University of Novi Sad
Faculty of Sciences
Department of Mathematics and Informatics

Student:
Milica Panić

Explainable Artificial Intelligence - Crop Classification

Master Thesis

Mentor:
Prof. Dušan Jakovetić

Novi Sad, January 2024

I would like to extend my heartfelt gratitude:

To my mentor, Dušan Jakovetić, whose guidance has been a fundamental part of my journey through academia and professional development. The knowledge I gained under his tutelage has been invaluable and has significantly shaped my understanding and skills in the field of Data Science.

To Sanja Brdar and Oskar Marko, for guidance and for instilling in me a passion for this field by demonstrating its vast applicability, from technology to agriculture.

I am also deeply grateful to all other professors, as this master's program has been a wonderful experience that has enriched my academic and personal growth.

Additionally, I owe a special thanks to my family and friends, whose unwavering support and encouragement have been a cornerstone of my strength and perseverance, guiding me through both challenges and triumphs with love and belief.

A shout out to my colleagues, particularly Jovana Stanimirović, Dragana Šorak, Tatjana Naumov, and Katarina Petranović, who were not only partners in numerous projects but also friends outside the academic world.

Lastly, to Vladan, whose constant support and encouragement have not only eased my academic endeavor but also made it a more delightful and enriching experience. He has truly been the wind beneath my wings throughout this journey.

Contents

List of Figures and Tables

1. Introduction

Agriculture is the foundation of human society, providing us food, supporting our economies, and bringing different cultures together. Since how well crops grow is really tied up with weather and soil health, it is important that we step up in monitoring our agricultural lands, especially with all the environmental changes happening.

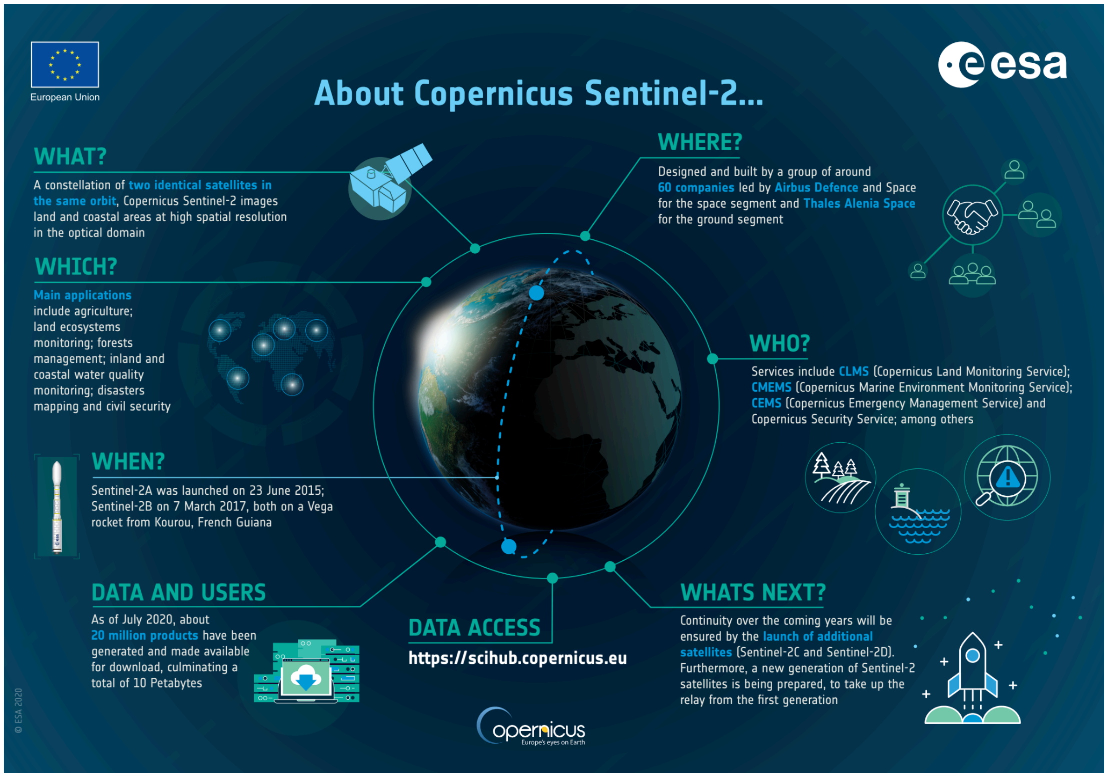
Copernicus is the Earth Observation component of the European Union's Space Programme, which monitors our planet and its environment for the ultimate benefit of the citizens of Europe [1]. This initiative, managed by the European Commission, collaborates with Member States, the European Space Agency (ESA), the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT), the European Centre for Medium-Range Weather Forecasts (ECMWF), EU Agencies, and Mercator Océan.

By using extensive global data from satellites and diverse measurement systems, including ground-based, airborne, and seaborne technologies, Copernicus offers invaluable information services. These services, which are both free and openly accessible, assist service providers, public authorities, and international organizations in improving the quality of life for European citizens and beyond.

In the field of technological advancements that define the 21st century, satellite Earth observation stands out as a powerful tool, casting light on our planet's most pressing challenges.

The European Union's important step in Earth observation, the Copernicus Sentinel-2, plays a significant role in this context. A constellation of two identical satellites orbiting in harmony, Sentinel-2A and Sentinel-2B, Sentinel-2 captures high spatial resolution images of terrestrial and coastal terrains. With applications spanning from agricultural monitoring to disaster mapping and civil security, its utility is incredibly useful.

Sentinel-2A was launched from Europe's Spaceport in Kourou (French Guiana) on 22nd June 2015 on top of a Vega launcher and Sentinel-2B followed on 7th March 2017 [3].



About Copernicus Sentinel-2...

WHAT?
A constellation of two identical satellites in the same orbit, Copernicus Sentinel-2 images land and coastal areas at high spatial resolution in the optical domain

WHERE?
Designed and built by a group of around 60 companies led by Airbus Defence and Space for the space segment and Thales Alenia Space for the ground segment

WHICH?
Main applications include agriculture; land ecosystems monitoring; forests management; inland and coastal water quality monitoring; disasters mapping and civil security

WHO?
Services include CLMS (Copernicus Land Monitoring Service); CMEMS (Copernicus Marine Environment Monitoring Service); CEMS (Copernicus Emergency Management Service) and Copernicus Security Service; among others

WHEN?
Sentinel-2A was launched on 23 June 2015; Sentinel-2B on 7 March 2017, both on a Vega rocket from Kourou, French Guiana

DATA AND USERS
As of July 2020, about 20 million products have been generated and made available for download, culminating a total of 10 Petabytes

DATA ACCESS
https://scihub.copernicus.eu

WHATS NEXT?
Continuity over the coming years will be ensured by the launch of additional satellites (Sentinel-2C and Sentinel-2D). Furthermore, a new generation of Sentinel-2 satellites is being prepared, to take up the relay from the first generation

Figure 1: Sentinel-2 infographics, source [2]

The Copernicus project's rich datasets, obtained from the Sentinel-2 satellites, serve as a good base for applying machine learning (ML) and deep learning (DL) algorithms for crop classification tasks [4]. Yet, the "black box" nature of many such models, especially deep neural networks, has called for advancements in Explainable Artificial Intelligence (XAI) to make the outcomes understandable and actionable to end-users, including farmers, agronomists, etc. [5].

The field of XAI serves for the decision-making processes of AI models. In the context of crop classification, where precision and reliability are paramount, XAI offers tools and techniques to interpret complex models, providing transparency and trust in their predictions [6].

In this context, the utilization of different ML models such as Random Forests, Naive Bayes, and K-Nearest Neighbors, alongside DL architectures like Convolutional Neural Networks (CNNs), has been explored.

SHAP (SHapley Additive exPlanations) is an advanced XAI framework that assigns each feature an importance value for a particular prediction.

The main goal of this thesis is to implement XAI techniques for accurate crop classification using Sentinel-2 satellite imagery. By implementing ML, DL algorithms and the SHAP library, this research aims to improve predictive accuracy while providing transparency in the model's decision-making processes. This work will demonstrate the practical application of XAI in agriculture, ensuring that the predictions are not only accurate but also interpretable and understandable.

2. Related work

The field of crop classification using satellite imagery has been an active area of research, given the significance of agriculture and the continuous advancements in both satellite technology and artificial intelligence. The merger of XAI with this domain further augments the understanding of model predictions, providing clarity and actionable insights.

The Naive Bayes classifier has been a popular choice in machine learning for its simplicity and effectiveness in probabilistic predictions. Presented in [7], they employed the Naive Bayes algorithm for crop classification. By assuming independence between these features, they were able to model the likelihood of each crop type and apply Bayes' theorem to estimate the posterior probabilities. The results showed its potential as a fast and efficient tool for initial crop type segmentation, despite some limitations in capturing inter-dependencies between features.

The k-Nearest Neighbors (KNN) algorithm is another influential method in the field of supervised learning, used for its simplicity and efficacy in classification tasks. In paper [8], KNN was utilized to classify crops. The study found that the optimal 'k' varied with the crop type and complexity. The paper also emphasized the importance of feature selection and dimensionality reduction in improving KNN's classification accuracy.

Study, published in 2022 [9], demonstrates the Random Forest (RF) algorithm's robustness in dealing with the complexities of satellite image data for crop classification. The RF model was selected for its ensemble approach, which integrates the decisions of numerous decision trees, each trained on a random subset of the data. This method significantly reduces the risk of overfitting, a common challenge in high-dimensional data like those of satellite images. The nature of RF, with multiple decision trees voting for the final output, enhances its reliability and generalization capabilities when applied to satellite imagery for crop classification.

While model accuracy is important, understanding the decision-making process of these models, especially in critical domains like agriculture, is equally vital. Introduced by [10], SHAP values provide consistent and locally accurate feature attributions, making them ideal for understanding complex models in different domains.

In the context of crop classification using satellite imagery, SHAP offers clarity on which features (or pixels) are most influential in determining a particular crop type. The use of explainable algorithms like SHAP is important to build more trust and provide transparency toward the use of ML models, thereby embracing data-driven models for its use in also, for example, drought early warning systems [11].

The convergence of machine learning, satellite technology, and XAI tools like SHAP promises a future where agriculture benefits from not just accurate predictions but also deep, interpretable insights.

While traditional machine learning models have been popular in crop classification, deep learning algorithms, especially CNNs, have revolutionized this domain. CNNs, designed for image data, exploit spatial hierarchies and automate feature extraction, thus making them exceptionally apt for high-dimensional satellite imagery.

A study by [12] showed application of CNNs on satellite images, highlighting their ability to detect and learn from crop patterns, often missed by traditional models.

3. Materials and Methods

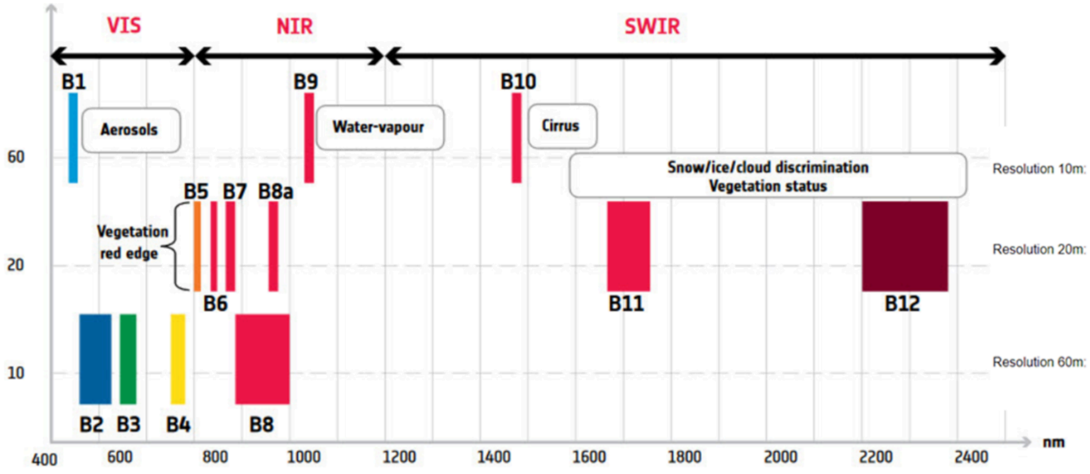
3.1. Data

As mentioned in the introduction, the Sentinel-2 mission is providing multispectral imagery at a high spatial resolution. This mission offers an array of spectral bands that span from the visible (VIS) to the near-infrared (NIR) and shortwave infrared (SWIR) regions of the electromagnetic spectrum. As illustrated in the provided data, Sentinel-2's imaging capabilities are organized into 13 distinct spectral bands, each targeting specific aspects of Earth's surface and atmosphere.

The spatial resolution of these bands varies, with the highest resolution being 10 meters per pixel for bands capturing visible and NIR wavelengths (B2 - Blue, B3 - Green, B4 - Red, and B8 - NIR). This high-resolution data is instrumental in applications requiring detailed observation of the land surface, such as agricultural monitoring and urban planning.

At a resolution of 20 meters, the red edge bands (B5 - Red Edge 1, B6 - Red Edge 2, B7 - Red Edge 3, B8a - Red Edge 4) and two SWIR bands (B11 - SWIR 1 and B12 - SWIR 2) provide critical information for assessing vegetation state and soil properties.

The remaining bands, B1 (Aerosols), B9 (Water vapor), and B10 (Cirrus), have a coarser spatial resolution of 60 meters. They are mainly used for atmospheric correction and cloud screening.



↑ Spatial resolution versus wavelength: Sentinel-2's span of 13 spectral bands, from the visible and the near-infrared to the shortwave infrared at different spatial resolutions ranging from 10 to 60 m on the ground, takes land monitoring to an unprecedented level

Figure 2: Spectral bands, source [13]

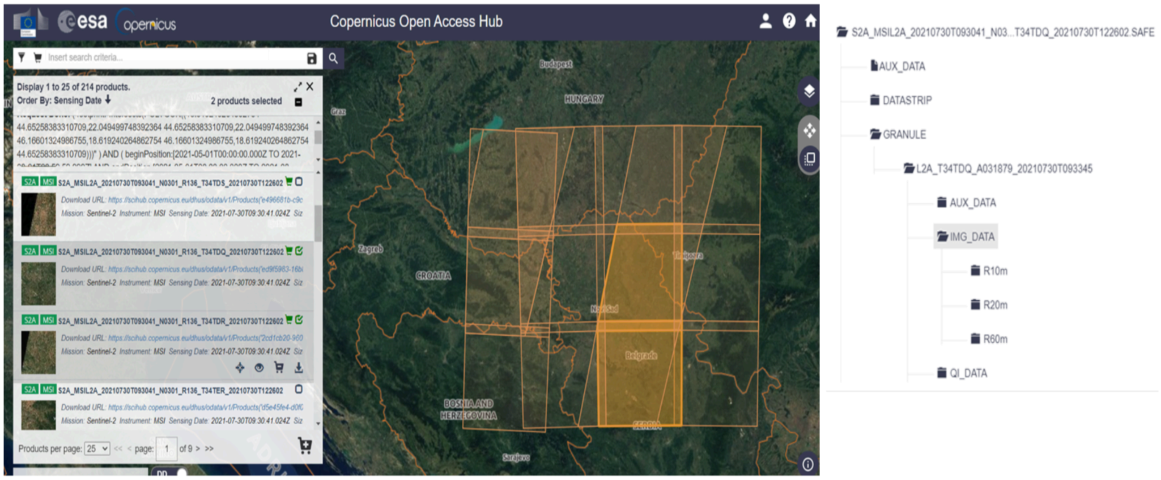
Figure 3 illustrates the interface of the Copernicus Open Access Hub, which provides access to satellite data from the Copernicus program's Sentinel satellites. Figure 3 shows how users can select and download multispectral images from the Sentinel-2 satellite for specific geographical regions and dates.

The upper left part shows the search interface where users can input specific criteria to locate satellite products. The search appears to be sorted by the "Sensing Date" which is typically the date when the satellite imagery was captured.

Below the search bar, there is a list of products displayed. Each entry provides details such as the product ID, which includes the sensing date, the mission name (Sentinel-2), the type of product (MSI for Multispectral Instrument), and the precise timing information. The listed URLs indicate where the data can be downloaded.

The right part presents a satellite view of a region with overlapping segments. Each segment represents the coverage area of a single Sentinel-2 image. The highlighted tiles are likely the ones that have been selected for download, which cover parts of Hungary, Slovenia, Croatia, Bosnia and Herzegovina, and Serbia.

