MVE441 Statistical learning for big data
Take-home examination-Q1

Sky Sunsaksawat

June 2024

Dataset1

For the first part of the exam question you will use a data set with 1866 observations and 6 features. The data set is contained in a tab-del file called "Fish.txt" posted on canvas. The data set contains information about different species of fish, such as weight, height and width. It also contains length measures of three types called L1, L2 and L3 that measure length from the "nose" to the beginning of the tail (L1) blue line in image below), nose to the notch of the tail (L2, yellow) and full length (L3, red).

Q-1a

The set of classifiers that are used to perform classification problem in this subtask comprise of 6 different classifiers with different degrees of complexity and character. It includes
1. Gradient boosting classifier: An ensemble method
2. Support vector classifier(SVC): A non-linear, parametric model
3. K-Nearest Neighbors (KNN): A non-parametric model
4. Decision tree: A non-linear, non-parametric model
5. Random Forest: An ensemble of decision trees (non-linear, non-parametric)
6. Neural Network : A non-linear, parametric model

- **Class imbalance**

Due to a small size dataset, Synthetic Minority Oversampling Technique (SMOTE) is introduced because the computational cost are not concerned in this dataset. This method is used to generate artificial samples for the minority class, ”Whitewish” in this case, which can re-balance dataset. The result after doing SMOTE is shown in Figure1.

- **Overall performance**

Table1 shows the overall performance for 6-classifiers. K-Nearest Neighbors, GradientBoostingClassifier, RandomForestClassifier, and MLPClassifier have outstanding performance, which its Cohen-Kappa score is approximately 0.90.

- **Class-level performance**

It can be concluded from Table2 that ”Bream”, ”Pike”, ”SilverBream” and ”Smelt” perform high level in both Sensitivity and Specificity. While, ”Perch”, ”Roach”, and ”Whitefish” have slight lower in sensitivity. Moreover, the confusion matrix of RandomForestClassifer, figure2, shows that Perch are frequently misclassified as Pike, and Roach as Perch. This suggests that there may be some overlap in the feature space between these species.

- **Class separation**

A t-distributed Stochastic Neighbor Embedding (t-SNE) plot was used to visualize the relationships between the labels (fish species in this case).It can be seen from the figure 3 that ”Smelt”, ”Pike”, ”SilverBream”, and ”Bream” are clearly differentiated. However, there is a minor overlap between ”SilverBream” and ”Bream” on the bottom left of the graph. Additionally, in these well-separated clusters, it has a few number points that have potential to be mislabeling data points. In the meantime, ”Whitewish”, ”Roach”, and ”Roach” are combined into a single group.

- **Training pipeline**

Prior to training the model, GridSerchCV() can also be used with cross-validation to fine-tune parameters. The pipeline includes standardization using StandardScaler, over-sampling using SMOTE, and the specified classifier (clf). Dimension reduction are not utilized in this dataset before perform classification task because it may loss of information for the minority class due to its size. Then, the dataset is split into training and testing using Stratified K-Fold Cross-Validation (StratifiedKFold) to ensure that each fold has the same class distribution as the original dataset. In the training process, it works by repeatedly splitting data into training and testing sets for each fold. The model learns from training data and generates predictions based on the testing set. A set of metrics, including the Cohen Kappa score, precision, recall, and confusion matrix, evaluate the model's performance across all classes. After processing all folds, the average values for each performance metric are computed

Classifier	**Cohen Kappa Score**	**Precision**	**Recall**	**F1 Score**	**Sensitivity**	**Specificity**
KNeighbors	0.90088	0.92729	0.91907	0.91989	0.91898	0.98970
GradientBoosting	0.89217	0.91469	0.91212	0.91251	0.88003	0.98886
SVC	0.86893	0.90398	0.89282	0.89411	0.88833	0.98602
DecisionTree	0.85199	0.88543	0.87890	0.88072	0.85544	0.98432
RandomForest	0.90446	0.92376	0.92230	0.92200	0.89136	0.99023
MLPClassifier	0.90069	0.92375	0.91908	0.91965	0.89764	0.98979

