Land Cover Classification Using Multispectral Satellite Data

Gonçalo Cardoso (102458) Guilherme Coimbra (102522) João Santos (102746) Rita Silva (103452) Tiago Santos (103738) Tiago Nóbrega (103863)

Prof. Maria do Rosário De Oliveira Silva, Prof. Catarina Padrela Loureiro

Multivariate Analysis

2024/2025

Abstract

Land cover classification is a complex exercise and is difficult to capture using traditional means. This project aims to classify land cover types using multispectral radiation data from a satellite. After obtaining the data, a preliminary analysis was performed to assess its patterns and characteristics. Various supervised learning methods were then applied to solve the classification problem in question. Considering that some input variables might be irrelevant to the classification problem, we applied dimensionality reduction to refine the dataset. The classification problem was then revisited using the processed dataset, and the results were analyzed to assess the impact of dimensionality reduction on the model performance. In the various classification tasks, we considered different training datasets to explore the impact of data balancing. We began by employing a non-parametric Naive Bayes classifier, allowing us to compare the results obtained from models trained on the original data with those trained on preprocessed and dimensionally reduced data. Subsequently, we conducted a classification task using Discriminant Analysis, employing both Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) classifiers. After this, we used a K-Nearest Neighbors (KNN) classifier. As expected, due to its simplicity the results were worse than for other more complex classifiers like the Logistic Regression. This was the last technique used and the one with the better results achieving an accuracy of 72% for the test set.

Keywords: *Envisat MERIS; CORINE Land Cover; Canonical Correlation Analysis; Principal Component Analysis; Naive Bayes; Discriminant Analysis; k-Nearest Neighbors; Logistic Regression;*

I. Introduction

1. Building the Dataset

This project originated from an unconventional idea, requiring us to create our own dataset from scratch. Firstly, obtaining radiance data from the satellite was challenging, as we were unsure which type of radiance data would be most suitable for our task. The dataset was also massive and in an unfamiliar format, which made processing it even more complex. Another difficulty was manually linking the coordinates of each satellite data point to its corresponding CORINE classification, a meticulous but essential step.

1.1 Satellite Data

In a first attempt we used the data *EN1_MDSI_MER_FRS_1P - Full Resolution Full Swath Geolocated and Calibrated TOA Radiance* [1]. Unfortunately, this dataset was not very useful for the proposed study. TOA means top of atmosphere, and there was not enough difference between the different labels. The lowest correlation between the different features was 95%, which made us take a step back.

This lead us to the used dataset: Envisat MERIS Full Resolution – Level 2 (*MER_FRS_2P/ME_2_FRG*) [2]. This dataset contained top values (topography) referent to the year 2020. The data was in a zip file containing several NetCDF files (Network Common Data Form), a data type commonly designed to store and manage large scientific datasets, particularly those used in geospatial, environmental, and atmospheric sciences. The challenge was to merge the geospatial data (latitude and longitude) with radiance values, as it consisted of more than 20 million points of data before filtering by location.

1.2 CORINE Land Cover

The labels for our dataset were extracted for each observation in the satellite data. To achieve this, we first downloaded a shapefile from the Copernicus Land Monitoring Service website [3], which contained the CORINE Land Cover 2012 classification for all of Europe. Next, to focus on data specific to Portugal, we obtained a shapefile corresponding to continental Portugal from the European Environment Agency website [4].

Using the Portugal shapefile, we filtered the satellite coordinates and then merged the filtered data with the CORINE shapefile to assign labels to our observations, thus constructing the final dataset.

2. Motivation and Objectives

When tasked with studying a dataset using the techniques learned in the Multivariate Analysis course, we aimed to find a meaningful dataset that would provide valuable insights and allow us to take pride in our work while applying the knowledge gained during the course.

This ambition led us to the challenging data mining process described earlier, where we utilized the CORINE Land Cover classification as labels. Our ultimate objective is to explore the possibility of predicting this classification using satellite data, which is obtained more frequently and quickly than the six years of meticulous work carried out by specialists across diverse fields, such as remote sensing.

While this is an ambitious goal, we also have a more pragmatic objective, we want to develop a technique that serves as the foundation for a more efficient approach. By leveraging the power of multivariate analysis, we aim to create a tool that can assist and enhance the efforts of the specialists responsible for the CORINE Land Cover classification.

3. Preliminary Analysis

3.1 Dataset Overview

The dataset consists of 13 spectral bands labeled M01 to M14, with band M11 missing. These bands represent radiance values collected from the MERIS satellite and cover specific spectral ranges. Each band is defined by its center wavelength (in nanometers) and its width. The range extends from 412nm (M01) to 900nm (M14), with applications spanning various purposes such as chlorophyll detection, sediment analysis, and atmospheric corrections. Our dataset consists of about 109,000 observations.

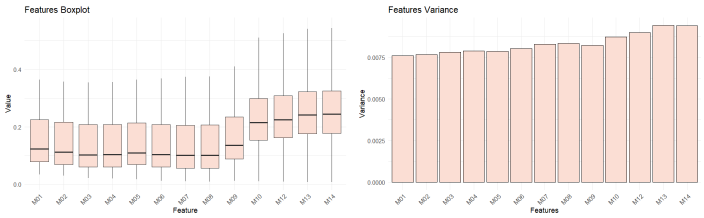
In terms of labels, the dataset includes a hierarchical labeling system organized into three classification layers. The first layer is the broadest categorization level, including land cover types such as artificial surfaces, agricultural areas, forests, wetlands, and water bodies. The two following layers are sub-categories providing more detail, such as rice fields, salines, and olive groves. In this study we focus on Layer 1, which allows us to broadly categorize the land cover types, enabling a solid understanding of the land use.

3.2 Summary Statistics

This subsection presents a statistical summary of the dataset's features to uncover patterns and tendencies. By analyzing measures like correlation and normality, we can study this dataset's suitability for classification and potential issues like skewness, outliers, and redundancy.

First, we look at the features distribution and variance shown in Figure 1. The box-plot reveals two distinct feature ranges: M01 to M09, with lower and tightly distributed values, and M10 to M14, showing greater variability and higher magnitudes. The inter-quartile range for M01 to M09 is compact (0.06 to 0.23), while for M10 to M14, it spans from about 0.15 to 0.32.

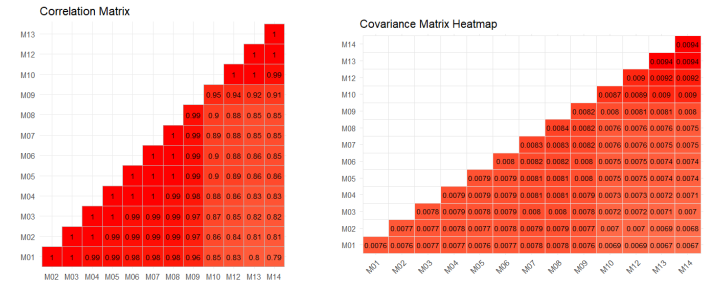
The variance plot shows relatively uniform variance across most features, with only a slight increase observed from lower to higher wavelengths.



(1) *Distribution* **(2)** *Variance*

Figure 1: *Features box-plot and variance visualized*

Next, we visualize how the features correlate.



(1) *Correlation Matrix* **(2)** *Covariance Matrix*

Figure 2: *Features correlation and covariance*

The correlation matrix shows high values among the 13 features, indicating a strong linear relationship between them. Again, there is a separation between the two groups of features, with the bands from M01 to M09 exhibiting almost perfect correlations with each other, with values nearing 1, and the bands M10 to M14 showing slightly lower but still significant correlations, with values ranging between 0.79 and 0.99. This distinction suggests that the features in the first group are more homogeneous and may carry redundant information, while the second group shows slightly more variability, potentially capturing different patterns in the data. This separation shows the possibility of considering dimensionality reduction techniques to simplify the dataset without losing critical information. It is also notable how the radiations close to each other in wavelength represent higher correlation values, which makes sense because the spectrum of radiance is emitted continuously.

The covariance matrix also carries valuable insights into the relationships between the spectral bands. Again, the covariance values are significant and gradually diminish as the distance from the main diagonal increases. In this matrix, M10 to M14 bands have notably higher covariance values than the other group, indicating that these features vary more strongly together compared to the other group of features.