There is an expanded view of the file directory for one of the products, which includes subfolders such as 'AUX_DATA', 'DATAStrip', 'GRANULE', with further subdivisions revealing the presence of 'IMG_DATA' and resolution-specific folders like 'R10m', 'R20m', and 'R60m', corresponding to different image resolutions as noted in the Figure 2.



Display 1 to 25 of 214 products.
Order By: Sensing Date ↓ 2 products selected

44.65258383310709,22.049499748392364 44.65258383310709,22.049499748392364 46.16601324986755,18.619240264862754 46.16601324986755,18.619240264862754 44.65258383310709)))") AND (beginPosition:[2021-05-01T00:00:00.000Z TO 2021-

S2A MSI S2A_MSIL2A_20210730T093041_N0301_R136_T34TDS_20210730T122602 □
Download URL: https://scihub.copernicus.eu/dhus/odata/v1/Products('e496681b-c9c
Mission: Sentinel-2 Instrument: MSI Sensing Date: 2021-07-30T09:30:41.024Z Siz

S2A MSI S2A_MSIL2A_20210730T093041_N0301_R136_T34TDQ_20210730T122602 ☑
Download URL: https://scihub.copernicus.eu/dhus/odata/v1/Products('ed9f5983-16b
Mission: Sentinel-2 Instrument: MSI Sensing Date: 2021-07-30T09:30:41.024Z Siz

S2A MSI S2A_MSIL2A_20210730T093041_N0301_R136_T34TDR_20210730T122602 ☑
Download URL: https://scihub.copernicus.eu/dhus/odata/v1/Products('2cd1cb20-960
Mission: Sentinel-2 Instrument: MSI Sensing Date: 2021-07-30T09:30:41.024Z Siz

S2A MSI S2A_MSIL2A_20210730T093041_N0301_R136_T34TER_20210730T122602 □
Download URL: https://scihub.copernicus.eu/dhus/odata/v1/Products('d5e45fe4-d0f
Mission: Sentinel-2 Instrument: MSI Sensing Date: 2021-07-30T09:30:41.024Z Siz

Products per page: 25 << < page: 1 of 9 > >>

```
S2A_MSIL2A_20210730T093041_N03...T34TDQ_20210730T122602.SAFE
    AUX_DATA
    DATASTRIP
    GRANULE
        L2A_T34TDQ_A031879_20210730T093345
            AUX_DATA
            IMG_DATA
                R10m
                R20m
                R60m
            QI_DATA
```

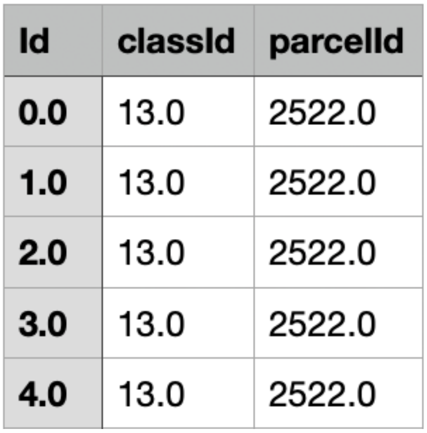
Figure 3: Copernicus Open Access Hub, source [14]

The BioSense Institute provided data that was converted into pixel format. Data was obtained through field work and field visits. Data is not open source and I got access to the data for joint research as part of BioSens' collaboration with students from the Faculty of Sciences.

This processed data was then organized into a CSV file format, providing us with pixel-level information where each pixel is labeled with a corresponding crop type identifier. The provided dataset facilitates further analysis and model training for crop classification.

The CSV file shown in Figure 4 contains three columns:

- Id: This is labeled as "pixel count".
- classId: This stands for "class label," which is a category for a type of crop.
- parcelId: This represents a "parcel identification number".



Id	classId	parcelId
0.0	13.0	2522.0
1.0	13.0	2522.0
2.0	13.0	2522.0
3.0	13.0	2522.0
4.0	13.0	2522.0

Figure 4: ‘train_data_labels.csv’

On Table 1 we can see all of the class labels. There are 11 classes in total: Maize, Wheat, Soybean, Sugar beet, Barley, Clover, Alfalfa, Oilseed rape, Vegetables, Other and Orchard. This is used to train a model to recognize different types of crops.

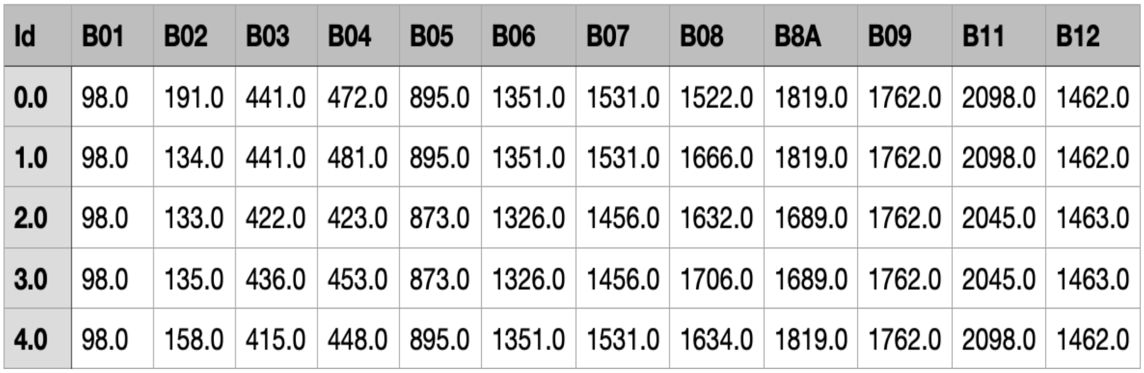
KUKURUZ	MAIZE	0
PŠENICA	WHEAT	1
SOJA	SOYBEAN	2
ŠEĆERNA REPA	SUGAR BEET	3
JEČAM	BARLEY	4
DETELINA	CLOVER	5
SUNCOKRET	ALFALFA	6
ULJANA REPICA	OILSEED RAPE	7
POVRĆE	VEGETABLES	8
OSTALO	OTHER	11
VOĆNJAK	ORCHARD	13

Table1: Classes represented in ‘train_data_labels.csv’

Figure 5 displays an example of a CSV file named with dates in the format 'yyyymmdd', indicating a time series of multispectral images from December 2020 to September 2021. Each file corresponds to a collection of 'granules', which are segments of the satellite imagery. The selected file, '20201205.csv', contains numerical data organized in a table with columns labeled 'Id' and 'B01' to 'B12', representing different spectral bands of the Sentinel-2 satellite data.

In total, there are 15 CSV files listed, each representing a set of data with a specific date. These file names suggest that they correspond to satellite data captured on the dates represented in their names, following a year-month-day format. The exact names of the CSV files are: ‘20201205.csv’, ‘20210223.csv’, ‘20210228.csv’, ‘20210409.csv’, ‘20210424.csv’, ‘20210504.csv’, ‘20210509.csv’, ‘20210623.csv’, ‘20210628.csv’, ‘20210708.csv’, ‘20210713.csv’, ‘20210807.csv’, ‘20210812.csv’, ‘20210911.csv’, ‘20210926.csv’.

The CSV files for the test dataset were available with the same 'yyyymmdd' date format as those used for the training data.

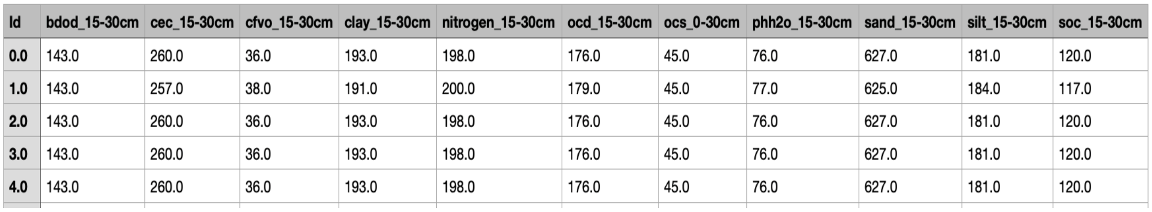


Id	B01	B02	B03	B04	B05	B06	B07	B08	B8A	B09	B11	B12
0.0	98.0	191.0	441.0	472.0	895.0	1351.0	1531.0	1522.0	1819.0	1762.0	2098.0	1462.0
1.0	98.0	134.0	441.0	481.0	895.0	1351.0	1531.0	1666.0	1819.0	1762.0	2098.0	1462.0
2.0	98.0	133.0	422.0	423.0	873.0	1326.0	1456.0	1632.0	1689.0	1762.0	2045.0	1463.0
3.0	98.0	135.0	436.0	453.0	873.0	1326.0	1456.0	1706.0	1689.0	1762.0	2045.0	1463.0
4.0	98.0	158.0	415.0	448.0	895.0	1351.0	1531.0	1634.0	1819.0	1762.0	2098.0	1462.0

Figure 5: Classes represented in ‘20201205.csv’

Figure 6 shows another csv file named "soilData.csv". This file contains soil data with various parameters, each column labeled with a specific soil attribute and measurement depth. Explanation of the data columns visible on the Figure 7:

- Id: Unique identifier for each data row.
- bdod_15-30cm: 'Bulk density' of the soil between 15 to 30 cm depth.
- cec_15-30cm: 'Cation exchange capacity' at the 15 to 30 cm depth.
- cfvo_15-30cm: 'Coarse fragments volume' between 15 to 30 cm depth.
- clay_15-30cm: ‘Clay’ at the 15 to 30 cm depth.
- nitrogen_15-30cm: Indicates the content of nitrogen in the soil at the depth of 15 to 30 cm.
- ocd_15-30cm: 'Organic carbon density' in the soil between 15 to 30 cm depth.
- oc_0-30cm: 'Organic carbon' content from the surface to 30 cm depth.
- phh2o_15-30cm: The pH measurement of the soil water extract between the depths of 15 to 30 cm.
- sand_15-30cm: The percentage of sand particles in the soil at the 15 to 30 cm depth.
- silt_15-30cm: This represents the silt content in the soil at the 15 to 30 cm depth.
- soc_15-30cm: 'Soil organic carbon' content between 15 to 30 cm depth.



Id	bdod_15-30cm	cec_15-30cm	cfvo_15-30cm	clay_15-30cm	nitrogen_15-30cm	ocd_15-30cm	ocs_0-30cm	phh2o_15-30cm	sand_15-30cm	silt_15-30cm	soc_15-30cm
0.0	143.0	260.0	36.0	193.0	198.0	176.0	45.0	76.0	627.0	181.0	120.0
1.0	143.0	257.0	38.0	191.0	200.0	179.0	45.0	77.0	625.0	184.0	117.0
2.0	143.0	260.0	36.0	193.0	198.0	176.0	45.0	76.0	627.0	181.0	120.0
3.0	143.0	260.0	36.0	193.0	198.0	176.0	45.0	76.0	627.0	181.0	120.0
4.0	143.0	260.0	36.0	193.0	198.0	176.0	45.0	76.0	627.0	181.0	120.0

Figure 6: Soil data in ‘soilData.csv’

3.2. Exploratory Analysis

In the exploratory analysis section, the idea was to look into the dataset in more detail, by plotting various graphs and exploring the classes. This preliminary step is crucial to understand the underlying structure of the data before applying more complex analysis or modeling techniques.

The provided pie chart (Figure 7) illustrates the distribution of different crop types. Maize is the most prevalent, accounting for 29.3% of the data, followed by wheat at 21.7%, which together exceed 50% of all data. Other crops like soybean, sugar beet, barley, and alfalfa have smaller proportions, with orchard and other categories making up the rest. This visualization helps in understanding the frequency of each crop type within the dataset.

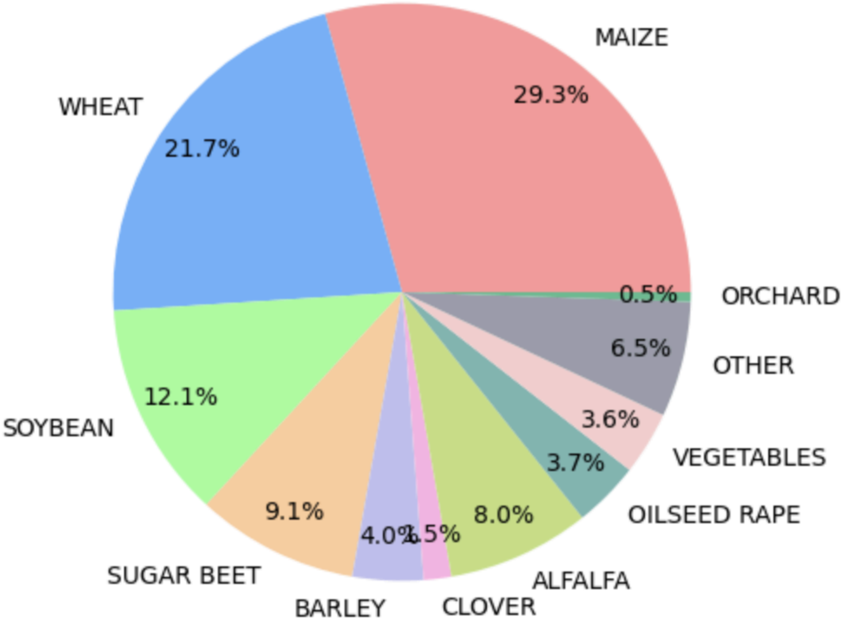


Figure 7: Percentage representation of classes

The bar graph (Figure 8) displays the numerical representation of different crop classes in the dataset. Maize and wheat are significantly more represented with higher counts, exceeding 800,000 and 600,000 respectively, while crops like orchard and oilseed rape have much lower counts, which gives us a first indication that this data is imbalanced.

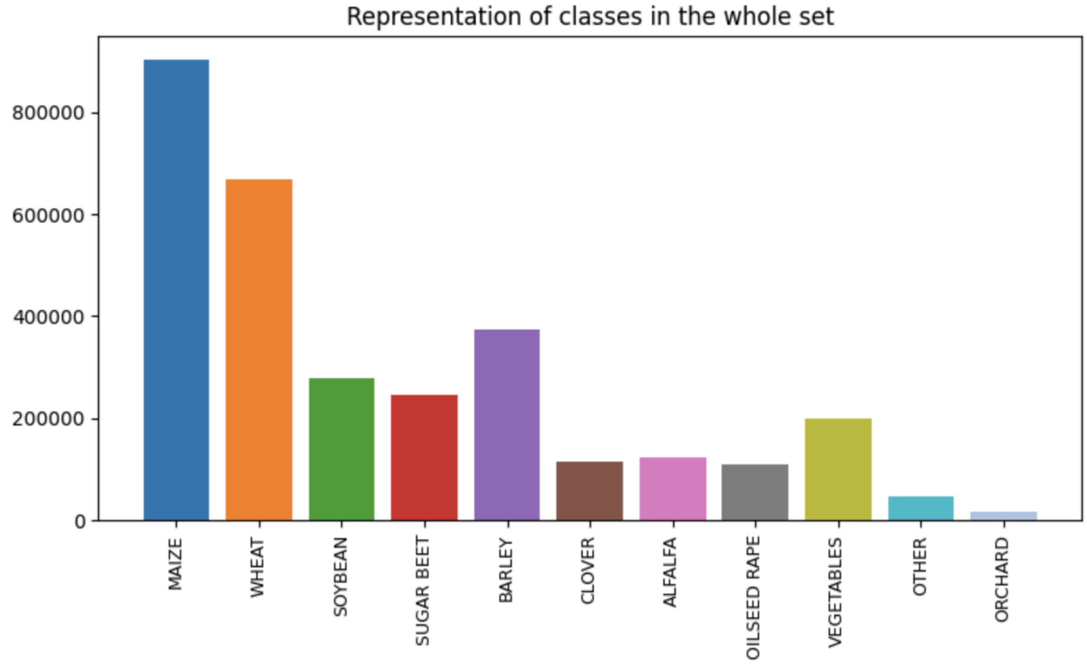


Figure 8: Numerical representation of classes in the whole set

In the next step, we conducted the analysis to understand the correlations and dependencies between different variables in the dataset. To do this, we plotted a heat map, which is a visual tool that uses color-coding to represent the values of a matrix. Here's how it works and how to read it:

- Each square in the heat map corresponds to a pair of variables.
- The color of each square indicates the strength and direction of the correlation between the two variables.
- Typically, a color gradient is used, where one color (like blue in this case) indicates a strong positive correlation, another (like red) indicates a strong negative correlation, and a neutral color (like white or light yellow) indicates no correlation. A legend is usually included that explains the color scale, often ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no correlation.

