# Evaluation of Machine Learning Methods for Yield Prediction

*Summary*

*Using a variety of field-, crop-, and climate related data sources, we test the performance of baseline machine learning regression and classification methods for predicting crop yield. Of all tested regressors, the RandomForestRegressor obtains the smallest test mean absolute error (7.65 hkg/ha for winter wheat (WW) and 6.45 hkg/ha for spring barley (SB)) and the highest test $R^2$ (0.35 for WW and 0.27 for SB). Similarly, the RandomForestClassifier obtains the highest accuracy (0.35 for WW and 0.35 for SB), and thus performs the best across all of the classification methods, even though the SVC and KNeighborsClassifier performance is almost as good. Including more features in the model, generally results in better performance. In particular, the inclusion of the climate history in addition to static field features results in a large improvement in performance. Overall, though, the yield regression results are somewhat poor showing low performance and severe overfitting which suggests that more data samples is needed to successfully fit a model that generalizes well. The relative success of the RandomForestRegressor compared to linear methods suggests that non-linear regressors are needed in order to separate the data set. The classification results are more satisfactory than the regressor results, even though, the classifiers also suffer from overfitting. An outline of possible avenues for future work to improve on these results is presented.*

## Introduction

Machine Learning holds great promise for the agricultural sector, and can in principle provide invaluable decision support to crop growers based on a wealth of different data sources. The first challenge, however, is knowing where to look; not all data is created equal, and some is more relevant for prediction than others.

Crop yield quantity summarizes all the decisions and influencing factors across the growing season, and has the simple interpretation that the higher the yield, the better the growing process. Having accurate models of crop yield allows for in-season optimization of field management, such that steps can be taken to directly maximize the field's earning potential. Yield quantities are either precisely measured by truck scales, or more approximately by harvester data or farmer rules of thumb. It is always measured in units of hkg/ha.

With a long-term view towards building systems for timely prediction of yield at the field level, we make a first step by undertaking a baseline analysis of yield predictability, taking various data sources into account.
Yields are, to various degrees of approximation, registered in SEGES' farm management software 'MarkOnline', and stored in the Danish Field Database (DMDB). In addition, DMDB stores various other field-specific information of agronomic importance, such as the soil type of the field, its crop and catch crop histories, all of which can be presumed relevant to the downstream prediction task. Furthermore, a field is also represented by remote sensing data, such as biomass - measured by Normalized Difference Vegetation Index (NDVI) - from optical satellites, and various time-series measurements of climate variables.

With a wealth of data on each field, we set out to evaluate the usefulness of each variable by performing various prediction tasks, using machine learning methods for regression and classification of yield given various sets of variables. Due to a shortage of measurements across different crops, we here restrict our analyses to yield prediction of winter wheat (WW) and spring barley (SB).

# Methods

This section describes the collection, validation, and preparation of the data, along with presenting the utilized regression and classification methods.

## Data collection

Data on each field was collected from DMDB, along with accompanying metadata to uniquely identify fields - both geographically and within the broader DMDB naming scheme.

DMDB stores data on over 500.000 fields each year, and only a small subset of these have yield measurements that can be validated. As a first step towards data validation, we restricted our attention to those fields with 'registered' measurements, indicating that registration was done by a human and not an automated process[1]. We also restricted the data set to registrations performed on the 30th of November, 2017.

For each field, we collected a number of variables describing properties which would not change over the course of the growing season. We think of these as 'static' variables, and they comprise the minimal set for downstream prediction.

- Static variables
    - ID of the farm (categorical)
    - Date of harvest (day of the year)
    - Date of sowing (day of the year)
    - Crop variety (categorical)
    - Soil type (categorical)
    - Field area (hectares)

In addition, we extracted each field's crop history - summarizing the various crop rotations performed on the field prior to the current yield measurement - by collecting data on crops and catch crops across a 7 year time-horizon. This data was extracted from DMDB:

- Crop history (for the last 7 years):
    - Crop type (categorical)
    - Harvest year of the individual crop (year)
    - Whether it was a catch crop (boolean)

Remote Sensing (satellite) data is available in the form of time-stamped measurements of NDVI across the field, see Figure 1.

[1] This turns out to mean that a human has confirmed the yield registry, which is not the same as having an accurate yield measurement. Many yield quantities were in fact automatically filled in by MarkOnline during the registration.