Table 1: Overall Performance Metrics for Various Classifiers

Classifier	**Bream**	**Perch**	**Pike**	**Roach**	**Silver Bream**	**Smelt**	**Whitefish**
Class Sensitivity							
KNeighbors	0.9853	0.7753	0.9706	0.8347	0.9669	0.9724	0.9278
GradientBoosting	0.9576	0.8262	0.9706	0.8206	0.9545	0.9724	0.6583
SVC	0.9779	0.7191	0.9706	0.7309	0.9669	0.9724	0.8806
DecisionTree	0.9077	0.7728	0.9706	0.7684	0.9378	0.9724	0.6583
RandomForest	0.9890	0.8262	0.9706	0.8062	0.9669	0.9724	0.7083
MLPClassifier	0.9779	0.8048	0.9706	0.8395	0.9628	0.9724	0.7556
Class Specificity							
KNeighborsClassifier	0.9683	0.9839	1.0000	0.9661	0.9988	0.9994	0.9858
GradientBoostingClassifier	0.9683	0.9725	1.0000	0.9710	0.9920	0.9988	0.9901
SVC	0.9683	0.9772	1.0000	0.9571	0.9932	1.0000	0.9770
DecisionTreeClassifier	0.9668	0.9611	0.9957	0.9643	0.9889	0.9964	0.9814
RandomForestClassifier	0.9683	0.9745	1.0000	0.9704	0.9975	1.0000	0.9934
MLPClassifier	0.9675	0.9785	1.0000	0.9667	0.9969	0.9994	0.9918

Table 2: Class level Sensitivity and Specificity for Various Classifiers

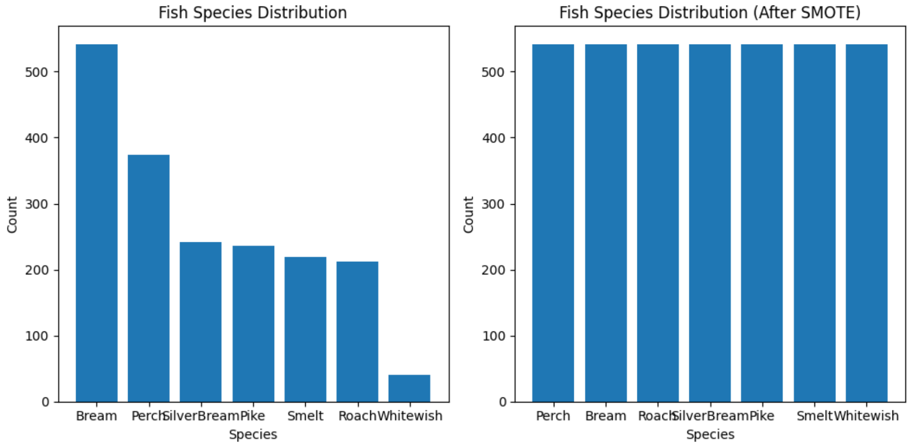

Figure 1: Fish Species Class Distribution Before and After SMOTE

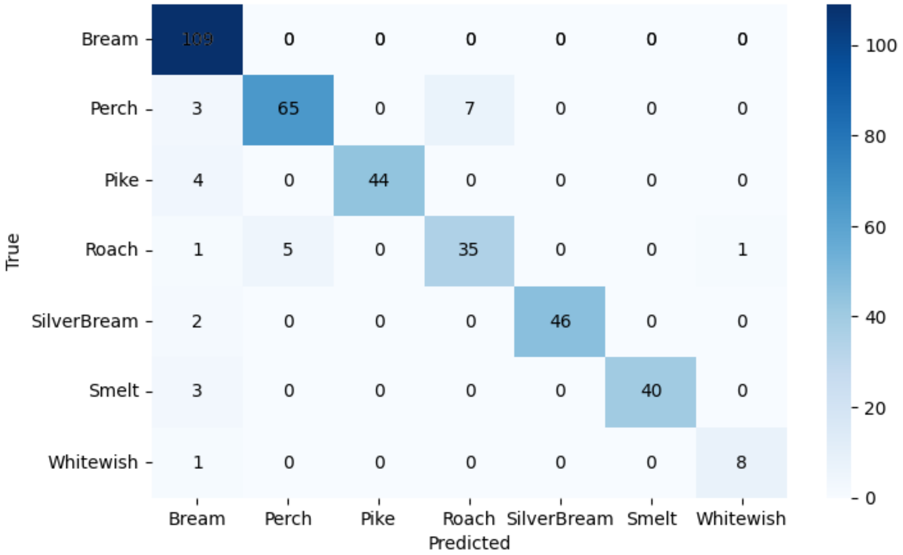

Figure 2: Confusion matrix after perform classification with RandomForestClassifier