3.3 Data Imbalance

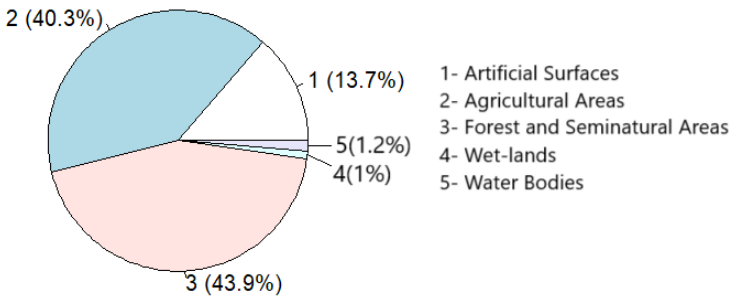


Figure 3: *Class distribution*

The class distribution depicted in the pie chart reveals a notable imbalance among the categories in the dataset. Class 3, representing forests, dominates with 43.9% of the data, followed by Class 2 (agricultural areas) at 40.3%. In contrast, Classes 1, 4, and 5, which correspond to artificial surfaces, wetlands, and water bodies, respectively, are significantly underrepresented, together accounting for less than 16% of the dataset. This imbalance is expected and reflects the natural distribution of land cover types in Continental Portugal. Forests and agricultural areas are typically more prevalent, while wetlands and water bodies are less common. This makes for a skewed distribution and will pose challenges for classification models, as some classes are very underrepresented.

3.4 Multivariate Normality Assumption

For a deeper understanding of the relationships between the dataset's features, we tested for multivariate normality. This statistical property, when verified, ensures that the joint distribution of two or more continuous variables follows a multivariate normal distribution. Multivariate normality is a critical assumption in many statistical techniques and machine learning algorithms, including Discriminant Analysis. The following plots include a reference line that represents the ideal relationship expected for data that satisfy the multivariate normality assumption.

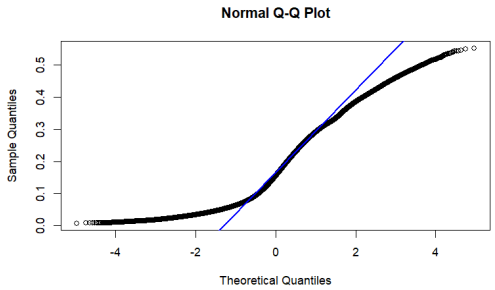
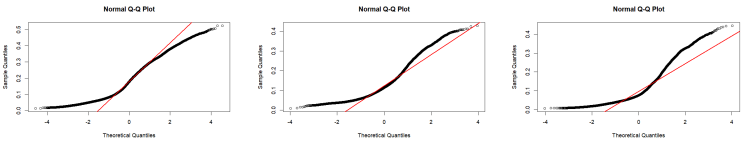


Figure 4: *QQ plot of the dataset*

Observing Figure 4, we conclude that the alignment of points along the reference line in the central quantiles suggests that the data is approximately normally distributed in this region. However, the deviations from the diagonal line on both ends of the plot indicate potential departures from normality, which could be attributed for example to skewed data or outliers.

Studying the normality of the data grouped by label classes, we observe similar plots for classes 1, 2, and 3. For classes 4 and 5, which have lower representation, the evidence for normality is weaker.



(1) *Class 1* **(2)** *Class 4* **(3)** *Class 5*

Figure 5: *QQ-Plots for observations of each label*

II. Preprocessing and Dimension Reduction

1. Data Splitting and Normalization

Having terminated the preliminary analysis, the dataset was then split into training, validation and test sets. This was achieved using stratified sampling to ensure proportional representation of the target variable across all subsets. A 70/15/15 split was applied, with 70% of the data assigned to the training set, while the remaining 30% was equally divided between validation and test sets.

Following the split, the features were standardized. This was done by subtracting the mean and dividing by the standard deviation. It is important that these two values are computed solely from the training set, in order to ensure that the model evaluates unseen data under identical conditions to those of the training data. This approach allowed us to generate both a scaled dataset and an unscaled dataset, enabling us to work with and compare the results from both versions.

2. Canonical Correlation Analysis

In this section, a canonical correlation analysis of the data will be done. Canonical correlation analysis (CCA) aims to analyze the relationships between 2 sets of variables. Since the information of what each radiance feature detects and it's target wavelength is available to us it's possible to study the shared structure of data in 2 different contexts, in an attempt to perform dimensionality reduction.

The first CCA will aim to relate bands 1–8, which represent water quality indicators, with bands 9–14 (keep in mind that 11 is not present in the data), representing vegetation and atmospheric indicators. Bands 1–8 are associated with detecting chlorophyll, suspended sediments, and dissolved organic matter, providing insights into aquatic ecosystem health and water clarity. Bands 9–14 capture vegetation reflectance, atmospheric corrections, and water vapor absorption, critical for understanding terrestrial processes and minimizing atmospheric interference.

No.	Band centre (nm)	Band width (nm)	Application
1	412.5	10	Yellow substance and detrital pigments
2	442.5	10	Chlorophyll absorption maximum
3	490	10	Chlorophyll and other pigments
4	510	10	Suspended sediment, red tides
5	560	10	Chlorophyll absorption minimum
6	620	10	Suspended sediment
7	665	10	Chlorophyll absorption & fluo. reference
8	681.25	7.5	Chlorophyll fluorescence peak
9	708.75	10	Fluo. reference, atmosphere corrections
10	753.75	7.5	Vegetation, cloud
11	760.625	3.75	O2 R- branch absorption band
12	778.75	15	Atmosphere corrections
13	865	20	Vegetation, water vapour reference
14	885	10	Atmosphere corrections
15	900	10	Water vapour, land

Figure 6: *Definition of Meris Bands*

This analysis will allow us to explore the relation between aquatic and terrestrial systems. Besides that, we can clearly see a division line between the correlation values when we compare the first 8 values with the last 5, so this will probably prove helpful to achieve dimensionality reduction.

Component	1	2	3	4	5
Correlation Value	0.9996	0.8541	0.5220	0.4465	0.1319

Table 1: *Canonical Correlation Values*

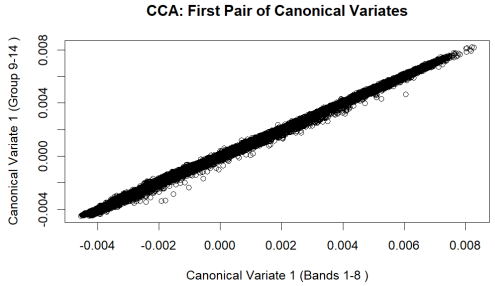


Figure 7: *Plot of the first pair of Canonical Variates*

From the first value of the canonical correlation, we can conclude that these 2 groups of variables represent a near-perfect correlation indicating a very strong linear relationship between each variable of each group. This suggests that changes in water properties are closely associated with changes in vegetation reflectance and atmospheric conditions. This can be because of land-water interactions and climate influence showing that for small territories these 2 sets of variables are highly interlined.

Even though it may have dimensionality reduction potential we still need to check how much of the original variance is explained by just the first canonical pair:

Pair	Canonical Correlation	Total Variance Explained
1	0.9996	0.9117
2	0.8541	0.0603

Table 2: *Total Variance Explained by the first 2 Canonical Correlation Pairs*

As expected, the total variance explained by just the first pair is almost total, with 91.1%. The 2nd pair explains 6%. It must be taken into account that the sum of the variance explained by all the pairs does not necessarily equal 100% because CCA focuses on maximizing correlation between the two groups at question.

The next CCA will be done in the context of the different wavelengths, for each radiation. It's going to be divided in 3 sets: bands 1-5 represent blue-green spectrum, while bands 6-9 represent yellow-red and the remaining bands cover near-infrared wavelengths.

Analysis	**1st Pair**	**2nd Pair**	**3rd Pair**	**4th Pair**
Blue-Green vs Yellow-Red	0.9998	0.9107	0.6441	0.2412
Blue-Green vs Near-Infrared	0.9935	0.6121	0.4153	0.0189
Yellow-Red vs Near-Infrared	0.9971	0.8182	0.3462	0.0929

Table 3: *Canonical Correlations for Different Analyses (Rounded to 4 Decimals)*

Although the values are all very high and similar to each other, we can check that radiations close to each other in wavelength represent higher correlation values. This makes sense, as the spectrum of radiance is emitted continuously. However, with such higher correlation values for CCA, one can conclude that this analysis was redundant and did not give any useful information about the dataset. Nevertheless, the first analysis was useful to understand the impact of the climate and homogeneity of the territory.

3. Principal Component Analysis

As an alternative dimension reduction technique to CCA, principal component analysis (PCA) was considered. The goal of PCA is to create new features (called principal components) by taking linear combinations of the original ones, and finding the orthogonal directions that maximize the variance of a given dataset, and, thus, can capture the most significant patterns present.