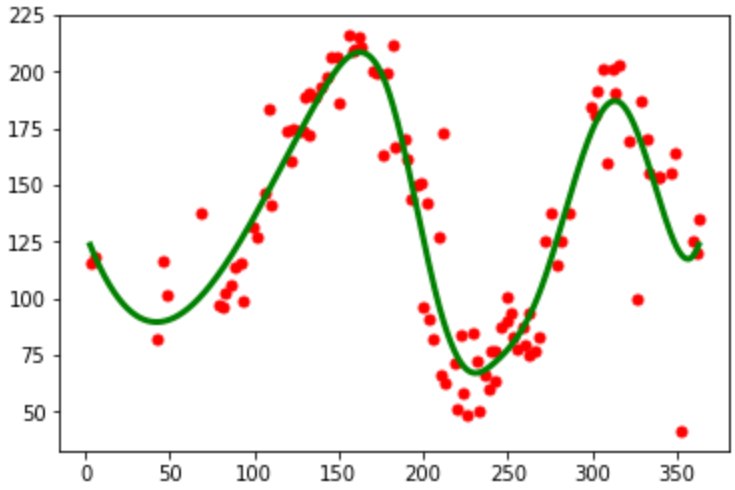



**Figure 1**: *Field growth curve as measured by NDVI. The x-axis is calendar day and the y-axis is NDVI value. The first peak from the left shows the growth curve of WW, the second peak shows the growth curve of the catch crop.*

To this end, we collected a one year history of measurements on each field, summarized into an average value per timestamp. This data was extracted from DMDB.

- NDVI history (for the past year):
    - date of measurement
    - average NDVI value on the field (number between 0-254)
    - standard error of the average (number between 0-254)

Lastly, we collected historical climate variables measured at daily resolution over the past two years from the harvest date.

- Climate history (past two years, each variable measured daily)
    - Mean air temperature (degrees celsius)
    - Global radiation (MJ/m²)
    - Minimum temperature (degrees celsius)
    - Maximum temperature (degrees celsius)
    - Precipitation (mm)

## Data validation

Each data set comes with its own host of outliers and malformed data entries which must be sieved out.

Since validated yield measurements bottleneck every other part of the analysis, we converted yield values to a common measurement scale (hkg/ha) and filtered out outliers (WW yield < 145 hkg/ha and SB yield < 125 hkg/ha). Of the remaining set, only those measurements with complete data in the static variables were kept. In other words, e.g. if a yield measurement lacked a soil type, it was discarded. A sample was also discarded if its harvest date preceded its sowing date.

After the static variables data set was finalized, the rest of the data sets were cleaned and prepared:

- Crop history
    - Catch crops were identified by their DMDB codes, being between 942 to 972, and annotated as such.
- NDVI history
    - Missing values were ensured not to exist.
- Climate history
    - Missing values were ensured not to exist.
    - Temperature variables were checked to make sure they lay within a reasonable range of values
    - Air temperature was checked to make sure it never exceeded the value in the maximum temperature variable, or was lower than the minimum value.
    - Global radiation was ensured to be strictly positive.
    - Precipitation was ensured to be strictly positive and never anomalously large.

## Data preparation

Once data sets were individually validated, we decided on various ways to compute *features* - representations of the data that align with the workings of the machine learning algorithms used downstream. Yield prediction is itself an unusual task, in that it doesn't easily lend itself to traditional regression or classification methods without some manipulation - data is recorded continuously over the growing season, but there is only one response variable available at the end.

We opted to summarize the time-evolving measurements by binning various time periods, see Figure 2 and computing summary statistics for each bin. Each of these summarizations becomes a feature in a regression or classification model, and its predictive importance can be divined from its coefficient in the model.

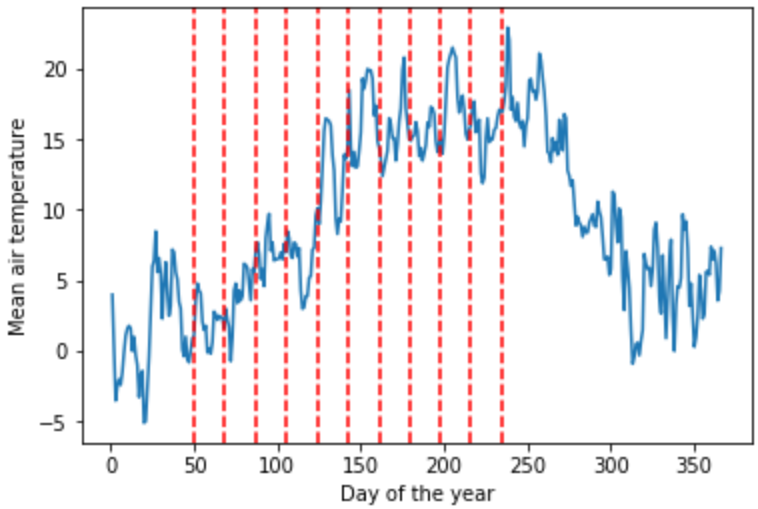



**Figure 2:** Mean air temperature measurements for a field across the year. Bins indicated by red lines; each bin defines an interval which is summarized into a feature value.