By examining the heat map, we could quickly assess which variables have strong correlations with each other. This is particularly useful for identifying potential features that could be predictors for modeling, as well as for spotting multicollinearity, where two or more variables are highly correlated and could potentially change the results of a predictive model.

The heatmap (Figure 9) suggests varying degrees of correlation between the different spectral bands (B01 to B12). The 'classId' row and column, which are mostly yellow, suggest that the class is not linearly correlated with the spectral bands, indicating that the class cannot be predicted by a simple linear combination of these bands alone. The 'parcelId' seems to be mostly disconnected from the spectral bands, given the yellow squares along its row and column, highlighting that parcel identification is likely independent of these spectral features.

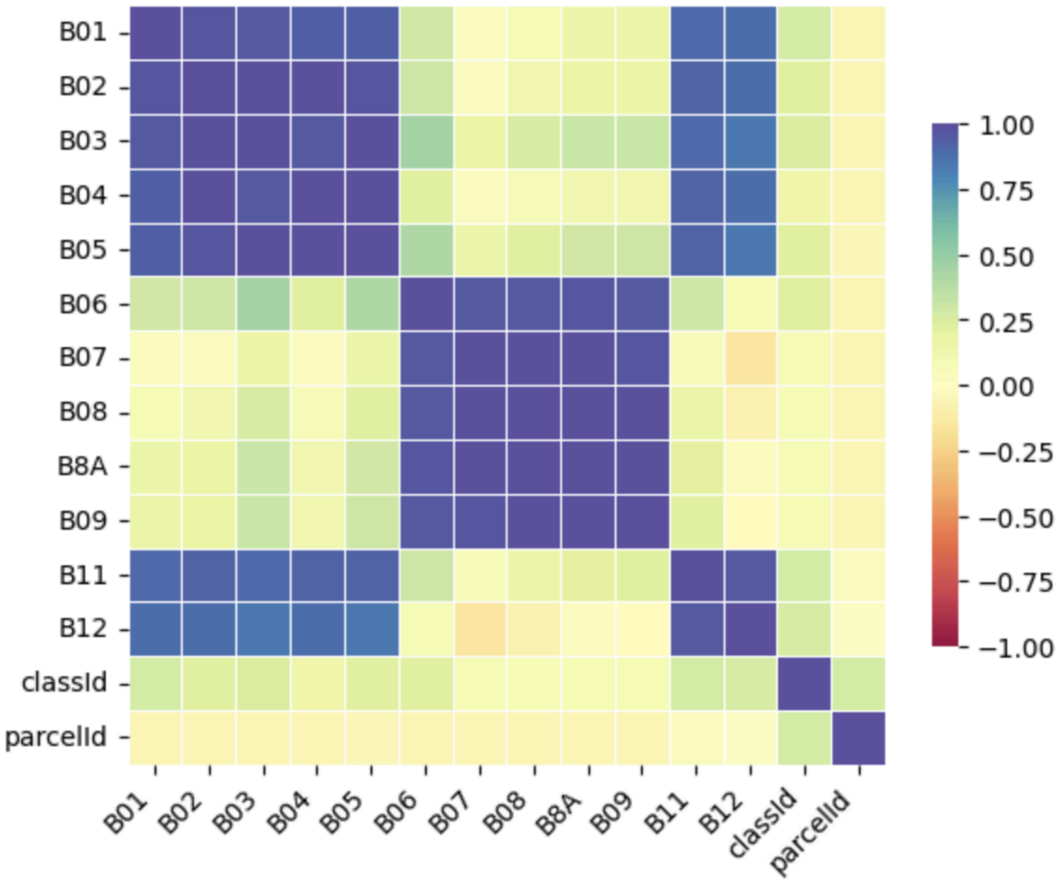


Figure 9: Correlation Matrix of Spectral Bands and Identifier

3.3. Data Preprocessing

Data preprocessing is a crucial step in the data analysis process, as it prepares the raw data for further processing and analysis. This stage ensures that the datasets are cleaned and formatted correctly, which can significantly improve the performance of machine learning and deep learning models.

In this phase, to construct a comprehensive training dataset, we combined multiple CSV files representing different time periods with a separate file containing the target labels. First logical approach was to prioritize the summer months for the initial merging of data. The rationale behind this decision was based on:

- Visibility of crops: During summer, crops are typically in full bloom, making them more visible and distinguishable in satellite imagery.
- Absence of snow cover: The lack of snow cover in the warmer months eliminates the risk of satellite images being more cloudy, which can occur in winter, leading to potential inaccuracies in the data.
- Peak growing season: Summer corresponds to the peak growing season for many crops.

By starting the data integration process with the summer months, we ensured that the dataset was rich with the most reliable and detailed information available for the crops.

Each CSV contained a set of features specific to a time period, and our objective was to bring them all together alongside their corresponding labels for model training. Here is an overview of how we merged these files:

1. Suffix assignment: To differentiate between features from different time periods, we assigned a unique suffix to each set of features corresponding to their respective CSV. This helps to avoid column name conflicts during the merge.
2. Preparation of DataFrames: We created a list named merged_dfs that contained all the data sets for training (‘20200708’ etc.) along with the labels (data_labels).
3. Initial DataFrame setup: We initialized the combined DataFrame, data_all_train, with the first dataset in the list to serve as the base for next merges.
4. Iterative merging:
    a. We looped through each DataFrame in merged_dfs starting from the second one (since the first was already initialized as the base).

b. For each DataFrame, we checked if there were any column names, excluding 'Id', that overlapped with those in the current data_all_train.
c. If no overlapping columns were found, we merged the DataFrame directly on the 'Id' column without adding any suffixes. This implies that each DataFrame had unique feature names and could be directly combined.
d. In case of overlapping column names, we performed the merge using suffixes to differentiate between features from different time periods.

The resulting data_all_train DataFrame was a horizontally combined set of all the features from different periods, neatly associated with their respective labels from the data_labels DataFrame. This merging process was crucial to ensure that the model had access to all necessary temporal features corresponding to the same 'Id' for its predictions.

In the process of data preparation, an attempt was made to augment the dataset by adding values as new rows to existing columns.

From the exploratory analysis, it was clear that the dataset was imbalanced, which means not all classes that needed to be predicted were represented equally. The most common class, class 0, was significantly overrepresented, while class 13 was the least represented. To resolve this issue, the Synthetic Minority Over-sampling Technique (SMOTE) algorithm was applied. SMOTE generated synthetic data, effectively balancing the class representation by ensuring that all classes had the same presence in the dataset.

We applied the MinMax Scaler to scale each feature between zero and one.

Similarly, the Standard Scaler was used to standardize the features to a mean of zero and a standard deviation of one.

3.4. Algorithms

In this paper, we wanted to try different machine learning and deep learning algorithms. At first, we tried Naive Bayes, KNN and Random Forest. The idea was to test these algorithms and how changing their parameters affects the results. After that, we created a Convolutional Neural Network, changing different layers, as well as parameters.

3.4.1. Naive Bayes

Naive Bayes (NB) is a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. It is a classification technique based on the Bayes' Theorem, which assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature [15]. The 'naive' in Naive Bayes comes from the assumption that the attributes of data are independent of each other, which is rarely the case in real-world data.

Bayes' Theorem (Figure 10) is used to find the probability of a hypothesis given some evidence. In mathematical terms:

- P(A|B) is the probability of hypothesis A given the evidence B
- P(B|A) is the probability of evidence B given that hypothesis A is true
- P(A) is the probability of hypothesis A being true (regardless of the evidence)
- P(B) is the probability of the evidence (regardless of the hypothesis).

In the context of classification, we can think of A as the class we want to predict and B as the observed data. Naive Bayes calculates the probability of each class given the data and classifies the data into the class with the highest posterior probability.

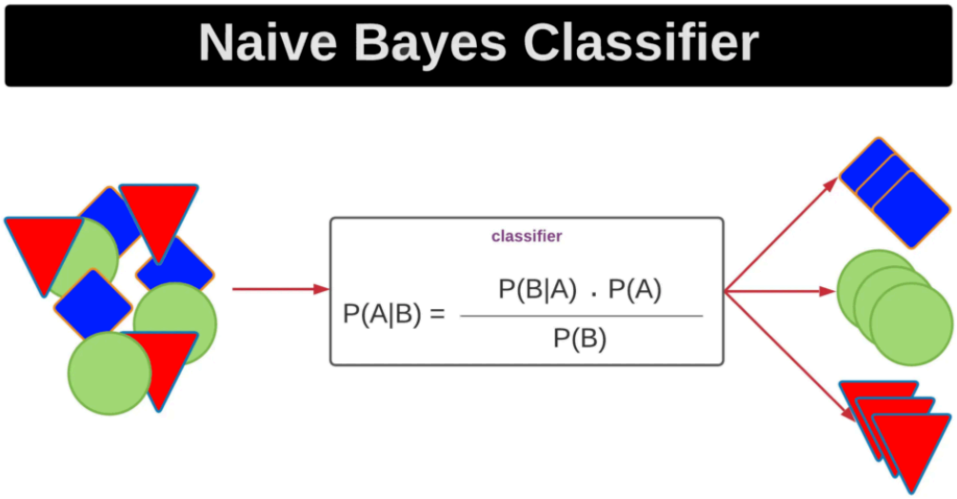


Figure 10: Naive Bayes Classifier, source [16]

Advantages of NB [17]:

- It is straightforward to implement and understand.
- It can be trained relatively quickly even with a large number of features.
- Works well with high-dimensional data.
- Often performs well in multi-class prediction.
- Can handle noise well if the noise is random and not biased toward any class.

Limitations of NB [17]:

- The assumption of feature independence is often violated in real-life scenarios, which can limit its effectiveness.
- If the probability of a categorical variable's feature is zero, it can nullify the entire probability.

Depending on the nature of the features, different types of Naive Bayes classifiers can be used:

1. *Gaussian*: Assumes that the continuous values associated with each class are distributed according to a Gaussian distribution.
2. *Multinomial*: Typically used for document classification, where the features are the frequencies of the words or tokens.
3. *Bernoulli*: Useful if your feature vectors are binary (i.e., zeros and ones).

3.4.2. KNN

K-Nearest Neighbors (KNN) is a supervised learning algorithm used for both classification and regression tasks [18]. It is a non-parametric, instance-based learning algorithm that makes predictions based on the majority class of its k nearest neighbors. Despite its simplicity, KNN has been widely applied in various domains.

Given an input vector, KNN searches the training dataset for the 'k' training examples that are closest to the point. The output is then determined by averaging (in regression problems) or by majority voting (in classification problems) [20].

The effectiveness of KNN is inherently tied to the distance metric used to find the nearest neighbors. Commonly used metrics include Euclidean distance, Manhattan distance, and Minkowski distance [21]. The choice of distance metric can significantly influence the performance of KNN and is often determined based on the nature of the data.

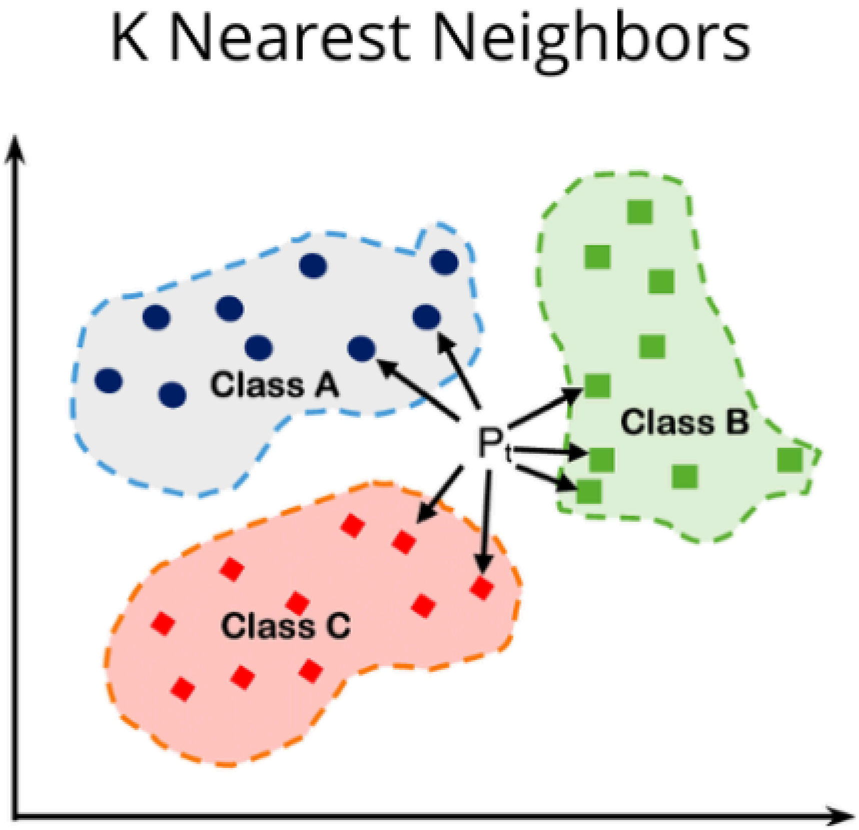


Figure (11): KNN, source [19]

KNN's primary strength lies in its simplicity and its ability to adapt to changes, as it makes decisions based on the entirety of the training data rather than a specific model [22]. However, it can be computationally intensive, especially with large datasets, as it requires the computation of the distance of a given point to all points in the training set.

The KNN algorithm, with its intuitive approach based on proximity and majority rule, has found applications across various domains. Its non-parametric nature, coupled with its adaptability, makes it a go-to algorithm for tasks where the relationship between features may not be linear or well-defined.

Advantages of KNN:

- The underlying principle of KNN is easy to understand – similar items are close to each other in the feature space, making it an intuitive algorithm for beginners.
- KNN is a lazy learner, meaning it does not have a training phase. It memorizes the training dataset and makes predictions at runtime, making it efficient for dynamic or frequently changing datasets.
- It can be applied to both classification and regression tasks, making it versatile for a wide range of applications.
- It can adapt well to changes in the dataset, making it suitable for datasets with evolving patterns.

Limitations of KNN:

- Calculating distances between the new data point and all existing points in the dataset can be computationally expensive, especially in large datasets.
- Features with larger scales can dominate the distance calculations, leading to biased predictions.
- It is not always simple and easy to choose an appropriate value for k.

3.4.3. Random Forest

Random Forest (RF) is an ensemble learning method predominantly used for classification and regression, which constructs a multitude of decision trees at training time to output the mode of the classes (classification) or mean prediction (regression) of the individual trees [23].

The fundamental concept of RF is to build several decision trees and let them work in harmony. Each tree is trained on a random subset of the dataset, and during prediction, the results from individual trees are combined to produce the final output. The final decision is made based on the majority voting from all the trees. By considering different subsets of features and data points for each tree, the algorithm reduces overfitting and increases the model's accuracy.

Each tree in the Random Forest is built from a sample drawn with replacement from the training set. This method, known as bootstrap aggregating or bagging, increases the diversity of the trees in the forest, leading to a more robust overall model [25].

