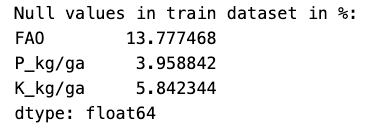
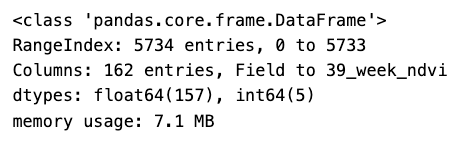
# **Analysis**

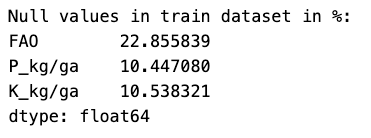
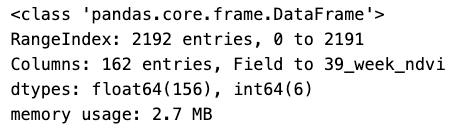
## **Data Loading and Exploration:**

* Read and loaded the training and test datasets.
* Explored the basic information of the datasets, including null values and duplications.

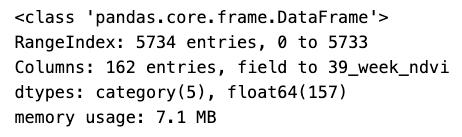
train\_data:



Test\_data:

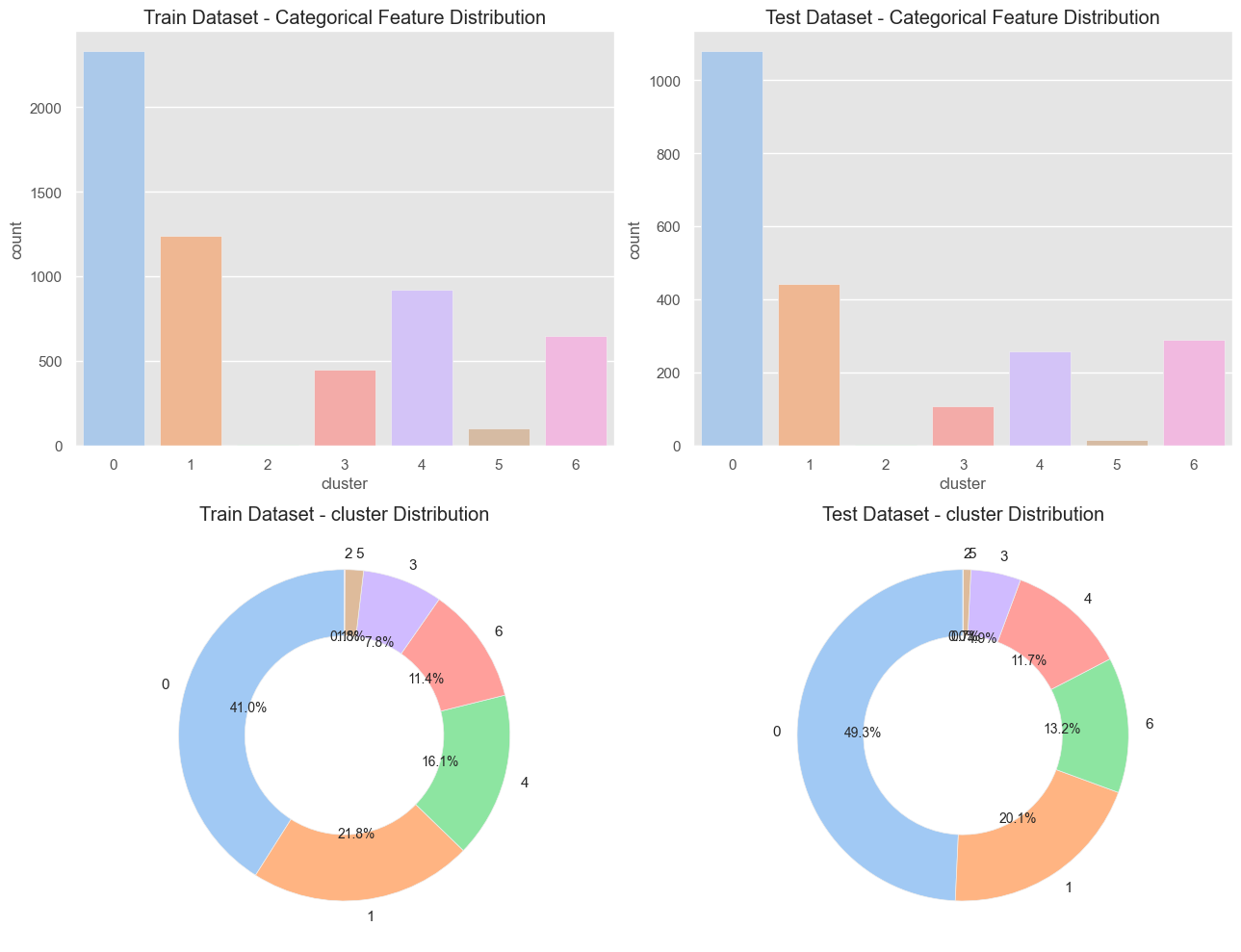


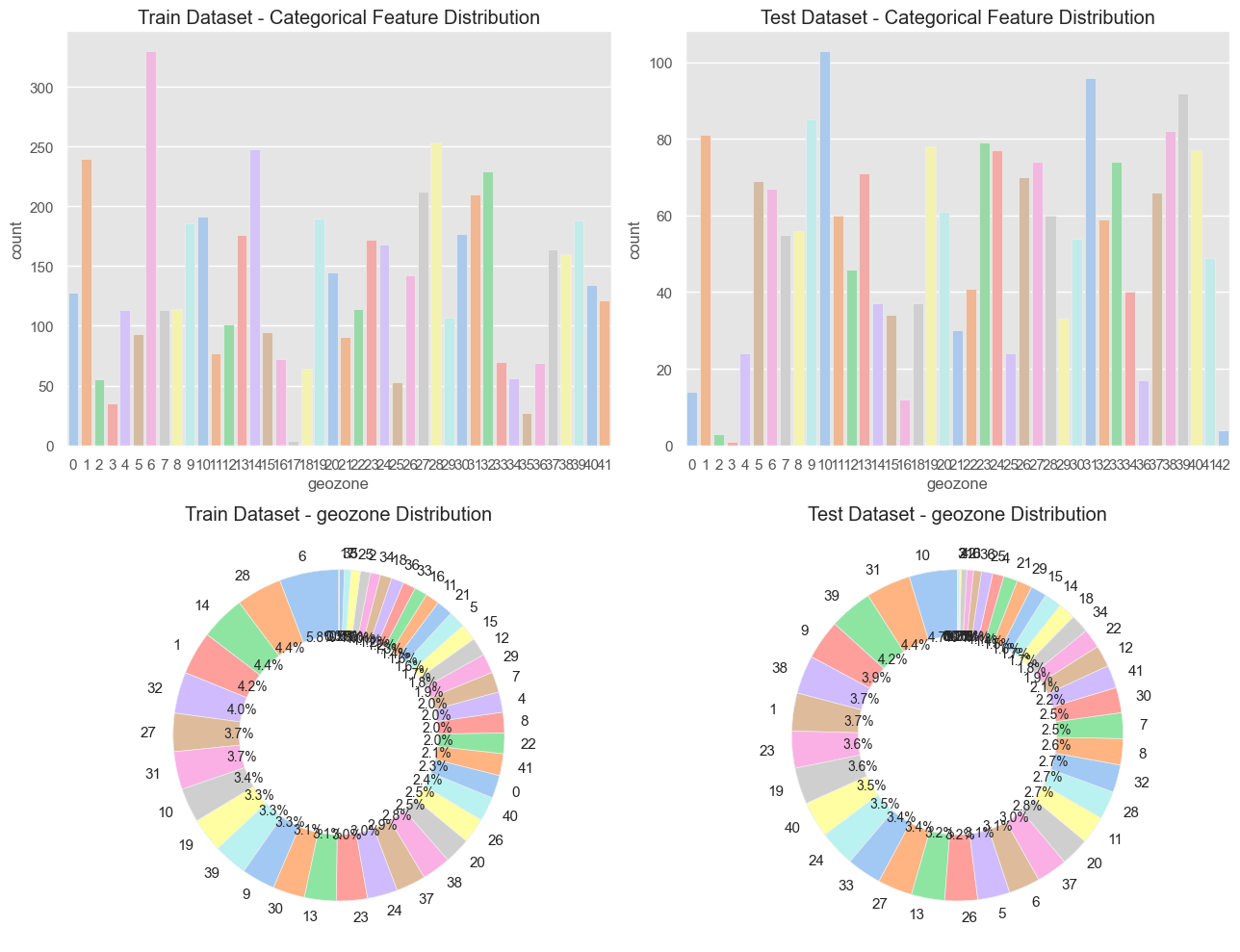