Since our original dataset has 13 features, we concluded that it would be relevant to perform PCA to try to effectively reduce the dimensions of our problem. To learn the PCA parameters only the training data set was used in order to prevent leakage of information between the train, validation and test sets that might compromise our classification results. The validation and test sets were then transformed as a posterior step. The results obtained for the first 3 principal components can be found in Table 4. Note that both the original data and the standardized version were used for comparison purposes.

	PC 1	PC 2	PC 3
Eigenvalue	1.01×10^{-1}	6.75×10^{-3}	2.81×10^{-4}
Cumulative Variance Explained (%)	93.46	99.69	99.98
Eigenvalue (Stand)	12.18	0.77	0.04
Cumulative Variance Explained (Stand, %)	93.73	99.68	99.96

Table 4: *Results obtained for the 3 first principal components for standardized (Stand) and non standardized data.*

The average of the eigenvalues obtained for the original dataset was 0.02 and for the standardized data was 2.17. As can be seen, the results achieved for the standardized and non standardized data were extremely similar, since all the features were in the same scale. We opted to use the standardized values since it is the more common approach when performing PCA, because it ensures that all the features contribute equally regardless of their magnitudes and no bias are introduced. By observing Table 4, we conclude that the first 2 principal components should be retained, since they are able to explain 99,7% of the total variance, which we considered to be a highly satisfactory value.

Moreover, in order to better understand the contribution of each feature to each principal component, the loadings for PC1 and PC2 were computed and the results can be found in Table 5.

Features	PC1 Loadings	PC2 Loadings
M01	-0.277	-0.268
M02	-0.279	-0.247
M03	-0.281	-0.225
M04	-0.282	-0.207
M05	-0.284	-0.144
M06	-0.284	-0.153
M07	-0.283	-0.159
M08	-0.283	-0.155
M09	-0.286	0.001
M10	-0.272	0.354
M12	-0.269	0.386
M13	-0.263	0.446
M14	-0.262	0.455

Table 5: *Loadings of each standardized feature for PC1 and PC2.*

All features contribute in an akin way to the first principal component, which is the direction that explains 93% of the variance. Thus, no feature will be dropped. We can conclude that PC1 captures the overall variance across the entire dataset, rather than focusing on specific feature groups and, thus, it probably is associated with a general pattern in radiation values, combining both water quality (features M01-M08) and vegetation and atmospheric contributions (M9-M14). For PC2, we can observe that features associated with water (M01-M08) and features related to vegetation and atmospheric conditions (M10-M14) exhibit different behaviors (feature M09 can be discarded from this analysis due to its small value). Therefore, it is possible to conclude that PC2 seems to capture the difference between these two groups of features, giving more relevance to the variance related to the vegetation and atmospheric data (since features M10-M14 possess larger PC2 loadings).

The projection of the standardized dataset onto the first and second principal components was computed and the plot in Figure 8 was obtained.

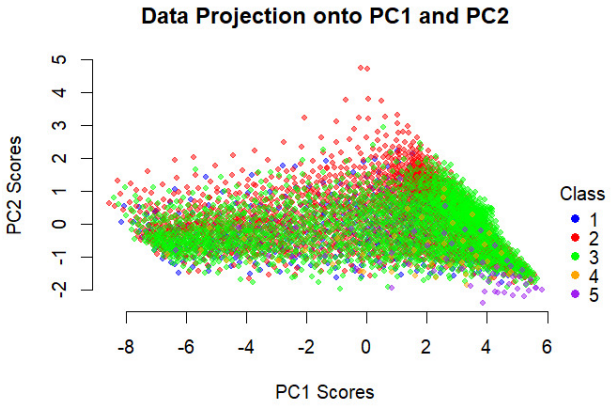


Figure 8: *Projection of the standardized dataset onto the first and second principal components.*

By observing the plot it is possible to conclude that even in the PCA space, the distinction between classes is not very clear. Nevertheless, we can infer that the majority of the observations with a high PC2 score belong to class 2, thus, we can conclude that agricultural areas are associated with a higher vegetation related variability (such as differences in chlorophyll absorption, fluorescence or atmospheric influences), as expected. The observations with higher PC1 score and lower PC2 score belong to class 5, which indicates that water bodies probably show more consistent and distinct radiation signals, possibly due to water absorption and scattering phenomena. Observations from classes 1, 3, and 4 do not exhibit a significant pattern and, even though class 3 scores for PC2 approximately range from -2 to 2, this does not distinguish them from observations belonging to the other classes.

By analyzing the plot we can also observe that PC1 captures a higher variance than PC2, as expected, however it does not imply a higher class separability, as most of the classes overlap across this axis. This difficulty to distinguish observations from different classes is in accordance to the high correlation that we can observe between features (which suggests a shared variance across the dataset), particularly within groups of water and vegetation and atmospheric indicators. This results in a reduced PCA's ability to differentiate the classes, since it focuses on finding common patterns. Note that the plot presented does not include all the observations but instead contains the average of the thirty closest points of each class for better interpretability.

3.1 PCA on the balanced dataset

In order to assess if balancing the standardized training dataset would improve the PCA performance and, thus, increase class separability in the transformed space, the PCA algorithm was applied to the balance dataset (Dataset 3 that is mentioned below in III). However, the results did not improve as can be seen by the plot in Figure 9 that represents the projection of the balanced dataset onto the first and second principal components. Akin to the PCA of the original dataset, the first principal component can explain 93.02% of the total variance and the second 99.63%. Thus, the first two principal components were retained as well.

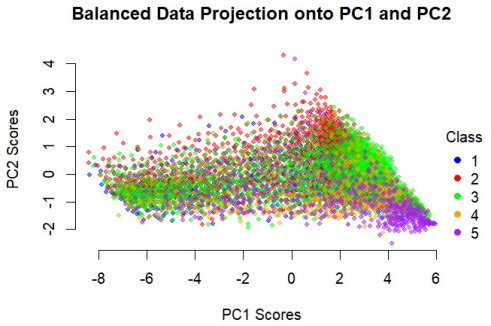


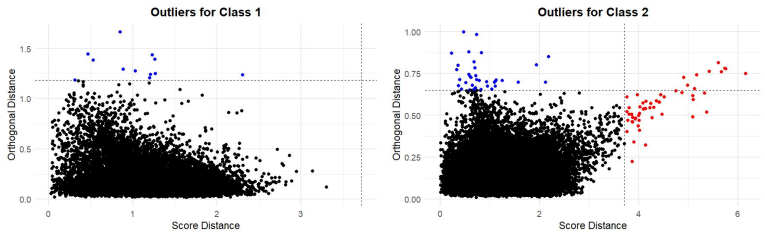
Figure 9: *Projection of the balanced standardized dataset onto the first and second principal components.*

Class separability in the PCA reduced space did not improve by using the balanced dataset, which further supports the claim that the variance in the dataset is probably dominated by features or patterns that are shared across all the classes (namely the majority classes of the original dataset) and is not directly influenced by the number of observations belonging to each class.

3.2 Outlier Detection

In addition to being used as a dimension reduction technique, PCA can also be implemented as an outlier detection method. To achieve this, the following plots in Figure 10 were obtained for each class (these plots only correspond to the results obtained for classes 1 and 2, the other plots can be found on the Appendix, in Figure 25).

The outliers were analyzed per class (in order to ensure the reliability of the results) by computing the score distance, SD, (which corresponds to the mahalanobis distance in the PCA space) and the orthogonal distance, OD, (which is the distance of an observation to the PCA space) for each point. The thresholds for the distances used to classify certain observations as outliers depend of the significance levels chosen, in our case it was $\alpha_{SD} = 0.001$ and $\alpha_{OD} = 0.00001$.



(1) *Outliers in class 1* **(2)** *Outliers in class 2*

Figure 10: *Outlier detection with PCA for classes 1 and 2. The outliers flagged only due to an excessive orthogonal distance are represented in blue and the ones identified due to their excessive score distance are colored in red. The black points are the regular observations.*

In class 1, 12 outliers were identified and in class 2, 90. In class 3, 191 outliers were found, in class 4, 2, and in class 5, 12. In total, we concluded that there were 307 outliers in the dataset. As the CORINE LandCover is from the year 2012, and the MERIS dataset is from 2020, the number of outliers obtained was expected, since some observations may not coincide with their given class. From 2012 to 2020 some observations have obviously changed class, as there was construction of houses, wild fires that devastated forests and some water bodies that might have reduced in size or completely disappeared due to climate change.

