# Supplement 1

## A. Introduction to MCYFS

The Joint Research Centre, Directorate for Sustainable Resources (JRC-D), of the European Commission (EC), and specifically the D.5 Food Security Unit, uses the MARS Crop Yield Forecasting System (MCYFS) to monitor growing conditions and status of crops, to evaluate effects of extreme meteorological events and to forecast crop yields and production in support of the management of the agricultural market of the European Union (EU). JRC experts, together with contracted partners, have been running and maintaining the MCYFS since 2000 (MARSWiki, 2020; Van der Velde and Nisini., 2019; Micale and Genovese, 2004; Lazar and Genovese, 2004). JRC staff provides the analysis and crop yield forecasts to DG-AGRI (the European Commission’s Directorate General for Agriculture and Rural Development) and to the public via the monthly JRC-MARS Bulletin. The same system is used to provide rapid science-based responses to ad hoc requests from DG-AGRI concerning extreme agro-meteorological conditions in specific regions of the EU. MCYFS includes two main components: i) the CGMS (Crop Growth Monitoring System, Supit and Van der Goot, 2003) which contains a suite of crop models and a tool-box for pre- and post processing of input and output data, and ii) CoBo (the statistical Control Board, Genovese and Bettio, 2004) which facilitates the analysts to perform statistical analysis and crop yield forecasts.

### A.1 CGMS

The main component of the CGMS is the World Food Studies (WOFOST) crop model (van Diepen et al., 1989). WOFOST is a simulation model for the quantitative analysis of the growth and production of annual field crops. It is a mechanistic, dynamic model that explains daily crop growth on the basis of the underlying processes, such as photosynthesis, respiration and how these processes are influenced by environmental conditions.

The crop simulation is fed by weather, soil and crop data. Observed meteorological data is interpolated on a regular 25 km grid using a method based on the distance, altitude and climatic region similarity between the center of grid cells and weather stations (see Van der Goot (1998)). Each crop has its own crop parameters that influence the response to weather and soil characteristics. They broadly fall into two categories. The first category describes the growth behaviour of each plant like partitioning of assimilates over plant organs and the second category describes the spatial and temporal variation in crop use, for instance the crop variety and sowing date for particular location and year. The soil influences the water available for crop growth in two ways. First, soil defines the physical limitations for roots (the rooting depth). Second, soil properties like texture determine the amount of water the soil can store.

WOFOST runs on the intersection between the 25 km meteorological grid and soil units based on the European soil map (<http://esdac.jrc.ec.europa.eu/>). In order to have the output data aggregated to administrative regions such as countries or provinces, simulation units are further intersected with the boundaries of these regions. The outputs at soil unit (STU) level are aggregated to grid level in an area weighted manner. Gridded simulations are aggregated to lowest NUTS level 3 considering the arable land area of each grid, derived from GLOBCOVER and CORINE Land Cover (Cerrani and Lopez Lozano, 2017). From NUTS3 to higher levels, crop area fractions for the current year, retrieved from Eurostat, are used to weight and aggregate the output (Cerrani and Lopez Lozano, 2017).

### A.2 CoBo

CoBo (Control Board) is a software tool supporting the JRC analysts in their statistical analysis and forecast yield for different crops and countries across Europe. CoBo supports analysts to explain past yield variability with predictors derived from weather observations, agro-meteorological indicators, remotely sensed variables (e.g. López-Lozano et al., 2015), and crop model outputs. CoBo provides analysts with different statistical methodologies including yield trend analysis, regression analysis (e.g. Bussay et al., 2015), and similarity analysis based on Principal Component Analysis (PCA) and Cluster Analysis (see Genovese and Bettio, 2004). In addition, MCYFS experts use their judgment based on information from other sources, such as farming magazines. Usually trend analysis provides a first crop yield forecast early in the growing season.

Various authors have proposed to subdivide crop yield into three components: mean yield, multi-annual trend and residual variation (e.g. Dagnelie et al. (1983)). It is assumed that the mean yield is determined by the interacting effects of climate, soil, management, technology, etc. Observed national, regional and sub-regional yields usually show a trend in time. The trend is mainly due to long-term economic and technological dynamics such as increased fertilizer application, improved crop management methods, new high yielding varieties, etc. The third component, the residual variation, is considered to be the variation among years (Dennett et al., 1980). Through multiple-linear regression, analysts try to explain this residual variation by predictors from weather observations, crop models and remotely sensed variables to forecast the end-of-season yield.

During similarity analysis, analysts identify historic years that have similar meteorological and crop conditions as the current year. The PCA and cluster analysis use predictor data of all available years including the current year to establish a similarity matrix among the years. Reported yields for the selected similar years are then weighed by the distance in the component space to come to a yield prediction (Genovese and Bettio, 2004). In case of technological trends, yields of the similar years are first corrected for the trend.

## B. Machine Learning

This section includes a brief summary of the machine learning concepts discussed in the paper. More detailed information on these concepts can be found in machine learning or deep learning books, such as Burkov, (2019) and Goodfellow et al. (2016). Mitchell (1997) provides a concise deﬁnition of machine learning: “A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.” The experience can be a set of training examples, the task can be crop yield prediction and the performance measure can be accuracy, root mean square error (RMSE), etc.

### B.1 Feature Design

Features are aggregations or summaries of input data for specific spatial and temporal windows. Feature design or feature engineering is the process of using prior (domain) knowledge to create features that influence or explain the variability in the labels.

### B.2 Supervised and Unsupervised Learning

Supervised machine learning uses a training set of examples which includes features or predictors as well as the output label (e.g. observed yield) to learn a function which relates feature values to the labels. When the label is a class (e.g. low, medium, high), the task is called classification; when the label is a real number, the task is called regression. Unsupervised learning extracts some information from a set of examples which do not have output labels. One approach is to provide a meaningful grouping of examples (e.g. clustering). Another approach is to extract the most important factors of variation (e.g. dimensionality reduction). The remainder of the section discusses concepts related to supervised learning.

*Ensemble Methods*: Ensemble methods are supervised learning methods that create a large number of weaker but statistically significant models. The final prediction is an average of the individual predictions. Random Forests (Breiman, 2001) and Gradient-boosted decision trees (see Friedman (2001); Hastie et al. (2009)) are ensembles of decision trees.

*Training, Validation and Test Sets*: Supervised machine learning algorithms are trained and evaluated on two disjoint sets. The common split is 70%-30% or 80%-20%, with the larger portion used as the training set and the smaller portion used as the test set. Training often involves iteratively improving the model by minimizing a loss function. Gradient-based methods are used to find the minimum or a reasonably low value of the loss function. Training is typically applied repeatedly to determine the optimal value of the hyperparameters of an algorithm. Hyperparameters are parameters that are not learned from data by the algorithm (e.g. the number and depth of decision trees in Random Forests). Hyperparameters are typically optimized using 5-fold or 10-fold cross-validation or custom *k*-fold validation. *K*-fold cross-validation or custom validation divides the training set into validation training and test folds. During 5-fold cross-validation, 4 folds are used as the training set and the 5th fold is used as the test set. The optimal values of hyperparameters are usually selected based on the average cross-validation performance metric. Note that the test set is never used during model training and hyperparameter optimization, including cross-validation.

