentially mislabeled observations (especially with overconfident classifiers such as decision trees). As a final investigation, I tried to understand if there were specific observations mislabeled by all models. The results of the exploration are shown in Table 0.2.

Observation ID	True class	Pred class (Logistic Regr)	Pred class (Random Forest)	Pred class (XGBoost)	Pred class (Decision Tree)	Pred class (SVM)	Pred class (KNN)
1462	SilverBream	Bream	Bream	Bream	Bream	Bream	Bream
1127	Whitewish	Bream	Bream	Bream	Bream	Bream	Bream
1582	Perch	Bream	Bream	Bream	Bream	Bream	Bream
1552	Smelt	Bream	Bream	Bream	Bream	Bream	Bream
820	SilverBream	Bream	Bream	Bream	Bream	Bream	Bream

Table 6: Potentially mislabelled datapoints, in common between all classes.

Compared to what we concluded by looking at the confusion matrices, the few potentially mislabeled observations in common with all models are all misclassified as *Bream*. The reason behind the potential mislabel can be due to different factors: for observation 1462 and 820, which are classified as *SilverBream*, the mislabelling can be caused by the species similarity with Bream; for other species, such as *Smelt*, the reason can lie to overlapping / similar measurements.