UiO : University of Oslo

FYS-STK4155 – Applied Data Analysis and Machine Learning

Project 3: Comparing selected machine learning approaches for mapping the distribution of vegetation types in Norway

December 16, 2020

Authors:

Håkon Berggren Olsen

Lasse T. Keetz

Lecturer:

Prof. Morten Hjorth-Jensen

Abstract

Since approximately two decades ago, the field of machine learning provides the ecological modelling community with new instruments that often outperform more conventional methodologies. However, as the field is still undergoing perpetual and rapid changes, the additional potentials for ecological research become more and more evident. In this report, we present a real-life case study of using supervised multi-class classification in the domain of Distribution Modelling. Specifically, we compare the performances of the Random Forest approach to an artificial neural network (ANN) using deep learning. The models are trained to predict the distribution of observed "Vegetation Types" across mainland Norway based on a set of 73 continuous and categorical "wall-to-wall" environmental rasters with a spatial resolution of 100m^2. We find that even though the overall modelling accuracy is not satisfactory for a general reliable vegetation map (best ANN model accuracy median on test data = 0.45), the results are promising and reveal possible future improvements. Most prominently, we show that adding a novel set of remotely sensed vegetation indices and spectral reflectances (SENTINEL-2A) in the feature matrix improves the classification performance. While the ANN performed slightly better compared to the RF (5-fold cross-validation medians of 0.45 ANN vs. 0.43 RF, respectively), the latter is considerably faster and arguably easier to use and interpret. In a second exercise of the report, we present a custom RF implementation written in Python that has similar performance to the RF implementation in scikit-learn, using the famous Iris flower dataset for benchmarking.

Contents

1 Introduction

The field of Machine Learning (ML) provides state-of-the-art toolboxes to answer and to generate research questions across scientific disciplines. Commonly, the versatile approaches are divided into three categories depending on the nature of the problem they address. **Supervised learning** deals with providing example inputs and associated labelled outputs in order to make the computer ”learn” potential underlying relationships (e.g. classification and regression). **Unsupervised learning**, on the other and, concerns automatically finding patterns in unlabeled and unstructured data (e.g. clustering). Finally, **reinforcement learning** aims at teaching a computer program to thrive towards a desired outcome by rewarding ”correct” behaviour while getting feedback from its dynamic environment (e.g. automated driving). One distinct characteristic of all ML algorithms is their capability to extract information from vast amounts of data without the need of being explicitly programmed for a specific task. This implies that possible relationships between input variables are automatically detected, i.e. inferred through analytical and/or statistical computations. In addition, the predictive performance of ML approaches is often significantly superior to conventional methods (e.g. Cutler et al., 2007).

This report presents a real-life example of applying machine learning within the field of bio-geography. We will use two different algorithms - Random Forest and artificial neural networks - to predict the spatial distribution of distinct vegetation types (VTs) across Norway. Accordingly, this case constitutes a supervised multi-class classification problem. The presence of a VT is spatially related to a set of 56 continuous and categorical environmental predictor variables (EVs). The theory behind the approach stems from the field of ”Distribution Modelling”, which is shortly introduced later in this chapter. The performance of the two algorithms will be compared and critically analyzed with a special emphasis on interpretability. In addition, we present a custom implementation of the Random Forest algorithm (Breiman, 2001), written in Python (Van Rossum and Drake Jr, 1995). Its performance is evaluated on a famous benchmark data set, the *Iris* flower data (Anderson, 1936; Fisher, 1936), and compared to the ”Random Forest Classifier” implementation contained in the popular Python package ”scikit-learn” (Pedregosa et al., 2011).

1.1 Aims and motivation

Land cover maps are a pivotal tool to monitor the state and changes of the earth surface (Jinru and Su, 2017), and mapping vegetation distribution in particular has high relevance for numerous environmental, societal, and economic demands (Bryn et al., 2018). Plants depict a key element in the surface energy balance by governing optical properties of the earth surface, most prominently the surface albedo (Duveiller et al., 2018). They affect the surface energy transport, e.g. by influencing turbulences through altering the surface roughness length (Richardson et al., 2013). Vegetation also influences hydrological cycles and evaporation rates on different scales, thereby mediating latent and sensible heat fluxes (Barrett et al., 2016). In addition, plants are an important component of the global carbon cycle. For instance, they withdraw carbon from the atmosphere and store it in their biomass both above and below ground through the process of photosynthesis. Consequently, knowing the currently realized vegetation distribution is a cornerstone for quantifying the climate system and the carbon cycle. Future feedbacks between terrestrial ecosystems and the global carbon cycle are currently considered one of the largest uncertainties in climate model projections (Müller et al., 2016; Forkel et al., 2019).

