ST443 Group 1 Project

Abstract

This report investigates the performance of several supervised learning models for a multi-class land-type classification task based on high-dimensional pixel data as task 1, and a feature selection task on high-dimensional neural activity recordings as task 2. We begin both tasks by pre-processing the dataset, including handling missing values, standardisation, and dimensionality reduction using Principal Component Analysis PCA. A range of classifiers is then trained and evaluated. Their generalisation performance is assessed using multiple evaluation metrics.

1 Task 1: multiclass classification

1.1 Methodology

The dataset, consisting of high-dimensional pixel data, is first cleaned by removing missing labels and extracting all spectral band features. We then split the data into an 80% training set and a 20% test set using stratified sampling to preserve class proportions. All models are implemented using `scikit-learn` pipelines, with standardisation to ensure consistent feature scaling. In addition to modelling on the raw predictors, PCA with 10 components is fitted on the training data and used to provide reduced-dimensional inputs for comparison.

For each classifier considered—*Discriminant Analysis*, *Logistic Classifier*, *Nearest Neighbour Classifier (k-NN)*, *Tree-type Models*, and *Support Vector Machine (SVM)*—performance is first estimated using 5-fold stratified cross-validation on the training set to further prevent over-fitting, computing *Accuracy*, *Balanced Accuracy*, *Macro-F1*, *Macro-AUC*, and *Confusion Matrices*. Each model is then retrained on the full 80% training set and evaluated once on the untouched 20% test set to obtain unbiased performance estimates and to select candidates for further analysis in the final binary classification task.

1.2 Visualisation and summary statistics

We detected the dimension of the dataframe to be 215,604* 223, with 218 band features. In addition, the data has no missing values or duplicates. However, with the legal range of [0,1] for reflectance rates in mind, we found around 0.84% of the raw data to be out of this range. Though the percentage is not large, we still capped all the invalid entries to be within [0,1] to ensure valid data modelling. Besides the invalid values, we also detected around 5.17% of outliers, with a threshold of 3 standard deviations away from the mean. The outlier amount is very small; the intuition behind their existence may be exceptionally high reflectance rates on certain surfaces, such as bright snow. Hence, we decided not to remove the outliers.

Now we move on to data visualisation. For a classification task, it is essential to check the distribution of each class and the correlation between them. From Figure 1, we can observe that the *alpine meadow* is the most frequent vegetation class, with around a quarter of the observations classified in this class. Also, *valley floor* and *alpine tundra* are quite prevalent, and about 58% of all the observations are classified into one of these vegetation types. Such an imbalance alerts us to be careful with the future fold splitting in Cross Validation, as a strictly even split may cause the disappearance of some classes within a fold.

From Figure 2, we can see that classes with clearly different spectral curves should be easier to separate, meaning most models are likely to classify them well. Where the lines overlap more, the classes are spectrally similar, so simple linear methods may struggle; nonlinear models may handle

Preprint. Under review.

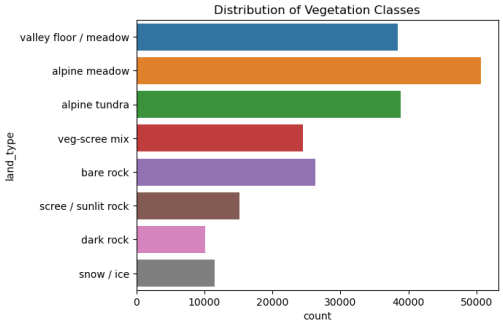

Figure 1: Class-wise distribution

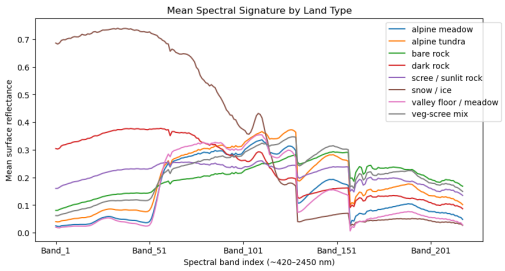

Figure 2: Mean Spectra

these cases better. The smooth changes across neighbouring bands also show that the bands are highly correlated, which supports using PCA or some form of regularisation later in the modelling.