*Training, Generalization Error*: After an algorithm is trained and optimized, it can be evaluated by making predictions on the training set and the test set. The error metric for predictions on the training set is called the training error; the error metric for predictions on the test set is called the test or generalization error.

*Bias, Variance, Underfitting, Overfitting*: If the training error is high, the algorithm is said to have a high bias. If the training error is low, but the test error is high, the algorithm has a high variance. An algorithm is said to underfit if it does not fit the data well and has high training and test set errors. On the other hand, an algorithm is said to overfit if the training error is low but the test error is high. Algorithms that perform well on the test set are said to generalize well.

*Linear and nonlinear methods*: Machine learning algorithms try to approximate the data-generating function - the unknown function modeling the process that produced the data. Different algorithms have different capacity based on the set of functions they can model. Linear methods can only model linear functions. Linear Regression is the most common linear method. Support Vector Machines (Boser et al, 1992; Cortes and Vapnik, 1995) can model both linear and nonlinear functions. Decision trees (Quinlan, 1986) and their variants can model nonlinear functions.

*Regularization:* Regularization is the technique to improve the generalization error of an algorithm without hurting the training error. In other words, regularization decreases the variance of a model without increasing the bias too much. Regularized versions of linear regression, such as Lasso, Ridge and Elastic Nets, add a parameter norm penalty term to the loss function. Lasso regression (Tibshirani, 1996) applies L1 penalty, which is the sum of absolute values of parameter weights. Ridge regression (Hoerl and Kennard, 1970) applies the L2 penalty, which is the sum of squared values of parameter weights. Elastic Nets (Zou and Hastie, 2005) use a combination of L1 and L2 penalties.

### B.3 Neural Networks and Deep Learning

A neural network is a simplified model of the biological neural network and typically consists of (i) an input layer where the data enters the network, (ii) one or more hidden layers where useful representation of the data is learned, and (iii) an output layer where the decision or prediction is made. Deep learning (LeCun et al., 2015) refers to learning using neural networks with many hidden layers.

*Representation learning*: It is the process of automatically extracting features that explain the underlying factors of variation. Deep neural networks are good at extracting data representations or features.

## C. Machine Learning Baseline Detailed Workflow

The overall workflow has two parts (*Figure 8*). The first part consists of preprocessing and feature design, which are specific to data sources, and splitting data into training and test sets. The second part, focusing on machine learning, is independent of data sources. Data from various sources, such as crop growth simulation outputs, weather observations and yield statistics, were homogenized and aligned to the same spatial and temporal resolutions. The data was split into training and test sets before designing features. Some data sources required feature design; others were directly used as features. Once we had features and labels, machine learning algorithms were trained and optimized on the training set and evaluated on the test set.

###### Timeline Description automatically generated

###### Figure 8: The high-level workflow

The overall workflow has two parts. The first part includes preprocessing and feature design. The second part includes machine learning.

### C.1 Preprocessing and Feature Design

#### C.1.1 Data Preprocessing

During data preprocessing, we homogenized data in terms of file format, filenames and required columns, and aggregated them to the same spatial and temporal resolutions. Input data can come in two forms: (i) static data (values do not change over time), and (ii) dynamic data (values change over time). The labels or targets were yield statistics. [*Section E*](#_j47esx90vh8s) provides a summary of input data and the assumptions about file format, file names and expected columns. Similarly, [*Section F*](#_kttbhj29ocsi) provides the preprocessing steps for each data source. Some data could be directly used as features. Others had to be aligned to the same spatial and temporal resolutions and aggregated to create features. In the baseline, the spatial resolution was different NUTS levels (Eurostat, 2016). Temporal resolution was daily, dekadal, etc. Some crops, such as winter wheat, have growing seasons that cross the calendar year boundary. In order to make the workflow support such crops, we transformed the data to align with the campaign year instead of the calendar year. A campaign year started after the end of the previous growing season and stretched up to the end of the current growing season. Data imputation (i.e. handling missing data) and outliers detection are two other preprocessing steps that could improve the accuracy of crop yield predictions. These steps require a thorough analysis of data sources and were not included in the baseline. The baseline did, however, filter out data with zero yield values because zero values cause problems, including division by zero.

Diagram

Description automatically generated

###### Figure 9: Preprocessing and feature design

Preprocessing and feature design were used to convert data into features and labels.

To keep the steps modular, data could be preprocessed independently using other software where appropriate (e.g. QGIS for shapefiles (QGIS, 2020)). The requirement for preprocessed data was that the data had the expected columns (e.g. NUTS region, calendar year) and was at the same spatial and temporal resolution.

#### C.1.2 Training, Validation and Test splits

We applied supervised learning, specifically supervised regression, to crop yield prediction. Supervised learning relies on training examples that include features as well as labels, such as yield statistics, to learn a function that relates features to labels. We split the full dataset into training and test sets. When using the yield trend, we added the last few years for each region to the test set (*Figure 10a*). This restriction was necessary because later years would contain yield trend estimated from earlier years and having earlier years in the test set would cause information leakage. When not using the yield trend, we could have used random splits. However, we needed the same test years for all regions to compare the predictions with MCYFS. Therefore, we added every *n*th year to the test set, with *n* determined by the test fraction. In both cases, we allocated 70% of the data for training and 30% for testing. We used the training set to train and optimize a model and the test set for final evaluation. We split the data into training and test sets before feature design because feature design relied on crop calendar information (see *Table 4*) and the average and standard deviation of the indicators shown in *Table 5*. We inferred the crop calendar and calculated indicator statistics for feature design only using the training set.

Timeline

Description automatically generated (a)

Timeline

Description automatically generated(b)

###### Figure 10: Training, validation and test splits when using yield trend

(a) For each region, we split the full dataset into training and test sets. (b) We further divided the training set into validation training and test sets for feature selection and hyperparameter optimization using a time-based 5-fold sliding validation.

We optimized the hyperparameters of feature selection (the number of features to select) and prediction algorithms (e.g. the number of neighbors for *k*-nearest neighbors) by dividing the training set into validation folds. When using the yield trend, we could not run cross-validation because the test fold could end up in a bin earlier than the training folds and that would cause information leakage. Therefore, we used a “time-based *k*-fold sliding validation (*Figure 10b*). For example, NL data was available from 1994 to 2018 and training years included 1994 to 2011. Assuming 5-folds, we trained the first iteration on data from 1994 to 2007, the second iteration from 1995 to 2008 for and so on until the fifth iteration from 1998 to 2011. When not using the yield trend, we applied regular *k*-fold cross-validation.