Moreover, mapping plant species and communities over large areas is an important tool to monitor the human impact on the natural environment and to survey and protect biodiversity. Accurate vegetation maps can, therefore, also contribute to sustainable environmental management and planning (Franklin et al., 2013; Henderson et al., 2014). Especially in areas with low population density and rugged terrain, such as Norway, acquiring sufficient land cover information from field-based sampling is often challenging, for example due to economic restrictions (Bryn et al., 2018). The field of distribution modelling offers a complementary alternative.

1.2 Distribution Modelling

Distribution modelling (DM) is a well-established tool in ecological research. In short, it implies predicting the distribution of organisms across space and time from environmental input data. The approach is strongly related to the "ecological niche theory", or the potential habitat suitability for different species with regards to prevailing biotic and abiotic environmental conditions (see e.g. Araújo et al., 2019). Accordingly, the field is by no means limited to terrestrial plant distribution and also deals with predicting animal habitats, for instance. DM for vegetation mapping implies predicting plant species or community occurrences (Bricher et al., 2013; Nieto-Lugilde et al., 2018) as a function of environmental predictors. Specifically, spatial occurrence or abundance data (e.g. from field surveys) is statistically related to variables such as mean temperature and precipitation at the same location. Furthermore, DM approaches are often methodically distinguished as presence-only and presence-absence approaches, each with individual advantages and limitations (Phillips et al., 2009).

However, the relationships between realized distribution (i.e. currently observed), potential distribution (i.e. theoretically "expected" based on environmental conditions), and measured environmental variables is often highly complex and non-linear. For instance, observed distributions may differ substantially from "expected" distributions due to biotic or anthropogenic interference, which can be complex to model (Guisan and Zimmermann, 2000).

Here lies the potential of ML approaches. Common techniques, including random forests and neural networks, have already been shown to significantly improve model accuracy in various domains in ecological modelling (Cutler et al., 2007). Applications include, for instance, modelling realized as well as potential plant species distribution (Bricher et al., 2013; Hengl et al., 2018), community level modelling (Nieto-Lugilde et al., 2018), and the spatial attribution of modelled land surface energy fluxes to environmental variables (Lian et al., 2018).

1.3 Vegetation Types

The target variable that represents the vegetation distribution in this report is referred to as "vegetation types" (VTs). This classification system spatially agglomerates plant communities based on species composition, indicator species, physiognomy, or a combination thereof. In theory, each VT occupies a unique niche in the ecological space and is, thus, the emergent outcome of the underlying biotic and abiotic environmental processes. The thematic detail hereby depends on the vegetation type mapping unit and aims at best representing and distinguishing relevant ecological processes at a desired spatial scale (Bryn et al., 2018).

The VTs from the data set we use in this project are part of a Norwegian national land cover classification scheme that is described by Rekdal and Larsson (2005). It comprises a total of 57 basic land cover classes out of which 45 describe vegetation cover. Table 4 in the appendix contains a full list of the land cover classes. For additional information in English, refer to Strand (2013).

1.4 Theory

The following section provides brief overviews over the theories behind the chosen machine learning methods, i.e. Random Forest and artificial neural networks. In addition, it describes a method that quantifies the influence of individual input features on the final classification performance, namely permutation feature importance.

1.4.1 Decision trees and Random Forest

Decision trees are an intuitive classification scheme in which the explanatory variables/features are bifurcated into leaves based on a certain threshold. A geometric and diagrammatic representation of the idea is shown in figure 1 (Protopapas et al., 2017).