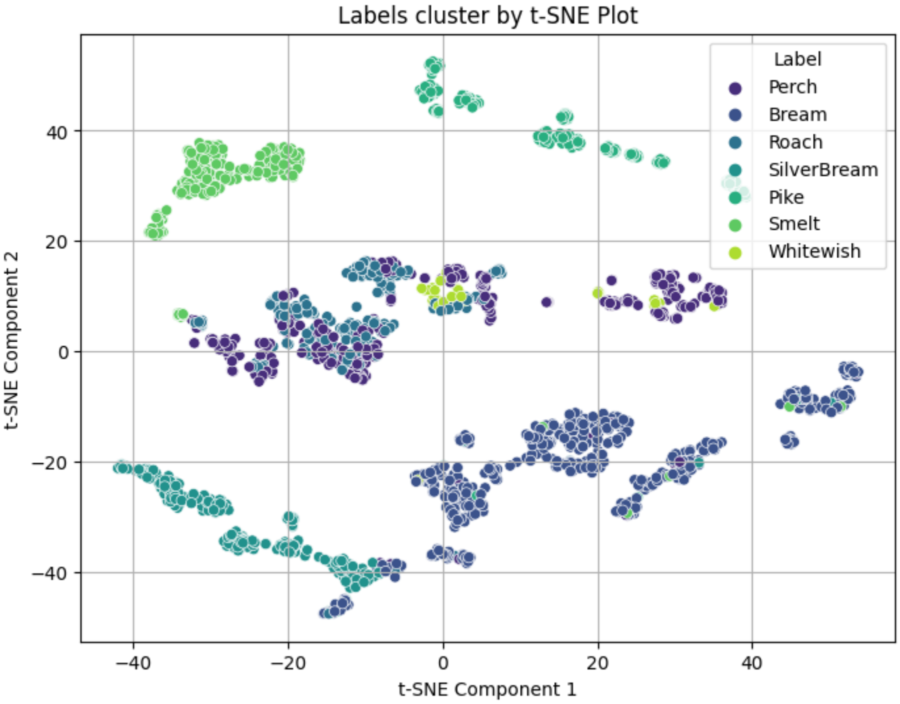

Figure 3: t-SNE plot showing the relationships between the labels

Q-1b

- **Method comparisons**

The 3 different feature selection methods that are used in this subtask containing Lasso, Ridge, and Elasticnet logistic regression. L1 regularization(Lasso) are likely to produce sparse models with many coefficients set to zero. In the meantime, coefficients of L2 regularization (Ridge) does not set to exactly zero. Lastly, Elasticnet combines both L1 and L2 penalties, so it provides a balance between the sparsity of Lasso and the stability of Ridge.

- **Result interpretion**

In the most left of figure4, it illustrates the average number of selected feature across three different methods. To attach confidence level, the error bars (including mean, standard deviation and confidence intervals) are computed averaged from 500 rounds running to visualize the selection stability. 95% confidence level are used in this case. Ridge model shows the most consistent feature selection. Lasso and Elasticnet Logistic Regression exhibit more variation in feature selection frequency. The frequency means are 4.60 with an interval of 0.63 and 5.50 with an interval of 0.30, correspondingly.

The Lasso model has the least number of specified features. Meanwhile, Ridge selects all features as expected. The average selection frequency of each features are also depicted in this figure. We can see that only 'L3' and 'Height' in Lasso model are consistency selected. In terms of Ridge, all feature are important. However, 'Weight' is the only feature in Elasticnet that has a lower frequency than other characteristics. In terms of variation, the number of selected feature are not vary between the mislabeled and correctly label data, as shown by the pink bar, which represents mislabeled observations, and the blue bar, which represents correctly labeled observations.