#### C.1.3 Feature Design

We incorporated agronomic principles from crop modeling to design features that had physical meaning in terms of their impact on crop growth and development. Based on the outputs of the WOFOST crop model (van Diepen et al., 1989; Supit et al., 1994), we selected 3 dekads (10-day periods) when significant changes occur in the crop’s development stage (DVS): (i) START\_DVS (DVS >= 0) is when the crop emerged from the soil, (ii) START\_DVS1 (DVS >= 100) is the middle of the flowering phase, and (iii) START\_DVS2 (DVS >= 200) is when the crop became ripe. (See de Wit et al. (2019) for a summary of how DVS is calculated.) Using these 3 dekads, we divided the crop season into 6 periods: (i) pre-planting window, (ii) planting window, (iii) vegetative phase, (iv) flowering phase, (v) yield formation phase, and (vi) harvest window (*Table 4*). In order to avoid information leakage, we inferred the crop calendar using the WOFOST data from only the training set.

|  |  |  |
| --- | --- | --- |
| **Period** | **Start Dekad** | **End Dekad** |
| Pre-planting window (p0) | min(1, avg START\_DVS - 11) | avg START\_DVS |
| Planting window (p1) | avg START\_DVS - 1 | avg START\_DVS + 1 |
| Vegetative phase (p2) | avg START\_DVS | avg START\_DVS1 |
| Flowering phase (p3) | avg START\_DVS1 - 1 | avg START\_DVS1 + 1 |
| Yield Formation phase (p4) | avg START\_DVS1 | avg START\_DVS2 |
| Harvest window (p5) | avg START\_DVS2 - 1 | avg START\_DVS2 + 1 |

###### Table 4: Crop calendar definition

We inferred the crop calendar from WOFOST outputs by selecting 3 dekads that signified important development stage changes. START\_DVS is when the crop emerges from the soil. START\_DVS1 is the middle of the flowering phase. START\_DVS2 is when the crop becomes ripe. The pre-planting window was restricted to a maximum of 12 dekads or 4 months.

|  |  |  |  |
| --- | --- | --- | --- |
| **Period** | **Maximum values** | **Average values** | **Counts of days or dekads with extreme values** |
| Pre-planting window |  | TAVG, PREC, CWB |  |
| Planting window |  | TAVG, PREC | RSM, TMIN, PREC |
| Vegetative phase | WLIM\_YB, TWC, WLAI | RSM, TAVG, CWB, FAPAR | RSM |
| Flowering phase |  | PREC | RSM, PREC, TMAX |
| Yield Formation phase | WLIM\_YB, WLIM\_YS, TWC, WLAI | RSM, CWB, FAPAR | RSM |
| Harvest window |  | PREC | PREC |

###### Table 5: Feature design using crop modeling principles

We identified indicators affecting crops during different crop calendar periods. Weather indicators included average temperature (TAVG), precipitation (PREC), climate water balance (CWB = precipitation - evapotranspiration), minimum temperature (TMIN) and maximum temperature (TMAX). WOFOST outputs included water-limited yield biomass (WLIM\_YB), water-limited yield storage (WLIM\_YS), water-limited leaf area index (WLAI), relative soil moisture (RSM) and total water consumption (TWC). Remote sensing indicators included the fraction of absorbed photosynthetically active radiation (FAPAR).

For each period of the crop calendar, we identified the weather indicators, crop growth model outputs and remote sensing indicators that affect or capture the state of crop growth and development (*Table 5*). Using these indicators, we designed 3 types of features: (i) maximum values for accumulative indicators (e.g. water-limited yield biomass), (ii) counts of days or dekads for indicators related to extreme conditions (e.g. maximum temperature), and (iii) average values for other indicators. [*Section E*](#_w1a9oqhh4s68) includes details about the data sources and the indicators used in feature design. Features for extreme conditions count days or dekads with values +/- 1 standard deviation and +/- 2 standard deviations from the average. We calculated the averages and standard deviations of indicators from the training data and used them to create both training and test features. By using the averages and standard deviations of indicators, we made the workflow generic and reusable. However, when crop-specific thresholds for different indicators are available, such data can be used to manually define more accurate and predictive features. For example, we could calculate the average precipitation during the pre-planting window as follows:

*avgPRECp0 : [‘AVG’, ‘PREC’, ‘4’, ‘14’],*

where *avgTAVGp0* is the name of the feature and 4 and 14 are the start and end dekads of the pre-planting window (*p0*). We could define other features taking the maximum or average values in a similar way. Likewise, we could define a feature to capture extremely hot conditions during the flowering phase as follows:

*TMAXp3gt32 : [‘TMAX’, ‘>’, ‘32’, ‘21’, ‘25’],*

where *TMAXp3gt32* is the feature name, 32 is the temperature threshold and 21 and 25 are the start and end dekads for the flowering phase (*p3*).

#### C.1.4 Yield Trend Estimation

We estimated a linear yield trend using a fixed 5-year trend window. We made the window length configurable to allow other options (e.g. 10 years). For every region and year, we used yield values of 5 previous years as yield trend features for machine learning. In the case of discontinuous data, the method selected the five previous years that were available(*Figure 11*).

**Continuous Data**

*Trend Window Prediction Year*

2013-2017 2018

2012-2016 2017

…

**Discontinuous Data**

*Trend Window Prediction Year Note*

2012, 2014, 2015, 2016, 2017 2018 Missing 2013

2010, 2012, 2014, 2015, 2016 2017 Missing 2011, 2013

…

###### Figure 11: Yield trend estimation using five-year trend window

To estimate the yield trend, we worked backward from the most recent year for each region. We used the yield values from 5 previous years to compute a linear trend. In the next iteration, we moved trend estimation by one year to the next most recent year. Likewise, the trend window moved by 1 year to the 5 previous years.

### C.2 Machine learning

The second part of the workflow applied machine learning to features and labels (*Figure 12*). For modularity, features from the first part of the workflow could be saved to a CSV file and loaded later for machine learning.

Diagram

Description automatically generated

###### Figure 12: Machine learning steps

Machine learning was applied using a pipeline consisting of feature scaling, feature selection and training (including hyperparameter optimization). A machine learning model was learned or fitted using only the training set. The fitted model was evaluated on the test set.

#### C.2.1 Feature Scaling

Feature scaling or normalization brings different feature values to similar ranges. The motivation is to prevent the sizes of feature values from affecting the learning process. Min-max normalization scales the features to [0, 1]: , where *x* is the old value and *x'* is the new value.

Standardization or z-score normalization transforms feature values to have mean 0 and standard deviation 1: , where *mean*(x) and are the mean and standard deviation of the feature values before standardization. In the baseline, we use z-score normalization to normalize feature values.

Feature scaling or normalization must be applied after training-test split to avoid information leakage. This is because calculating statistics like min, max, mean and standard deviation must not use the test data.