For each of the extraneous data sets, we decided on the following “featurization strategies”:

- Crop history:
    - A 7-year window backwards in time was extracted from the field’s crop history. Each year in the history was interpreted as a categorical variable denoting the main crop grown that year. If a particular year was missing a value - possibly due to the field having existed for fewer than 7 years - it was given its own “unknown” category, and kept in the feature set.
- NDVI history:
    - The two-year NDVI history was, for each field, characterized by very occasional measurements. Values for missing days of the growing season were filled in by nearest neighbor imputation.
    - Values within each bin were averaged to produce a feature value.
- Climate history:
    - Some fields lacked a climate history due to complications with the historical weather data. We imputed their feature values by filling in the global mean of each bin.
    - For each bin (date range) we computed one summary value per variable, in the following manner:
        - Mean air temperature was averaged
        - Global radiation was summed
        - Minimum temperature was summarized by the minimum value in the whole bin
        - Maximum temperature was summarized by the maximum value in the whole bin
        - Precipitation was summed

We chose bin placement by examining the approximate growth curves across different crops, and placed bins linearly within a date range that captured the interesting dynamics of the curve - periods with rapid positive or negative growth. For winter wheat, we chose bins uniformly between 19th of February and the 22nd of August. For spring barley, we chose bins uniformly between 20th of March and 27th of August. For consistency, the same bins were used to compute features for the NDVI data set and the climate data set.

All categorical features are normalized by using a label encoding. That is, for each categorical feature, all categories are enumerated from 0 to the number of categories minus one, e.g. if a categorical feature spans five different classes, these classes are encoded using the integers from 0 to 4.

The SB data set contains 5104 samples, whereas the WW data set contains 6040 samples.

## Feature sets

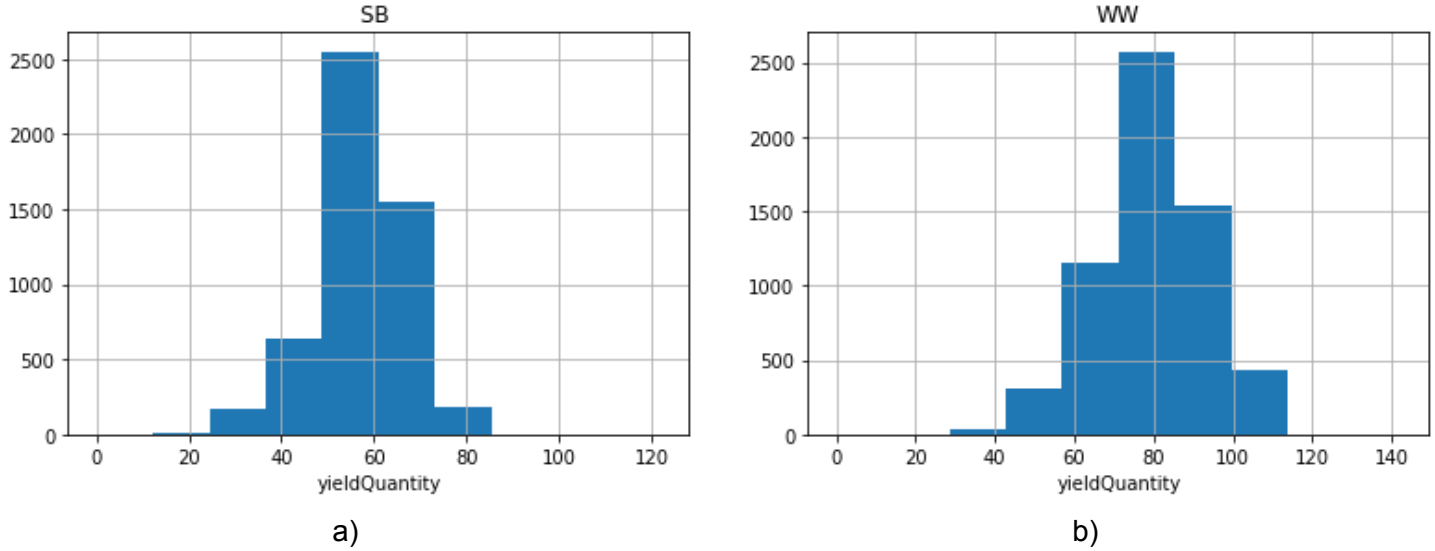
We divide our features into different data sets, to investigate the relation between the amount and what features are given the machine learning methods and the prediction performance of each method. Thereby, we construct the following Feature Sets (FS), where each feature is for a given field at a given harvest year.

| Name of feature set | Features contained | Number of features |
|---|---|---|
| FS1 | Static features | 6 |
| FS2 | FS1 and 7 years of crop history | 20 |
| FS3 | FS1 and NDVI history | 16 |
| FS4 | FS1 and climate history | 171 |
| FS5 | All features | 195 |

## Response variables

The response variable we want to predict with our machine learning methods is: the yield of a given field at a given harvest year. This response variable can simply be given to our utilized regression methods directly.

