

---

# ST443 Group 1 Project

---

## Abstract

This report investigates the performance of machine learning models across two high-dimensional classification tasks. Task 1 addresses multi-class land-type classification using pixel-level image features, while Task 2 focuses on feature selection for binary decision classification based on neural activity recordings. Both tasks begin with exploratory data analysis, covering data structure, distributions, and feature characteristics. A range of models is then trained and evaluated using multiple performance metrics. Task 1 also includes a brief real-world application, whereas Task 2 emphasises identifying informative features in a highly sparse, high-dimensional dataset.

## 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 (plus a further subset within test data for specific models) 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 complete 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 identified the dataset dimension as  $215,604 \times 223$ , including 218 band features. There are no missing values or duplicates. Since reflectance values should lie within  $[0,1]$ , we checked for invalid entries and found that about 0.84% of values were outside this range. Although the proportion is small, we clipped these values to the valid interval to ensure consistency. We also detected roughly 5.17% of outliers using a 3-sigma threshold. Given the small proportion and the fact that extreme reflectance can naturally occur (e.g., bright snow), we chose not to remove them.

For visual exploration, we first examined class distributions and inter-class relationships. From Figure 1, *alpine meadow* is clearly the most frequent class, accounting for about a quarter of all observations. *Valley floor* and *alpine tundra* are also common, with these three classes together forming around 58% of the dataset. This imbalance highlights the need for careful stratification in cross-validation, as uniform data splitting could otherwise cause some minority classes to disappear in certain folds.

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

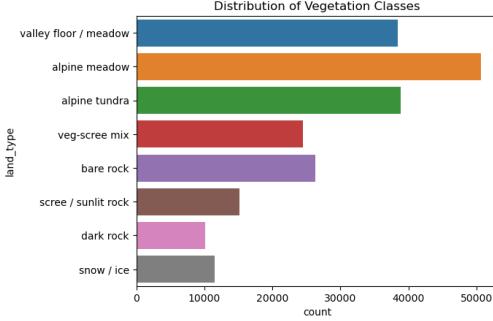


Figure 1: Class-wise distribution

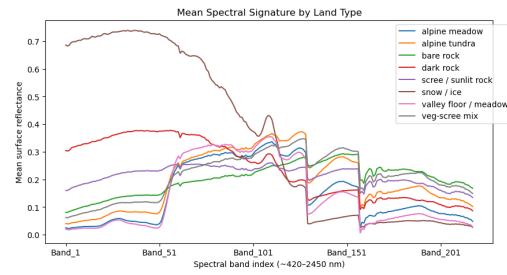


Figure 2: Mean Spectra

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 sufficiently independent for us to conduct PCA? To gain more intuitive evidence on this, we plot the correlation heatmap between adjacent samples and all bands:

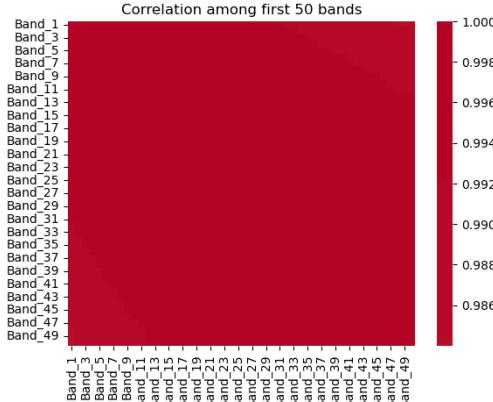


Figure 3: Adjacent correlation heatmap

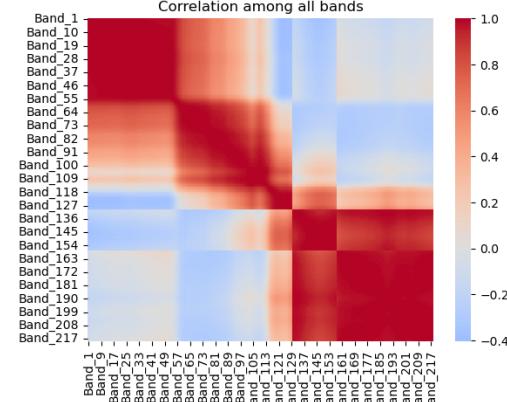


Figure 4: All bands correlation heatmap

From Figure 3, the heatmap shows strong correlations among adjacent spectral bands, meaning many channels capture almost the same information. In contrast, the blue regions in Figure 4 represent bands from different spectral ranges, showing lower or even negative correlation. This block-like pattern highlights substantial multicollinearity among the raw features. It motivates using PCA to compress redundant bands into a smaller set of orthogonal components while retaining most of the spectral variation.

Before fitting models, we apply PCA and clustering as exploratory steps. PCA reduces the high-dimensional, highly correlated feature space into a smaller set of informative components. Clustering on these leading components is more interpretable than clustering on all raw bands and helps reveal any broad structure in the data. Together, PCA and clustering provide insight into dimensionality and underlying patterns before proceeding to classification.

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 much in visualising the features (Figure 5), the clusters appear 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 vegetation type based on pixel reflectance values. This is a multi-class classification problem, and we will use several types of classifiers. To encode categorical variables as numerical, we will use Scikit-Learn's LabelEncoder. The methodology will

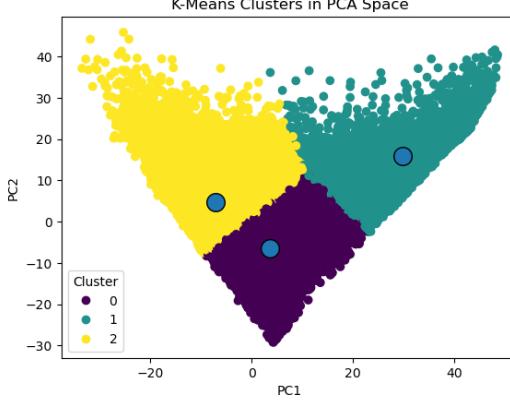


Figure 5: K-Means clustering for the 2 testing principal components

be as follows: we will fit the models and report cross-validation errors on the training set to tune the hyperparameters. After selecting the best model, we will fit it to the entire training dataset 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 reliably assess model performance, we use 5-fold stratified cross-validation on the training set. Stratification ensures that each fold preserves the class distribution, which is vital 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 we take the average across classes, which is sensitive to class imbalance. While *ovr* means that we treat one class as positive, the rest as negative for all the classes. 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:

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

### 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:

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

In addition, to ensure parameter convergence, we trained the classifier for 2000 iterations to achieve faster, more stable convergence.

#### 1.3.4 Tree-type models

Gradient Boosting Decision Trees (GBDT) build an additive ensemble of shallow trees, where each new tree is trained to correct the residual errors of the previous ones. By iteratively minimising a differentiable loss via gradient descent in function space, GBDT can capture nonlinear relationships and class-specific patterns. Although more sensitive to hyperparameters, it often performs strongly on high-dimensional structured data:

$$\begin{aligned} F_m(x) &= F_{m-1}(x) + \nu h_m(x), \quad m = 1, \dots, M, \\ F_0(x) &= \arg \min_c \sum_i \ell(y_i, c), \end{aligned}$$

An alternative is the Random Forest model, which leverages multiple simple decision trees to reduce variance when averaging the trees.

#### 1.3.5 Nearest Neighbour Classifier (k-NN)

k-Nearest Neighbours is a simple, non-parametric classification algorithm that predicts the label of a new sample by examining the labels of its  $k$  closest training samples under a chosen distance metric (usually Euclidean distance). In high-dimensional settings like task 1, distances between points become less informative, thereby reducing k-NN's discriminative power:

$$\psi_{\text{kNN}}(x) := \arg \max_k \sum_{i=1}^K \mathbf{1}\{Y^{(i)} = k\},$$

However, due to the high dimensionality of the data, k-NN runs out of memory. To address this while preserving fair comparability across models, we train k-NN on a randomly selected subset of the training data. Crucially, the final evaluation is still performed on the same 20% untouched test set used by all other models, ensuring consistency in evaluation and avoiding data leakage.