#### C.2.2 Feature selection

The motivation for feature selection is to remove features that might be correlated or those that capture the noise in training data, leading to overfitting. In the baseline we used three methods for feature selection: (i) Random Forest (RF), (ii) Recursive Feature Elimination (RFE), and (iii) Mutual Information Regression (MI). RF (Breiman, 2001) uses an ensemble of weak decision tree models built by picking a random subset of candidate features to build the trees. The final prediction is an average of the predictions of individual trees. RFE (e.g. Granitto et al., 2006), recursively eliminates unimportant features by evaluating a machine learning algorithm which provides feature weights or feature importances. In the baseline, we used RFE with Lasso regression (Tibshirani, 1996). Lasso applies the L1-norm penalty on feature weights to drive weights of unimportant features to zero. MI (Shannon, 1948) is a univariate feature selection method, similar to pearson’s r (see Benesty et al. (2009)), that calculates the information content of individual features.

During feature selection, we optimized the number of features as a hyperparameter. We included feature selection in a pipeline consisting of feature scaling, feature selection, and training and evaluation. The training and evaluation stage used a standard machine learning algorithm. We passed the pipeline to a grid search method using 5-fold cross-validation or a time-based 5-fold sliding validation to optimize the hyperparameters. For each combination of hyperparameters, a 5-fold cross-validation or sliding validation was run to determine the mean score. Hyperparameter values with the best mean scores were selected as optimal. In the baseline, we focused on optimizing the number of features and used reasonable default values of hyperparameters for training. However, it is possible to do a more thorough hyperparameter optimization if required.

We created a pipeline consisting of feature scaling, feature selection and training stages to avoid information leakage during feature selection (Muller and Guido, 2016). The pipelines ensured each stage of training and optimization used only the training data. In effect, the parameters for scaling features (e.g. mean and standard deviation), the number of features to select and the feature weights for the trained model were learned from the training set.

#### C.2.3 Training and Evaluation

During training, we took the selected features and labels and re-optimized an algorithm’s hyperparameters using grid search and 5-fold cross-validation or sliding validation. In this step, we used a pipeline consisting of feature scaling and training. We selected the model with optimal values of hyperparameters for final evaluation. In the baseline, we optimized a small set of hyperparameters. It is possible to do a more thorough hyperparameter optimization and analysis if required. We evaluated the performance of four algorithms: (i) Ridge Regression (Ridge), (ii) K-nearest neighbors (KNN), (iii) Support Vectors Regression (SVR), (iv) Gradient Boosted Decision Trees (GBDT). Ridge (Hoerl and Kennard, 1970) is a linear model with L2-norm as the regularization penalty. KNN (Cover and Hart, 1967; Aha et al., 1991) is an instance-based method relying on similarity of data points in the feature space. In the baseline, KNN is weighted by distance. SVR, an extension of the support vector machines algorithm, is based on the principle of structural risk minimization (Guyon et al., 1992; Burges 1998). For linear data, SVR finds a hyperplane with the largest margin of separation. For nonlinear data, SVR uses kernel functions to map the data to a high dimensional space and finds a separating hyperplane there. In the baseline, we applied SVR with the RBF (radial-basis function). GBDT is an ensemble method which uses gradient boosting (Friedman, 2001) to grow the trees and often provides more accurate results compared to random forests (see Hastie et al., 2009).

We compared the performance of the algorithms with a poor man’s method (the “null” method). When the yield trend was not used, the null method was equivalent to the ZeroR algorithm (see Baskin et al., 2017) that predicts the average of the training set. When the yield trend was used, the null method predicted the estimated linear yield trend (using a 5-year window). We evaluated all algorithms based on the mean absolute error (MAE), mean absolute percentage error (MAPE), root mean squared error (RMSE) and the coefficient of determination or R2. For MAE and RMSE, we calculated their normalized counterparts to compare them in percentage terms. The normalized errors were calculated by dividing the mean error with the mean true yield.

Mean Absolute Error (MAE):

Normalized MAE was calculated as MAE divided by the mean of the true yield values.

Mean Absolute Percentage Error (MAPE):

Root Mean Squared Error (RMSE):

Normalized RMSE was calculated as RMSE divided by the mean of the true yield values.

Coefficient of determination or R2 score (R2):

## D. Results: Supplemental Figures

|  |  |  |
| --- | --- | --- |
| **Chart** | **Chart** | **Chart** |
| Chart | Chart | Chart |
|  |  | Chart |
| Chart | Chart | Chart |
| Chart | Chart | Chart |

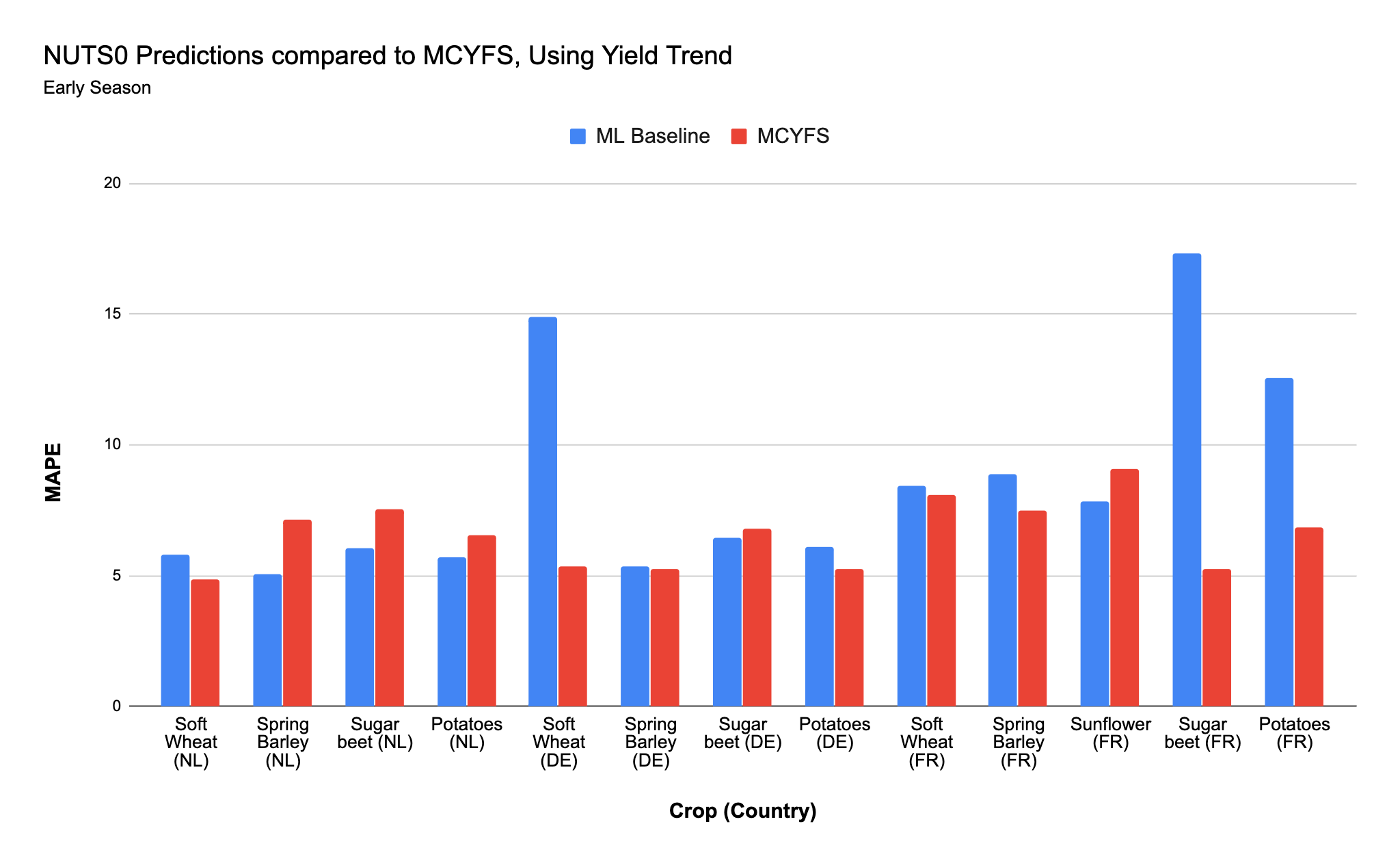
###### Figure 13: Yield Trend vs. No Yield Trend