III. Classification

To solve our classification problem, we decided to use supervised learning methods. The following subsections present each of these models. A classification task was performed both before data preprocessing and dimensionality reduction, and again after these procedures. Given the highly imbalanced data, we considered different training datasets with varying balances during these tasks. The training dataset was balanced through oversampling and undersampling, i.e., by increasing the representation of the minority classes and decreasing the number of observations of the majority classes to match the

desired representation percentages. This was achieved by duplicating existing samples. Balancing the training data aims to ensure that the classification models are not overly biased towards the majority class and can perform more effectively across all classes. For the Bayes, Discriminant Analysis and k-Nearest Neighbors classifiers, we considered 3 different train datasets:

- Dataset 1: the original imbalanced training data;
- Dataset 2: derived from the original training data to achieve a more balanced dataset with proportions of 15%/35%/40%/5%/5% for classes 1, 2, 3, 4, and 5, respectively;
- Dataset 3: derived from the original training data to achieve an even more balanced dataset with proportions of 20%/30%/30%/10%/10% for classes 1, 2, 3, 4, and 5, respectively.

For the Logistic Regression, we only considered Dataset 1 and 2, because the results obtained with Dataset 2 and 3 were very similar and showed no significant differences in overall performance. To evaluate the performance of each classification task, we decided to consider the following traditional metrics:

1. **Precision**: measures the proportion of correctly predicted positive observations out of all predicted positive observations. It is calculated as:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

   Precision is essential when the cost of false positives is high.

2. **Recall**: measures the proportion of correctly predicted positive observations out of all actual positive observations. It is calculated as:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

   Recall is crucial when the cost of false negatives is significant.

3. **F1-score**: is the harmonic mean of Precision and Recall, providing a single metric that balances both concerns. It is calculated as:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

   The F1-score is particularly useful when dealing with imbalanced datasets.

4. **Accuracy**: measures the proportion of correctly classified observations out of the total number of observations. It is calculated as:

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Observations}}$$

In our case, accuracy could be a misleading metric when working with imbalanced data. For example, if we have a class representing 50% of the observations and our classifier assigns all the observations to this class, we would achieve an accuracy of 50%, despite the model's poor performance across other classes.. Thus, analyzing the values of Precision, Recall, and F1-score provides a more robust evaluation of the model's performance.

1. Naive Bayes Classifier

To address the classification problem, we started by employing a Naive Bayes classifier. Naive Bayes classification is a simple yet powerful probabilistic machine learning algorithm based on applying Bayes' theorem. It assumes conditional independence between features given the class label and performs better when data exhibit a normal distribution and are balanced. Given these assumptions and considering the data characteristics, including high feature correlation, limited multivariate normality, and significant class imbalance, we anticipated poor performance from the classifier.

To address these limitations, we utilized a non-parametric variant of the Naive Bayes classifier. Specifically, we employed the R function `nonparametric_naive_bayes` from `naivebayes` package. This customized version calculates class-conditional probabilities non-parametrically using a kernel density estimator (KDE) and accommodates features with continuous values (numeric or integer). By default, the smoothing bandwidth is determined using Silverman's "rule of thumb" with a Gaussian kernel. While we will not delve into the technical details, this approach provides a more flexible alternative to the traditional parametric Naive Bayes classifier.[5]

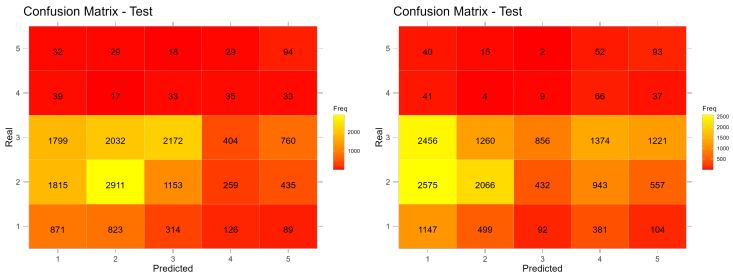
Original Data

Table 6: *Metrics for Test Data with Overall Accuracies: 0.37, 0.32, and 0.26.*

Class	Trained w/ Dataset 1			Trained w/ Dataset 2			Trained w/ Dataset 3		
	P	R	F1	P	R	F1	P	R	F1
1	0.19	0.39	0.26	0.19	0.44	0.26	0.18	0.52	0.27
2	0.50	0.44	0.47	0.52	0.38	0.44	0.52	0.31	0.40
3	0.59	0.30	0.40	0.61	0.21	0.31	0.62	0.12	0.20
4	0.04	0.22	0.07	0.05	0.35	0.05	0.05	0.46	0.08
5	0.07	0.47	0.12	0.05	0.46	0.05	0.05	0.46	0.07

We confirm our assumptions by observing the values presented in Table 6. Analyzing the overall evolution of the metrics with the increase in the balance of the training dataset, we conclude that this increase leads to worse classification results. The confusion matrices displayed in Figure 11 depict the differences between the classifications obtained using Train Dataset 1 and Train Dataset 3. From the matrices, it is possible to confirm the very poor performance of the classifier.

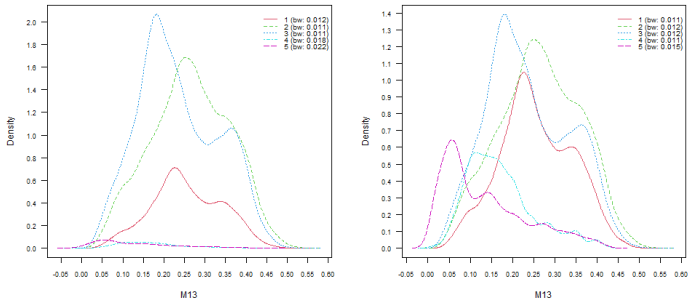
The oversampling and undersampling of the training data led the model to fail in capturing the characteristics of class 3 observations, as seen in the reduced number of predictions assigned to this class. This behavior can be attributed to the inherent simplicity of the Naive Bayes algorithm, which struggles with imbalanced data and highly correlated features.



(1) *Trained w/ Dataset 1* **(2)** *Trained w/ Dataset 3*

Figure 11: *NB Classification - Confusion Matrices for test data*

Further analysis reveals that balancing the training data negatively affected the model's capacity to predict even the more representative classes. The model allowed us to plot the fitted distributions for each class across different features. In Figure 12, we present the fitted distributions for the *M13* band in two scenarios: training with the imbalanced data (Dataset 1) and training with Dataset 3. These plots demonstrate the effects of balancing, with increased representation for underrepresented classes, but at the cost of likely misrepresenting the patterns in the original data.



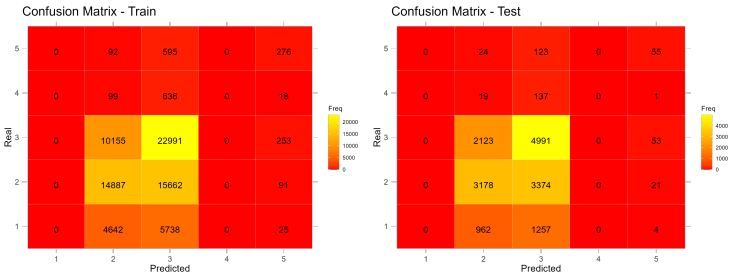
(1) *Trained w/ Dataset 1* **(2)** *Trained w/ Dataset 3*

Figure 12: *NB Classification - Fitted distributions for M13 band for each class*

The algorithm's assumption of feature independence, the characteristics of the data, and the impact of synthetic sampling techniques likely contributed to the observed performance degradation.

Preprocessed and Dimension Reduced Data

Using the datasets after applying preprocessing methods and dimensionality reduction, we re-evaluated the classifier. The confusion matrices obtained for the training (Dataset 1) and test sets are shown in Figure 13.



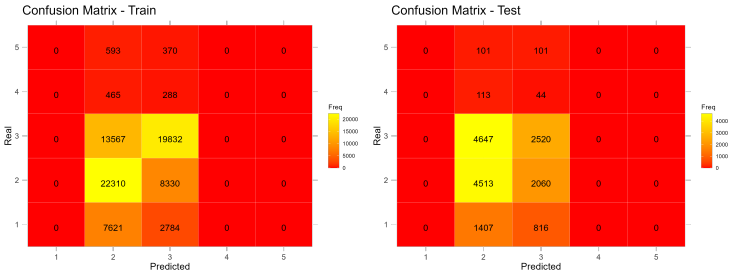
(1) *Train* **(2)** *Test*

Figure 13: *NB Classification after preprocessing and dimensionality reduction- Confusion Matrices for train and test data*

From the confusion matrices, we observe a deterioration in performance. This suggests that the dimensionality reduction did not positively impact the model. The classifier predicts observations only for Classes 2 and 3. Given the high correlation and imbalance in the data, the two principal components retained, which represent 99.7% of the data's variance, proved ineffective in this case.