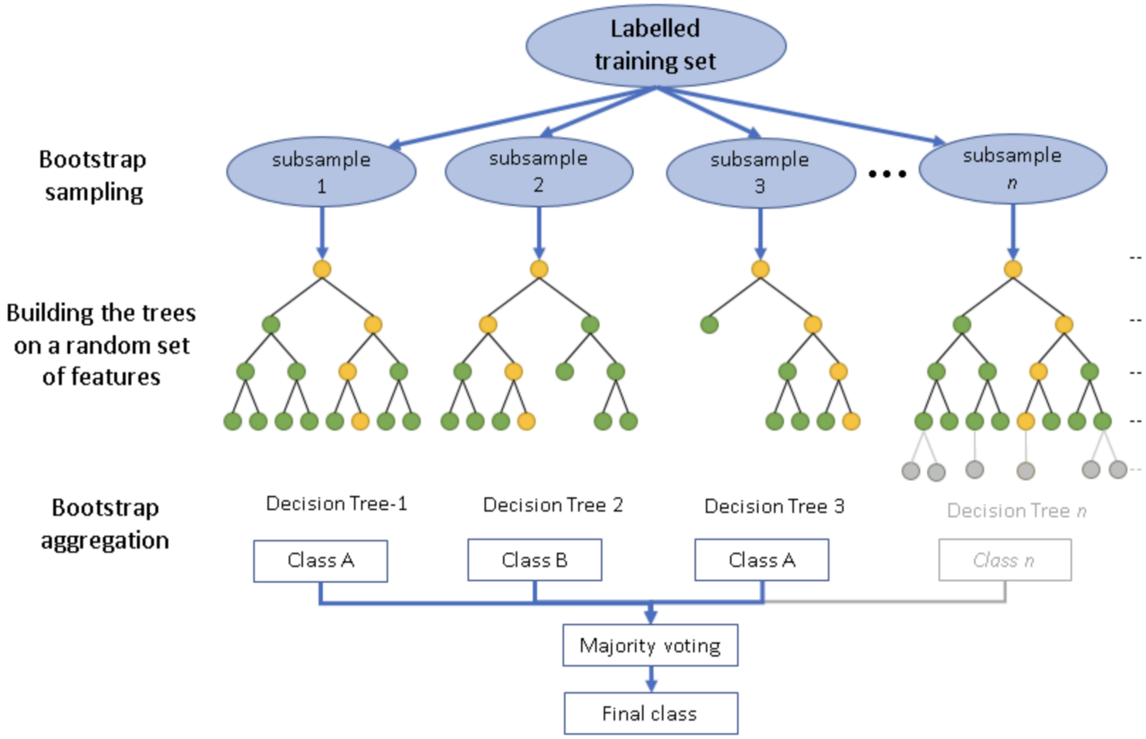


Figure (12): Simplified scheme of Random Trees, source [24]

When individual trees are grown, only a random subset of features is considered for splitting each node. The randomness in feature selection ensures that the trees are de-correlated, which makes the forest's predictions more reliable by reducing the variance [26].

Advantages of Random Forest:

- It is less prone to overfitting. Each individual tree may overfit the data, but collectively, the forest tends to balance this out [23].
- RF can handle unbalanced data sets by balancing the error in the minority class [27].
- It provides a straightforward indicator of feature importance, making it a valuable tool for feature selection [28].

Limitations of Random Forest:

- A large number of trees can increase the overall training time, especially when dealing with large datasets.
- The decision process of a RF is not as easily interpretable as that of a single decision tree, which can be a downside in applications where understanding the model's decision-making process is crucial.
- It can be viewed as a "black box" model. Understanding how individual trees arrive at a specific prediction can be challenging.

Hyperparameters in Random Forest:

- N_estimators: Determines the number of trees in the forest. While a higher number often leads to better performance, it also increases computational demands.
- Criterion: Defines the function to measure the quality of a split. "Gini" measures Gini impurity, while "entropy" calculates information gain.
- Max Depth: Specifies the maximum depth of each tree. Deeper trees can capture more complex patterns but increase the risk of overfitting.
- Max Features: Determines the number of features considered for each split. Choosing an appropriate value is crucial to balance model complexity and accuracy.
- Bootstrap: Decides whether to use bootstrap samples (randomly sampled with replacement) when building trees. Setting it to "False" means using the entire dataset for each tree.
- N_jobs: Specifies the number of processors used during training. Utilizing multiple processors can significantly speed up the training process.

3.4.4. Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a class of deep neural networks. They are particularly known for their ability to recognize patterns and structures within images, making them extremely useful for image and video recognition, image classification and more.

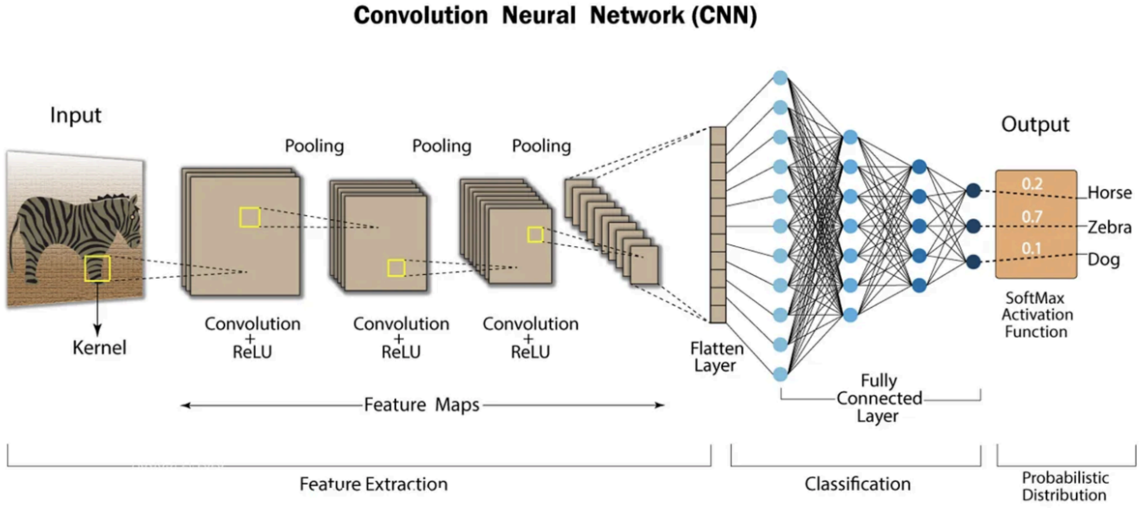


Figure (13): CNN architecture, source [29]

We will try to explain architecture of CNN, by explaining the hyperparameters in CNNs:

1. Dropout: Dropout is a regularization technique used to prevent overfitting in neural networks, including CNNs [30].
2. Max Pooling: Max pooling reduces the spatial dimensions (width and height) of the input, thus reducing the number of parameters and computational complexity in the network. This operation retains the most important features while discarding less relevant information.
3. Batch Normalization: Batch normalization is a technique used to improve the training stability and speed up the convergence of neural networks. It normalizes the inputs of each layer in a mini-batch to have a zero mean and unit variance.
4. Activation Functions: Activation functions introduce non-linearities into the network, allowing it to learn complex patterns. Common activation functions include ReLU (Rectified Linear Unit), which replaces all negative values with zero, and variants like Leaky ReLU and Parametric ReLU that allow small negative values to pass through. Sigmoid and Tanh functions squash the output between 0 and 1 or -1 and 1, respectively, making them suitable for binary classification tasks.
5. Fully Connected Layers: Fully connected layers, also known as dense layers,

are traditional neural network layers where each neuron is connected to every neuron in the previous and subsequent layers. These layers are often added at the end of CNN architectures to map the extracted features to the output classes.

6. Softmax Activation: Softmax activation is commonly used in the output layer of a classification network. It converts the raw output scores of the network into probability values, ensuring that they sum up to 1.

Advantages of CNNs:

- CNNs are capable of recognizing patterns regardless of their position in the input images, providing spatial invariance and making them robust to variations in object position, scale, and orientation.
- Pretrained CNN models can be fine-tuned on specific tasks with limited data, leveraging knowledge learned from large, general datasets.
- While initially designed for image-related tasks, CNN architectures have been adapted and extended for various other data types, including sequences and graphs, showcasing their adaptability and versatility.

Limitations of CNNs:

- Computational Intensity: Training deep CNNs, especially large architectures, requires substantial computational resources, including powerful GPUs.
- Need for Large Datasets: CNNs, particularly deep ones, thrive on large, diverse datasets for effective training. Acquiring and annotating such datasets can be challenging and time-consuming.

4. Explainable AI

Explainable AI (XAI) refers to the set of techniques and methods designed to make complex machine learning and deep learning models interpretable and understandable to users. As machine learning models, particularly deep learning models, become increasingly sophisticated, there is a growing need to understand the decisions they make. XAI aims to explain these "black box" models, providing insights into how they arrive at specific predictions or decisions. It helps answer questions about why a certain decision was made, building confidence in the model's predictions.

In sensitive domains like healthcare and criminal justice, it is crucial to ensure that AI models are not biased or discriminatory. Explainability allows practitioners to identify and mitigate biases, promoting fairness and ethical use of AI.

XAI plays an important role in enhancing our comprehension of complex processes in agriculture. It not only contributes to the advancement of agricultural science but also helps in addressing broader environmental challenges. Through the insights provided by XAI, we can develop more resilient agricultural systems that are better adapted to the changing climate and environmental conditions, ultimately expanding our knowledge of the intricate interactions within nature.

Explainable models are more likely to be adopted in real-world applications, especially when users need to act based on the model's recommendations. If users understand the reasoning behind the predictions, they are more likely to accept, or not accept, and act upon them.

In the field of XAI, there are different techniques based on the models. For machine learning, SHAP (SHapley Additive exPlanations) is a powerful technique, while Integrated Gradients offers a method for deep learning networks.

Since the focus of this paper is on machine learning models, we will explore the SHAP technique.

4.1. Introduction to SHAP

SHAP, based on game theory, is a powerful technique in the field of XAI. It provides a measure of feature importance, attributing a value to each feature indicating its contribution to a particular prediction. SHAP values consider all possible feature combinations, ensuring fair and consistent attribution.

SHAP builds Shapley values which are used to quantify the contribution of each feature to the prediction of a particular instance. Shapley values fairly distribute the "value" of a coalition among its members. In the context of machine learning, features are considered as players in a cooperative game, and the value to be distributed is the model's prediction for a specific instance [10].

SHAP provides consistent feature attribution, meaning that the sum of SHAP values for all features equals the difference between the model's prediction for the instance and the average prediction. This consistency ensures reliable interpretation across different instances and models.

Methods in this family are concerned with explanations for the model f at some individual point x^*. They are based on a value function e_S where S is a subset of variable indexes $S \subseteq \{1, ..., p\}$ [35]. Typically, this function is defined as the expected value for a conditional distribution in which conditioning applies to all variables in a subset of S

$$e_S = E[f(x)|x_S = x_s^*].$$

This function is also often defined as the model prediction at x^* after zeroing out the values of variables with indices outside S. Whichever definition is used, the value of e_S can be thought of as the model's response once the variables in the subset S are specified [35].

SHAP can be used with any machine learning model, as it is model-agnostic. It is implemented in Python with the 'shap' library [36], which provides a high-level interface for applying SHAP to models. The library includes specific algorithms optimized for different types of models, such as tree-based models (TreeSHAP), which allow efficient computation of SHAP values [37].

Advantages of SHAP:

- If a model changes in such a way that a feature's contribution increases or stays the same, regardless of the other features, the attributed importance to that feature should not decrease.
- Each feature's attributed importance adds up to the actual prediction minus the average prediction over the dataset.
- SHAP values naturally account for missing data using the Shapley value's properties [38].

Limitations of SHAP:

- Calculating SHAP values can be computationally expensive, especially for models with a large number of features and complex interactions.
- Approximations of SHAP values might be necessary, which can introduce inaccuracies.
- Interpretation of SHAP values can sometimes be non-intuitive, especially in the presence of highly correlated features.

5. Metrics and validation

Evaluating machine learning models involves a range of metrics, each providing different insights into the model's performance. Here's an overview of used evaluation metrics:

1. Accuracy: The ratio of correctly predicted observations to the total observations. It is a useful measure when the target classes are well balanced.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

2. Precision: The ratio of correctly predicted positive observations to the total predicted positive observations. It is particularly important in contexts where False Positives are a larger concern than False Negatives.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

3. Recall (Sensitivity): The ratio of correctly predicted positive observations to all the observations in actual class. It is important in contexts where False Negatives are more concerning than False Positives.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

4. F1 Score: The weighted average of Precision and Recall. This score takes both false positives and false negatives into account, and is a more informative measure than Accuracy on imbalanced class problems.

$$F1 = 2\frac{Precision * Recall}{Precision + Recall}$$

5. Macro F1 Score: Is a way of computing the F1 score that gives equal weight to all classes, regardless of their size. It's particularly useful when you're dealing with imbalanced datasets. The Macro F1 score is calculated by taking the average of the F1 scores of each class:

$$Macro\ F1 = \frac{1}{N}\sum_{i=1}^{N} F1_i$$

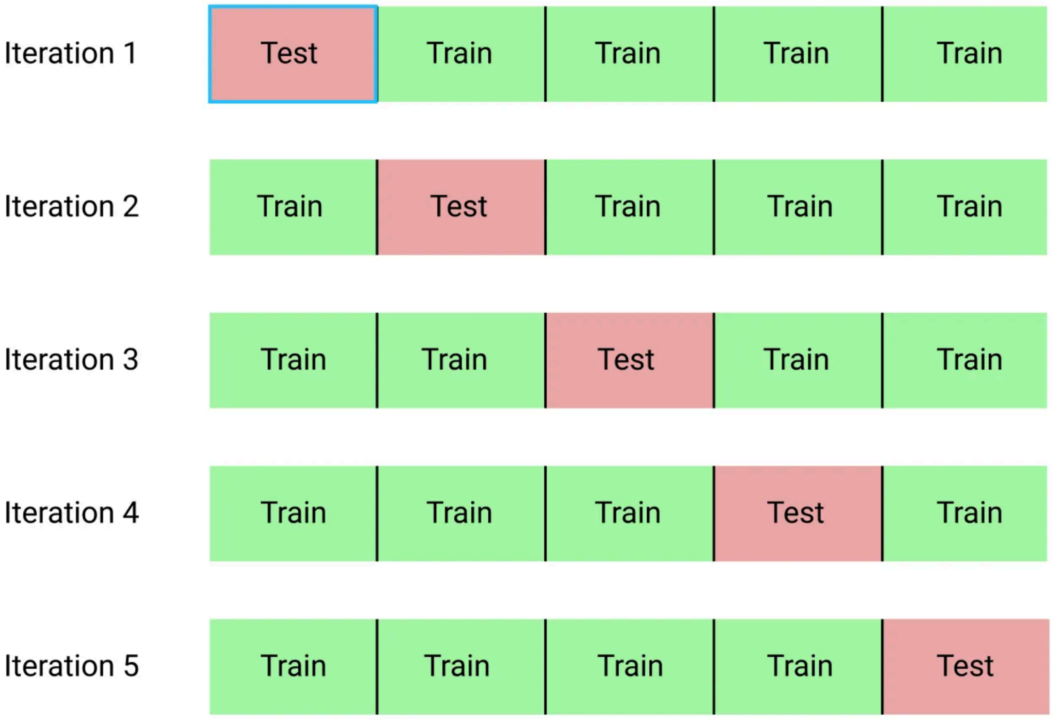
where N is the number of classes and $F1_i$ is the F1 score for class i.

We would highlight that part of this thesis was part of a project in subject Large Scale Data Mining. For that purpose, we were uploading our results to the Kaggle competition page [31], where predictions were evaluated by Macro F1-Score.

In machine learning, the robustness and generalizability of a model are very important. Cross-validation is a very useful methodology, providing insights into how a model performs on unseen data. Among its variations, Group K-Fold cross-validation stands out.

Group K-Fold cross-validation is a technique that ensures the same group is not represented in both the training and validation sets [32]. This is crucial for scenarios where the data may contain groups or clusters, and it is essential to evaluate the model's performance on groups that it has not seen during training.

The dataset is split into 'k' groups, with each fold containing approximately the same number of distinct groups. The data is partitioned such that each fold is used once as a validation while the 'K-1' remaining folds form the training set. This process iterates until each group has been in the validation set exactly once [34].



Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

Figure (14): k-fold Cross Validation, source [33]

In this thesis, GroupKFold is employed with 'n_splits' equal to 3, 5 and 7, indicating the data is partitioned into 3, 5 and 7 groups, respectively. The 'groups_train' array indicates which group each sample belongs to, ensuring that samples from the same group are not split amongst the training and validation sets.

In summary, Group K-Fold cross-validation is a powerful tool in machine learning, offering a more stringent and realistic evaluation of a model's predictive capabilities. By ensuring that the validation sets contain only unseen groups, it provides a rigorous test of the model's ability to generalize beyond the data it was trained on.

6. Results

In this study, the idea was to develop models for crop classification using satellite imagery. The ultimate goal was to create a model that not only predicts with high accuracy but also whose decisions can be explained and justified. To this end, over 250 models were developed, including different machine learning and deep learning models, such as Naive Bayes, K-Nearest Neighbors (KNN), Random Forest (RF), Convolutional Neural Networks (CNN), and Multi-Layer Perceptron (MLP).

Model performance was evaluated based on the Macro F1 score, a harmonic mean of precision and recall that considers both false positives and false negatives, making it a balanced metric for multi-class classification problems. The Random Forest model showed as the top performer for the given dataset, achieving a Macro F1 score of 93.8%, followed by CNN at 88.4%, KNN at 87.8%, MLP at 87.1%, and Naive Bayes at 75.2%.