* Validated and cleaned column names, converting them to lowercase and removing special characters. (renamed fertilizers column names and deleted special characters from column names)
* Checked and converted data types of categorical columns.

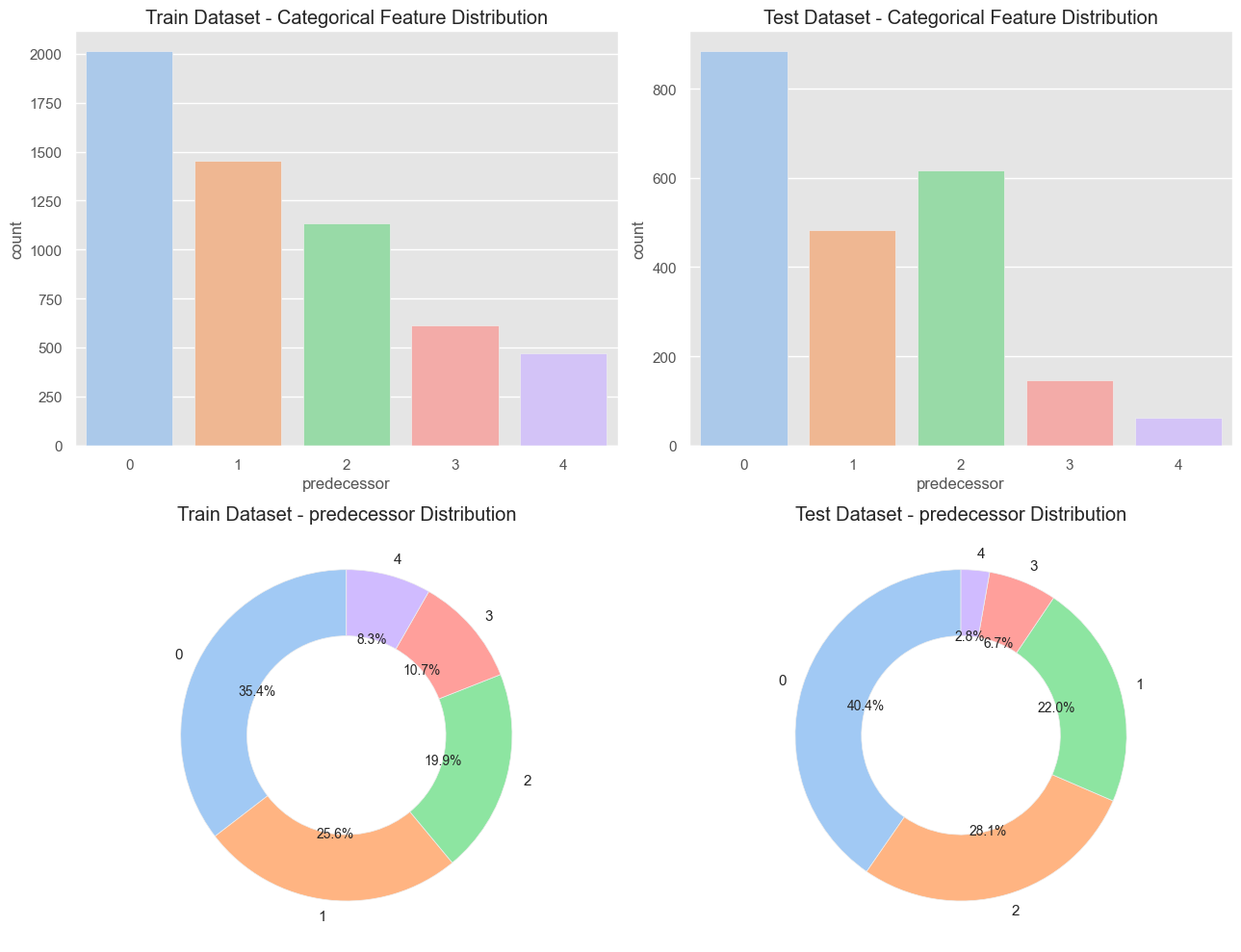


* Explored categorical features' distributions using count plots and pie charts.

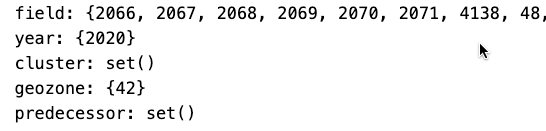








* Checked for unseen categories between train and test datasets.