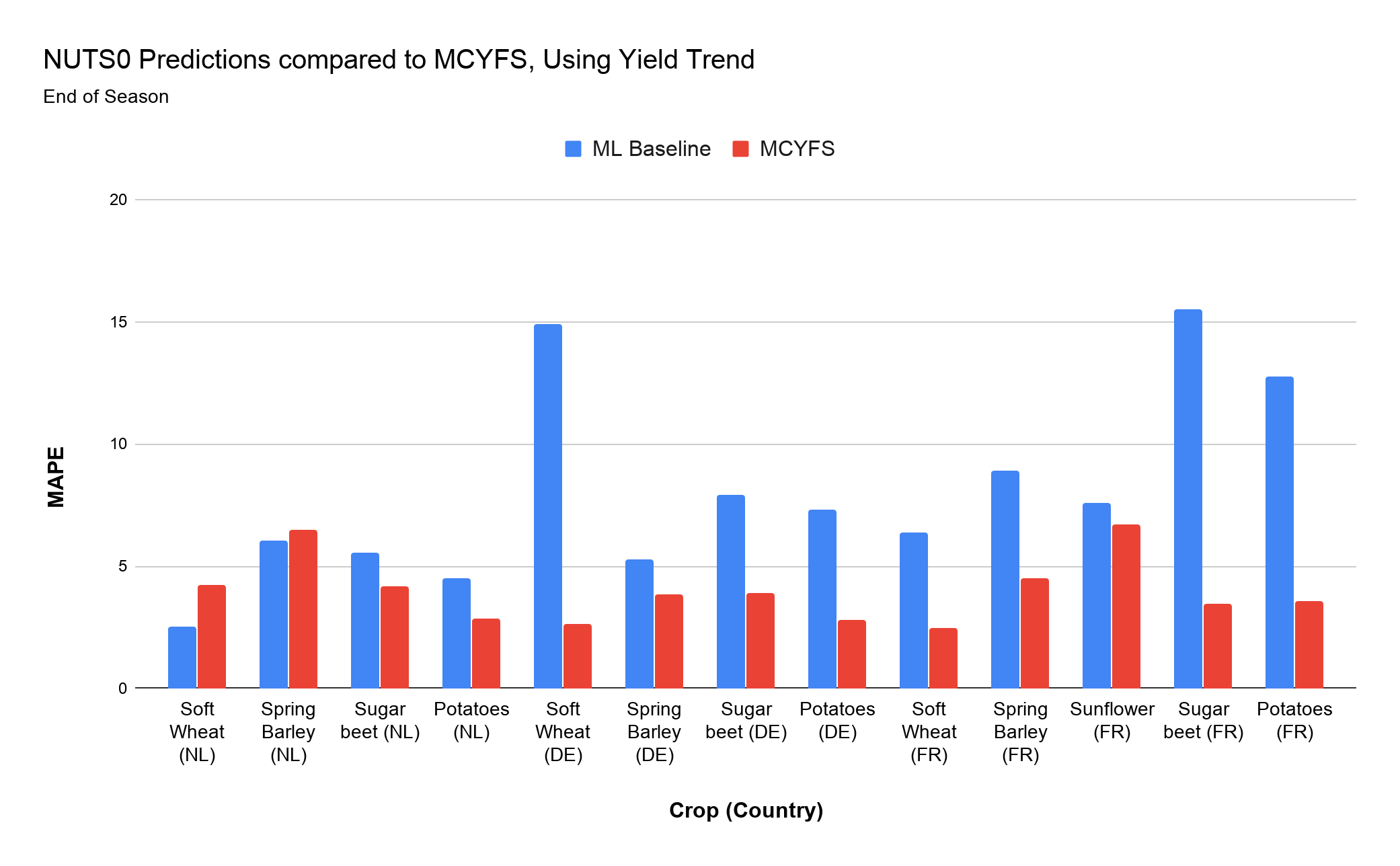
The MAPE of Gradient Boosted Decision Trees was compared with the null method.

|  |  |  |
| --- | --- | --- |
| **Chart** | **Chart** | **Chart** |
| Chart | Chart | Chart |
|  |  | Chart |
| Chart | Chart | Chart |
| Chart | Chart | Chart |

###### Figure 14: Early season prediction using a 5-year yield trend

The MAPE of Gradient Boosted Decision Trees for early and end of season predictions.

(a)

(b)

###### Figure 15: Comparing machine learning baseline with past MCYFS forecasts

MAPE for a) Early season predictions (30 days after planting), and b) end of season predictions, both using a 5-year yield trend.

## E. Input Data

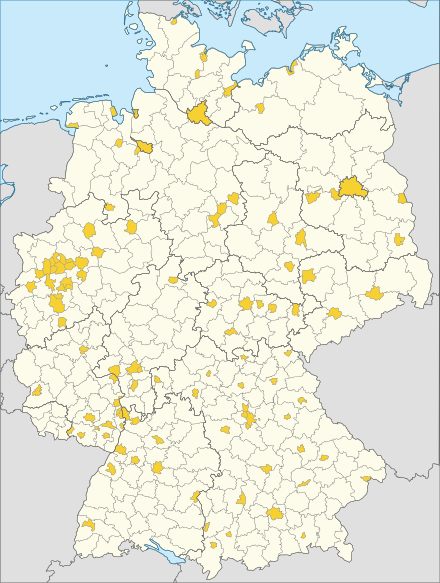
Input data included WOFOST simulation outputs, weather observations, remote sensing, soil, region centroid coordinates and distance to coast, crop area fractions and yield statistics.