Table 2: Overall performance of models

Model	Macro F1 Score
Random Forest	93.8%
CNN	88.4%
KNN	87.8%
MLP	87.1%
Naive Bayes	77.7%

To delve deeper into the Random Forest model's predictive power, a comprehensive analysis of all performance metrics was undertaken. This analysis revealed that the model not only had an excellent F1 score but also scored highly across other metrics, demonstrating its robustness and reliability as a classifier for this application.

Table 3: Performance of Random Forest

	Accuracy	F1 score	Precision	Recall
Random Forest	0.91	0.94	0.94	0.86

As shown on Table 3, the RF model achieved an accuracy of 0.91, indicating a high level of overall correctness in predictions. The F1 score was 0.94, while precision was 0.94 and recall 0.86. These metrics collectively underscore the model's robustness and reliability as a classifier in our specific application.

The model's training was conducted on diverse data sets, a strategy that aimed to enhance its generalization capabilities. The data corresponding to the summer months resulted with the highest results. This seasonal variation in predictive performance suggests a potential correlation between the model's effectiveness and seasonal factors, which could be a subject of further investigation.

To further enhance our understanding of the model's predictive behavior, the next step involves the explainable AI (xAI). By employing xAI techniques, we aim to interpret and explain how the model makes its decisions. Understanding the model's decision-making process is crucial, especially in applications where transparency and trust are crucial.

MinMax Scaler normalization proved highly beneficial for models like neural networks, where data within a specific range significantly improved convergence and overall performance. After normalization Macro F1 score for CNN model increased from 83.1% to 88.4%.

Similarly, when we employed the Standard Scaler to standardize the features to a mean of zero and a standard deviation of one, we noticed a marked improvement in models that assume a Gaussian distribution of the features. This standardization helped in stabilizing the training process and improving model performance. This technique proved most useful for the Naive Bayes model, where the Macro F1 score increased from 72.9% to 77.7%.

7. Model explanations

Among all the models tested, the Random Forest (RF) algorithm demonstrated the most superior performance on the given dataset, achieving the highest Macro F1 score of 93.8%, indicating exceptional ability in balancing precision and recall across all classes in a multi-class classification scenario. We applied SHAP to this model to gain deeper insights into its decision-making process. The following figures will present the results of this implementation, showcasing how SHAP values elucidate the influence of different features on the model's predictions.

In summary, SHAP summary plot provides a snapshot of which features are driving the model's predictions, in which direction, and to what extent, based on the dataset used for this analysis.

We will show the Summary plots for all classes, so let's see how we can interpret the results:

- Feature Importance: Each row on the plot represents a feature (B01, B02, etc.) from the dataset.
- SHAP Values: The SHAP value measures the impact of a feature on the model's output. The plot displays a dot for each instance in the dataset.
- Value Color: The color represents the feature value for that instance. In these plots, pink dots indicate high feature values, while blue dots represent low feature values.
- Impact Direction: If a dot is on the right of the center (0 on the X-axis), it means that this feature's value for that instance had a positive impact on the model's prediction. Conversely, dots to the left indicate a negative impact.
- Feature Impact Distribution: The spread of the dots shows the distribution of the impacts. A wide spread means that the feature's impact varies greatly across instances, while a narrow spread suggests it has a more consistent impact.
- Overall Importance: Features with a higher concentration of dots far from the center are generally more important to the model's output. The plot generally orders features by their importance.

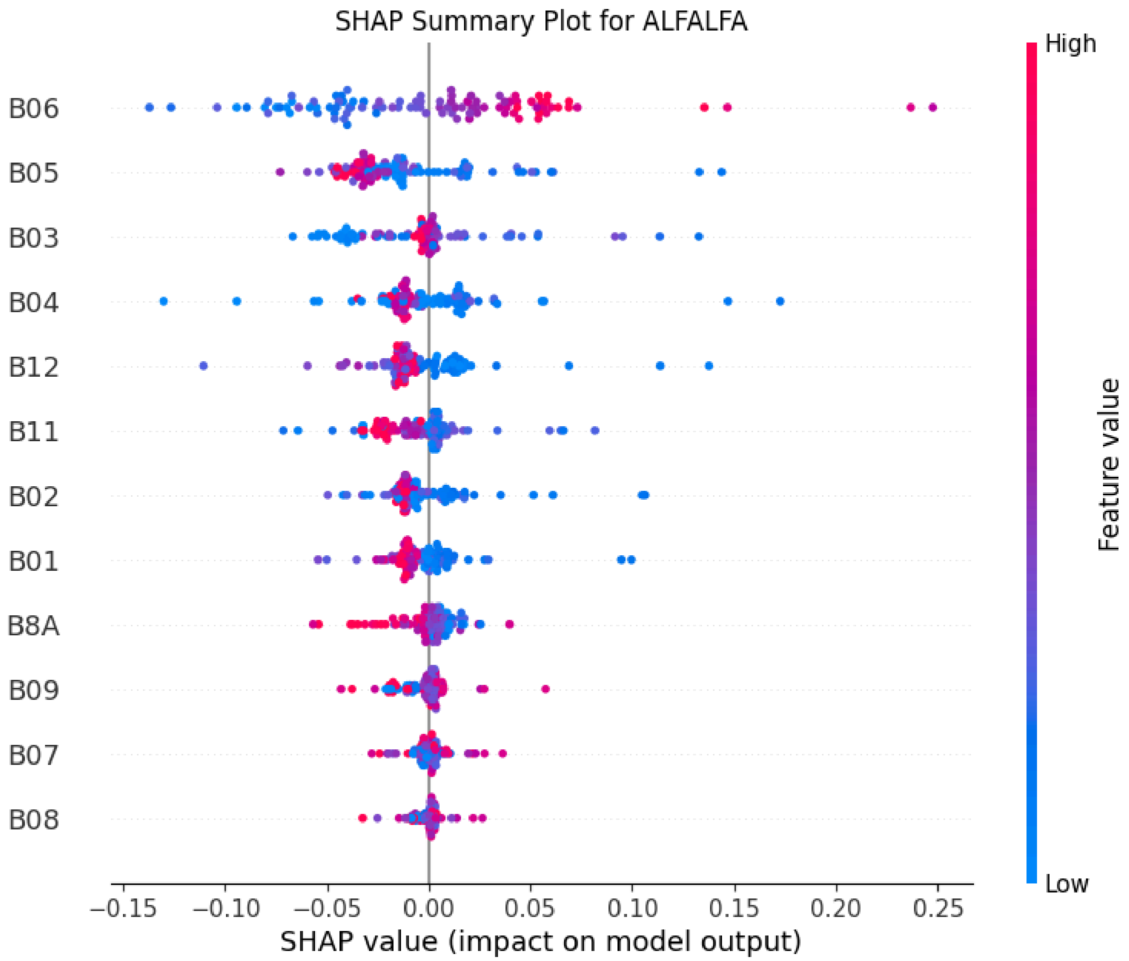


Figure (15): SHAP Summary Plot for ALFALFA

Let us explain Figure(15):

Feature B06 appears to have the most significant impact on the model's output. With a large spread of SHAP values primarily on the positive side, it suggests that higher values of B06 tend to increase the predicted outcome.

The pink dots (indicating high feature values) for feature like B06 are mostly on the positive side of the SHAP value axis. This implies that when these features have higher values, they are likely to push the model's predictions higher.

Conversely, for the features B03, B04, the blue dots (indicating lower feature values) tend to be on the negative side, suggesting that lower values of these features contribute to a decrease in the predicted outcome.

Features like B01, B8A, and B09 show a cluster of dots close to the zero center line, indicating that these features have a more consistent but smaller impact on the model's predictions.

Features such as B12 and B11 display a mix of positive and negative impacts on the model's output, as indicated by the spread of blue and pink dots on both sides of the zero line. This indicates that for different observations, these features can either increase or decrease the prediction, depending on their values.

Features like B07 and B08 have dots concentrated around the center with lower spread, which suggests these features have a lesser impact on the model output compared to features like B06.

This model seems to be most sensitive to variations in B06, with considerable but varied sensitivity to B05, B03, and B04.

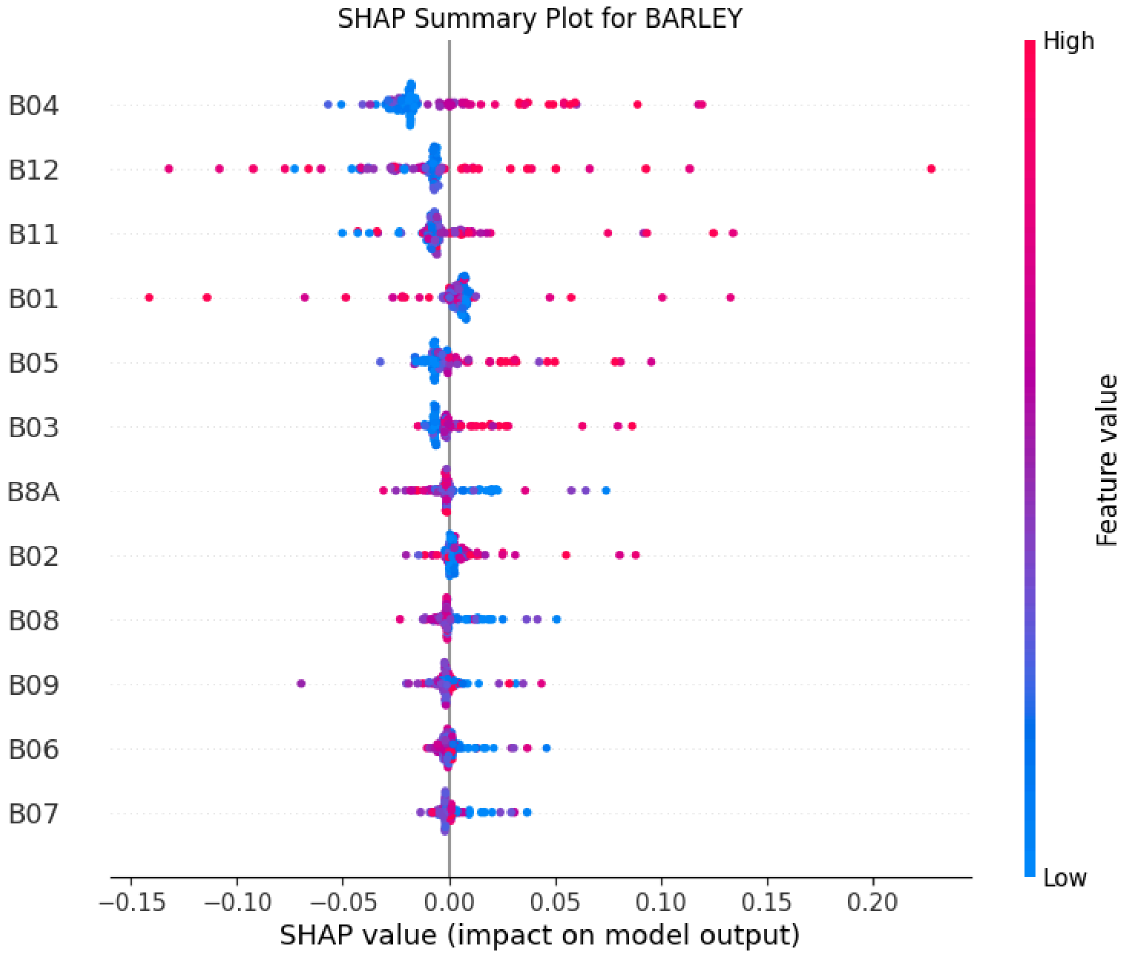


Figure (16): SHAP Summary Plot for BARLEY

Let us explain Figure(16):

Feature B04 seems to have the highest impact on the model's output, with a mix of positive and negative SHAP values. However, a greater density of points on the positive side suggests that higher values of B04 are more likely to increase the model's predicted value.

The spread of SHAP values for B12 is relatively narrow, indicating that while it may have some impact on model predictions, its effect is more uniform across different data points.

For features like B04, B11, B01, and B05, we see that higher feature values (indicated by pink dots) tend to push the SHAP value to the positive side, suggesting these features positively influence the prediction when they have high values.

The blue dots for these same features indicate that lower values tend to have a negative impact on the model's prediction, pushing the SHAP values to the left of the plot.

Some features, such as B03 and B8A, show a high variability in their impact on the model's output, as evidenced by the widespread of SHAP values. This suggests that the effect of these features on the prediction can vary significantly from one instance to another.

Features like B09, B06, and B07 seem to have a cluster of points close to the zero center line and thus may have a less significant impact on the model's prediction.

For the next nine plots, we will not write an interpretation. They are interpreted in the same way as the previous two.