However, the classification methods have to be given a variable spanning a finite number of classes. Therefore, we perform binning of the response variable to 10 bins, i.e. 10 classes to classify. The variable do not have a uniform distribution, as illustrated by the histogram in Figure 3 using the equal spaced intervals on the response variable. This distribution of classes in the data set, i.e. where few classes contain many samples and vice versa, have the disadvantage of making the classification methods biased towards the classes with many samples.




a) b)

**Figure 3**: *These histograms illustrate the yield quantity in hkg/ha where the bins are equally spaced intervals. a) illustrates fields with spring barley and b) illustrates fields with winter wheat.*

To prevent this disadvantage, we perform a binning to obtain a nearly uniform distribution of the samples over the 10 classes, by splitting the bins on each 10'th percentile of the samples, as illustrated by the histogram in Figure 4.

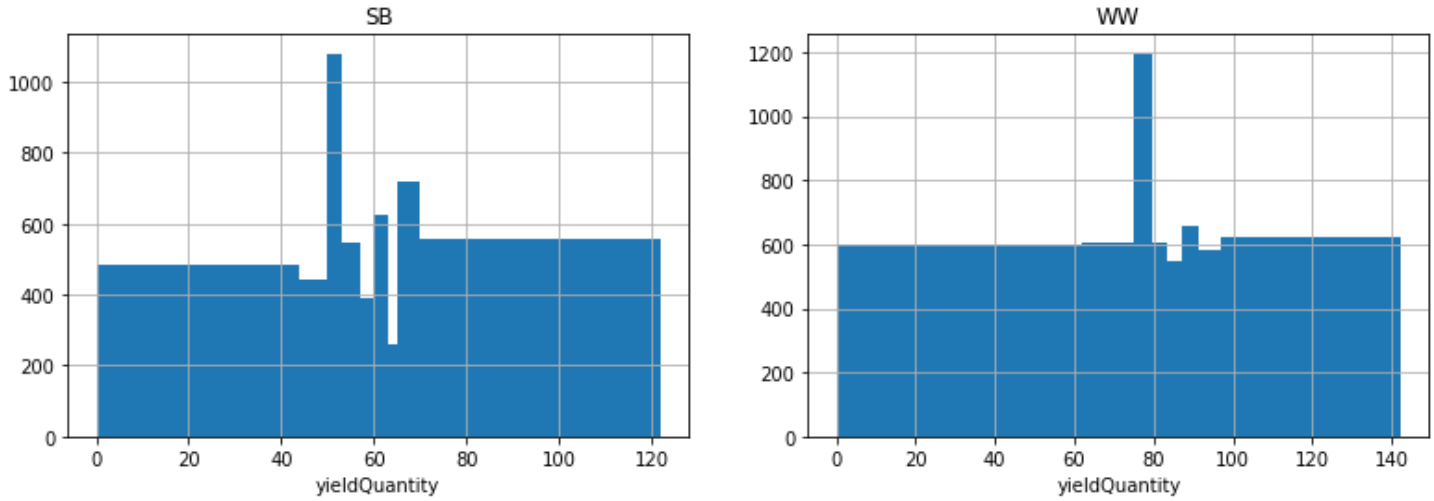



**Figure 4**: *These histograms illustrate again the yield quantity in hkg/ha, but with bins with unequally spaced intervals, for obtaining a uniform distribution. a) illustrates fields with string barley and b) illustrates fields with winter wheat.*

## Regression and classification methods

In this initial test of machine learning methods for regression and classification of yield, we restrict our attention to well-proven baseline methods. In selecting the methods, we lean towards the recommendations given in the Scikit-Learn Machine Learning Map[2]. We use the implementations of the methods provided in Scikit-Learn and use all methods in their (Scikit-Learn) default configuration. In terms of regressors, we consider methods that are suitable for smaller samples sizes, i.e. we consider the regressors listed in the table below.

| **Regressor Name** | **Scikit-Learn Class Name** | **Type** |
|---|---|---|
| Linear Regression | linear_model.LinearRegression | Generalized Linear |
| Elastic Net | linear_model.ElasticNet | Generalized Linear |
| LASSO | linear_model.Lasso | Generalized Linear |
| Ridge Regression | linear_model.Ridge | Generalized Linear |
| Epsilon-Support Vector Regression (SVR) | svm.SVR | Support Vector Machine |
| Random Forest Regressor | ensemble.RandomForestRegressor | Ensemble |

As a baseline benchmark we also consider a dummy regressor that always predicts the mean value of the (training) target variable.

[2] http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Similarly, we consider the below listed well-proven classification methods.

| Classifier Name | Scikit-Learn Class Name | Type |
|---|---|---|
| Logistic Regression | linear_model.LogisticRegression | Generalized Linear |
| k-nearest Neighbors | neighbors.KNeighborsClassifier | Nearest Neighbors |
| Random Forests Classifier | ensemble.RandomForestClassifier | Ensemble |
| Multi-Layer Perceptron (MLP) | neural_network.MLPClassifier | Neural Network |
| C-Support Vector Classification (SVC) | svm.SVC | Support Vector Machine |

As a baseline benchmark we also consider a dummy classifier that uses the “stratified” classification strategi, i.e. it generates random predictions based on the (training set class) distribution of the target variable.