But is the data really independent enough to some extent for us to conduct PCA? To gain more intuitive evidence on this, we plot the correlation heatmap between adjacent samples and all bands:

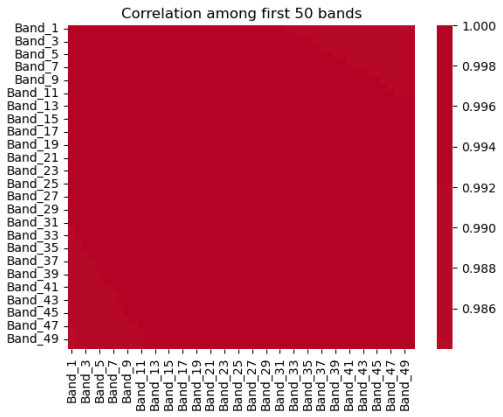

Figure 3: Adjacent correlation heatmap

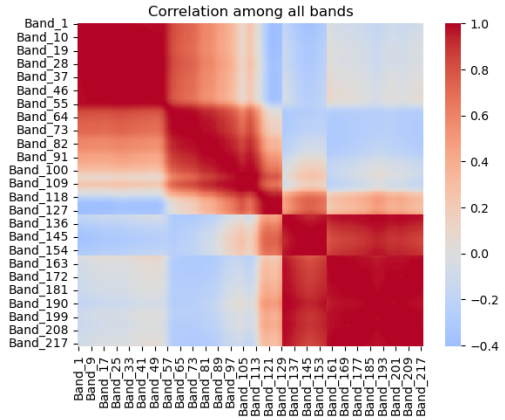

Figure 4: All bands correlation heatmap

From Figure 3, the heatmap reveals very strong correlation among adjacent spectral bands, indicating that many channels capture nearly identical information. In contrast, the blue regions in Figure 4 correspond to bands measuring different spectral regions, showing lower or even slightly negative correlation. This block-like pattern confirms substantial multicollinearity in the raw features and motivates the use of PCA, which compresses these redundant bands into a smaller set of orthogonal components while preserving most of the spectral variation.

Before model fitting, we apply PCA and clustering. PCA reduces the high-dimensional, highly correlated feature space to a smaller set of informative components. Clustering on these leading principal components is more interpretable than clustering on all raw bands, and it provides a clearer view of potential group structure in the data. Together, PCA and clustering help assess dimensionality reduction and visualise underlying patterns prior to training the classification models.

We first performed a test PCA of 4 components, which explained 99.87% of the data variance. While proceeding to the K-Means clustering with the first 2 components does not aid us too much in visualising the features (Figure 5), clusters seem arbitrarily separated. As such, we will only use PCA for dimensionality reduction and model fitting.

1.3 Model development

In this part of the investigation, we will try to predict the Vegetation type based on the reflectance values of the pixels. This is a multi-class classification problem, and we will use several types of

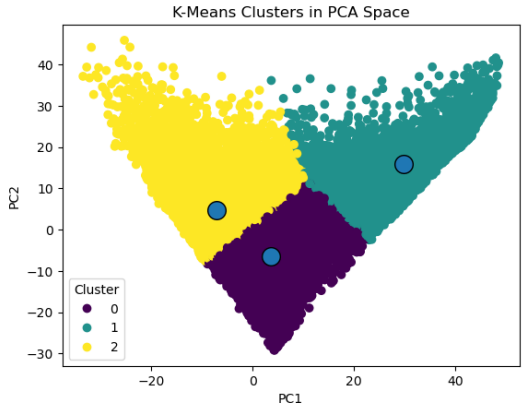

Figure 5: K-Means clustering for the 2 testing principal components

classifiers for this task. To encode the variables from categorical to numerical, we will use Scikit Learn's Label Encoder. The methodology will be the following: we will fit the models and report the cross-validation errors on the training set to adjust the hyperparameters. After finding the best model, we will fit it to the whole training data and report its performance on the test set. The test set will be used only to report the performance.

1.3.1 Inputs and evaluation framework

We use all available spectral band measurements as the feature matrix *X*, and the land-type label for each pixel as the target variable *y*. Before model fitting, we standardise all features to ensure comparability across bands, and we evaluate models both on the raw feature space and on a reduced-dimensional space obtained via PCA with 10 components.