Dimensionality Reduction After Data Balancing

Given the poor results obtained previously, we explored another approach. We balanced the training dataset to match the proportions of Dataset 3 and then performed PCA. Using the results from PCA, we applied dimensionality reduction to the original data, retaining two dimensions. We then performed classification again. However, the results, shown in Figure 14, exhibit the same patterns as those obtained when using PCA on unbalanced data, as seen in Figure 13. The classifier continues to assign observations predominantly to classes 2 and 3, similar to the previous case.



(1) *Train* **(2)** *Test*

Figure 14: *NB Classification After Dimensionality Reduction Using Balanced Data - Confusion Matrix for Train and Test Data*

Discussion

We verified a poor performance of the Naive Bayes classifier across all metrics, especially when applied to more balanced datasets. The simplistic assumption of conditional independence between features was not compatible with the data under study which exhibit high feature correlations. Balancing made the problem worse, as the adjustments distorted the underlying probability distributions.

Dimensionality reduction using PCA did not improve performance, despite retaining over 99% of the variance. This result shows that PCA looks at overall variation but fails to capture the specific details of each class. Furthermore, applying PCA after balancing gave similar results to applying it before balancing.

These results suggest that Naive Bayes is not a good fit for this classification problem, particularly when dealing with datasets that include complex relationships between features and overlapping distributions. Consequently, we decided to discard dimensionality reduction for the remaining classification tasks.

2. Discriminant Analysis

Discriminant Analysis is one of the oldest and most well-established classification techniques [6]. It includes methods such as Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA), each tailored to specific data characteristics and assumptions. These methods are grounded in probability and linear algebra principles, making them interpretable and computationally efficient.

LDA assumes that the data within each class is normally distributed and shares the same covariance matrix. QDA, on the other hand, does not assume that data shares the same covariance.

In our case, where we have a high correlation among features, deviations from normality, and class imbalance, we anticipate challenges in applying these methods effectively. In the following topics, we analyze the performance of these discriminant methods on our dataset.

2.1 LDA - Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a classification method that finds linear decision boundaries between classes. Given that we have 5 classes in our dataset, the model generates 4 discriminant functions. To perform this task we utilized the R package `caret` [7] which relies on the `MASS` package for Linear Discriminant Analysis.

Original Data

We performed LDA using the three different train datasets. The precision, recall, and F1-score metrics for each class for the test data are presented in Table 7. The overall accuracies for the test data were 0.71, 0.70, and 0.71, respectively.

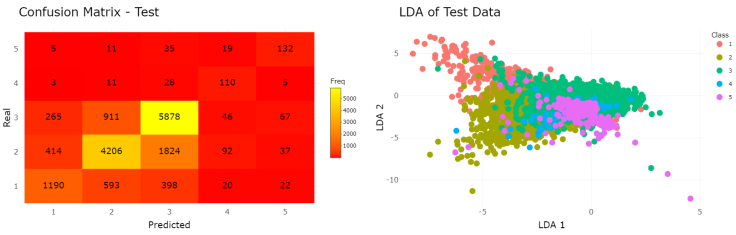
Table 7: *Original data: Metrics for Test Data with Overall Accuracies: 0.71, 0.70, and 0.71.*

Class	Trained w/ Dataset 1			Trained w/ Dataset 2			Trained w/ Dataset 3		
	P	R	F1	P	R	F1	P	R	F1
1	0.74	0.43	0.54	0.72	0.46	0.56	0.63	0.54	0.58
2	0.73	0.65	0.68	0.74	0.62	0.67	0.73	0.64	0.68
3	0.70	0.86	0.77	0.69	0.86	0.77	0.72	0.82	0.77
4	0.58	0.49	0.53	0.51	0.62	0.56	0.38	0.70	0.50
5	0.48	0.54	0.51	0.51	0.59	0.55	0.50	0.65	0.57

From Table 7, we observe variations in the metrics across the three datasets. There is a notable improvement in the recall metrics for the underrepresented classes as the balance of the training data improves. This improvement can likely be attributed to the increased number of observations for these classes, allowing the model to better learn their characteristics and enhance its ability to predict their occurrences. However, this comes at the cost of decreased precision for these same classes. The oversampling process may have introduced redundancy, potentially leading the model to overestimate the presence of these classes in the data.

An inverse pattern is observed for the metrics of the most representative classes. As their proportion in the training data decreases, their recall declines, while precision shows relative stability or slight improvement. Overall, the F1-scores remain comparable across the three cases, balancing the opposing trends in recall and precision for each class.

To further illustrate the classifier's performance, we analyze the model trained with Dataset 3, as shown in Figure 15. This figure presents the confusion matrix and a visualization of the projected data onto the first two LDA directions, providing insight into class separability and the model's predictive behavior.



(1) *Test - Confusion Matrix* **(2)** *Projection onto First Two LDA Directions*

Figure 15: *Original Data: LDA using train dataset 3. Test - Confusion Matrix and Projected Data Visualization*

From the confusion matrix in Figure 15 (a), we observe that the classifier shows high accuracy for Class 3. However, the minority classes (Classes 4 and 5) still experience significant misclassifications, as evident from the scattered off-diagonal values. The visualization of the projected data onto the first two LDA directions in Figure 15 (b) shows that there is a noticeable overlap between the minority classes and others. This overlap explains the confusion observed in the classification results for Classes 4 and 5. Overall, the balanced dataset improves class separability, particularly for underrepresented classes, but challenges remain due to inherent overlaps in feature space.

Preprocessed Data

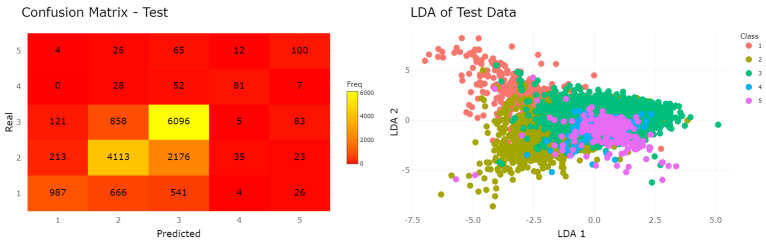
Considering the preprocessed data, we repeated the classification task, training the model with both the imbalanced training dataset and training dataset 3. The respective precision, recall, and F1-score metrics are presented in Table 8.

From the table, we observe similar behavior to the results obtained with the original dataset. The precision for less representative classes decreases when balancing is applied, while recall improves for the same reasons discussed previously. However, the overall F1-score is worse for the classification using training dataset 3, which was not the case with the original data. This leads us to conclude that using a more balanced dataset with standardized variables is not beneficial for this classification task.

Table 8: *Standardized Data: Metrics for Test Data with Overall Accuracies: 0.70 and 0.70.*

Class	Trained w/ Dataset 1			Trained w/ Dataset 3		
	P	R	F1	P	R	F1
1	0.75	0.44	0.56	0.64	0.54	0.59
2	0.72	0.63	0.67	0.73	0.62	0.67
3	0.68	0.85	0.76	0.71	0.81	0.76
4	0.59	0.48	0.53	0.34	0.73	0.46
5	0.42	0.48	0.45	0.40	0.51	0.45

The confusion matrix and the projection of the test data onto the first two LDA directions are displayed in Figure 16.



(1) *Test - Confusion Matrix* **(2)** *Projection onto First Two LDA Directions*

Figure 16: *Standardized Data: LDA using train dataset 1. Test - Confusion Matrix and Projected Data Visualization*

Analyzing the confusion matrix and the graphic of the projections onto the first two LDA directions of the test data, we observe similar results to those in Figure 15. There remains a significant overlap of points projected onto the first two LDA directions, making it difficult to establish clear linear separation boundaries. Furthermore, the data imbalance exacerbates the model's difficulty in capturing patterns within the observations.

2.2 QDA - Quadratic Discriminant Analysis

Like LDA, Quadratic Discriminant Analysis (QDA) seeks to maximize the separation between classes. However, it achieves this by allowing for more complex decision boundaries. Unlike LDA, QDA models each class with its own covariance structure, leading to quadratic decision boundaries. This flexibility enables QDA to handle cases where the variance of features differs significantly across classes. Once again, we used R's `caret` package to perform this task.

Original Data

We repeated the same classification procedure performed for LDA, using QDA with the three training datasets. Table 9 summarizes the precision, recall, and F1-scores for the test data, along with the overall accuracies for each case, which were 0.70, 0.69, and 0.69, respectively.