## Performance indicators

A set of performance indicators, i.e. score functions, is used in the evaluation of the performance of the regression and classification methods. Specifically, we evaluate the performance of the regression methods in terms for the mean absolute error and the coefficient of determination ($R^2$). The mean absolute error (implemented by the function sklearn.metrics.mean_absolute_error) is a measure of the average absolute difference between the predicted value and the true value. Ideally, the mean absolute error equals 0. The coefficient of determination (implemented by the function sklearn.metrics.r2_score) is a measure of the fraction of variation in the data set explained by the model - a goodness of fit measure. Ideally, the coefficient of determination equals 1. The performance of the classification methods is evaluated using the accuracy score (implemented by the function sklearn.metrics.accuracy_score), i.e. the fraction of predictions that match the corresponding true value. Ideally, the accuracy score equals 1.

We perform 10-fold cross validation to evaluate the performance of the different machine learning methods. K-fold cross validation is a technique to execute a prediction method on K partitionings of the data set. Thus, with 10-fold cross validation, the data set, containing the features and true values, is partitioned into 10 individual data sets. The machine learning method is then executed 10 times, where each execution rotates the data sets, such that the nine of the 10 data sets is used as the training set and the one of the 10 is used as the test set. As a result of this technique, the mean of the performance indicators over all 10 executions is reported.

## Computational environment

All experiments were conducted on a workstation featuring an Intel Xeon E3-1270v5 CPU and 32 GiB RAM. The workstation was running Microsoft Windows 10 Enterprise, 64-bit, build 1703. All computations were done in double precision floating point representation using scientific Python packages. Specifically, we used the packages available in the Anaconda Python Distribution[3] version 5.0.1 based on Python 3.6.

[3] https://www.anaconda.com/download/

# Results

Figure 5 illustrates the mean absolute error performance for all regression methods using the different feature sets (which are overlaid - not stacked). Both train and test results are shown for WW as well as for SB. The RandomForestRegressor obtains the smallest test mean absolute error (7.65 hkg/ha for WW and 6.45 hkg/ha for SB), and thus performs the best across all of the regression methods. Generally, a lower mean absolute error is obtained when including more features in the model. In particular, the inclusion of the climate history results in a large improvement in performance.