To answer the question that which set of features are optimal classification performance, Recursive feature elimination with cross-validation (RFECV) are introduced. We can concluded from figure 5 that the f1 score is maximum at 0.80, for the feature selector in this subtask, when the number of selected feature is 6. The model will be maximized by using all features in prediction.It is a similar trend when compared to the three classifiers in 1a, including Gradient Boosting Classifier, Decision Tree Classifier, and Random Forest Classifier. The information are summarized in the table 3. Surprisingly, the number of optimal feature in Decision Tree Classifier is only 5. L2 feature is excluded. Decision Trees use a greedy search to determine the best split at each node. It can be implied that one feature was less significant in reducing the impurity at each split compared to the other features.

For each class, Figure6 consists of three heatmaps that show the feature coefficients for different fish classes. Each cell in the grid reflects the magnitude of the coefficient for a specific feature-class combination, with the color intensity denoting the magnitude. It can highlight features that are influential for classifying each fish species. ”L3” and ”Height” are often important feature across multiple species. ”Weight” of Smelt and Roach is a key characteristic. ”Width” is importance across classes except Silverbream.According to Smelt and Perch, ”L1” is dominant. The only class that is influenced by ”L2” is Roach.

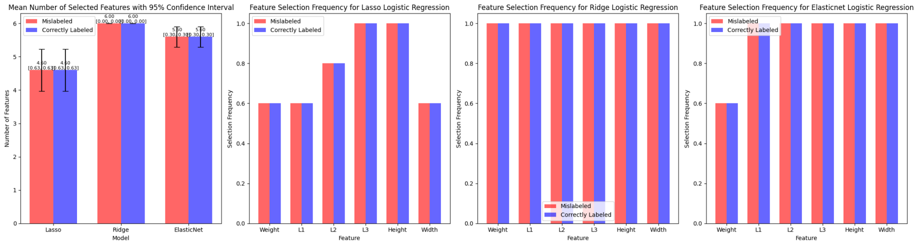

Figure 4: The mean number of selected features across three different logistic regression models (Lasso, Ridge, and Elasticnet Logistic Regression) with 95% confidence level

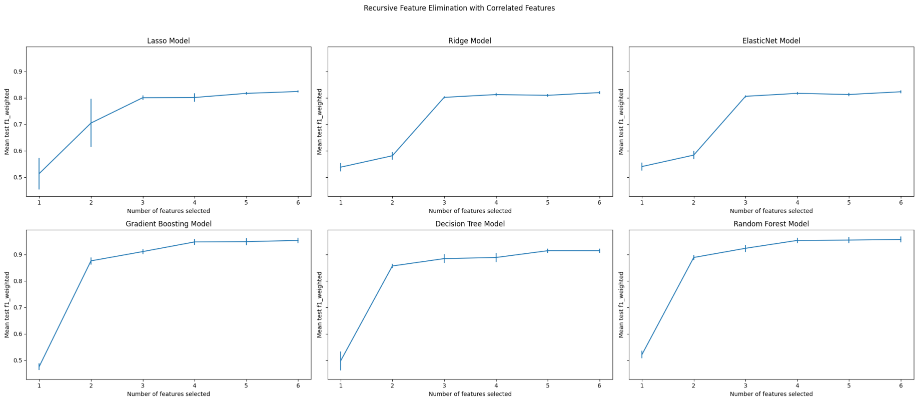

Figure 5: Plots comparing the mean test f1-weighted for different numbers of selected features using Recursive Feature Elimination (RFE) with six different methods: Lasso, Ridge, ElasticNet, Gradient Boosting and Random Forest

Feature Selection	**Optimal Number of Features**	**Selected Features**
Lasso	6	Weight, L1, L2, L3, Height, Width
Ridge	6	Weight, L1, L2, L3, Height, Width
ElasticNet	6	Weight, L1, L2, L3, Height, Width
Gradient Boosting	6	Weight, L1, L2, L3, Height, Width
Decision Tree	5	Weight, L1, L3, Height, Width
Random Forest	6	Weight, L1, L2, L3, Height, Width