### E.1 NUTS Regions



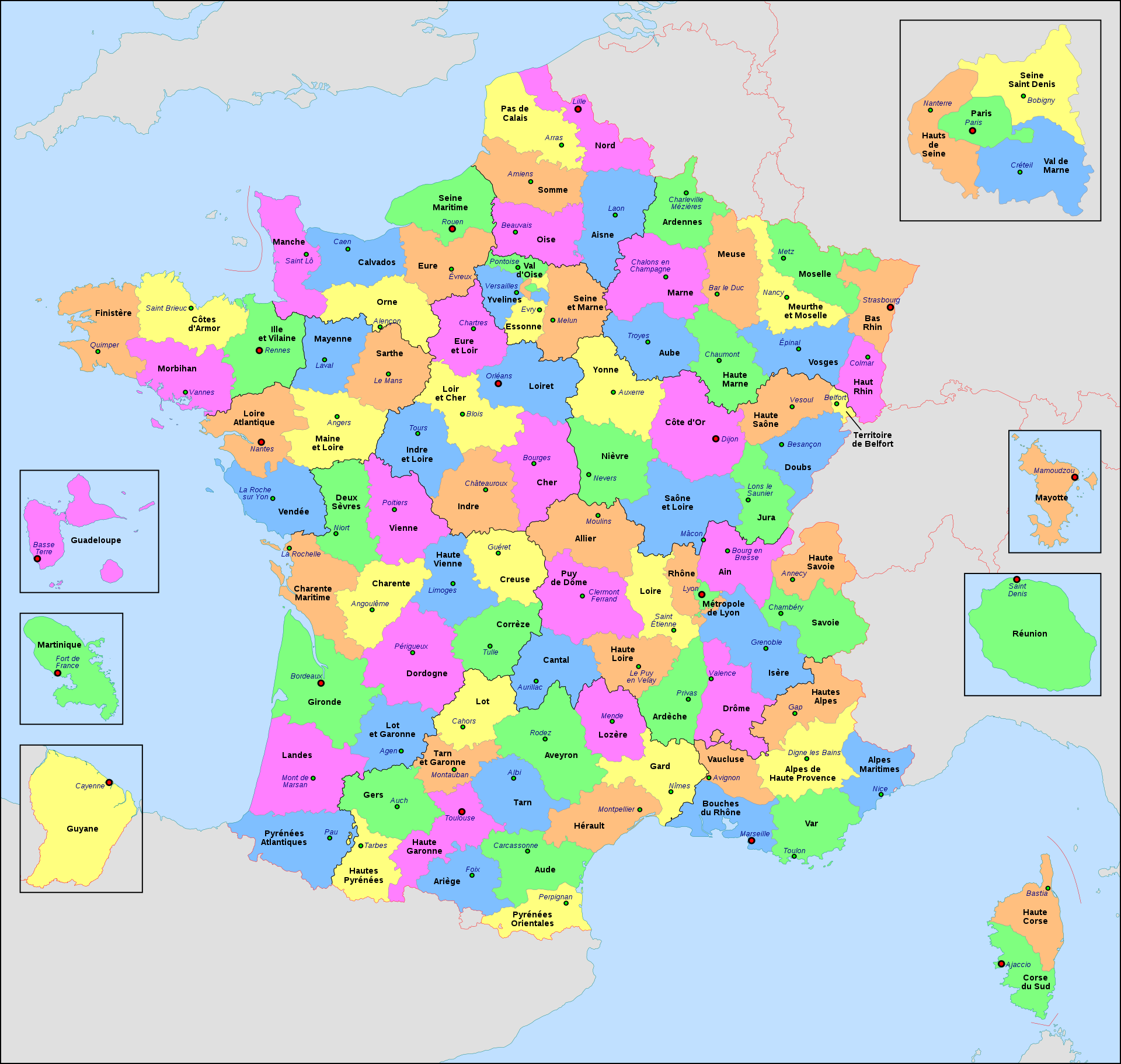
###### Figure 16: NUTS2 regions (Provincies) of the Netherlands

12 NUTS2 regions (see <https://en.wikipedia.org/wiki/Provinces_of_the_Netherlands>)



***Figure 17: NUTS3 regions (Kreise OR Districts) of Germany***

350 NUTS3 regions (see <https://en.wikipedia.org/wiki/Districts_of_Germany>). District-free cities are shown in yellow, districts in white.



***Figure 18: NUTS3 regions (Départements) of France***

101 NUTS3 regions (see <https://en.wikipedia.org/wiki/Departments_of_France>)

### E.2 Details of Data Sources

#### E.2.1 WOFOST Indicators

* POT\_YB (potential dry weight biomass (kg ha-1))
* POT\_YS (potential dry weight storage organs (kg ha-1))
* WLIM\_YB (water-limited dry weight biomass (kg ha-1))
* WLIM\_YS (water-limited dry weight storage organs (kg ha-1))
* PLAI (potential leaf area divided by surface area (m2 m-2))
* WLAI (water-limited leaf area divided by surface area (m2 m-2))
* DVS (development stage (0-200))
* RSM (percentage of soil water holding capacity)
* TWC (sum of water limited transpiration (cm))
* TWR (sum of potential transpiration (cm))

#### E.2.2 Meteo Indicators

* TMAX (maximum daily air temperature (℃))
* TMIN (minimum daily air temperature (℃))
* TAVG (average daily air temperature (℃))
* VPRES (average daily vapour pressure (hPa))
* WSPD (average daily wind speed at 10 m (m s-1))
* PREC (sum of daily precipitation (mm))
* ET0 (sum of daily evapotranspiration of short vegetation (Penman-Monteith, Allen et al., 1998) (mm))
* RAD (sum of daily global incoming shortwave radiation (KJ m-2 d-1))
* RELH (average daily relative humidity (%))
* CWB (climate water balance, calculated as PREC - ET0)

#### E.2.3 Remote Sensing Indicators

FAPAR (Fraction of Absorbed Photosynthetically Active Radiation (Smoothed))

#### E.2.4 Soil Moisture Indicators

* SM\_WP (wilting point)
* SM\_FC (field capacity)
* SM\_SAT (saturation)
* DEPTH (rooting depth)
* SM\_WHC (water holding capacity)

#### E.2.5 Region Centroid Information

* CENTROID\_X (longitude)
* CENTROID\_Y (latitude)
* DIST\_COAST (distance to coast)

#### E.2.6 Crop Area Fractions

* FRACTION

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Description** | **Source** | **Parameters used** | **Years** |
| Outputs of WOFOST crop model aggregated to NUTS regions | JRC (see Lecerf et al. ( 2019), *Section A.1*) | WLIM\_YB, WLIM\_YS, WLAI, DVS, RSM, TWC | 1979-2018 |
| Daily or dekadal gridded weather observations aggregated to NUTS regions only including the arable land area | JRC (see EC-JRC (2020), *Section A.1*) | TAVG, TMIN, TMAX, PREC, ET0, CWB | 1979-2018 |
| Remote sensing indicators aggregated to NUTS regions only including the arable land area | JRC (see Copernicus Global Land Service, 2020) | FAPAR | 1999-2018 |
| Soil data aggregated to NUTS regions only including the arable land area | JRC (see Lecerf et al. (2019), *Section A.1*) | SM\_WHC | Static |
| Centroids of NUTS region | Eurostat (Eurostat, 2020b) | CENTROID\_X, CENTROID\_Y, DIST\_COAST | Static |
| Reported regional (NUTS2 or NUTS3) yield statistics | NL: CBS (NL-CBS, 2020)  FR: AGRESTE (FR-Agreste, 2020) DE: Regionaldatenbank (DE-RegionalStatistiks, 2020) | Yield statistics for NUTS regions | NL: 1994-2018,  FR: 1989-2018,  DE: 1999-2017 |
| Eurostat national (NUTS0) yield statistics | Eurostat (Eurostat, 2020a) | Yield statistics for NUTS0 level | NL: 1971-2018  FR: 1971-2018  DE: 1975-2018 |
| Crop Area Fractions | JRC (see Lecerf et al., 2019) | Area fraction | 1979-2018 |
| Past MCYFS Yield Forecasts | JRC (See Van der Velde and Nisini, 2019) | Forecast date, Yield forecast | Mostly 1995-2018 |

###### Table 6: Data Summary

Most of the data sources came from the MCYFS database or Eurostat.

