Project Part 1: Comprehensive Analysis of Hyperspectral Data

Team Name: Spectral Data Soft Sensor (B)

Team Member: Bahadir Bagci, Hassan Majeed, Zoheb Rahman

Introduction

For the first project, our team has chosen the Hyperspectral Soft Sensor project B and we did an in-depth exploration of the given dataset. This dataset includes a detailed collection of spectral data covering wavelengths from 450 to 2500 nm, pre-processed to remove water absorption bands (1351-1430, 1801-2023, 2451-2501 nm) and smoothed using the Savitzky-Golay filter. It also contains 20 leaf and canopy traits, such as chlorophyll content and equivalent water thickness, gathered from 42 global studies, totalling 12,180 observations across 1,741 variables (approximately 1,721 spectral bands and 20 traits).

Aims & Objectives

Our work in this report focuses on

1. Setting up effective communication channel,
2. Importing and validating the dataset,
3. Identifying key challenges,
4. Creating insightful visualizations,
5. Performing a Principal Component Analysis (PCA) on the spectral predictors,
6. Developing a pretreatment plan.

Methodology

For this analysis, we have selected Python as our primary programming language due to its robust ecosystem of data science libraries, including pandas for data manipulation, numpy for numerical computations, and matplotlib/seaborn for visualizations, alongside scikit-learn for PCA implementation.

Result

Communication and Code Sharing:

To ensure seamless collaboration, we have set up a GitHub repository as our primary platform for code management. We also use Microsoft Teams for real-time communication, document sharing, and scheduling weekly progress meetings, ensuring a structured and efficient workflow among team members.

Data Import:

The dataset was successfully imported into the Python environment using the pandas library with the command df = pd.read_csv("data/data_part_1.csv", index_col=0), creating a structured DataFrame with 12,180 rows representing individual samples and 1,741 columns comprising the variables. The first column serves as a unique sample identifier, while the remaining columns include spectral bands (serving as predictors) and trait data (reserved for response variables). We conducted initial validation using df.shape and df.head() to confirm the data structure and integrity.

Data Challenges:

Our preliminary analysis revealed several challenges within the dataset. Notably, there is a significant presence of missing values in the trait data, with variables such as Anthocyanin showing approximately 91% gaps (1,044 non-null entries out of 12,180) and Chlorophyll with 2,114 non-null entries. The spectral data, while largely complete, exhibits high collinearity among the 1,721 bands, with correlations often exceeding 0.95, suggesting potential redundancy. The traits are well-defined with clear units (e.g., Chlorophyll in µg/cm^2), and the spectral bands represent reflectance values across the specified wavelength range. As a static dataset collected from diverse ecosystems, no temporal synchronization issues are present, though addressing missing data and multicollinearity will be critical.

Visualization and Comments:

The dataset comprises 12,180 observations sourced from a variety of global ecosystems, including forests, croplands, and tundra regions. The spectral data exhibits low variance (typically 0.01-0.05 reflectance units), while traits such as Chlorophyll display a broader range (mean 28.37 µg/cm^2, range 0.45-29.50 µg/cm^2). Our visualization efforts have highlighted the smoothed nature of the spectral data, which is conducive to predictive modelling, though the high incidence of missing trait values poses a challenge. These visual tools provide valuable insights into data distribution and relationships.

Exploratory Data Analysis with PCA:

We performed a Principal Component Analysis (PCA) on the spectral band data to reduce dimensionality and identify underlying patterns. The process involved standardizing the data and computing the first 10 principal components. Figure 1 shows the correlation matrix of wavelengths, which highlights strong multicollinearity across large regions of the wavelengths indicating that PCA is an appropriate method for this dataset. From Figure 2, we can observe that 95% of variance is captured by the first four components (PC4), meaning that only four components are sufficient to represent the majority of the data.

The biplots in Figure 3 visualises the distribution of observations and their loading along PC1 and PC2. The **1746nm** wavelength can be seen to be highly correlated with PC1 as it has +ve sign, and is almost horizontal, while **869nm** wavelength shows high correlation with PC2 as it is almost vertical. The bar plot in Figure 4 further illustrates the loadings w.r.t PC1 and PC2. In particular, the wavelength region 800-1300 nm has strong correlation with both PC1 and PC2 making it a strong candidate for selection in downstream analysis and representation.