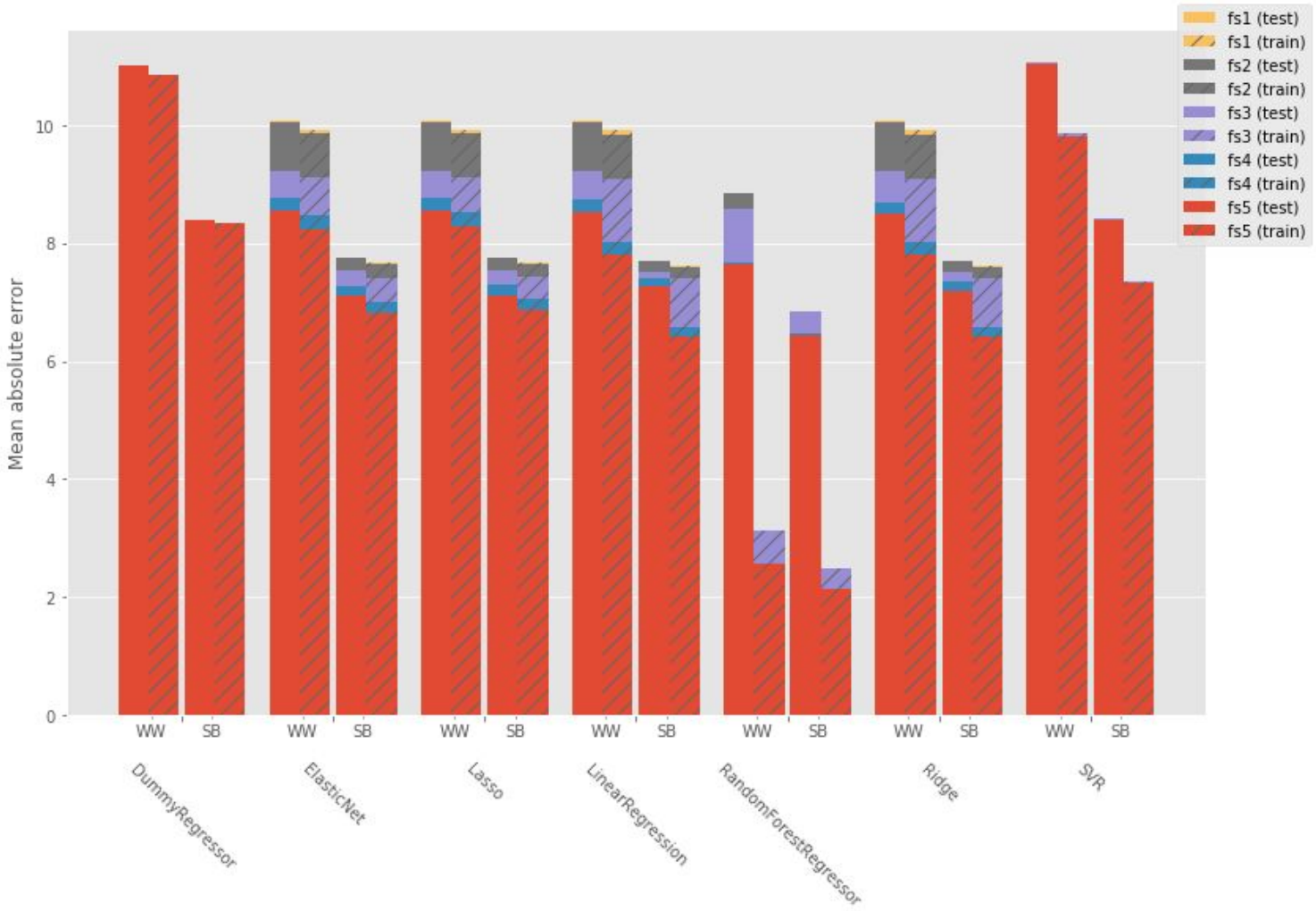



**Figure 5**: Mean absolute error performance for all regression methods using the different feature sets for WW and SB. Note that the feature sets are overlaid (not stacked).

Figure 6 illustrates the coefficient of determination ($R^2$) in a composition similar to the one used in Figure 5. The RandomForestRegressor shows the highest test $R^2$ (0.35 for WW and 0.27 for SB), and thus performs the best across all of the regression methods.