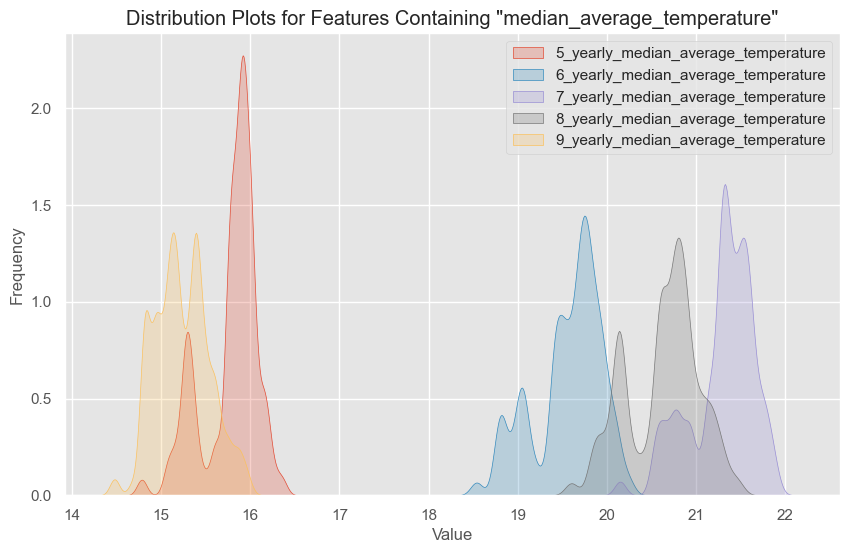


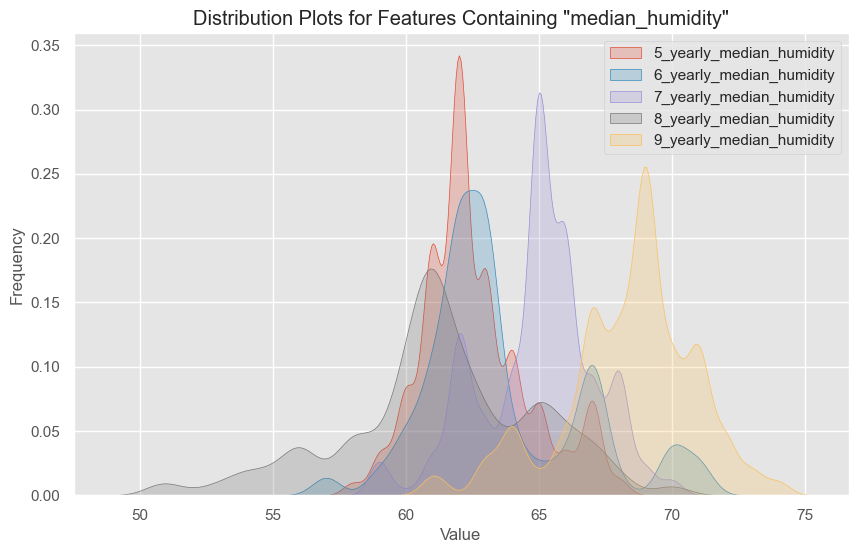
## **Outlier detection:**

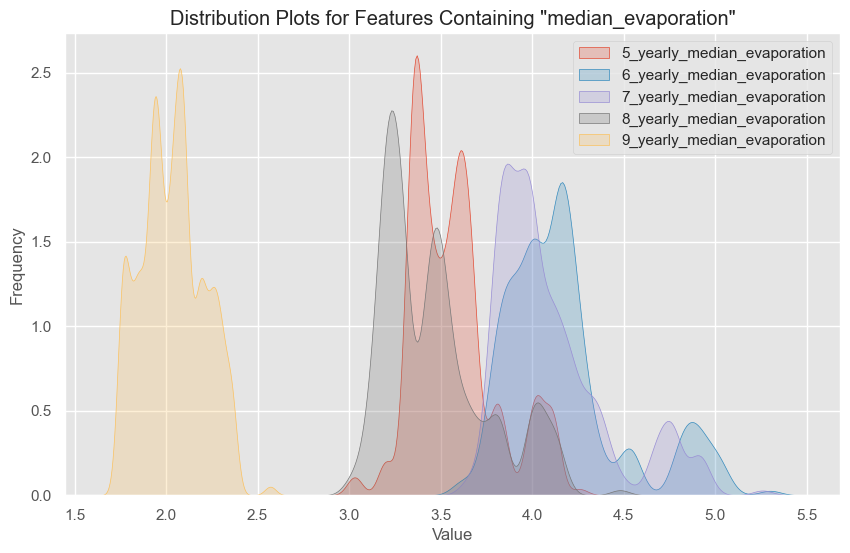
* When inspecting the maximum and minimum values for numeric data (.describe()), no outliers were found, so there was no point in building boxplots

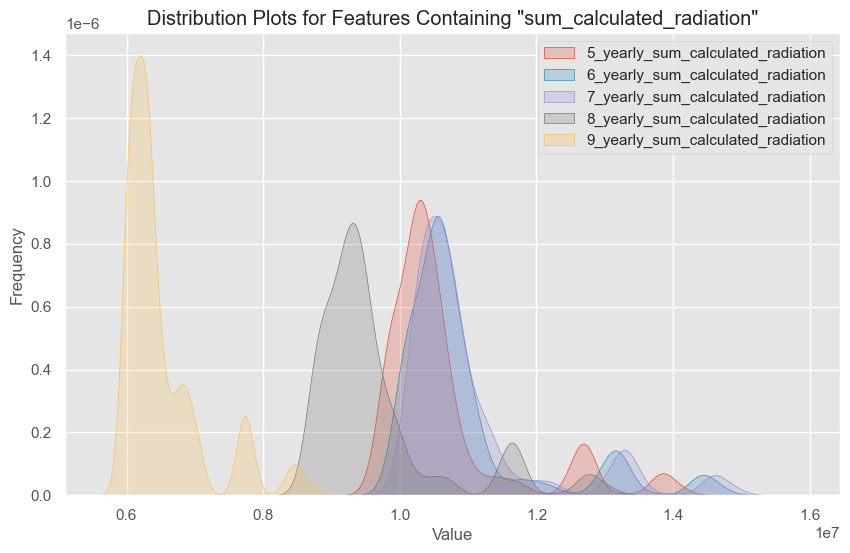
## **Numerical Features Analysis:**

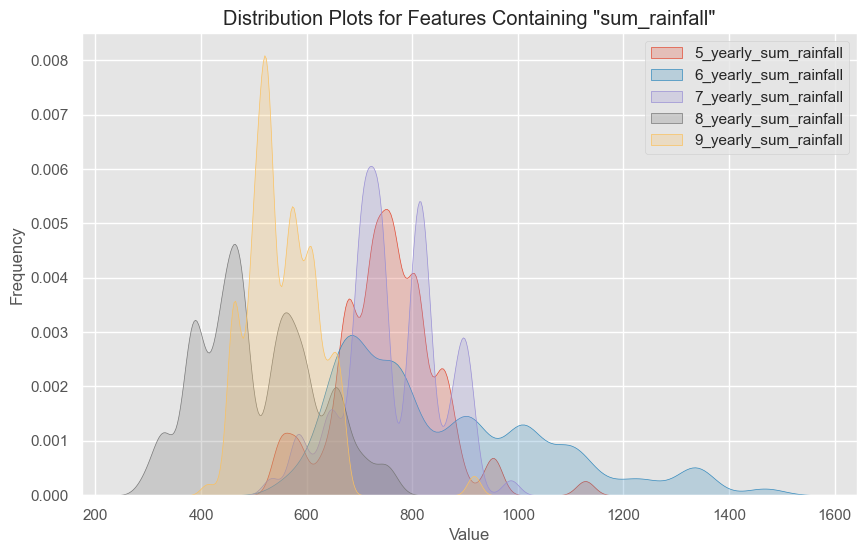
* Created combined distribution plots for various numerical features.