Table 9: *Class-wise Metrics for QDA Test Data with Overall Accuracies: 0.70, 0.69, and 0.69.*

Class	Trained w/ Dataset 1			Trained w/ Dataset 2			Trained w/ Dataset 3		
	P	R	F1	P	R	F1	P	R	F1
1	0.70	0.47	0.56	0.68	0.49	0.57	0.61	0.56	0.58
2	0.74	0.60	0.67	0.75	0.58	0.66	0.76	0.58	0.66
3	0.67	0.86	0.75	0.67	0.86	0.75	0.69	0.83	0.75
4	0.67	0.55	0.61	0.51	0.71	0.59	0.36	0.71	0.48
5	0.55	0.54	0.54	0.50	0.59	0.54	0.45	0.64	0.53

From Table 9, we observe that QDA exhibits similar behavior compared to LDA. As the balance of the training data increases, there is a slight improvement in recall for the underrepresented classes, likely due to the increased number of observations allowing the model to better learn their patterns. However, this comes at the expense of precision, which declines for these classes, potentially caused by the oversampling process leading the model to overestimate their prevalence. The F1-scores for the more balanced training datasets are slightly worse than those obtained with the unbalanced dataset. This suggests that the trade-off between precision and recall does not consistently favor improved overall performance when balancing the training data.

Since the overall F1-scores are highest for the model trained with the unbalanced data, we present the confusion matrix for the test data in Figure 17. It is notable that the confusion matrix is similar to that obtained for the LDA classification. However, there is a higher number of observations incorrectly predicted as Class 3, which is a clear consequence of the imbalanced training data.

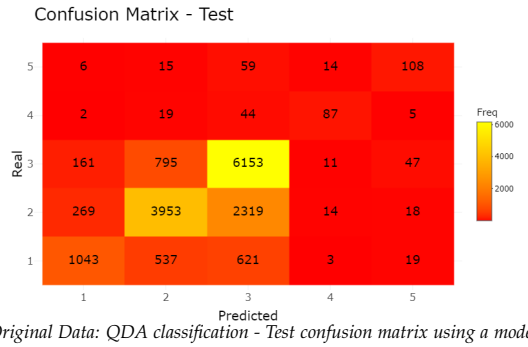
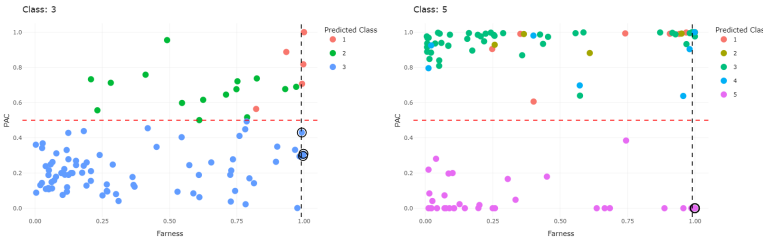


Figure 17: *Original Data: QDA classification - Test confusion matrix using a model trained with dataset 1*

To provide a different visualization of the results for this model, we selected 100 observations from each class in the unused validation dataset. For these observations, we calculated the probabilities of the alternative class (PAC) and the farness from the true class. Using these values, we plotted a class map, which is a graphical representation of the PAC against the farness for the observations of each class. The resulting plots for Classes 1, 3, and 5 are shown in Figure 18. For further details about this methodology, we refer to the paper *Class Maps for Visualizing Classification Results* [6].

(1) *Class 1*



(2) *Class 3* **(3)** *Class 5*

Figure 18: *QDA- Class maps for a small validation sample*

From the class map for Class 5, we observe that the PAC values are either close to 0 or close to 1. This indicates that the misclassified observations tend to have posterior probabilities close to 1 for the predicted class and/or very low posterior probabilities for the true label. Comparing the class maps for Classes 1 and 3, we notice a higher number of correctly predicted observations for Class 3. This is expected, as Class 3 is the most represented, allowing the model to predict it more accurately. For Class 1, we observe a smaller number of observations misclassified as Classes 4 and 5, likely due to the reduced representation of these classes in the training dataset used to train the model.

Preprocessed Data

We repeated the classification task using preprocessed data. The model was again trained using training datasets 1 and 3. The metrics obtained are shown in Table 10.

Table 10: *Standardized Data: Metrics for Test Data with Overall Accuracies: 0.69 and 0.69.*

Class	Trained w/ Dataset 1			Trained w/ Dataset 3		
	P	R	F1	P	R	F1
1	0.70	0.48	0.57	0.61	0.57	0.59
2	0.75	0.59	0.66	0.76	0.58	0.66
3	0.67	0.86	0.75	0.69	0.83	0.75
4	0.71	0.58	0.64	0.39	0.78	0.52
5	0.44	0.45	0.44	0.36	0.56	0.44

From the table, we observe that the recall values for less representative classes increase with oversampling, while precision decreases. However, this does not translate into improved classification performance, as indicated by the similar overall F1-scores for the model trained with dataset 3. As observed with the original data, increasing the balance of the dataset does not positively impact the classification task.

The confusion matrix for the test data obtained from the model trained with dataset 1 is shown in Figure 19.

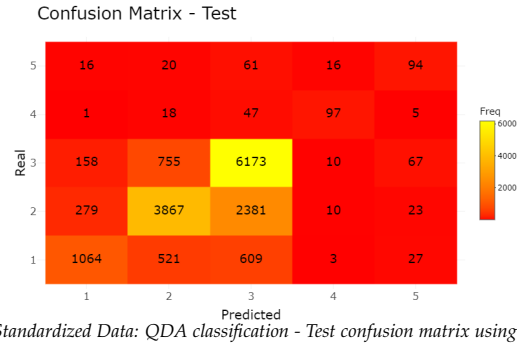
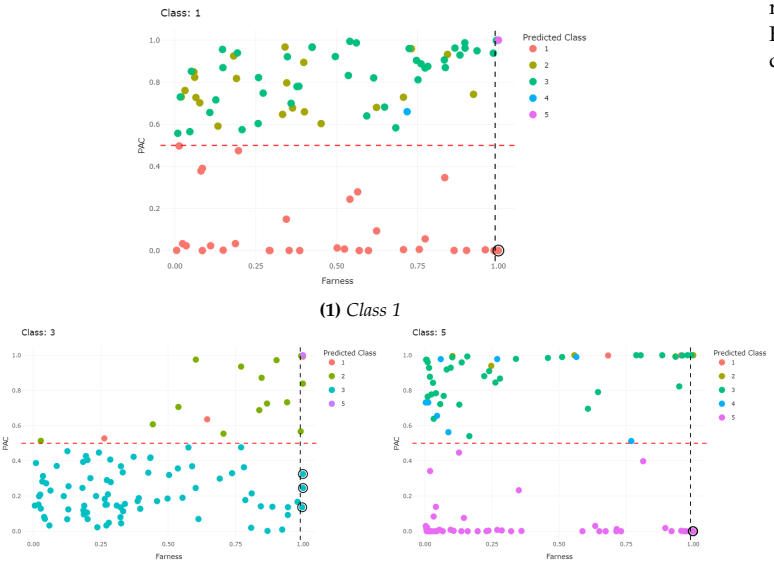


Figure 19: *Standardized Data: QDA classification - Test confusion matrix using a model trained with dataset 1*

Comparing the confusion matrices in Figures 17 and 19, we again observe that they are similar. From this, we infer that standardizing the dataset does not have a notable impact on the classification task using QDA. To facilitate further comparison between the classification results obtained with the original data and the standardized data, we present the class maps for classes 1, 3, and 5 in Figure 20. To make it possible to compare, a seed was defined so that the observations chosen from the validation set were the same.



(1) *Class 1*

(2) *Class 3* **(3)** *Class 5*

Figure 20: *Standardized Data: QDA- Class maps for a small validation sample*

As expected, the class maps display behavior similar to those obtained for the original data. However, a notable difference is observed in the plot for class 1. Since these graphics are generated using a small sample, definitive conclusions cannot be drawn. Nonetheless, we note that standardization appears to have led to a higher number of misclassifications for class 1 observations.

Discussion

LDA showed stable performance, achieving reasonable precision and recall, particularly for the dominant classes. The assumption of shared covariance matrices across all classes matched well with the dataset's observed covariance structures, which showed some similarities between classes for some features. Balancing improved recall for underrepresented classes, indicating that LDA effectively used the increased representation of these classes. However, precision decreased, suggesting overfitting to the new repeated samples.

QDA struggled due to its reliance on class-specific covariance matrices. This flexibility can be advantageous in cases of well-defined class separations, however, here it underperformed because minority class covariance matrices were poorly estimated due to insufficient samples. Balancing destabilized QDA even further, most likely due to exaggerated differences in covariance structures introduced by synthetic balancing.