**File format**: CSV. **File name format**: <source>\_<NUTS level>\_<country>.csv

### E.3 Data Sources Requirements for the Workflow

*Expected columns*

1. **Static Data**: IDREGION is required. Other columns depend on the data source. It is best to have IDREGION as the first column. If preprocessing is required, preprocessing code has to be added or updated to handle other columns. Other columns:
   1. **Centroids data**: IDREGION, CENTROID\_X, CENTROID\_Y, DIST\_COAST.
   2. **Soil data**: IDREGION, SM\_WP, SM\_FC, SM\_SAT, DEPTH.
2. **Dynamic Data**:
   1. **Daily Meteo, Dekadal Meteo, Remote Sensing data**: IDREGION, DATE are required. The date must be in YYYYMMDD format. Other columns depend on the data source. It is best to have IDREGION, DATE as the first two columns. Other columns:
      1. **Daily Meteo data**: TMAX, TMIN, TAVG, VPRES, WSPD, PREC, ET0, RAD, RELH.
      2. **Dekadal Meteo data**: All indicators are averages of daily values.
      3. **Remote Sensing data**: FAPAR.
   2. **WOFOST data**: CROP\_ID, IDREGION, DATE. It is best to have these as the first 3 columns. Other columns: POT\_YB, POT\_YS, WLIM\_YB, WLIM\_YS, PLAI, WLAI, DVS, RSM, TWC, TWR.
   3. **Area fractions data**: Area fractions data is required to aggregate predictions to NUTS0. For NUTS2, area fractions for NUTS1 and NUTS2 regions are required. For NUTS3, area fractions for both NUTS1, NUTS2 and NUTS3 are required. AREA\_FRACTIONS data requires these columns: CROP\_ID, IDREGION, FYEAR, FRACTION. Note it is possible to support CROP (name) instead of CROP\_ID, but this requires updates to preprocessing code.
3. **Yield Data**: CROP, IDREGION are required. The preferred format also requires FYEAR and YIELD. Another supported format is: CROP, IDREGION, 1994, 1995, … YIELD data comes with CROP (name) and WOFOST with CROP\_ID. It is possible to add a preprocessing step to support name and ID in both cases.
4. **MCYFS Yield forecasts data**: CROP, IDREGION, FYEAR and YIELD\_PRED are required. MCYFS has multiple yield predictions for each year. Therefore, PREDICTION\_DATE is also required to determine the prediction dekad. The PREDICTION\_DATE must be in DD/MM/YYYY format.

## F. Data Preprocessing Details

1. **Centroids data**: First shapefiles for NUTS level were loaded into QGIS application and centroid coordinates were exported. Distance to coast was also calculated using QGIS.
2. **Soil data**: Soil data were aggregated from soil units to regions in an area weighted manner only including arable land. During preprocessing, SM\_WHC was calculated as (SM\_FC - SM\_WP). Other columns including SM\_FC and SM\_WP were dropped.
3. **Daily Meteo, Dekadal Meteo data**: During preprocessing, FYEAR and DEKAD were extracted from DATE and DATE is dropped. CWB was calculated as (PREC - ET0). METEO\_DAILY was optionally converted to dekadal data by taking averages in most cases. For TMAX, max was taken, for TMIN min was taken. For PREC, ET0, CWB sum was taken. VPRES, WSPD and RELH were dropped. NOTE the dekadal meteo data (without preprocessing from daily meteo data) is the average of all daily meteo indicators.
4. **Remote Sensing data**: During preprocessing, FYEAR and DEKAD were extracted from DATE and DATE is dropped. If the yield prediction NUTS level was NUTS3, NUTS3 regions inherited remote sensing data from NUTS2.
5. **WOFOST data**: During preprocessing, FYEAR and DEKAD were extracted from DATE and DATE was dropped. Summary step created DVS summary for selected crop and country by region and year.
6. **Yield data**: For data in the first format (CROP, IDREGION, FYEAR and YIELD), CROP\_ID was looked up. After filtering data for the selected crop, CROP and CROP\_ID were dropped. For data in the second format (CROP, IDREGION, 1994, 1995, ...), data in year columns were exploded to create separate rows. This converted data to the first format (CROP, IDREGION, FYEAR and YIELD). After filtering data for the selected crop, CROP and CROP\_ID were dropped.
7. **Eurostat NUTS0 yield data**: NUTS0 yield data was preprocessed to keep CROP, IDREGION, FYEAR and YIELD, and CROP\_ID was looked up. After filtering data for the selected crop, CROP and CROP\_ID were dropped.
8. **MCYFS yield forecasts data**: PREDICTION\_DATE was used to determine the PRED\_DEKAD. PREDICTION\_DATE was renamed to PRED\_DATE.

### 

## References

Aha, D.W., Kibler, D., Albert, M.K., 1991. Instance-based learning algorithms. Machine Learning 6, 37–66. doi:10.1007/BF00153759.

Allen, R.G., Pereira, L.S., Raes, D., Smith, M., et al., 1998. Crop evapotranspiration – guidelines for computing crop water requirements., in: Irrigation and Drainage, paper 56, FAO, Rome.

Baskin, I.I., Marcou, G., Horvath, D., Varnek, A., 2017. Benchmarking machine- learning methods. Tutorials in Chemoinformatics , 209–222doi:10.1002/ 9781119161110.ch13.

Benesty, J., Chen, J., Huang, Y., Cohen, I., 2009. Pearson correlation coefficient, in: Noise Reduction in Speech Processing. Springer, pp. 1–4. https://link.springer.com/content/pdf/10.1007/ 978-3-642-00296-0\_5.pdf, Last accessed: May 11, 2020.

Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A training algorithm for op- timal margin classifiers, in: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM New York, NY, USA. pp. 144–152.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Burges, C.J., 1998. A tutorial on support vector machines for pattern recog- nition. Data Mining and Knowledge Discovery 2, 121–167. doi:10.1023/A: 1009715923555.

Burkov, A., 2019. The hundred-page machine learning book. Andriy Burkov Quebec City, Can.