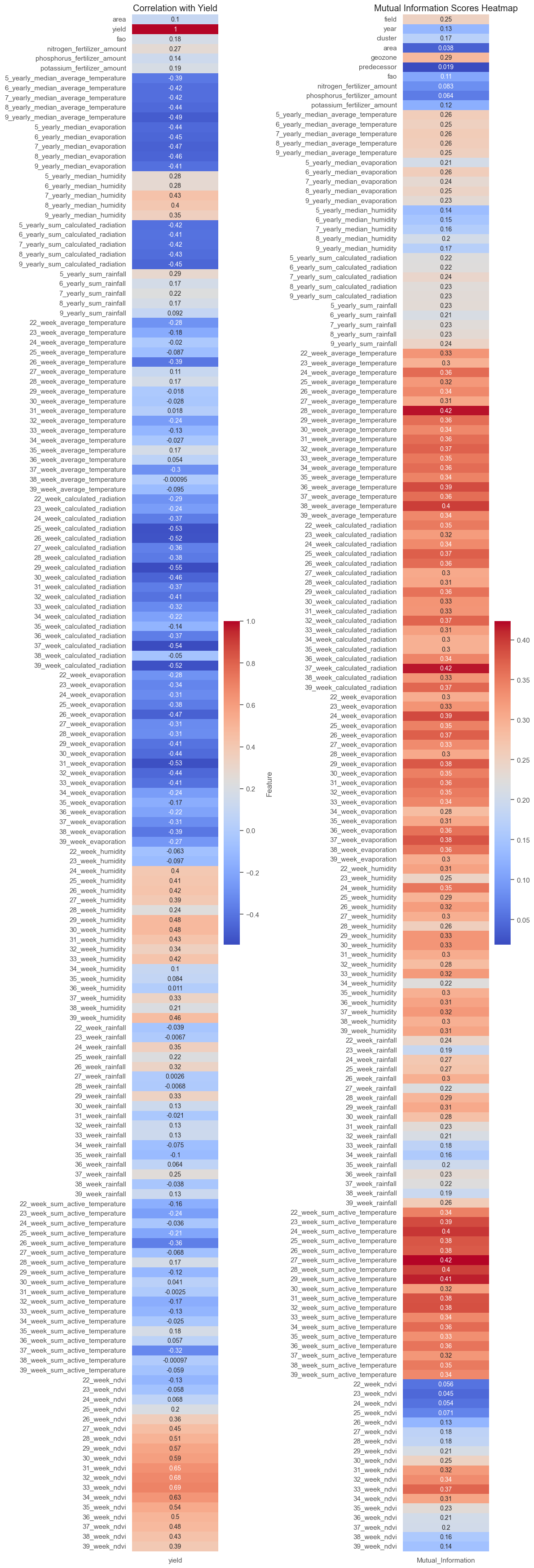




* Handled missing values using KNN imputation. The **KNN imputer** model for null values imputation was chosen as this model has proven itself well, based on my experience

## **Correlation Analysis:**

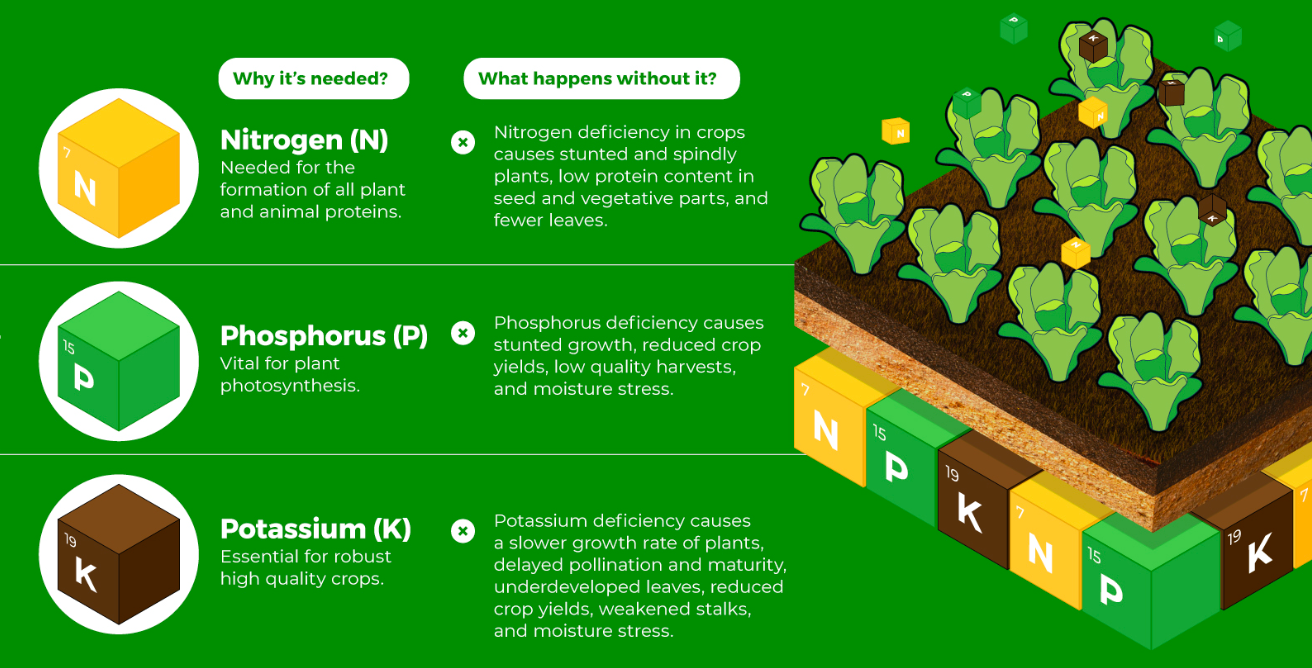
* Checked linear correlation and mutual information to understand relationships between variables and **yield** feature