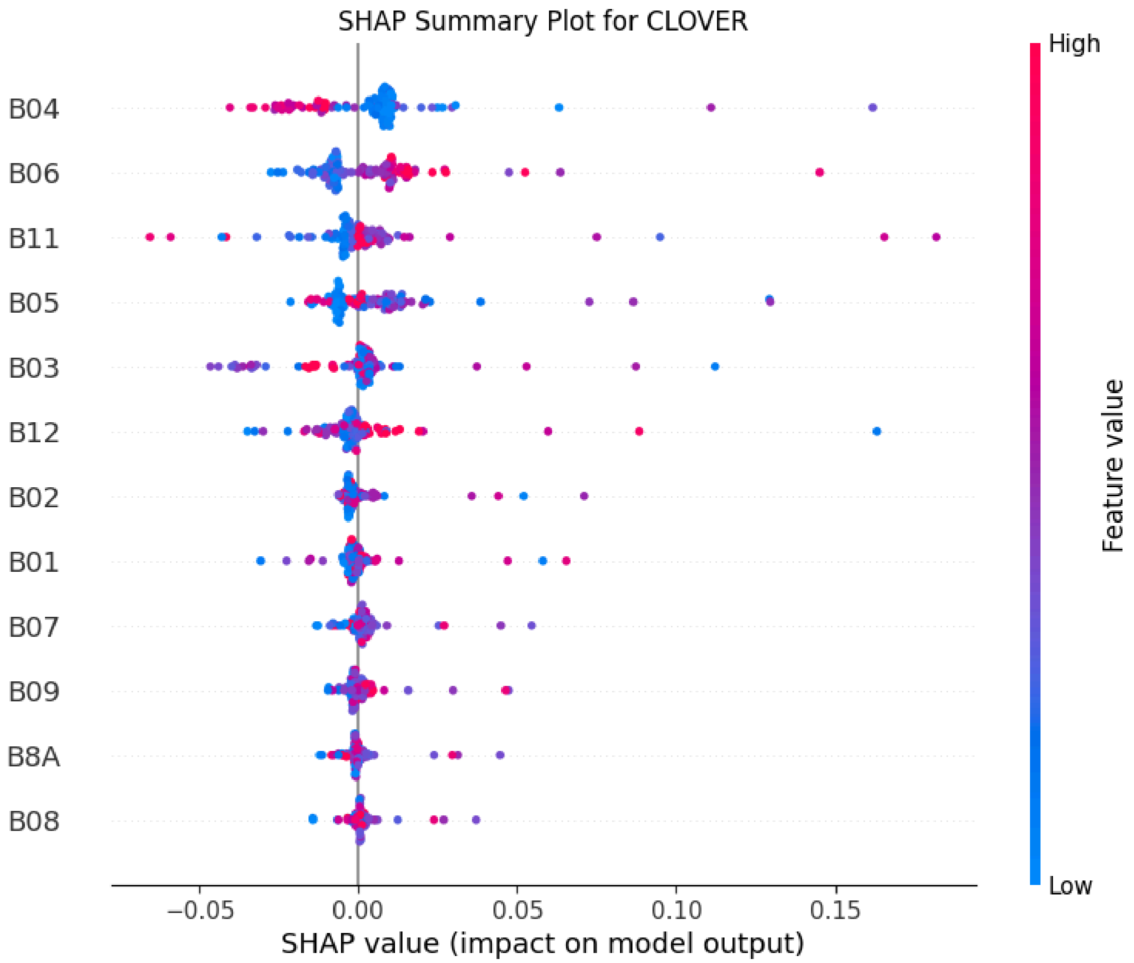


Figure (17): SHAP Summary Plot for CLOVER

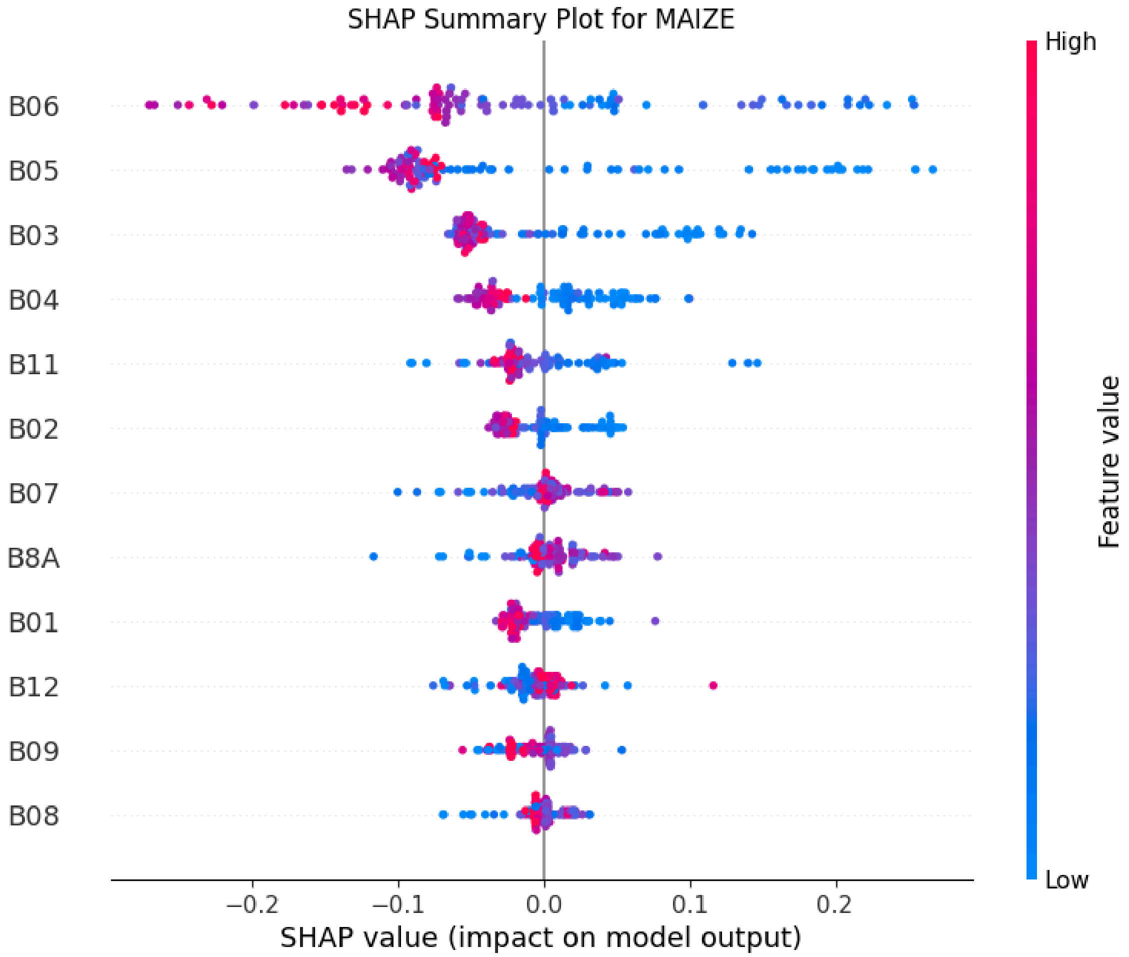


Figure (18): SHAP Summary Plot for MAIZE

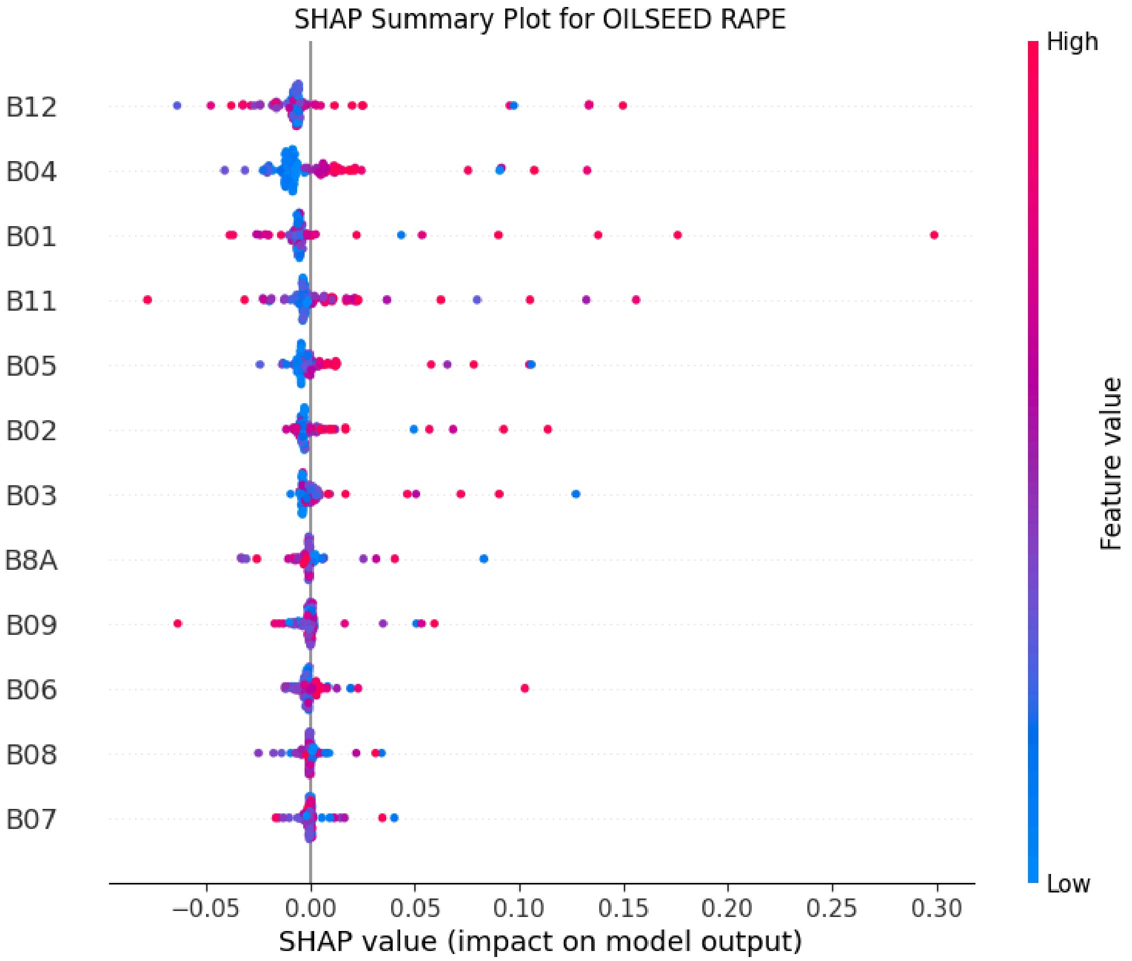


Figure (19): SHAP Summary Plot for OILSEED RAPE

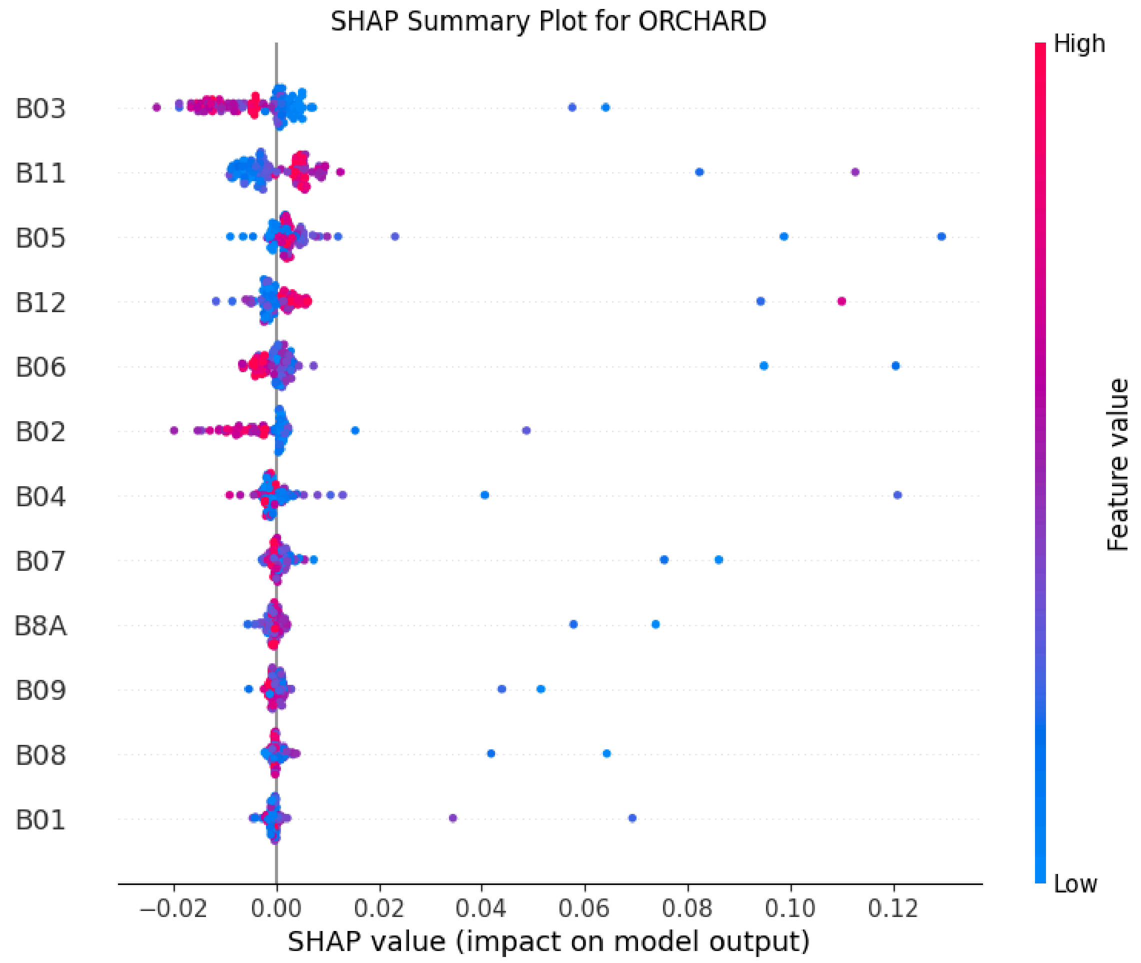


Figure (20): SHAP Summary Plot for ORCHARD

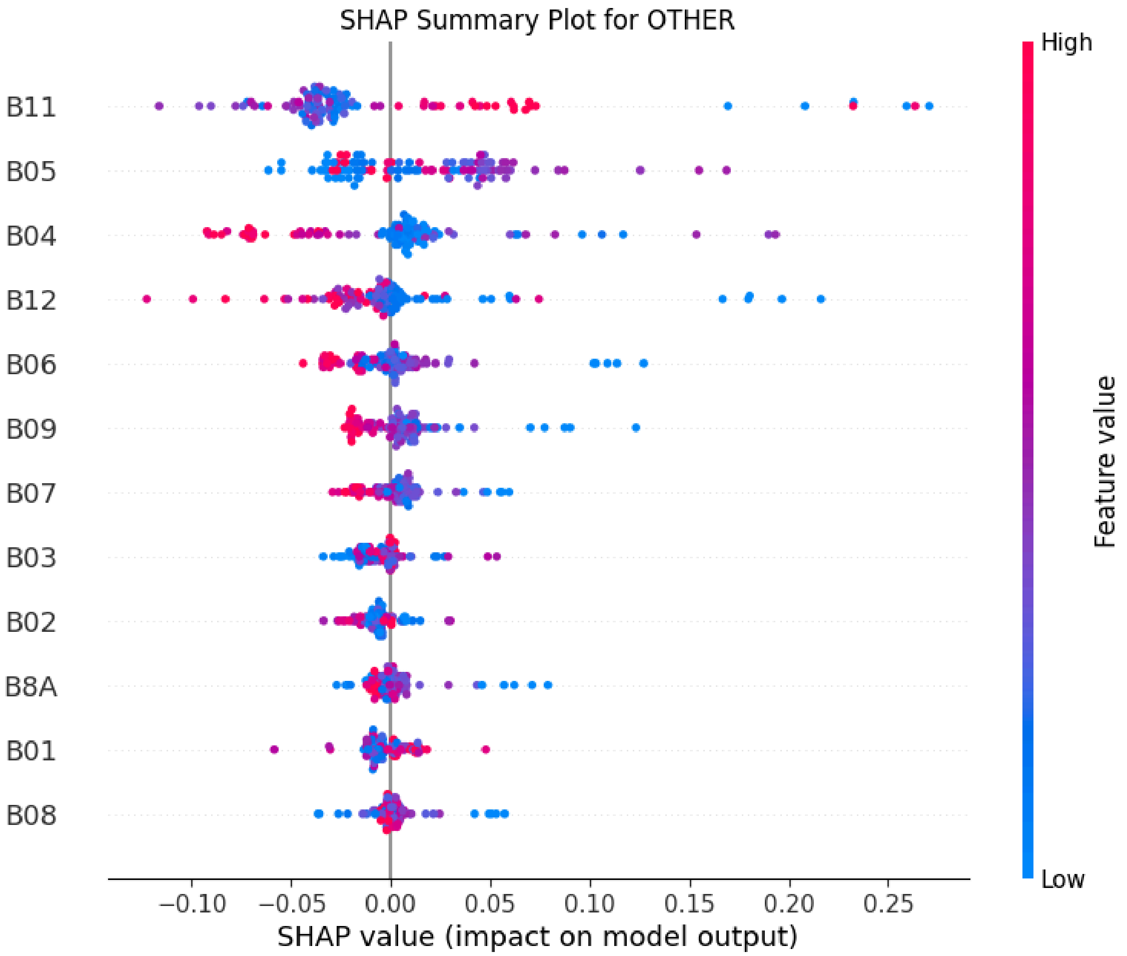


Figure (21): SHAP Summary Plot for OTHER

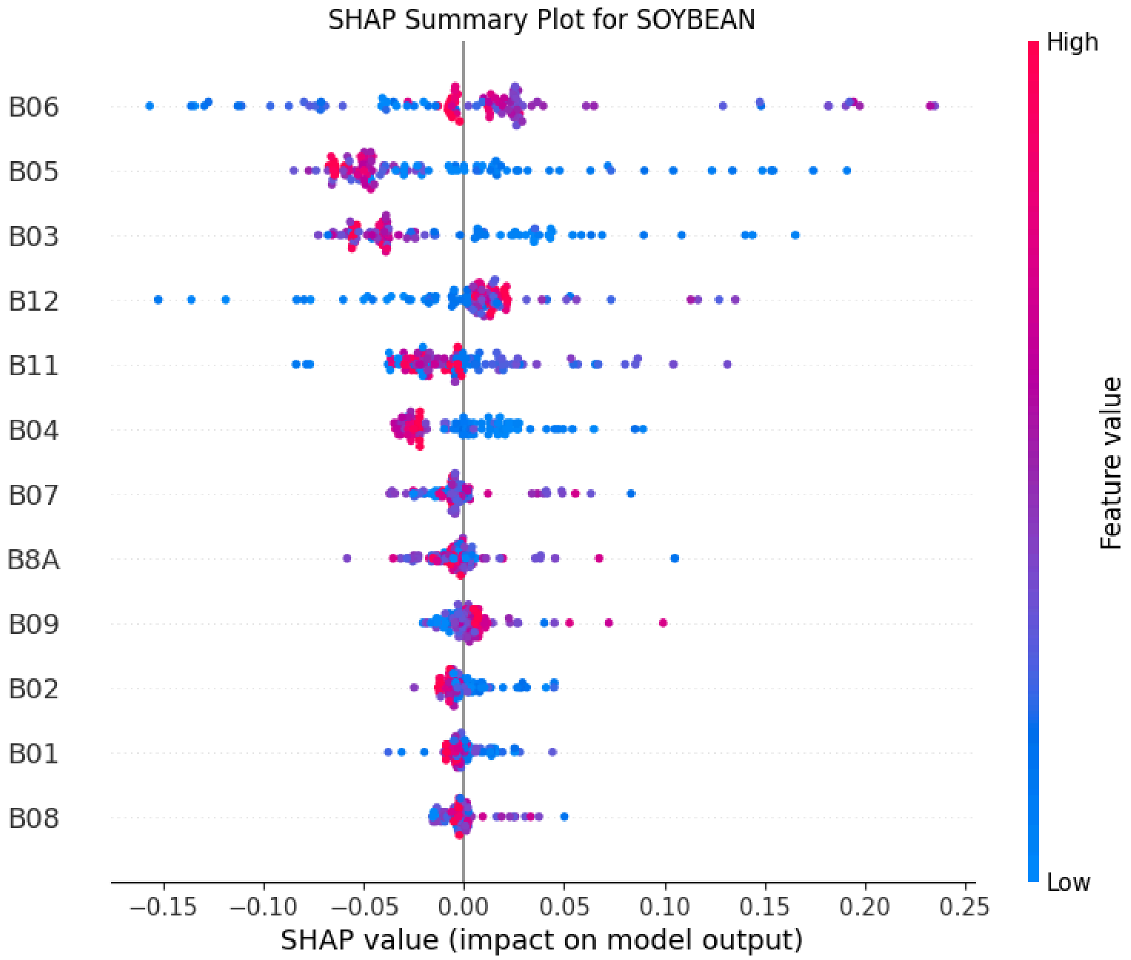


Figure (22): SHAP Summary Plot for SOYBEAN

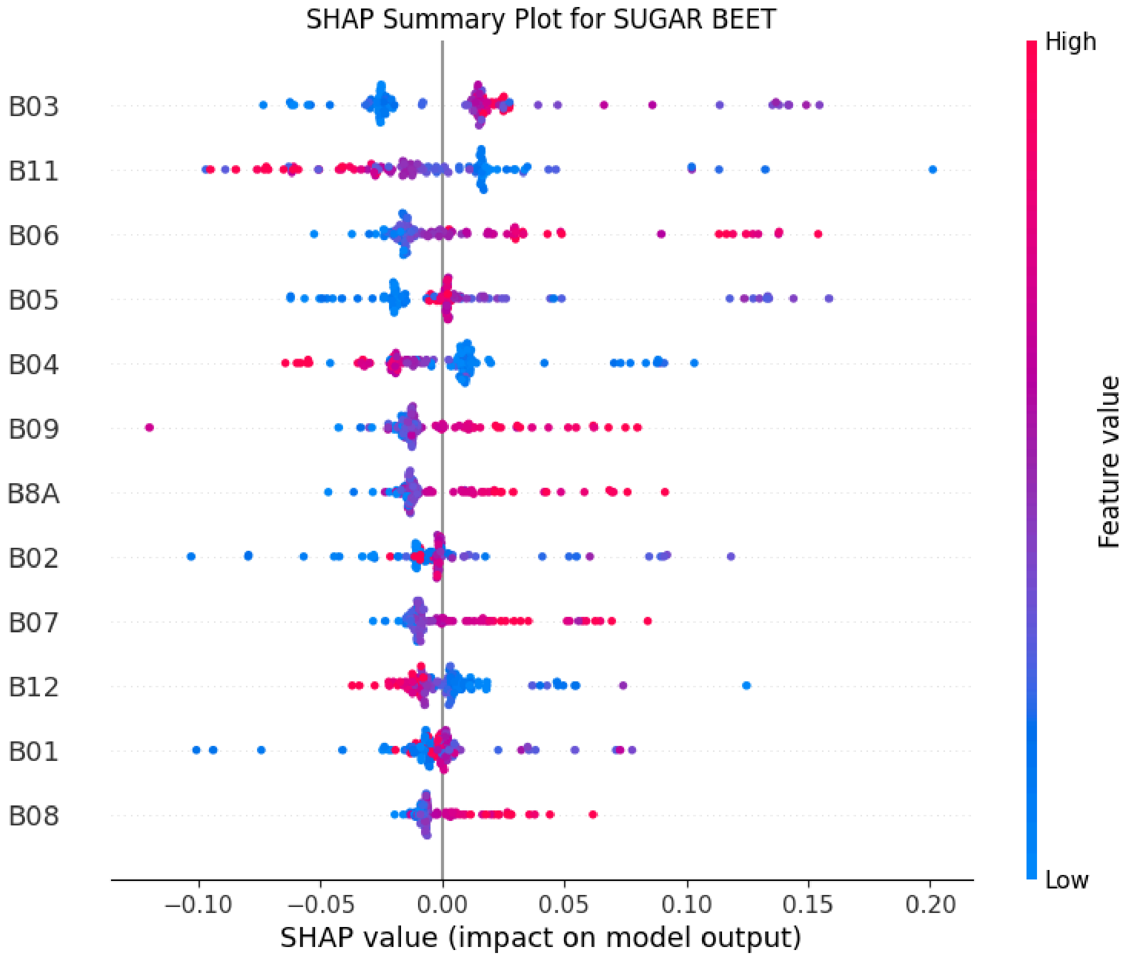


Figure (23): SHAP Summary Plot for SUGAR BEET

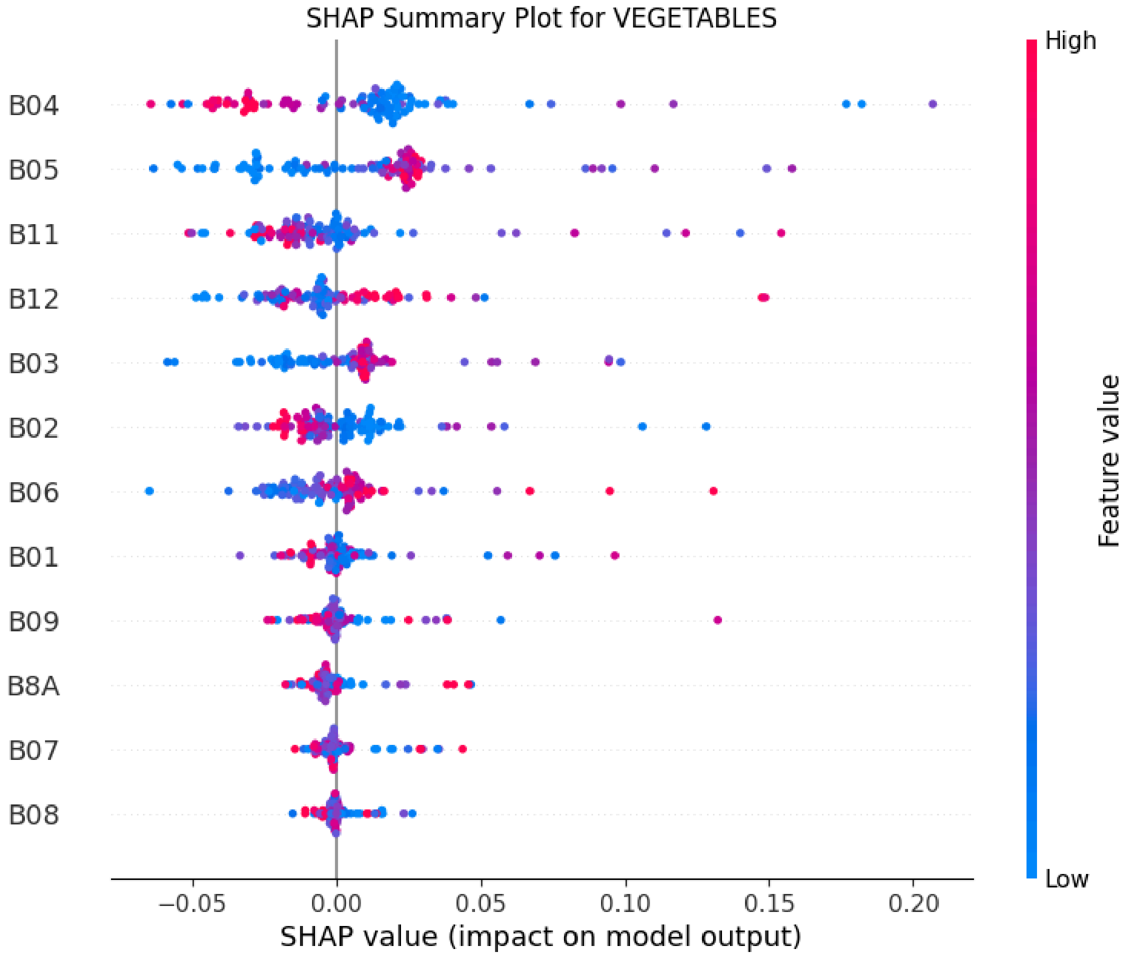


Figure (24): SHAP Summary Plot for VEGETABLES

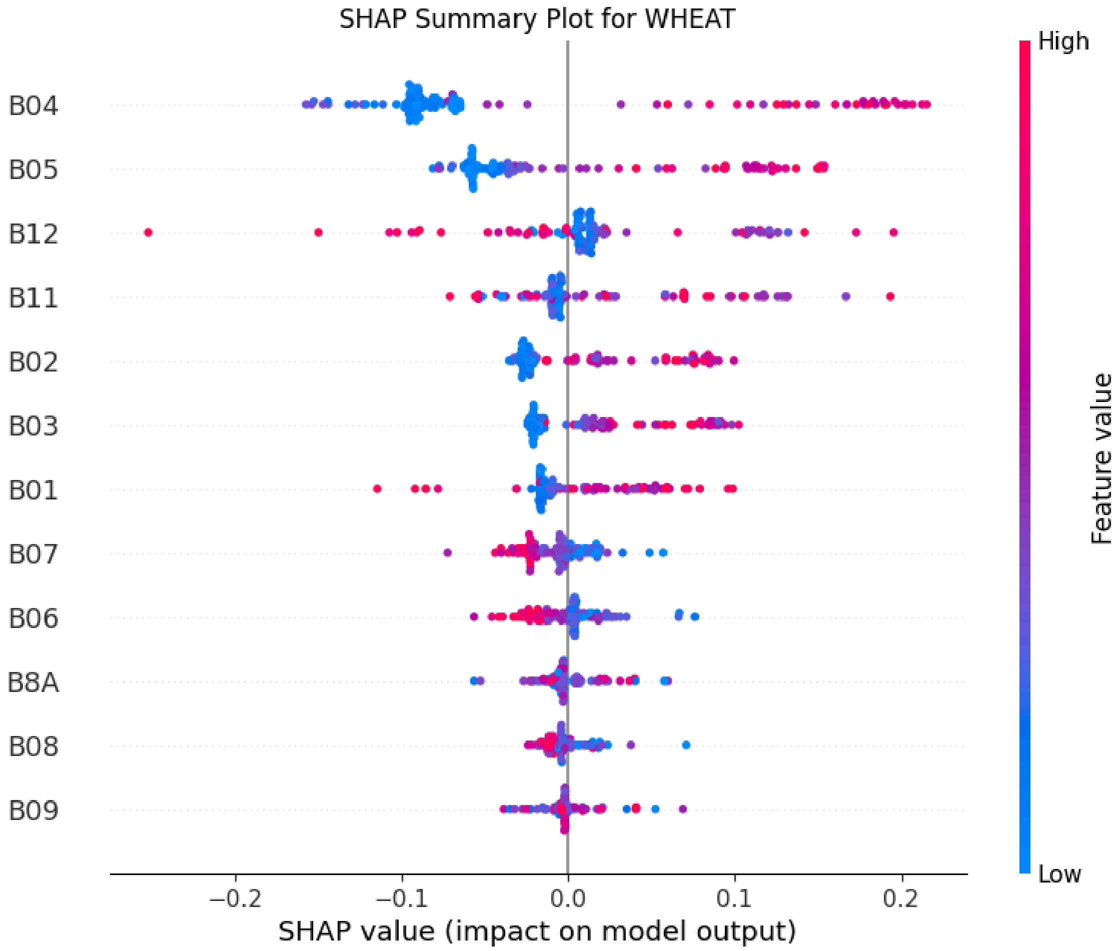


Figure (25): SHAP Summary Plot for WHEAT

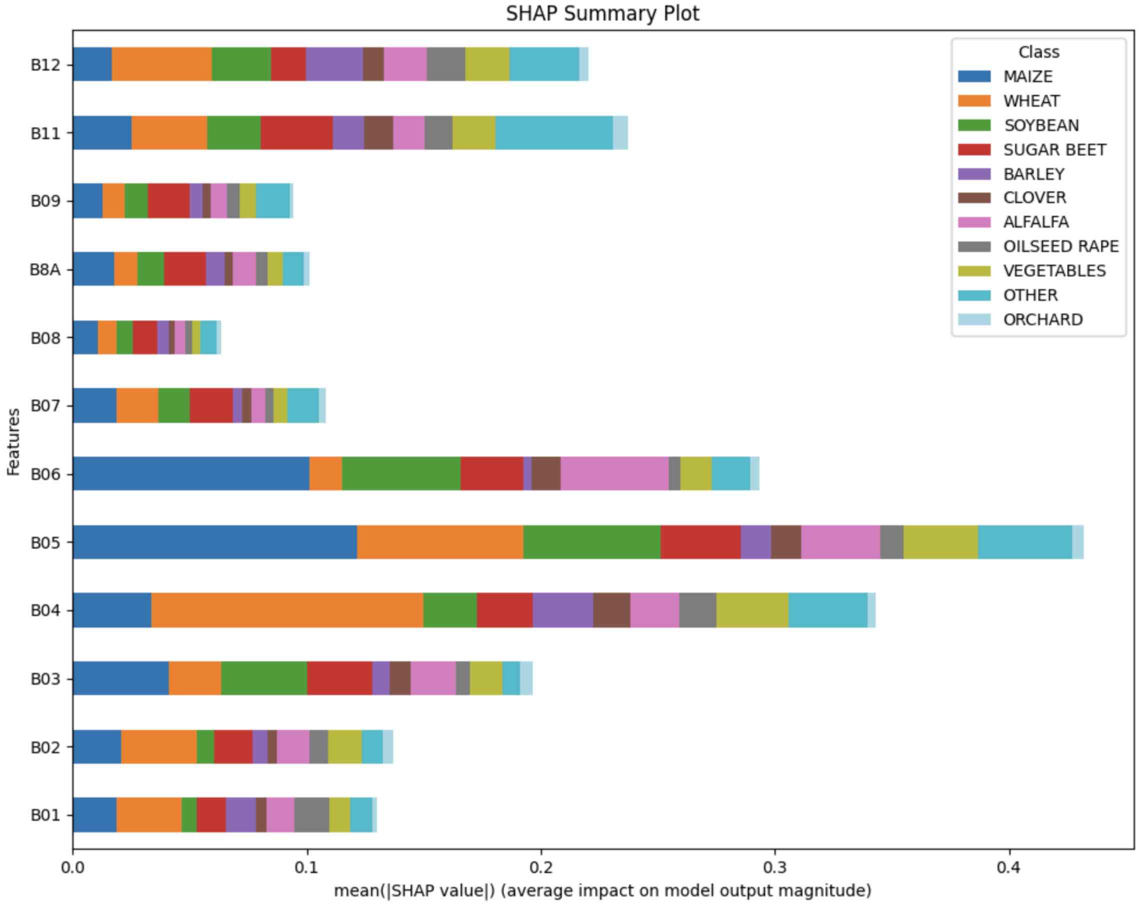


Figure (26): SHAP average impact on model output

Figure (26) displays the average impact of each feature on the model's output across multiple classes.

Figure (26) that features B05 and B04 have the most significant impact across all crop classes. B05 has the highest influence for the "Maize" class, while B04 is a significant predictor for the "Wheat" class. Feature B06 is observed to have a strong effect on the "Maize", “Soybean”, and “Alfalfa” classes. The features B11 and B12 are especially important for the "Other" category. The remaining features (B01, B02, B8A, B07, B09, B08) display varying levels of influence across the different classes, with none being as dominant as B05 and B04. This suggests a nuanced relationship between features and crop classes, where some features are universally important across classes, while others exhibit more specificity in their influence on model output.