#### 1.3.6 Support Vector Machine (SVM)

*Support Vector Machines* classify data by finding the optimal separating hyperplane that maximises the margin between classes. Only a subset of the training points—the support vectors—determine this boundary, making SVM robust and effective in high-dimensional feature spaces:

$$\psi_{\text{SVM}}(x) = \text{sign}\left(\beta_0^* + \sum_{i \in \mathcal{S}} \alpha_i^* y_i k(x, x_i)\right),$$

To ensure comparability across models, the SVM uses the same training–testing configuration as k-NN mentioned before.

### 1.4 Model performance

As mentioned in the methodology, we split the data into two parts: 80% for training (and a further subset for SVM and k-NN) and 20% for testing, and used 5-fold cross-validation to prevent overfitting. Now that we have fitted all the desired models, it's time to check each of their performances using various performance metrics, based on the same 20% untouched data:

As shown in Table 1a, most raw models perform very well across all four metrics, though with some noticeable differences. Logistic Regression, SVM, and RF consistently achieve high *Accuracy*, *Balanced Accuracy*, *AUC*, and *F1 score*, suggesting they handle the high-dimensional structure of the data effectively. In contrast, LDA and QDA perform weaker across most metrics, which may be due to their stronger assumptions on class covariance that do not hold for this dataset. GBDT and k-NN fall somewhere in between, performing reasonably but not reaching the top-performing models.

Model	Acc	BalAcc	AUC	F1	Model	Acc	BalAcc	AUC	F1
Logistic	0.9922	0.9911	0.99995	0.99116	Logistic_PCA10	0.9916	0.9923	0.99994	0.99248
LDA	0.8613	0.8598	0.99080	0.86179	LDA_PCA10	0.8406	0.8331	0.98853	0.83837
QDA	0.9217	0.9183	0.99558	0.91886	QDA_PCA10	0.9311	0.9284	0.99637	0.92789
RF	0.9849	0.9832	0.99987	0.98353	RF_PCA10	0.9764	0.9729	0.99966	0.97388
GBDT	0.9475	0.9521	0.99842	0.95305	GBDT_PCA10	0.9534	0.9451	0.99828	0.94638
KNN	0.9613	0.9595	0.99869	0.96081	KNN_PCA10	0.9624	0.9605	0.99869	0.96172
SVM	0.9904	0.9895	0.99995	0.98986	SVM_PCA10	0.9892	0.9888	0.99993	0.98918

(a) Raw Models

(b) PCA-10 Models

Table 1: Performance comparison of models with and without PCA.

After applying PCA with 10 components (Table 1b), the overall ranking of models remains similar. Logistic Regression and SVM stay stable and strong, while RF sees a slight drop, which is expected since tree-based models do not always benefit from linear dimension reduction. LDA and QDA continue to struggle even after PCA, while k-NN performance remains broadly consistent.

### 1.5 Application: glacier-ice detection

For this application, we aim to treat *glacier-ice* as the only significant (positive) land type, while setting all other land types as negative, turning the previous multi-class classification into a binary classification task.

Among the four performance metrics, we choose to focus on the *F1 score*, as it integrates precision and recall and provides a valuable basis for comparison with previous testing (AUCs remain high across all models, with little practical value). Then, based on the model performances in the prior section, the three models with the highest *F1 scores* are: PCA Logistic Regression (0.99248), raw SVM (0.98904), and raw RF (0.98353).

## 2 Task2: feature selection

### 2.1 Methodology

### 2.2 Exploratory Data Analysis (EDA)

### 2.3 Model development