Table 3: Optimal Number of Features and Selected Features for Various Models

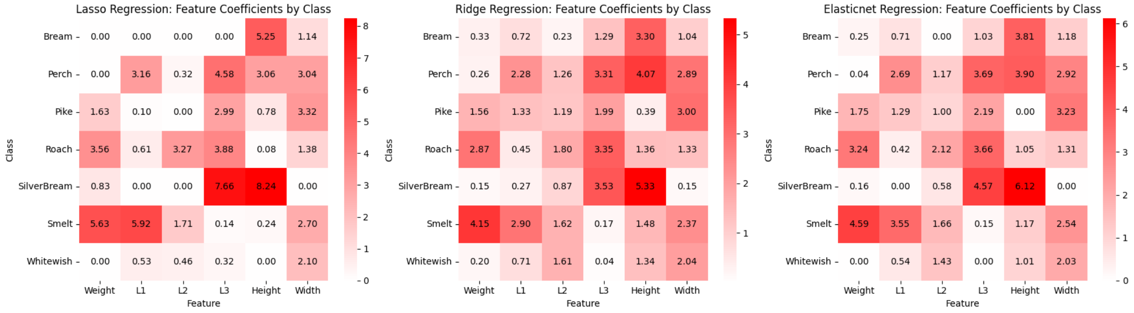

Figure 6: The figure consists of three heatmaps that show the feature coefficients for different fish classes

Q-1c

To perform this task, I write two functions to add simulated features to the dataset. One is for unrelated feature. It is simply add a specified number of unrelated (random) features to the dataset. Another function have more complexity.This function augments a given dataset by generating a specified number of new features that are correlated with the existing ones. Four different methods are utilized to form a new correlated feature to dataset. It contains
1.Linear Combination: Adds two existing features together with some noise.
2.Polynomial Feature: Squares an existing feature and adds some noise.
3.Add Noise: Adds noise directly to an existing feature.
4.Statistical Transformation: Applies a statistical transformation (logarithm, exponential, or square root) to an existing feature and adds noise.

Set of classifier that are used in this subtask include K-Nearest Neighbors classifier(KNN), Support Vector Classifier (SVC), and Random Forest Classifier(RF). Lasso and Elasticnet logistic regression are two feature selection approaches used here. The performance of each simulation are measured by Cohen-Kappa score.

- **Performance**

From the figure7, the performance of K-Nearest Neighbors classifier(KNN) are significantly decreased over number of added features.The Cohen Kappa Score drops from above 0.8 to around 0.3, indicating a substantial decline in classification accuracy. KNN is sensitive to the curse of dimensional. Adding unrelated features increases the depth of the feature space, making it more difficult for the algorithm to discover the closest neighbors effectively.Support Vector Classifier(SVC)'s and Random Forest Classifier's score have minor drop over the range. Random Forest is an ensemble of decision trees, which helps in averaging out the noise and redundancy.It can be implied that these two classifiers are more robust to the addition of unrelated features than KNN.

On the other hand, adding correlated features performance have very minimal impact on the Cohen-kappa score of all three classifiers. The figure8 shows that classification performance remains relatively stable with a modest fluctuation as the number of correlated features increases.

- **Impact on feature selection**

Figure 9 presents the impact of adding unrelated feature on feature selection. Interestingly, the number of feature that two techniques pick growing linearly as increasing of added feature number. The strategy utilized in unrelated feature construction may be the reason of this linear expansion. Elasticnet, which combines both L1 and L2 regularization, tends to select a larger number of features, including more unrelated features, compared to Lasso.

For the adding correlated features, Lasso picks the smaller number of feature, even when correlated features are introduced. Figure 10 depicts the trend of effect after adding correlated feature. Elasticnet's selected feature number when 50 features are added is nearly double that of Lasso's.This is due to the differences in their regularization approaches.