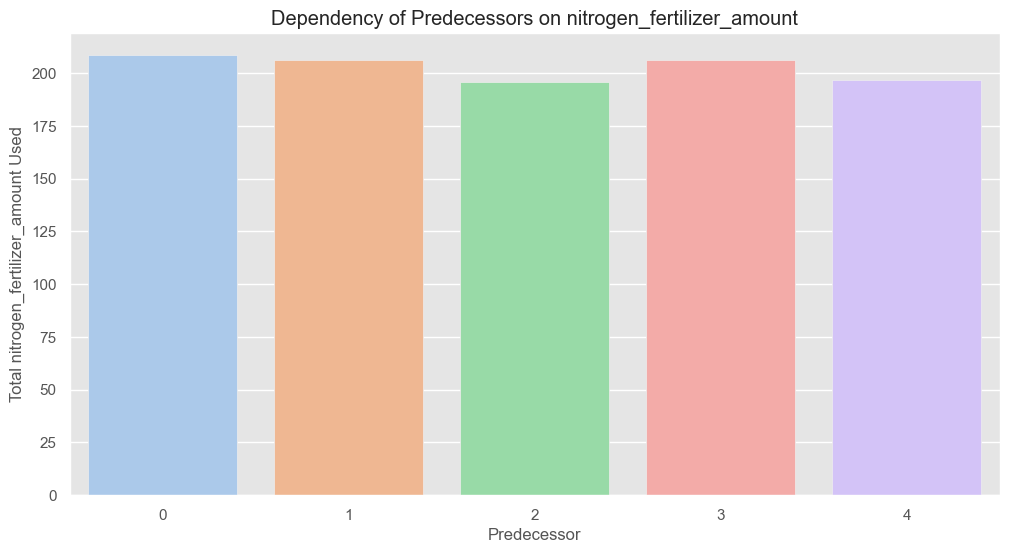


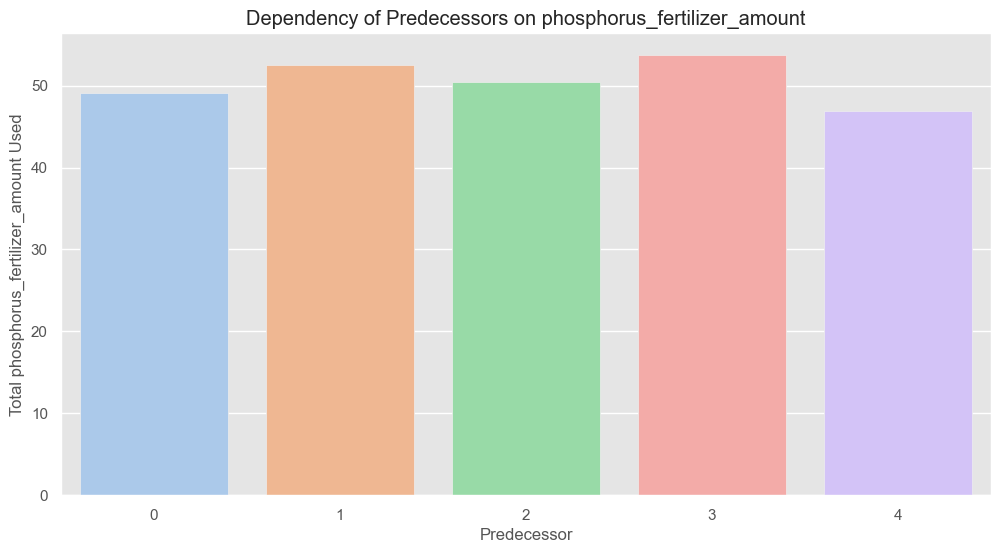
Also, a graph of mutual information score was plotted to find non-linear dependencies

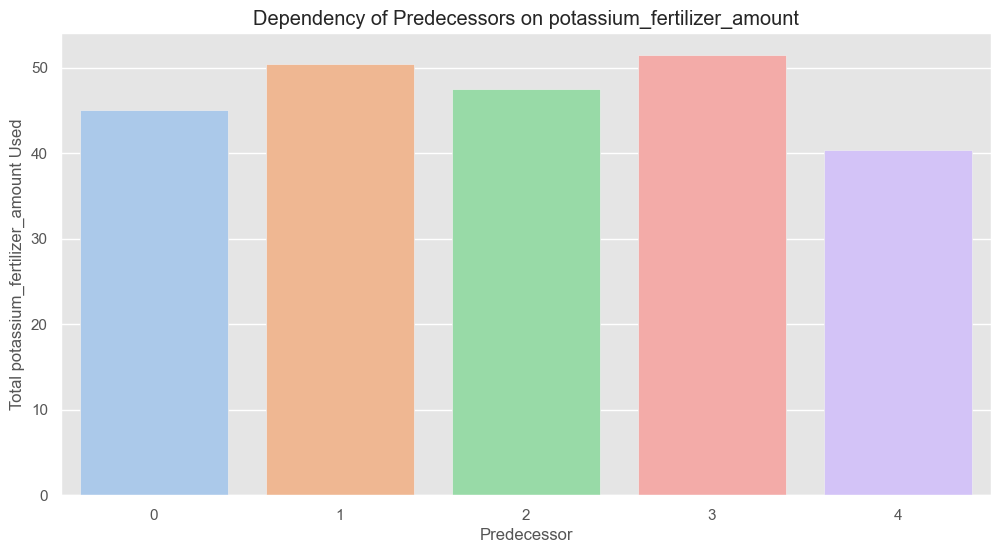
## **Predecessor Impact Analysis:**

* Explored the impact of predecessor on the usage of nitrogen, phosphorus, and potassium fertilizers.









There was no particularly clear correlation between the previous crop and the amount of fertilizer applied this year

## **Feature Selection:**

* Initially considered using mutual information scores for feature selection (optional)
* Selected specific features based on correlational analysis. Included categorical columns, metadata and ndvi values. This helped to reduce dataset size 6.2 times with little loss of model accuracy.

## **Time Series Compression (Optional):**

* Considered aggregating metrics over different periods for time series features (optional). Tried to compress the temporal data (for each metric aggregated 18 weeks into 6 periods of 3 weeks), which helped to reduce the size of the data by a factor of three, also with little loss in model accuracy

## **Principal Component Analysis (PCA) (Optional):**

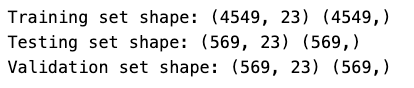
* Performed Principal Component Analysis on the data (optional). It gave some results, but PCA is really bad for explainability, because we are using decomposed vectors, not original ones.