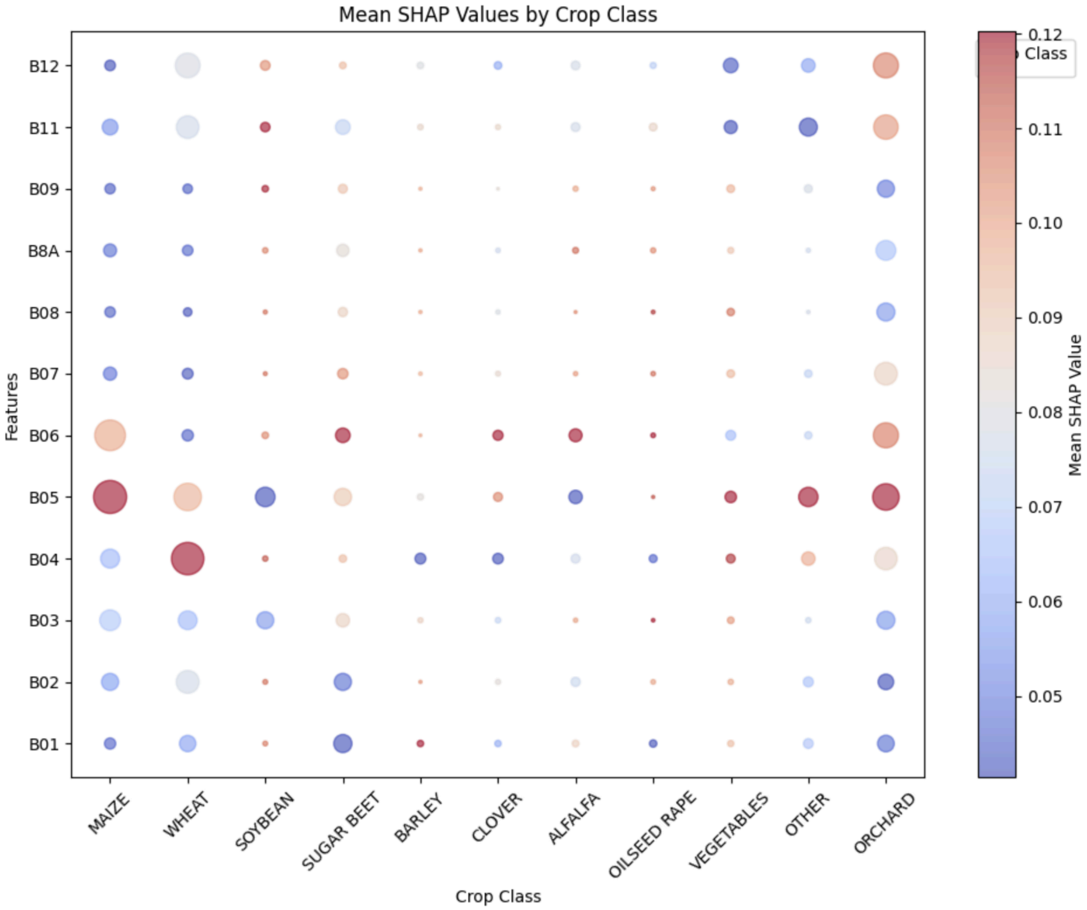


Figure (27): Mean SHAP values by Crop Class

The SHAP summary plot provides insights into the impact of different features on the predictive model's output across various crop classes. Each dot represents the mean absolute SHAP value of a feature for a specific class, which quantifies the feature's contribution to the model's prediction.

Larger dots indicate a greater average impact of a particular feature on the model's output for that class, meaning the feature is more influential for predicting that crop class. The color intensity reflects the magnitude of the SHAP value: lighter colors indicate lower impact, while darker colors signify higher impact.

From the Figure 27, it can be observed that:

Certain features have a more significant impact across multiple crop classes, as indicated by larger dots spread across the classes.

Some features have a strong influence on specific crop classes but not others, which could be seen from large dots concentrated under particular crop types.

More or less, B4, B5 and B6 show all classes. Maize, wheat and orchard are the most visible over all features.

The plot is useful for understanding which features the model relies on when making predictions for different crop types. For instance, features with consistently high SHAP values across many crop classes might be considered more globally important, whereas features with high SHAP values for specific classes could be indicative of specialized characteristics that distinguish those classes.

The interpretability of the Random Forest model was enhanced by applying SHAP, which provided valuable insights into the features that most significantly influenced the model's predictions. SHAP values revealed that certain spectral bands were particularly influential in the classification of specific crops.

For instance, features such as B04 and B05, which capture the red-edge and near-infrared spectral regions, respectively, were found to be critical in distinguishing between crop types. The application of SHAP not only validated the Random Forest model's high performance but also uncovered opportunities for further refinement.

By understanding the model's decision-making process, we were able to address potential overfitting issues, ensuring that the model generalized well to unseen data. This was particularly important in the context of satellite imagery, where varying conditions such as cloud cover, shadow, and changes in lighting conditions can significantly impact model performance.

8. Conclusion

The primary objective of this research was to compare various machine learning and deep learning models for crop classification and to leverage explainable AI techniques to interpret these results. Various sets of data, based on the date time, were used in training the model. The use of MinMax and Standard scalers techniques also showed that it can affect the model's performance. Attributes from 'soils.csv' brought specific insights into the model; however, their influence fluctuated markedly from month to month, leading to overfitting. To mitigate this, the decision was taken to narrow the focus to spectral bands, ensuring model stability and reducing overfitting risks. The class distribution within the dataset presented another hurdle; Class 0 was the most prevalent, while Class 13 was the least, which affected the model's learning process. This imbalance was addressed by employing the Synthetic Minority Over-sampling Technique (SMOTE), which equalized the representation of classes and provided a more balanced training ground for the model. The exploration of various machine learning models in this study has revealed promising results for the application of satellite imagery in crop classification. The models employed, including Naive Bayes, K-Nearest Neighbors, Random Forest, Convolutional Neural Networks, and Multi-Layer Perceptron, all provided solid foundations for addressing the classification problem at hand. The Random Forest, in particular, stood out with its high Macro F1 score, indicating its robustness and reliability. Each model brought a unique perspective to the table, with their strengths and weaknesses contributing to a comprehensive understanding of the problem. The application of SHAP provided a huge benefit. It afforded transparency in the model's decision-making, thereby building trust in its predictions, and it offered a way to model optimization, improving the model's predictive accuracy. The analysis reveals that the spectral bands B5 (Red Edge 1) and B4 (Red) emerge as the most influential across all crop classes, with B5 having the impact on the "Maize" class and B4 being predictive for the "Wheat" class. Additionally, B6 (Red Edge 2) exerts an effect on "Maize", "Soybean", and "Alfalfa" classes, demonstrating its significance in crop classification. The prominence of B4 and B5 can be attributed to their capability to capture critical information about vegetation health and chlorophyll content, which are vital for distinguishing between different crop types. Looking ahead, future work will be dedicated to the Convolutional Neural Network (CNN) model, investigating the power of interpretability through the Integrated Gradients method. This approach is expected to unveil the layers of CNN's decision-making, offering even deeper insights into the complex patterns learned from satellite imagery and driving the frontiers of agriculture forward.

Bibliography

[1] Copernicus: Europe's eyes on Earth. Available online: https://www.copernicus.eu/sites/default/files/2023-07/Copernicus%20General%20EN.pdf

[2] Sentinel-2 infographics. Available online: https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-2

[3] Sentinel-2 launched. Available online: https://www.copernicus.eu/sites/default/files/2018-10/Copernicus_Sentinel-2_Factsheet_13042018.pdf

[4] Drusch M., Del Bello U., Carlier S., Colin O., Fernandez V., Gascon F., Meygret A. *Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services.* Remote Sensing of Environment, 2012.

[5] Samek W., Wiegand T., Müller K. R. *Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models*, 2017.

[6] Arrieta A. B., Díaz-Rodríguez N., Del Ser J., Bennetot A., Tabik S., Barbado A., Herrera F. *Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI.* Information Fusion, 2020.

[7] Chavan J., Pawade N., Tale A., Kadam A., Gujar A. *Crop Yield Prediction Using Naive Bayes Algorithm.* IJSRCSEIT, 2022.

[8] V . Latha Jothi, Neelambigai A., Nithish Sabari S., Santhosh K. *Crop Yield Prediction Using KNN Model,* 2020.

[9] Khuong H. Tran, Hankui K. Zhang, John T. McMaine, Xiaoyang Zhang, Dong Luo. *10 m crop type mapping using Sentinel-2 reflectance and 30 m cropland data layer product*, 2022.

[10] Lundberg S., Lee S.-I. *A Unified Approach to Interpreting Model Predictions*. Proceedings of NeurIPS, 2017.

[11] Abhirup D., Biswajeet P.. *Interpretable and explainable AI (XAI) model for spatial drought prediction*, 2021.

[12] Kim J., Park H. *Deep CNNs in Satellite Imagery: Unraveling the Magic.* Journal of Satellite Image Analysis, 2016.

[13] Spectral bands illustration. Available online: https://esamultimedia.esa.int/docs/EarthObservation/Sentinel-2_ESA_Bulletin161.pdf

[14] Copernicus Open Access Hub, Available online: https://dataspace.copernicus.eu/explore-data

[15] Siddharth M., Hao L., Jiabo H. *Machine Learning for Subsurface Characterization,* 2020.

[16] Naive Bayes Classifier. Available online: https://mlarchive.com/machine-learning/the-ultimate-guide-to-naive-bayes/

[17] Witten I. H., Frank E., Hall M. A., Pal C. J. *Data Mining: Practical Machine Learning Tools and Techniques*, 2016.

[18] Cover T., Hart P. *Nearest neighbor pattern classification*. IEEE Transactions on Information Theory, 1967.

[19] KNN illustration. Available online: https://medium.com/@sachinsoni600517/k-nearest-neighbours-introduction-to-machine-learning-algorithms-9dbc9d9fb3b2

[20] Aha D. W., Kibler D., Albert, M. K. *Instance-based learning algorithms*, 1991.

[21] Duda R. O., Hart P. E., Stork D. G. *Pattern Classification (2nd ed.*), 2001.

[22] Larose D. T. *Discovering Knowledge in Data: An Introduction to Data Mining*, 2005.

[23] Breiman L. *Random Forests*. Machine Learning, 2001.

[24] Simplified Schematization of Random Trees. Available online: https://catalyst.earth/catalyst-system-files/help/concepts/focus_c/oa_classif_intro_rt.html

[25] Breiman L. *Bagging predictors*. Machine Learning, 1996.

[26] Ho T. K. *The Random Subspace Method for Constructing Decision Forests*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998.

[27] Chen C., Liaw A., Breiman L. *Using Random Forest to Learn Imbalanced Data*. University of California, Berkeley, 2004.

[28] Louppe G., Wehenkel L., Sutera A., Geurts P. *Understanding variable importances in forests of randomized trees*. In Advances in neural information processing systems, 2013.

[29] CNN architecture. Available online: . https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-lea

rning-b3921bdd82d5

[30] Srivastava N., Hinton G. E., Krizhevsky A., Sutskever I., Salakhutdinov R. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. Journal of Machine Learning Research, 2014.

[31] Kaggle: https://www.kaggle.com/competitions/crop-classification/submissions

[32] James G., Witten D., Hastie T., Tibshirani, R. *An Introduction to Statistical Learning*. Springer, 2013.

[33] k-fold Cross Validation. Available online: https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85

[34] Varma S., Simon R. *Bias in error estimation when using cross-validation for model selection*. BMC Bioinformatics, 2006.

[35] Holzinger A. , Saranti A., Molnar C., Biecek P., Samek W. *Explainable AI Methods - A Brief Overview,* 2022.

[36] 'Shap' library. Available online: https://shap.readthedocs.io/en/latest/

[37] Lundberg S. M., Erion G., Lee, S. I. *Consistent Individualized Feature Attribution for Tree Ensembles*, 2018.

[38] Datta A., Sen S., Zick Y. *Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems*. In IEEE Symposium on Security and Privacy, 2016.

Biography

Milica Panić was born on 4th of December 1994 in Zrenjanin, Serbia. She attended elementary school "Vuk Karadžić" and grammar school "Zrenjaninska gimnazija" in Zrenjanin. Milica's passion for Mathematics led her to pursue a Bachelor of Science in Applied Mathematics at the Faculty of Sciences, University of Novi Sad, which she completed in 2020. Continuing her academic journey in the same year, she enrolled in a Master's program in Data Science at the same faculty. Milica began her career as a Researcher in June 2022, working on the "Horizont2020 project CYRENE" under the mentorship of Dušan Jakovetić. In January 2023, she also started the position of Junior Data Scientist at TX Services in Belgrade, where she applies her academic knowledge to real-world data challenges.

UNIVERZITET U NOVOM SADU ● Prirodno-matematički fakultet
21000 Novi Sad, Trg Dositeja Obradovića 3

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj, RBR:	
Identifikacioni broj, IBR:	
Tip dokumentacije, TD:	Monografska dokumentacija
Tip zapisa, TZ:	Tekstualni štampani materijal
Vrsta rada, VR:	Master rad
Autor, AU:	Milica Panić
Mentor, MN:	prof. Dušan Jakovetić
Naslov rada, NR:	Objašnjiva veštačka inteligencija - Klasifikacija useva
Jezik publikacije, JP:	Engleski
Jezik izvoda, JI:	Engleski
Zemlja publikovanja, ZP:	Republika Srbija
Uže geografsko područje, UGP:	Vojvodina
Godina, GO:	2024.
Izdavač, IZ:	Autorski reprint
Mesto i adresa, MA:	Novi Sad, Departman za matematiku i informatiku, PMF, Trg Dositeja Obradovića 2
Fizički opis rada, FO:	Poglavlja (8), strana (54), literaturnih citata (38), tabela (3), slika (27)
Naučna oblast, NO:	Matematika
Naučna disciplina, ND:	Primenjena matematika

Ključne reči, KR:	Python, Mašinsko učenje, Duboko učenje, Usev, Klasifikacija, Random Forest, KNN, Naivni Bajes, Konvolutivne neuronske mreže, SHAP, xAI
Univerzalna decimalna klasifikacija, UDK:	
Čuva se, ČU:	Biblioteka Departmana za matematiku i informatiku Prirodno-matematičkog fakulteta, u Novom Sadu
Važna napomena, VN:	
Izvod, IZ:	U ovoj tezi istražujemo različite modele mašinskog učenja i dubokog učenja za klasifikaciju useva. Primenjena su četiri algoritma i rezultati su upoređeni kroz različite metrike evaluacije. SHAP metod je korišćen za razumevanje outputa i za upoređivanje efekata spektralnih opsega na svaku klasu useva.
Datum prihvatanja teme od strane NN veća, DP:	05.02.2024.
Datum odbrane, DO:	
Članovi komisije, KO: Predsednik: Mentor: Član:	 dr Oskar Marko, naučni saradnik Instituta BioSense, Novi Sad Prof. Dr Dušan Jakovetić, vanredni profesor, PMF, Novi Sad dr Marko Panić, naučni saradnik Instituta BioSense, Novi Sad

UNIVERSITY OF NOVI SAD ● Faculty of Science 21000 Novi Sad,
Trg Dositeja Obradovića 3

KEY WORDS DOCUMENTATION

Accession number, ANO:	
Identification number, INO:	
Document type, DT:	Monograph type
Type of record, TR:	Textual material, printed
Contents code, CC:	Master thesis
Author, AU:	Milica Panić
Mentor, MN:	prof. Dušan Jakovetić
Title, TL:	Explainable Artificial Intelligence - Crop Classification
Language of text, LT:	English
Language of abstract, LA:	English
Country of publication, CP:	Republika Srbija
Locality of publication, LP:	Vojvodina
Publication year, PY:	2024.
Publisher, PU:	Author's reprint
Publ. place, PP:	Novi Sad, Faculty of Sciences, Department of Mathematics and Informatics, Trg Dositeja Obradovića 2
Physical description, PD:	Chapters (8), pages (54), references (38), tables (3), figures (27)

Scientific field, SF:	Mathematics
Scientific discipline, SD:	Applied Mathematics
Key words, KW:	Python, Machine learning, Deep learning, Crop, Classification, Random Forest, KNN, Naive Bayes, Convolutional neural networks, SHAP, xAI
Universal decimal classification, UDC:	
Holding data, HD:	Department of Mathematics and Informatics' Library, Faculty of Science, Novi Sad
Note, N:	
Abstract, AB:	In this thesis we research different machine learning and deep learning models for crop classification. Four algorithms were tested and the results were compared through different validation scores. SHAP was used to explain the output and to compare the effect of spectral bends on each crop class.
Accepted by the Scientific Board on, ASB:	05.02.2024.
Defended on, DE:	
Thesis defend board, DB: Chairperson: Mentor: Member:	dr Oskar Marko, scientific associate of BioSense Institute, Novi Sad Prof. Dr Dušan Jakovetić, professor at PMF, Novi Sad dr Marko Panić, scientific associate of BioSense Institute, Novi Sad