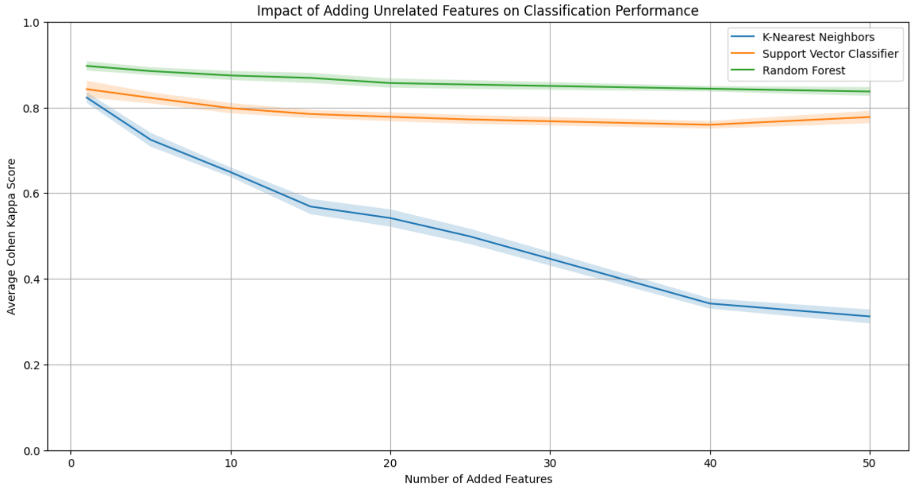

Figure 7: The impact of adding unrelated features on the classification performance of three different classifiers: K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), and Random Forest (RF).

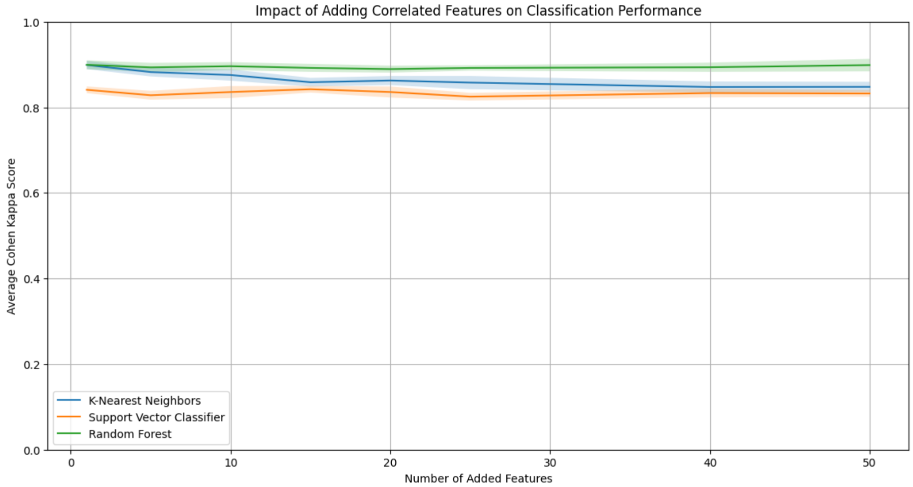

Figure 8: The impact of adding correlated features on the classification performance of three different classifiers: K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), and Random Forest (RF).

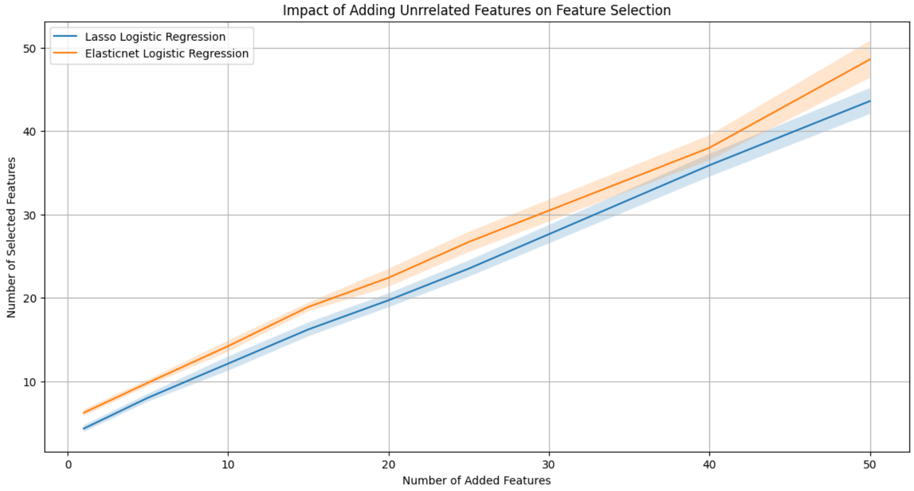

Figure 9: The plot illustrates the impact of adding unrelated features on two feature selection methods:Lasso Logistic Regression and Elasticnet Logistic Regression