## **Metric Creation (Optional):**

* Defined and created various regression metrics for model evaluation. Used MAE, MSE, RMSE, r2, MAPE, WMAPE

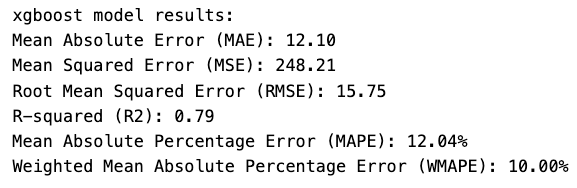
## **Train-Test Split:**

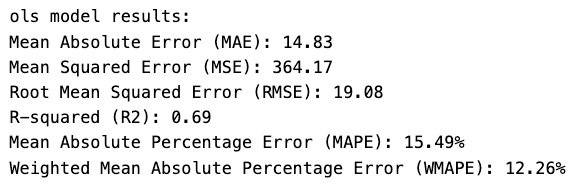
* Split the data into training, testing, and validation sets.

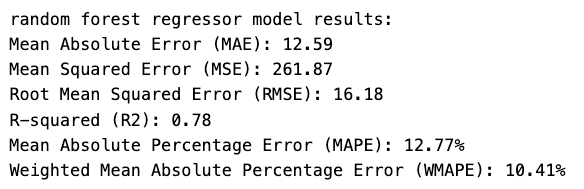
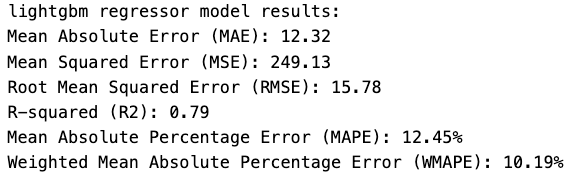


## **Modeling:**

* Used XGBoost, Ordinary Least Squares (OLS), Random Forest, and LightGBM for regression modeling (out of the box models without hypertuning)
* Evaluated models using regression metrics.

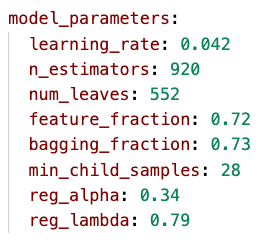




## **Hyperparameter Tuning (Optional):**

* Performed hyperparameter tuning for LightGBM using Optuna



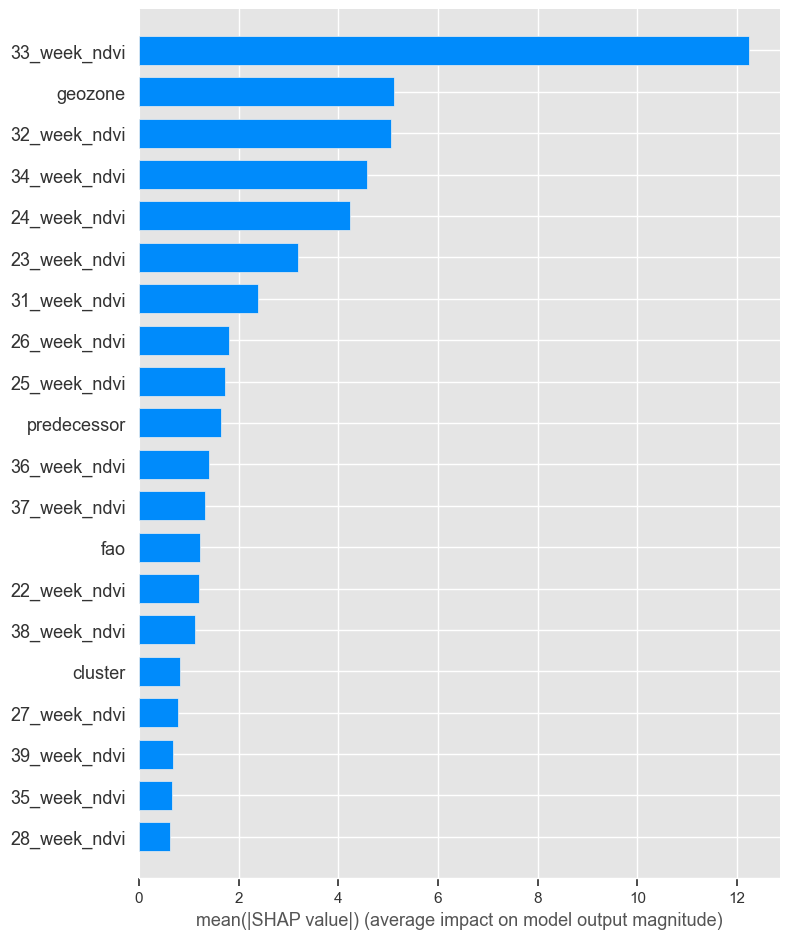
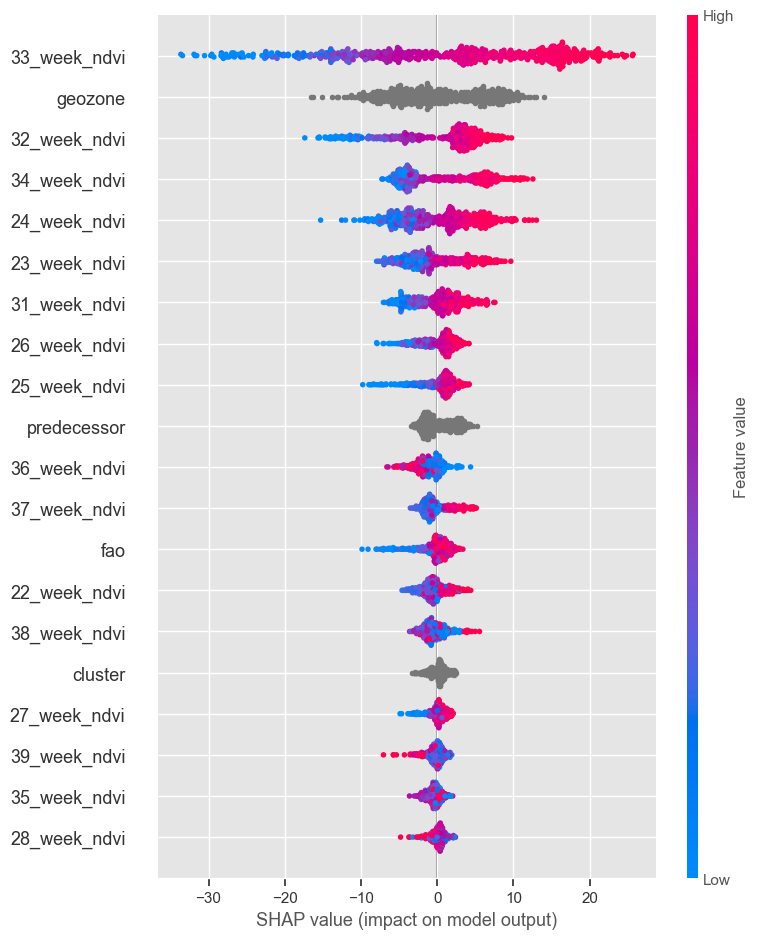
## **Final Model Evaluation:**

* Evaluated the final LightGBM model using regression metrics on the test set.