3. k-Nearest Neighbors (kNN)

k-Nearest Neighbors (kNN) is a straightforward supervised classification algorithm. It classifies an unlabeled observation by identifying its k closest data points (its nearest neighbors) based on a chosen distance metric. The observation is then assigned to the most frequent class among these neighbors. While this approach can result in ties, introducing a weighting factor, such as $\frac{1}{\text{distance}^2}$, for the k nearest neighbors helps to mitigate this issue, in our case we do not introduce weighting.

Unlike many machine learning algorithms, kNN does not require a training or optimization process. It can handle non-linear problems effectively, as it can define non-linear decision boundaries. Its simplicity allows for easy implementation with basic computational tools.

However, the lack of a dedicated training step means that for every new observation, the algorithm must compute distances to all points in the training set. This results in a high computational cost that grows with the size of the training set. Additionally, selecting the optimal value for k is often a trial-and-error process, and a poorly chosen k can lead to suboptimal results. Another challenge lies in selecting an appropriate distance metric.

In our implementation, we have consistently used the Euclidean distance as the distance metric.

For all different datasets we ran a tuning algorithm that leveraged the cross validation technique and chose an optimal k according to the validation accuracy. We obtained similar results for all datasets and opted to chose $k = 17$. Here we show an example of the tuning results for dataset 1 of the original data:

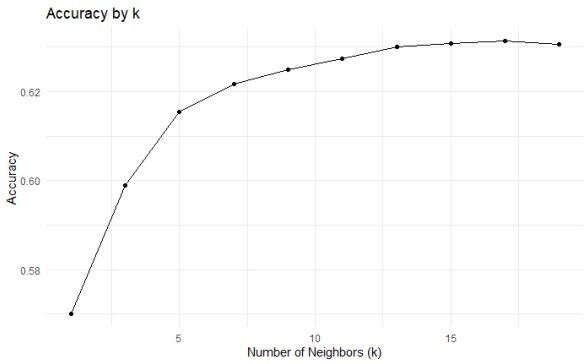


Figure 21: *Maximum accuracy obtained from cross-validation for each k*

We can clearly see from 21 that the optimal k is 17. **Original Data**

The metrics presented in Table 11 reveal varying performance of the k-Nearest Neighbors (kNN) classifier across the three datasets. The overall accuracy is highest for Dataset 1 (0.64), followed by Dataset 2 (0.62) and Dataset 3 (0.56). However, significant differences in performance metrics can be observed for individual classes.

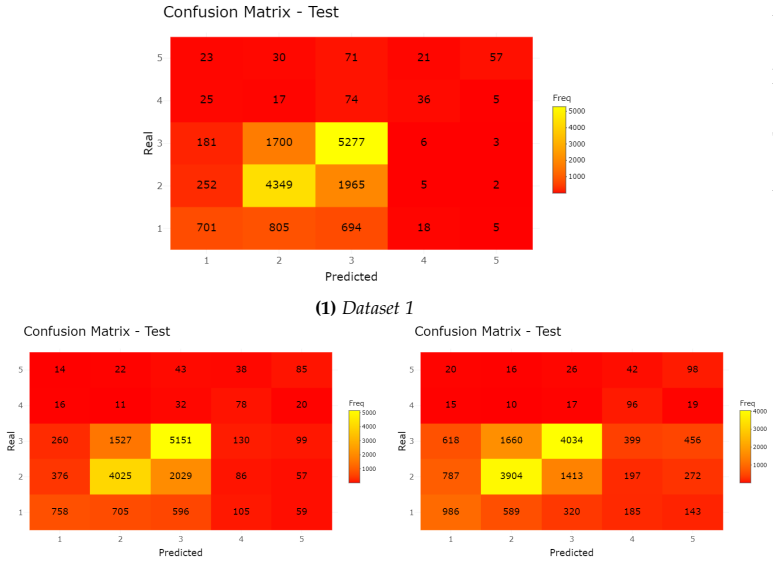
Table 11: *Metrics for Test Data with Overall Accuracies: 0.64, 0.62, and 0.56. And k = 17*

Class	Trained w/ Dataset 1			Trained w/ Dataset 2			Trained w/ Dataset 3		
	P	R	F1	P	R	F1	P	R	F1
1	0.59	0.32	0.41	0.53	0.34	0.42	0.41	0.44	0.42
2	0.63	0.66	0.65	0.64	0.61	0.63	0.63	0.60	0.61
3	0.65	0.74	0.44	0.66	0.72	0.69	0.70	0.56	0.62
4	0.42	0.23	0.23	0.18	0.50	0.26	0.1	0.61	0.18
5	0.80	0.28	0.41	0.27	0.42	0.33	0.10	0.49	0.17

Class 1 demonstrates relatively consistent precision across all datasets, but its recall is notably low (around 0.3–0.4). This indicates that the classifier struggles to correctly identify observations from Class 1, leading to moderate F1-scores. Class 2, on the other hand, exhibits balanced precision, recall, and F1-scores across all datasets, showcasing consistent classification performance. Class 3 achieves the highest recall values, particularly for Dataset 2 (0.72), suggesting that observations from this class are more distinguishable in the feature space. However, the F1-scores remain moderate, indicating some room for improvement. Class 4 suffers from the poorest performance, with F1-scores below 0.3 across all datasets. While Dataset 3 achieves a higher recall for this class (0.61), it comes at the expense of precision. Lastly, Class 5 shows high precision for Dataset 1 (0.80), but its recall is consistently low across all datasets, resulting in low F1-scores.

The confusion matrices in Figure 22 provide further insights into the classification performance. For Dataset 1, the classifier demonstrates strong diagonal dominance, particularly for Class 3, indicating a high number of correct predictions. However, significant misclassification is observed for Class 5. In Dataset 2, the predictions are more scattered, with reduced diagonal dominance, but some improvements are noted for Class 3. Dataset 3, which exhibits the poorest overall accuracy, shows substantial misclassification across all classes, with Class 5 being particularly challenging to classify correctly.

These results highlight the dependence of kNN's performance on the dataset used for training which noticebly needs improving.



(1) *Dataset 1*

(2) *Dataset 2* **(3)** *Dataset 3*

Figure 22: *Confusion Matrix of the test set for a KNN classificator*

Preprocessed Data

We ran the classification algorithm with the standardized datasets and got the exact same results, this is expected in this case, where we did not use any type of weighting making the classification invariant to standardization.

4. Logistic Regression

Logistic regression with more than two classes, known as multinomial logistic regression, is a statistical method used to predict which class an outcome belongs to when there are three or more possible classes. It works by modeling the relationships between the input features and the probabilities of each class. The method calculates the likelihood of an outcome belonging to each class using a function that ensures all probabilities add up to 1. The class with the highest probability is chosen as the prediction. To perform this classification task we utilized the *multinom* function from the `nnet` library[8].

Original Data

Initially, we performed multinomial logistic regression on Dataset 1. The precision, recall and F1-score metrics for each class for the train and test data are presented in Table 12. The overall accuracies for the train and test data were 0.71 and 0.72 respectively.

Table 12: *Metrics for Train and Test Data with Overall Accuracies: 0.71, and 0.72.*

Class	Train data			Test data		
	P	R	F1	P	R	F1
1	0.73	0.44	0.55	0.74	0.43	0.55
2	0.70	0.70	0.70	0.71	0.71	0.71
3	0.72	0.83	0.77	0.72	0.83	0.77
4	0.75	0.46	0.57	0.68	0.43	0.53
5	0.70	0.39	0.50	0.72	0.36	0.48

From the table, we can see a reasonably good performance for most classes, with with F1-scores ranging from 0.50 to 0.77 for train data and 0.48 to 0.71 for test data.

Class 3 stands out as having the highest precision, recall, and F1-score in both training and testing datasets, indicating that the model predicts this class more effectively, most likely due to its domination of the dataset.

However, performance for Class 5 is weaker, with relatively low recall and F1-scores, suggesting difficulty in correctly identifying instances of this class, most likely because they only account for roughly 1% of the dataset.

The overall accuracy for the training data (0.71) and the test data (0.72) are very close, indicating that the model is not overfitting. This consistency suggests that the model is generalizing well to unseen data and maintaining similar performance across both train and test data.

Our next step was to perform multinomial logistic regression on Dataset 2. The precision, recall and F1-score metrics for each class for the train and test data are presented in Table 13. The overall accuracies for the train and test data were 0.70 and 0.72 respectively.

