**README – Crop Yield Prediction**

This project contains a series of Python notebooks and scripts designed to facilitate data preparation, model training, and evaluation for predicting crop yield of **Maize** and **Soy** using the Producer Data dataset. The workflow includes preprocessing, feature engineering, XGBoost-based regression, and bootstrapped accuracy assessment.

**1\_PD\_CF\_Data\_prparation.ipynb**

Prepares the dataset for crop yield prediction through cleaning, aggregation, and feature generation.

**Main Tasks:**

* Load raw data containing yield, vegetation, and climate indicators.
* Filter low-quality samples based on NDVI and coverage thresholds.
* Compute field-level statistics over time (e.g., mean, max, std).
* Export a cleaned, feature-rich dataset ready for modeling.

**2\_Fit\_Classifier.ipynb**

Trains an XGBoost regression model to predict crop yield.

**Main Tasks:**

* Load and shuffle the prepared dataset.
* Add engineered features (e.g., sine/cosine of week, vegetation growth rates).
* Select predictors and labels dynamically.
* Apply **Sequential Forward Selection (SFS)** to choose relevant features.
* Train the XGBoost model and evaluate with R², MAE, and RMSE.
* Save the trained model and selected features for reuse.

**3\_Simulation.ipynb**

Runs predictions using the pre-trained model and evaluates crop yield performance.

**Main Tasks:**

* Load the trained model and feature list.
* Recreate engineered features as used in training.
* Predict crop yield for Maize and Soy.
* Generate scatter plots and time series charts to compare observed vs. predicted yield.

**4\_Simulation.py**

Script version of the simulation notebook for batch prediction and performance evaluation.

**Main Tasks:**

* Load pre-trained model and selected features.
* Apply cyclical transformations and vegetation growth rates.
* Run predictions on the dataset.
* Evaluate model performance week by week.
* Save results as CSV.

**4\_SimulationBoots.py**

Adds a robust layer of evaluation through bootstrapped accuracy estimation.

**Main Tasks:**

* Compute bootstrapped R² scores by crop and week.
* Calculate standard deviation of bootstrapped metrics.
* Smooth results using Gaussian filters.
* Save the accuracy table as CSV and plots as PNG.

**General Instructions**

**Environment Setup**

Required Python packages:

* pandas, numpy, scikit-learn, xgboost, matplotlib, seaborn
* Run notebooks in **Jupyter Notebook**.
* Execute .py scripts in a **Python 3 environment**.

**Workflow**

1. Run 1\_PD\_CF\_Data\_prparation.ipynb to clean and prepare the input dataset.
2. Train the model using 2\_Fit\_Classifier.ipynb.
3. Use 3\_Simulation.ipynb or 4\_Simulation.py to generate and evaluate predictions.
4. Optionally, run 4\_SimulationBoots.py to compute bootstrapped weekly accuracy per crop.

**Outputs**

* Cleaned datasets and prediction results (.csv)
* Trained model files (.joblib)
* Feature selection output (.json)
* Accuracy plots and scatter plots (.png)

**Notes**

* Raw .csv input files are excluded from the repository due to GitHub file size restrictions.
* All intermediate and final outputs are saved in the designated output folder.