## 

## **SHAP (SHapley Additive exPlanations) Analysis:**

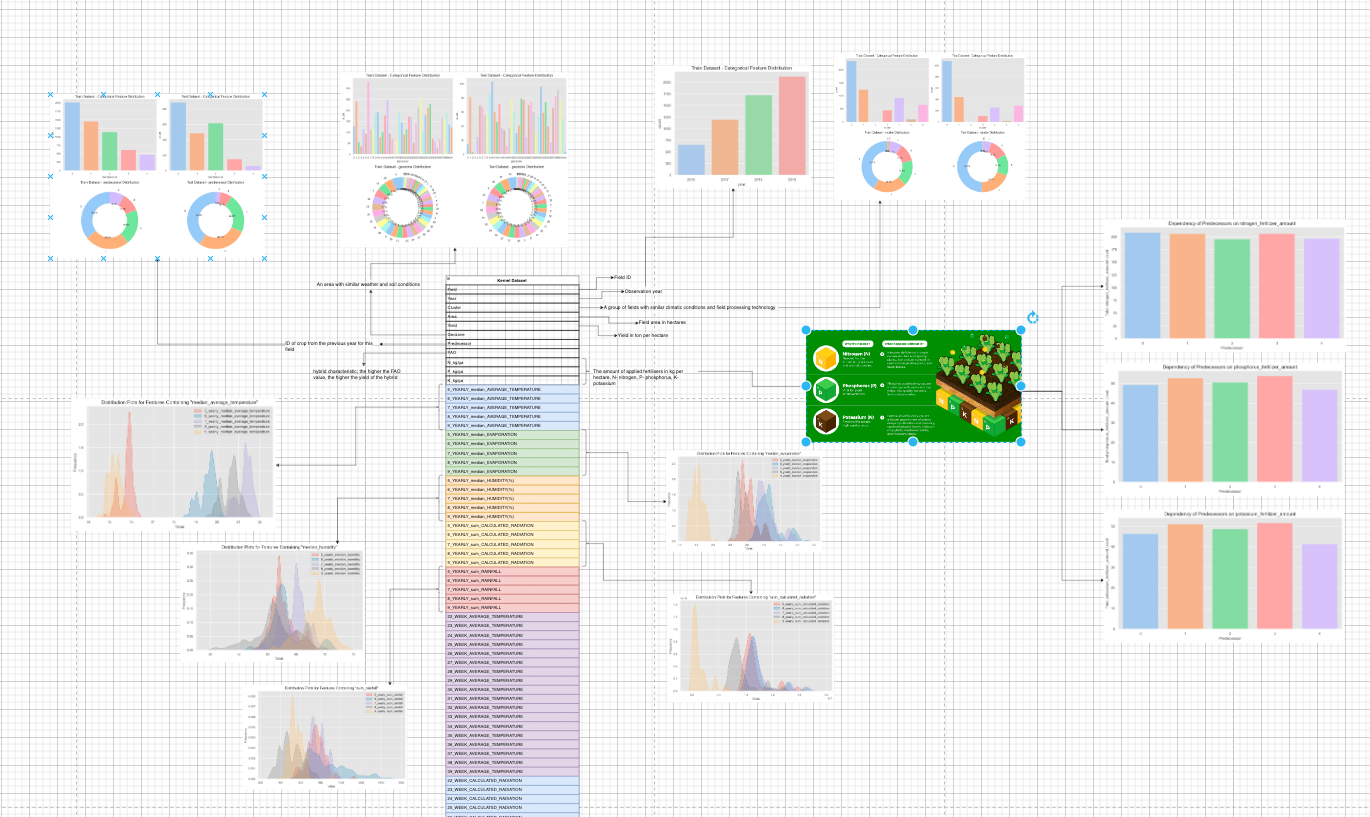
* Utilized SHAP values to interpret the model's output and feature importance
* Generated summary plots for feature importance.

## **Comments:**

It should be remembered that for the final result **not all predictors were used**, but those that were chosen by me based on the correlation analysis and the method of multiple model runs, which made it possible to reduce the size of the data by a factor of 6.2. If all predictors were used, the RMSE would be about 14-14.5 and 85% R2.

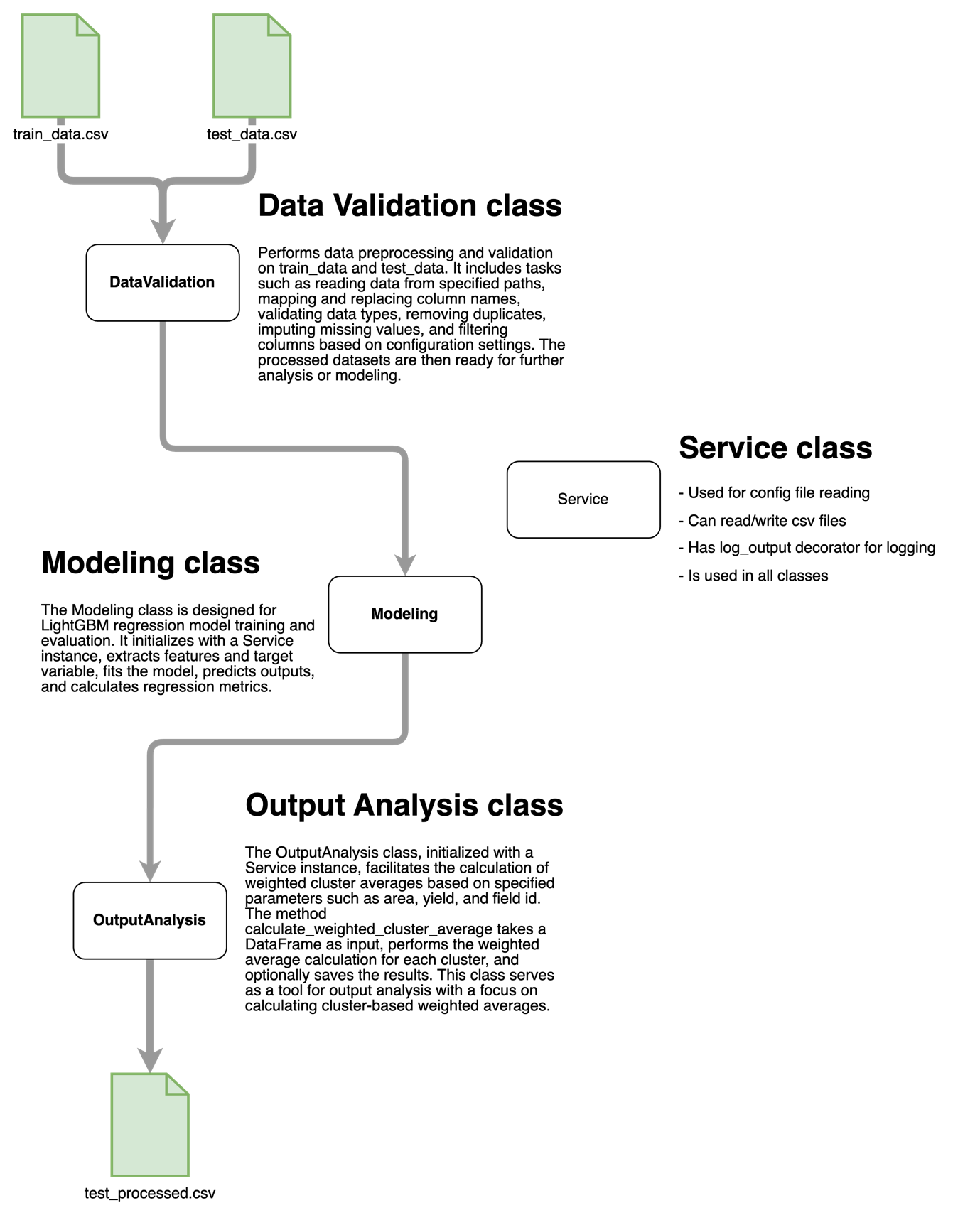
The analysis with **SHAP** showed us the same results as if we had run them on all predictors. NDVI value plays a huge role in model predictions. Actually, it is logical because they showed high values in the correlation analysis