Table 13: *Metrics for Train and Test Data with Overall Accuracies: 0.70, and 0.72.*

Class	Train data			Test data		
	P	R	F1	P	R	F1
1	0.73	0.49	0.58	0.72	0.48	0.57
2	0.68	0.67	0.67	0.73	0.68	0.70
3	0.69	0.83	0.75	0.72	0.83	0.77
4	0.79	0.69	0.74	0.53	0.66	0.59
5	0.76	0.53	0.62	0.59	0.60	0.60

Comparing the metrics from the previous table to this one, we observe that the overall accuracy has slightly decreased for the training data (from 0.71 to 0.70) but slightly improved for the test data (from 0.72 to 0.72). This indicates that the model's performance on the test data has become more consistent with its training performance, suggesting improved generalization.

Notably, the F1-scores for individual classes show less variation in this table, with most classes performing more uniformly. This is likely due to the more even distribution of classes in this dataset, which helps the model allocate resources more equitably across classes, improving its ability to generalize predictions.

Class 3 still achieves the highest F1-scores in both tables, however classes 4 and 5, which had lower performance in the earlier table, now show improved recall and F1-scores, particularly for the test data. This suggests that a more even class distribution has reduced the model's bias toward certain classes, allowing it to better capture the characteristics of the underrepresented ones.

The confusion matrix obtained from the test data from the model trained with Dataset 2 is displayed in Figure 23.

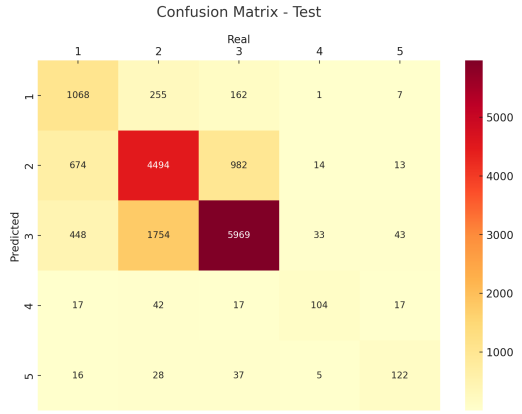


Figure 23: *Original Data: Logistic regression - Test confusion matrix using a model trained with Dataset 2*

From the confusion matrix, it is possible to see that class 3 presents the highest accuracy with most predictions correctly classified, while classes such 4 and 5, that have fewer samples, show more scattered misclassification across other classes.

Preprocessed Data

We repeated this classification task with standardized data. The metrics obtained by the model trained by Dataset 1 are displayed in Table 14. The overall accuracies for the train and test data were 0.72 and 0.71 respectively.

Table 14: *Metrics for Train and Test Data with Overall Accuracies: 0.72, and 0.71.*

Class	Train data			Test data		
	P	R	F1	P	R	F1
1	0.74	0.44	0.55	0.74	0.45	0.56
2	0.71	0.70	0.70	0.70	0.69	0.70
3	0.72	0.83	0.77	0.71	0.82	0.76
4	0.75	0.50	0.60	0.77	0.53	0.63
5	0.69	0.39	0.5	0.72	0.38	0.50

The metrics in the table, trained with standardized data, show no significant differences compared to the original data. The overall accuracies (0.72 for training and 0.71 for testing) are almost identical to the earlier results, and the performance trends for each class remain consistent, with Class 3 performing the best and Class 5 the weakest. This suggests that standardizing the data did not have a major impact on the model's performance, indicating that the original data distribution was likely already well-suited for the model.

The confusion matrix obtained from the test data from the model trained with Dataset 2 is displayed in Figure 24.

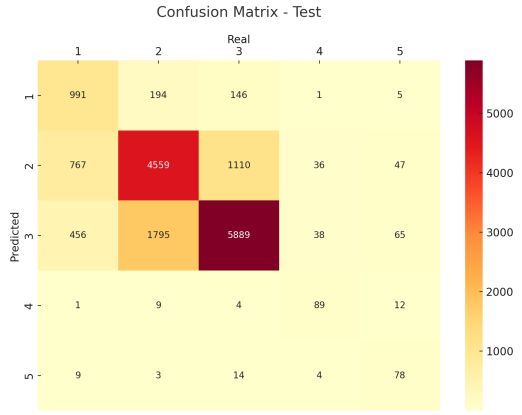


Figure 24: *Preprocessed Data: Logistic regression - Test confusion matrix using a model trained with Dataset 1*

Although this confusion matrix was generated with the standardized Dataset 1, it shows performance patterns similar to the one represented in figure 23. Class 3 still achieves the highest accuracy, while classes 4 and 5, show slight improvements in correct predictions, particularly for class 5.

Just like with Dataset 1, the results with Dataset 2 showed no significant differences in overall performance or class-specific metrics. Given the similarity in outcomes and the lack of any noteworthy distinctions, we decided not to include these results in the report.

IV. Conclusions

In conclusion, our analysis of the MERIS satellite radiance data aimed to classify CORINE land cover levels and address the challenges posed by the highly imbalanced dataset. Through the implementation of various preprocessing and classification techniques, we gained valuable insights into the problem and its complexities.

The dataset was extremely imbalanced, with one class constituting only 0.9% of the observations, requiring careful handling. Techniques like stratified splitting ensured the representation of all classes in train, validation, and test sets. Balancing methods, showed potential to improve results but also posed risks of overfitting.

Several models were evaluated, with results indicating that the majority class was consistently well predicted, but minority classes suffered from poor classification. This highlighted the inherent difficulty of accurately modeling such an imbalanced problem.

The radiance levels captured by the MERIS satellite proved to be informative, yet their discriminatory power varied across different land cover classes. This suggests potential for further feature engineering or integration of additional data sources to enhance the classification process.

We have then gattered some possible improvements to our work that can help achieve our main objective:

1. Integrating additional datasets. An example that might be helpful is using satellite images of our observations' coordinates to decompose into rgb values and have these values join our dataset as new features.

2. Use data from a bigger area. For example the whole europe to represent better minority classes and allow us to explore the other land cover classification levels.

3. Ensure the labeling was obtained at a time closer to the sattelite's data to avoid outliers and irregularities.

4. Use classifiers that are able to learn more complex data like convolutional neural networks or transformers.

To sum up this project serves as a stepping stone for a more refined and robust analysis and helped us consolidate our multivariate analysis knowledge and understanding.

References

[1] NASA EOSDIS LANCE. EN1_MDSI_MER_FRS_1P Product, 2022.

[2] European Space Agency (ESA). Envisat MERIS Full Resolution Full Swath Level 2 (MER_FRS_2P), 2022.

[3] Copernicus Land Monitoring Service. Corine land cover - copernicus land monitoring service.

[4] European Environment Agency (EEA). Portugal shapefile - eea reference grids.

[5] Majka Michal. Naive bayes classification, 2023.

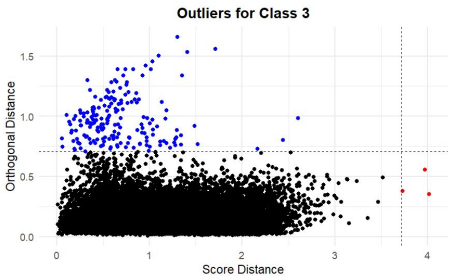
[6] Peter J. Rousseeuw Jakob Raymaekers and Mia Hubert. Class maps for visualizing classification results. *Technometrics*, 64(2):151–165, 2022.

[7] Kuhn and Max. Building predictive models in r using the caret package. *Journal of Statistical Software*, 28(5):1–26, 2008.

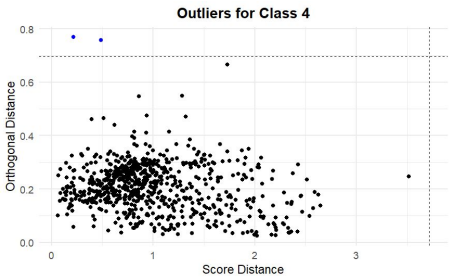
[8] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

Appendix

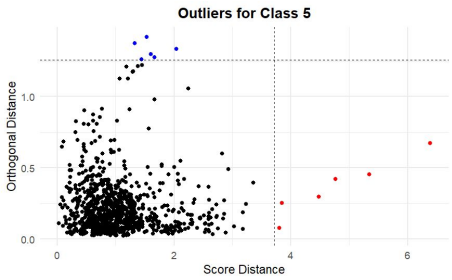
The following plots were the ones obtained for classes 3, 4 and 5 when performing the outlier analysis.



(1) *Outliers in class 3.*



(2) *Outliers in class 4.*



(3) *Outliers in class 5.*

Figure 25: *Outlier detection with PCA for classes 3, 4 and 5. The outliers flagged only due to an excessive orthogonal distance are represented in blue and the ones identified due to their excessive score distance are colored in red. The black points are the regular observations.*