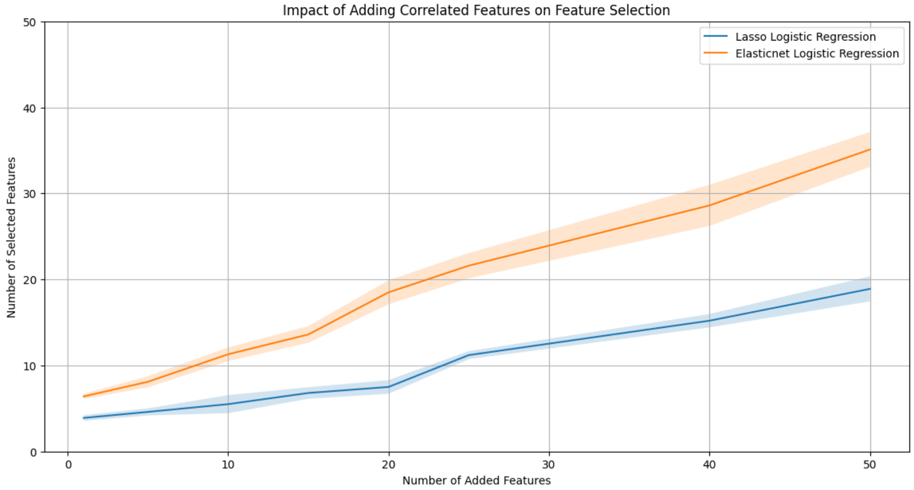

Figure 10: The plot illustrates the impact of adding correlated features on two feature selection methods:Lasso Logistic Regression and Elasticnet Logistic Regression

Q-1d

To complete this task, it is necessary to determine if each observation in the dataset can be classified into one specific class with high confidence or it is better suited to a ”set prediction” if some observations belonging to multiple possible classes.

Firstly, training model on the dataset is a starting point. We can consider a confidence or probability of each observation. Luckily, it can be obtained from attribute of each classifier. Then, threshold setting is a crucial step. In my case, the confidence threshold is set to 0.8. It can be decided on a threshold at which an observation is considered confidently classified. For example, if the probability of the projected class is greater than 0.8, it can be classified it with confidence. For observations in which no one class exceeds the confidence level, it will be considered as a set prediction.

The results of classified with confidence and set prediction are shown in figure 11. Gradient boosting classifier has the leastest proportion of set prediction. It means that the majority of prediction have confidence level over the threshold. Gradient Boosting, which combines many weak learners, is more resistant to over-fitting than other approaches. This can lead to more accurate and confident predictions. In the meantime, KNN have the smallest confidence prediction amount. This classifier heavily depends on the choice of k, the number of neighbors. If k is not chosen appropriately, it might lead to less confident predictions.

The table 4 shows the all common incorrect predictions with high confidence from all classifiers. It can be observed that the confidence of prediction is exceptionally high at approximately 1.00. Surprisingly, Bream are frequently misclassified with a very high confidence level. If we additionally look at figure 12, which shows the label cluster by t-SNE plot, we can find that there are some other label in the Bream cluster, which are potential mislabeling. Moreover, there are other mislabeled spot that result in wrong prediction with high confidence in table 4. Consequently, with these evidences, it is clear that the dataset contains mislabeled observations.