In the same way, a data and analysis schema was built to fully understand the problem. The draw.io file can be found in this repository 

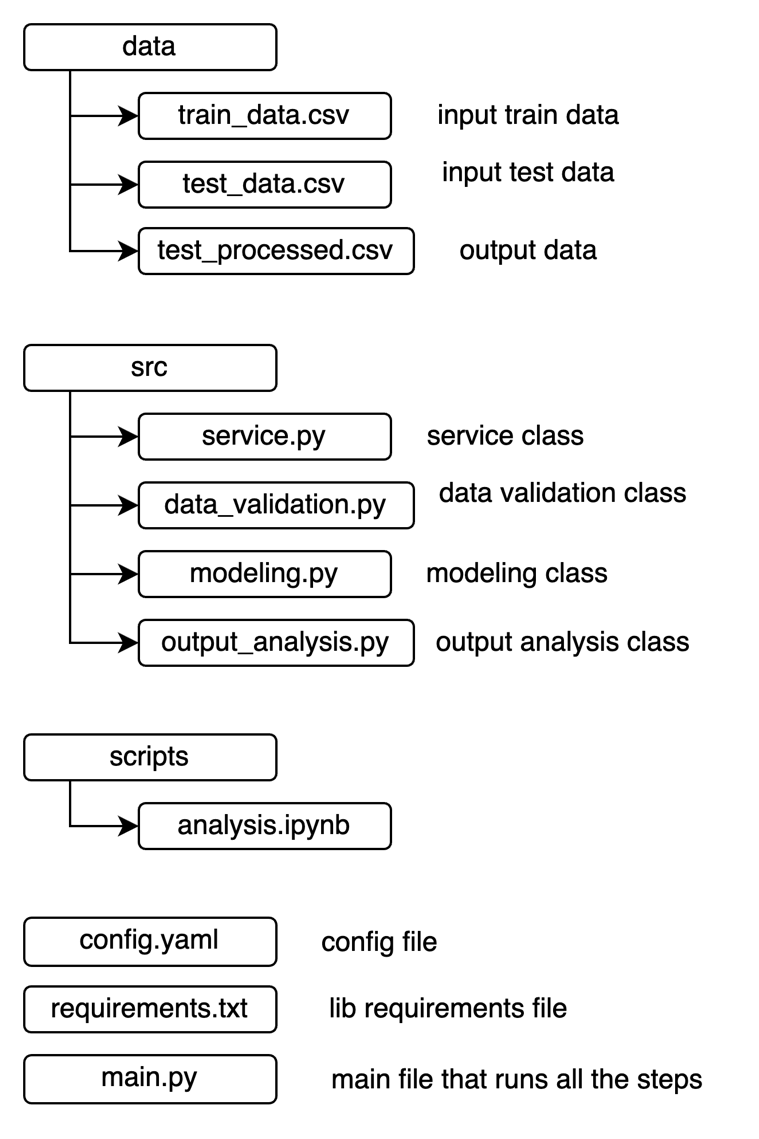
# **Code Development**

Having chosen the best parameters for the model, it was decided to build the codes as a structured project. Further it will be shown what happens in each module of the code and what is the relationship between them

## **Pipeline structure:**



## **Project folder structure:**



## **Config file structure:**

data\_validation:

read\_path:

train\_data: data/train.csv

test\_data: data/test.csv

column\_mapping:

N\_kg/ga: 'nitrogen\_fertilizer\_amount'

P\_kg/ga: 'phosphorus\_fertilizer\_amount'

K\_kg/ga: 'potassium\_fertilizer\_amount'

replacement\_dict:

\(\%\): ''

categorical\_columns: ['field', 'year', 'cluster', 'geozone', 'predecessor']

imputation:

n\_neighbors: 5

columns\_to\_impute: ['fao', 'phosphorus\_fertilizer\_amount', 'potassium\_fertilizer\_amount']

columns\_to\_include: [

'field', 'year', 'cluster', 'area', 'yield', 'geozone', 'predecessor',

'fao','22\_week\_ndvi', '23\_week\_ndvi','24\_week\_ndvi', '25\_week\_ndvi','26\_week\_ndvi',

'27\_week\_ndvi', '28\_week\_ndvi','29\_week\_ndvi', '30\_week\_ndvi', '31\_week\_ndvi',

'32\_week\_ndvi', '33\_week\_ndvi', '34\_week\_ndvi', '35\_week\_ndvi', '36\_week\_ndvi',

'37\_week\_ndvi', '38\_week\_ndvi', '39\_week\_ndvi'

]

modeling:

split\_parameters:

target\_variable: 'yield'

ignore\_columns: ['year', 'field']

model\_parameters:

learning\_rate: 0.042

n\_estimators: 920

num\_leaves: 552

feature\_fraction: 0.72

bagging\_fraction: 0.73

min\_child\_samples: 28

reg\_alpha: 0.34

reg\_lambda: 0.79

output\_analysis:

wca\_parameters:

cluster: 'cluster'

field: 'field'

area: 'area'

yield: yield

output\_path: data/test\_processed.csv