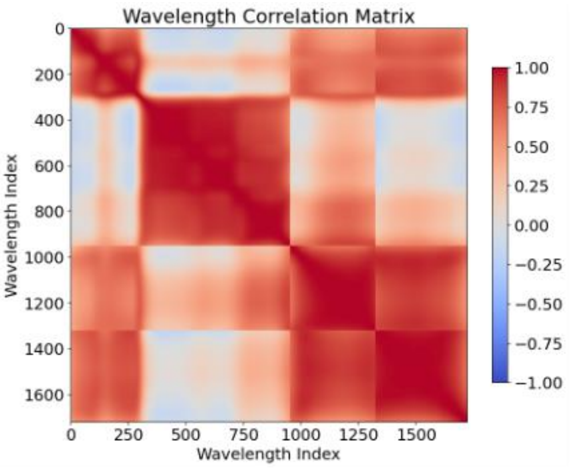

Figure 1: Correlation Plot of different wavelengths

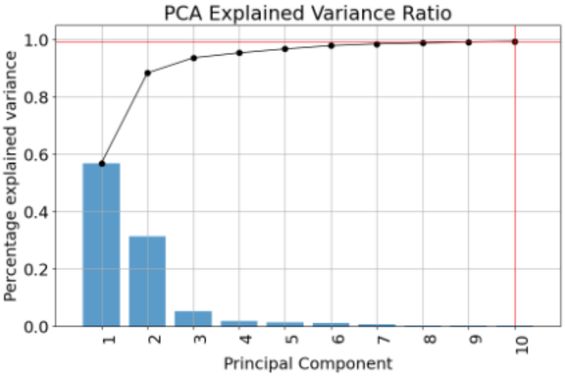

Figure 2: Scree Plot Showing Number of Components vs Variance

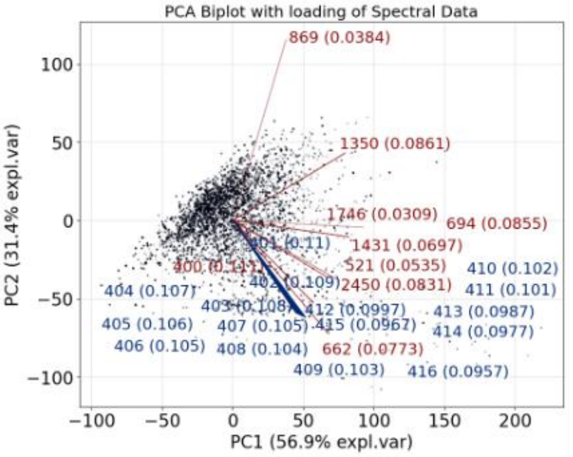

Figure 3: PCA Biplot including loading

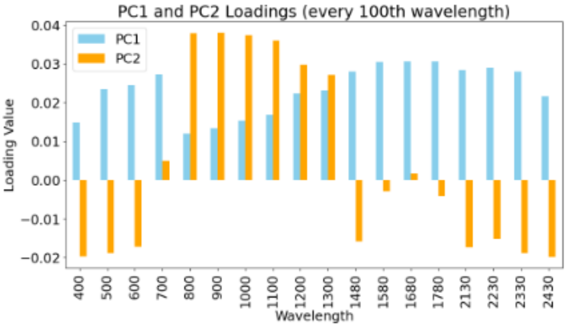

Figure 4: Loadings w.r.t PC1 and PC2

Pretreatment Plan:

To address the identified challenges, we have developed a comprehensive pretreatment strategy. This includes imputing missing trait values using multivariate regression techniques, applying standardization to the spectral data for consistency, detecting and mitigating outliers, preserving the existing Savitzky-Golay smoothing while monitoring for additional noise, and selecting a subset of the most informative spectral bands based on PCA results. Given the static nature of the data, no temporal adjustments are required. This plan will be implemented and refined in the subsequent phase of the project.

Pretreatment steps and plan:

- Inspect data for missing values; treat them by either removing or interpolating
- Standardise all predictor variables to remove scale effects
- Remove irrelvant or constant variables (if there are any extra variables)
- Smooth out/filter data if there is noise present
- Outlier handling by clipping