Bussay, A., van der Velde, M., Fumagalli, D., Seguini, L., 2015. Improving operational maize yield forecasting in hungary. Agricultural Systems 141, 94–106. doi:10.1016/j.agsy.2015.10.001.

Cerrani, I., Lopez Lozano, R., 2017. Algorithm for the disaggregation of crop area statistics in the MARS crop yield forecasting system. https://agri4cast.jrc.ec.europa.eu/DataPortal/Resource\_Files/PDF\_Documents/31\_rationale.pdf, Last accessed: Oct 8, 2020.

Copernicus Global Land Service, 2020. Fraction of Absorbed Photosynthetically Active Radiation. https://land.copernicus.eu/global/products/fapar, Last accessed: Oct 19, 2020.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20, 273–297. doi:10.1007/BF00994018.

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Trans- actions on Information Theory 13, 21–27. doi:10.1109/TIT.1967.1053964.

Dagnelie, P., Palm, R., Istasse, A., 1983. Pr ́evision de productions agricoles dans les dix pays de la Communaute Economique Europeenne. Technical Report. Faculte des Sciences Agronomiques de l’Etat, Gembloux , Belgium.

DE-RegionalStatistiks, 2020. Regionaldatenbank deutschland. https://www.regionalstatistik.de/genesis/online/data, Last accessed: May 11, 2020.

De Wit, A., Boogaard, H., Fumagalli, D., Janssen, S., Knapen, R., van Kraalingen, D., Supit, I., van der Wijngaart, R., van Diepen, K., 2019. 25 years of the WOFOST cropping systems model. Agricultural Systems 168, 154–167. doi:10.1016/j.agsy.2018.06.018.

Dennett, M., Elston, J., et al., 1980. Weather and yields of tobacco, sugar beet and wheat in europe. Agricultural Meteorology 21, 249–263. doi:10.1016/ 0002-1571(80)90070-9.

EC-JRC, 2020. JRC Agri4Cast Data Portal. https://agri4cast.jrc.ec. europa.eu/DataPortal/Index.aspx, Last accessed: May 11, 2020.

Eurostat, 2016. Nomenclature of territorial units for statistics. https://ec.europa.eu/eurostat/web/nuts/background, Last accessed: May 11, 2020.

Eurostat, 2020a. Eurostat - agricultural production - crops. https://ec.europa.eu/eurostat/statistics-explained/index.php/ Agricultural\_production\_-\_crops, Last accessed: May 11, 2020.

Eurostat, 2020b. Eurostat - geographical information and maps.

https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/ administrative-units-statistical-units/nuts, Last accessed: May 11, 2020.

FR-Agreste, 2020. Agreste web data portal. https://agreste.agriculture. gouv.fr/agreste-web/, Last accessed: May 11, 2020.

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of Statistics , 1189–1232 https://www.jstor.org/stable/ 2699986, Last accessed: May 11, 2020.

Genovese, G., Bettio, M., 2004. Methodology of the MARS Crop Yield Forecasting System. Vol 4. Statistical Data Collection, Data Processing and Analysis. Official Publications of the European Communities, Luxembourg.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org, Last accessed: May 11, 2020.

Granitto, P.M., Furlanello, C., Biasioli, F., Gasperi, F., 2006. Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products. Chemometrics and Intelligent Laboratory Systems 83, 83–90. doi:10.1016/ j.chemolab.2006.01.007.

Guyon, I., Vapnik, V., Boser, B., Bottou, L., Solla, S.A., 1992. Structural risk minimization for character recognition, in: Advances in Neural Information Processing Systems, pp. 471–479.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Hoerl, A.E., Kennard, R.W., 1970. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 12, 55–67. doi:10.1080/00401706. 1970.10488634.

Lazar, C., Genovese, G., 2004. Methodology of the MARS crop yield forecasting system Vol. 2. Agrometeorological Data Collection, Processing and Analysis. Office for Official Publications of the European Communities, Luxembourg.

Lecerf, R., Ceglar, A., Lopez-Lozano, R., Van Der Velde, M., Baruth, B., 2019. Assessing the information in crop model and meteorological indicators to forecast crop yield over europe. Agricultural Systems 168, 191–202. doi:10. 1016/j.agsy.2018.03.002.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. doi:10.1038/nature14539.

MARSWiki, 2020. MARS Crop Yield Forecasting System. https://marswiki.jrc.ec.europa.eu/agri4castwiki/index.php/Welcome\_to\_WikiMCYFS, Last accessed: May 11, 2020.

Micale, F., Genovese, G., 2004. Methodology of the MARS crop yield fore- casting system. Meteorological Data Collection, Processing and Analysis, in: Publication EUR 21291 EN, Office for Official Publications of the EU, Lux- embourg. pp. 5–26.

Mitchell, T.M., 1997. Machine Learning. McGraw Hill, New York.

Muller, A.C., Guido, S., 2016. Introduction to machine learning with Python: a guide for data scientists. O’Reilly Media, Inc.

NL-CBS, 2020. Cbs open data portal. https://opendata.cbs.nl/statline/ #/CBS/nl/?fromstatweb, Last accessed: May 11, 2020.

QGIS Development Team, 2020. QGIS Geographic Information System. http: //qgis.osgeo.org, Last accessed: May 11, 2020.

Quinlan, J.R., 1986. Induction of decision trees. Machine Learning 1, 81–106. doi:10.1007/BF00116251.

Shannon, C.E., 1948. A mathematical theory of communication. Bell System Technical Journal 27, 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x.

Supit, I., van der Goot, E., 2003. Updated system description of the WOFOST crop growth simulation model as implemented in the crop growth monitoring system applied by the European Commission. Treemail Pubilshers, Heelsum.

Supit, I., Hooijer, A., Van Diepen, C., 1994. System description of the WOFOST 6.0 crop simulation model implemented in CGMS. vol. 1. theory and algorithms., in: EUR Publication No. 15959 EN, Office for Official Publications of the European Communties, Luxembourg. p. 146.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58, 267–288. doi:10. 1111/j.2517-6161.1996.tb02080.x.

Van der Goot, E., 1998. Spatial interpolation of daily meteorological data for the Crop Growth Monitoring System (CGMS), in: Proceedings of Seminar on Data Spatial Distribution in Meteorology and Climatology 28 September - 3 October 1997, Office for Official Publications of the EU, Luxembourg. pp. 141–153.

Van Diepen, C., Wolf, J., Van Keulen, H., Rappoldt, C., 1989. WOFOST: a simulation model of crop production. Soil Use and Management 5, 16–24. doi:10.1111/j.1475-2743.1989.tb00755.x.

Van der Velde, M., Nisini, L., 2019. Performance of the MARS-crop yield forecasting system for the European Union: Assessing accuracy, in-season, and year-to-year improvements from 1993 to 2015. Agricultural Systems 168, 203–212. doi:10.1016/j.agsy.2018.06.009.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: series B (Statistical Methodol- ogy) 67, 301–320. doi:10.1111/j.1467-9868.2005.00503.x.