To assess model performance reliably, we employ 5-fold stratified cross-validation on the training set. Stratification ensures that each fold preserves the class distribution, which is important given the imbalance between land-type classes observed earlier. For each model, we compute multiple performance metrics: *Accuracy*, *Balanced Accuracy*, *Macro-F1 Score*, and *Macro-AUC*. The term *Macro* means that we are taking the average over the number of classes, sensitive to class imbalance. While *ovr* means that we treat one class as positive, the rest as negative for all the classes

Additionally, we use confusion matrices to visualise class-wise errors. After cross-validation, each model is retrained on the 80% training split and evaluated once on the untouched 20% test set to obtain unbiased final performance estimates.

1.3.2 Linear and Quadratic Discriminant Analysis (LDA & QDA)

LDA assumes that each class follows a multivariate Gaussian distribution with its own mean vector but a shared covariance matrix. It leads to linear decision boundaries. The discriminant function is:

$$\delta_k(x) = x^\top \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + \log \pi_k,$$

QDA relaxes the *LDA* assumption by allowing each class to have its own covariance matrix. This yields more flexible quadratic decision boundaries. The discriminant function is:

$$\delta_k(x) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x - \mu_k)^\top \Sigma_k^{-1} (x - \mu_k) + \log \pi_k,$$

Both classifiers predict based on the rule of:

$$\psi_(x) = \arg\max_k \ \delta_k(x).$$

From Table 1, QDA combined with PCA10 yields the strongest performance across all metrics, likely because PCA stabilises QDA's class-wise covariance estimates in this high-dimensional setting. In contrast, LDA slightly deteriorates after PCA. However, some more complex models can be tested.

Table 1: *Discrimiant Analysis* performances

Model	Accuracy	Bal. Acc	AUC (macro-ovr)	F1 (macro)
LDA	0.861483	0.860111	0.990794	0.862057
LDA (PCA10)	0.840588	0.833129	0.988525	0.838366
QDA	0.839197	0.842769	0.984779	0.839298
QDA (PCA10)	0.916653	0.916828	0.994898	0.915836

1.3.3 Logistic Classifier

As one of the most fundamental models in machine learning, the *Logistic Classifier* estimates the probability of each class given the input features. In the binary case, it models the log-odds of class membership as a linear function of the predictors. Now for multi-class problems, logistic regression uses the softmax function to model probabilities across all classes:

$$P(y = k \mid x) = \frac{\exp(\beta_k^\top x)}{\sum_{j=1}^{K} \exp(\beta_j^\top x)}.$$

With 1000 iterations, we obtain the following summary table:

Table 2: *Logistic Classifier* performance

Model	Accuracy	Bal. Acc	AUC	F1
Logistic	0.990399	0.987690	0.999926	0.987823
Logistic_PCA	0.987964	0.985349	0.999899	0.985476

We can see that the classifier has very high performance on this dataset, despite its large size. One possible reason is due to the fact that in image recognition, the signal-to-noise ratio is very high, as compared to, for example, financial time series, where the ratio is relatively low.

1.3.4 Nearest Neighbour Classifier (k-NN)

1.3.5 Tree-type models

One of the most widely used tree-based models is the *Gradient Boosting Decision Tree (GBDT)*. *GBDT* builds an additive ensemble of shallow trees, where each new tree is fitted to correct the residual errors of the previous ones. By iteratively minimising a differentiable loss through gradient descent in function space, it captures subtle nonlinearities and class-specific patterns. Although more sensitive to hyperparameters, *GBDT* often delivers strong predictive performance on high-dimensional structured data:

$$F_m(x) = F_{m-1}(x) + \nu\, h_m(x), \quad m = 1, \ldots, M,$$

$$F_0(x) = \arg\min_c \sum_i \ell(y_i, c),$$

An alternative is the *Random Forest* model, leveraging multiple simple decision trees to reduce the variance when averaging the trees. With these two models implemented, we check their test performances:

Table 3: *Tree-type* models performances

Model	Accuracy	Bal. Acc	AUC (macro-ovr)	F1 (macro)
LDA	0.946	0.971	0.952	0.998
LDA (PCA10)	0.952	0.968	0.943	0.998
QDA	0.984	0.990	0.982	1.000
QDA (PCA10)	0.975	0.984	0.972	1.000

1.3.6 Support Vector Machine (SVM)