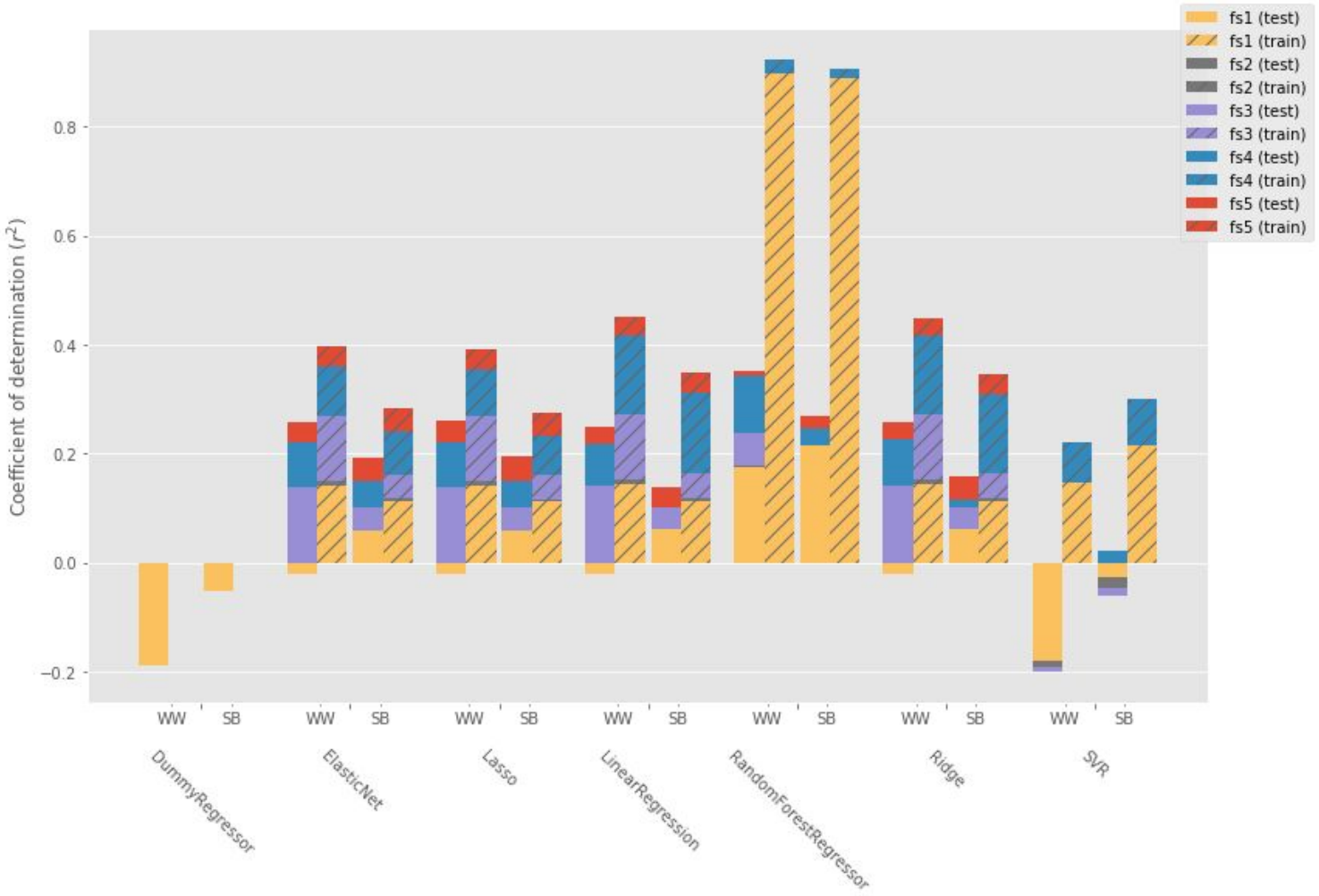



**Figure 6**: Coefficient of determination ($R^2$) performance for all regression methods using the different feature sets for WW and SB. Note that the feature sets are overlaid (not stacked).

Figure 7 illustrates the accuracy performance for all classification methods using a composition similar to the one used in Figure 5. The RandomForestClassifier obtains the highest accuracy (0.35 for WW and 0.35 for SB), and thus performs the best across all of the classification methods, even though the SVC and KNeighborsClassifier performance is almost as good. As was the case for the regression results, a significant improvement in classification accuracy is obtained when including more features in the model. In particular, the inclusion of the climate history results in a large improvement in classification accuracy.

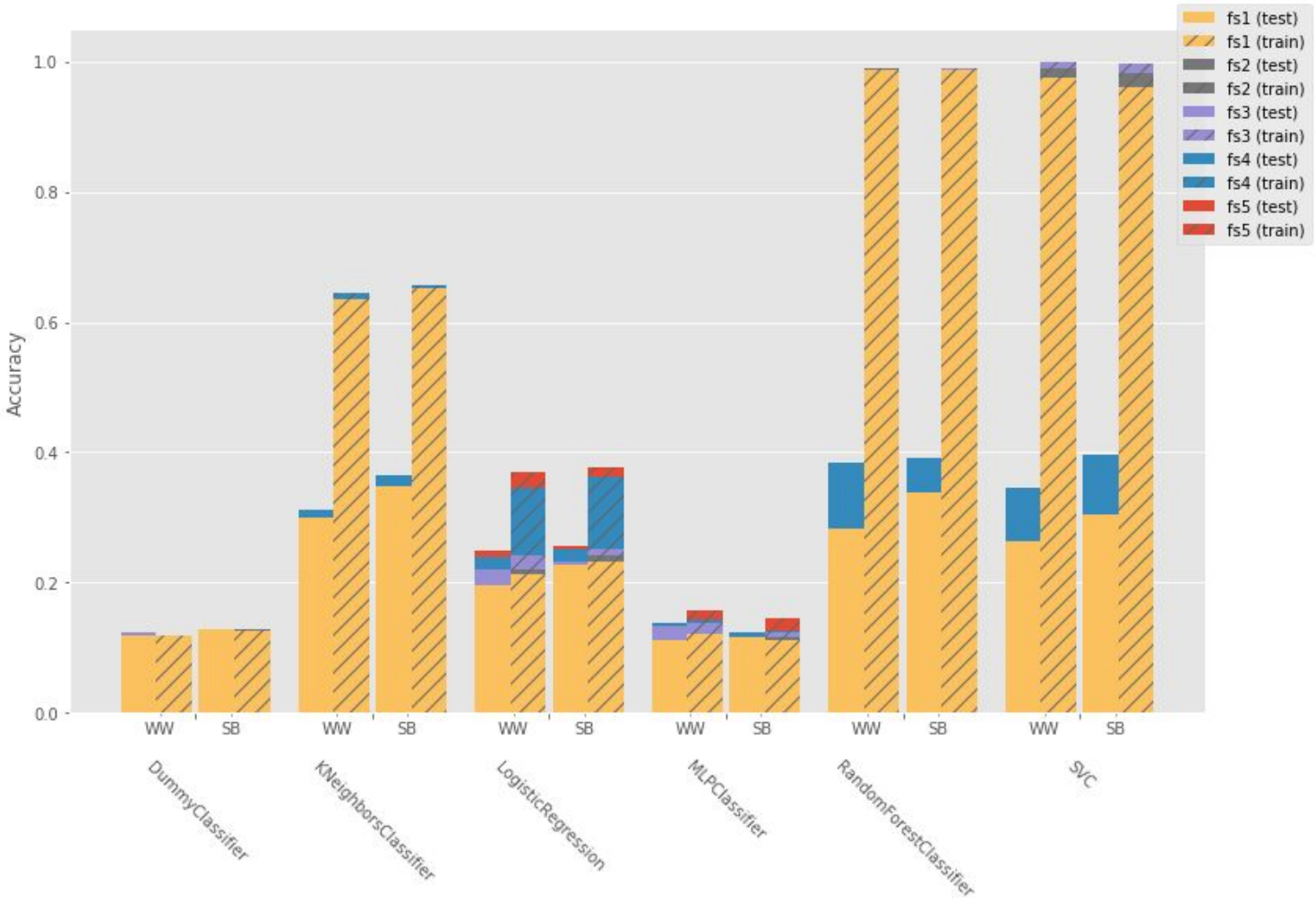



**Figure 7**: Accuracy performance for all classification methods using the different feature sets for WW and SB. Note that the feature sets are overlaid (not stacked).