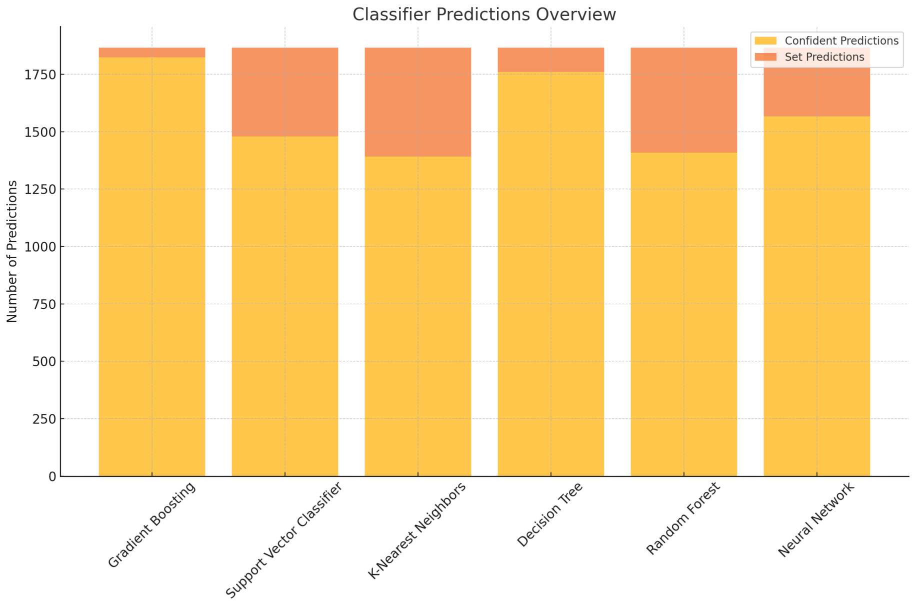

Figure 11: The plot shows comparison of the performance of different classifiers. The chart has two categories of predictions for each classifier: Confidence Prediction and Set Prediction

Table 4: Classifier Predictions with Confidence Scores and Actual Labels

Index	Predicted	Confidence	Actual
118	Bream	0.999474	Pike
302	Bream	1.000000	Perch
388	Bream	1.000000	Roach
447	Bream	1.000000	Smelt
655	Bream	1.000000	Perch
659	Roach	1.000000	Perch
672	Bream	1.000000	Roach
926	Perch	1.000000	Roach
947	Bream	1.000000	Perch
955	Roach	1.000000	Perch
1048	Bream	0.999999	Pike
1091	Roach	0.999990	Perch
1101	Bream	1.000000	Perch
1112	Bream	1.000000	SilverBream
1339	Bream	1.000000	Pike
1415	Perch	1.000000	Roach

Index	Predicted	Confidence	Actual
1462	Bream	1.000000	SilverBream
1495	Bream	1.000000	Smelt
1507	Bream	1.000000	Perch
1552	Bream	1.000000	Smelt
1555	Bream	1.000000	Perch
1596	Bream	1.000000	SilverBream
1603	Bream	1.000000	Smelt
1642	Bream	1.000000	Smelt
1706	Bream	1.000000	Pike

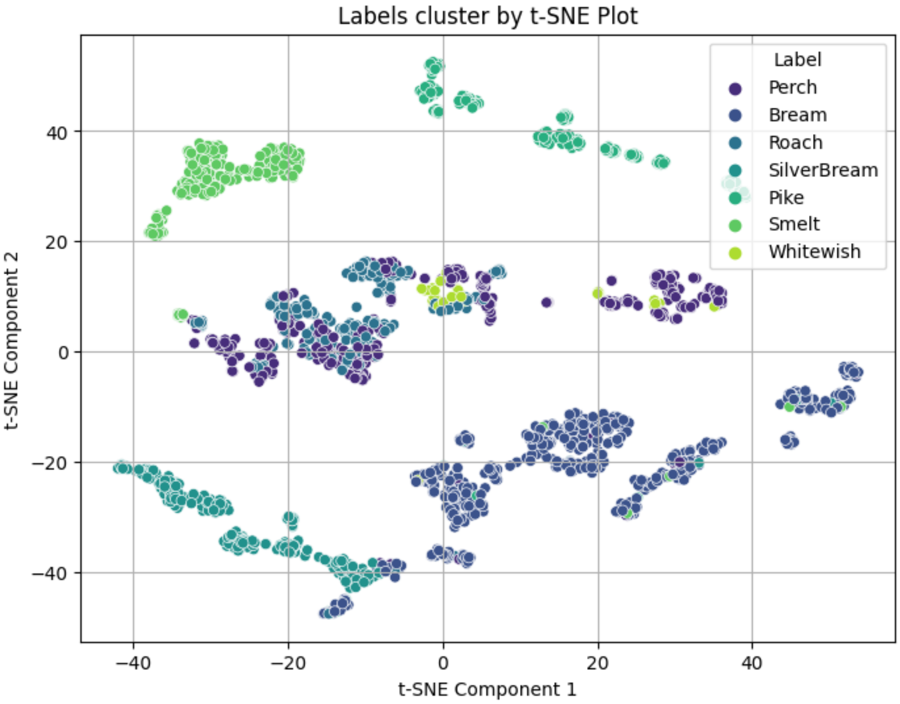

Figure 12: t-SNE plot showing the relationships between the labels