# Discussion

The yield regression results are generally somewhat poor. Performance levels are only marginally better than the DummyRegressor. All regressors show a mean absolute error on the order of 6 to 11 hkg/ha for the test data. The only regressor that obtains a low mean absolute error and high coefficient of determination on the training set is the RandomForest regressor. Unfortunately, this successful separation of the training data does not generalize to the test data showing that the RandomForest regressor suffers from severe overfitting. The use of larger feature sets generally improves the test results. Of the additional features, the climate history appears to contribute the most to obtaining a better separation of the data set. Overall, the results are unsatisfactory suggesting that more data samples are needed to successfully fit a model that generalizes well. Furthermore, the failure of the linear methods and the training success of the RandomForest regressor suggests that non-linear regressors are needed in order to separate the data set.

The classification results are somewhat more satisfactory than the regressor results. Both the RandomForestClassifier and the SVC are able to separate the training set and perform significantly better than the DummyClassifier. Unfortunately, overfitting is still a major concern for these classifiers as seen in the large difference in accuracy between training and test data. Again, this suggests the need for more data records as well as more (or other) features that provide a more clear separation of the classes in order to fit usable model.

In the data collection and transformation process, our choice of methods inherently implies various approximations and assumptions that may impact our final results in negative ways. Specifically, future work should consider ways to handle the following potential issues:

- Handling of missing values: The data set suffers from a significant number of missing values. We handle this problem by partly leaving out some of the records while imputing others. If all records suffering from missing values are simply left out, the size of the resulting data set becomes critically small. Ideally, these missing values are filled using data from other sources. If the data simply does not exist, one should consider ways to define optimal imputation strategies.
- Ordering of categorical features: The current encoding of categorical features (integer labels from 0 to “max_value - 1”) implies an implicit ordering and distance between the features. Such an ordering may not be justifiable, e.g. an ordering of farms based on their (arbitrarily) assigned farm IDs cannot be justified. Ideally, in such cases, one should use a One-Hot encoding or another encoding strategy that does not imply any order among the categorical features.
- Assertion of integrity of data: Currently, we assume that all records marked as “registered” have been verified. In this set of verified records, yield quantities of 50.0 hkg/ha for Vaarbyg and 75.0 hkg/ha for Vinterhvede appears to be significantly overrepresented. Such anomalies in verified records should be further investigated.
- Choice of regressor and classifier parameters: All regressors and classifiers have been used in their default Scikit-Learn configuration which may be suboptimal for the problem at hand. Ideally, a grid search or similar strategy should be used to tune the hyperparameters to the problem at hand.
- Handling of poorly represented classes in classification: In order to optimally train a classifier, a significant number of examples of all possible classes must be included in the training set. Currently, this is not the case for the available registered data. Ideally, one should collect more data to adequately represent all possible classes. If this is not feasible, the problem may be somewhat mitigated using oversampling strategies.
- Justification of the field/year independence assumption: Currently, all data records are indexed by their field ID and the harvest year. We assume that all such records are independent which is a rather naive assumption. However, our use of crop-, ndvi-, and climate histories potentially captures all relevant dependencies. Ideally, this issue of capturing the dependence among data records should be further investigated.

In a addition to finding strategies for handling these potential issues, one may also consider the following avenues for future work on improving the current regression and classification results:

- Including more data sources, e.g.
    - Location of fields (e.g. UTM coordinates).
    - Soil samples.
    - Field management history (e.g. use of fertilizer and pesticides).
    - Elevation and orientation of field (if on a hillside).
    - High resolution climate data (hourly measurements).
    - Raw spectral data or additional vegetation indices from Sentinel satellites.
- Restructuring of features, e.g.
    - Using degree days instead of average temperature.

    - Using the crop type as a feature instead of training separate models for each crop type.
- Using other machine learning methods. Our results indicate that non-linear models are needed in order to separate the different classes from each other. We have only considered the very basic non-linear models. This leaves a substantial set of other non-linear models that may potentially be used to obtain improved regression and classification results.
- Incorporation of additional information obtained from inspection of and/or unsupervised learning from the training data.
- Adaptation of methods to allow for prediction of the response variables at time instances placed throughout the harvest year, e.g. predicting yield midway through the growth season.