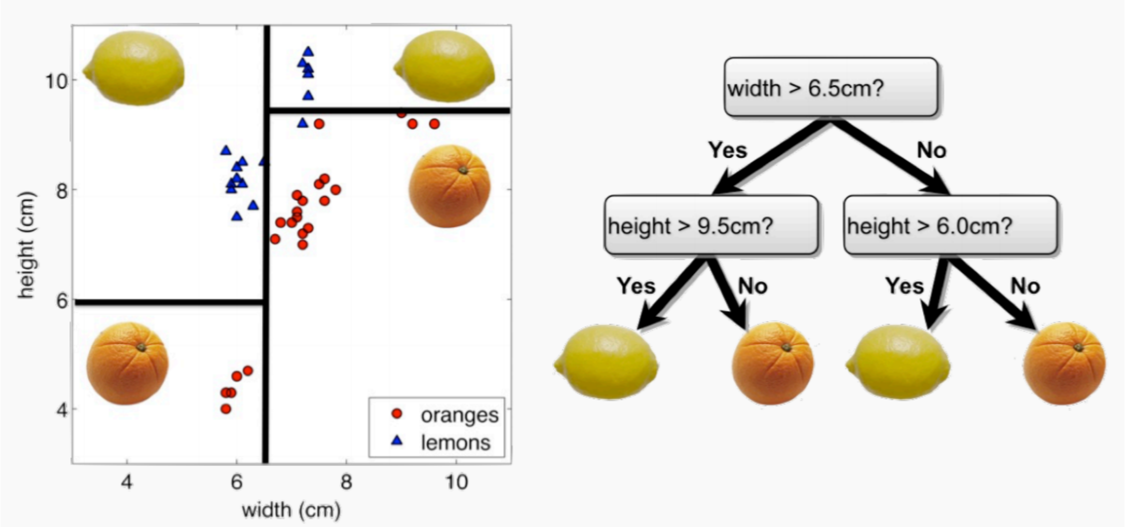

Figure 1: *To the left is the feature space where the regions predict the label. To the right is the diagrammatic tree which shows the thresholds for the features. Figure from Protopapas et al. (2017).*

Gini index is a measure of the variance in the classification, and are used to measure the purity/impurity of a threshold split (Hastie et al., 2017). It is given by taking the probability of a training example being in region m (so the m-th node of the tree), being classified as class k.

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{1}$$

Similarly, entropy is also a measure of the purity/impurity of a node m:

$$D = \sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk}) \tag{2}$$

Even as the equation for entropy in a node carries a different meaning, the numerical behaviour is still similar and we would likewise get low values for high purity nodes, and higher values for impure nodes.

Decision trees are bound to suffer from a high amount of variance as they can be perfectly fitted to the training data. A way to curb the variance is to do bagging (bootstrap-aggregating). A new data set is created with bootstrapping, leaving some of the data points out and creating a tree based on the bootstrapped data sets. By utilizing all the difference predictions, one would lower variance. This is done by either averaging the results (regression) or having each of the model vote on the final predictions (classification). The out of bag samples (OOB) can be evaluated to estimate the test error of the model (Hastie et al., 2017).

Random forest (RF) is a collection of decision trees like those created by bagging. However at each split, only a subset, m, of the explanatory variables/features are considered, usually $m \approx \sqrt{p}$, where p are all the available features. This is called decorrelating, as the model is allowing less than half of the features, $m \approx \sqrt{p}$, and thereby prohibits the strong predictors (features with strong correlation to the outcome) to be presented in the outcome Hastie et al. (2017).

```
Data: $X_{train}$, $y_{train}$, mtry, num_trees
Result: Fit random forest
Split data into training and testing;
for $tree \leq num\_trees$ do
    Create bootstrap sample from $X_{train}$ and $y_{train}$;
    Randomly select specified number (mtry) of columns (features) from $X_{train}$;
    Use subset of data and features to grow the current tree;
    while $depth_{tree} \leq depth_{max}$ and Gini index in node $\neq 0$ do
        Split data into new nodes the best way possible (measured by lowest Gini index) by testing
          all available features and thresholds;
        Determine and store predicted classes of each node (most frequent class based on samples
          that are assigned to the node);
    end
end
```

Same, but for prediction:

```
Data: X$_{pred}$, mtry, num_trees
Result: Predict with random forest
Input data for prediction;
for $tree \leq num\_trees$ do
    Retrieve values from the feature matrix from the subset of features that were used to grow the tree (based on mtry);
    while $Current\ node \neq terminal\ node$ do
        Check current node with feature values;
        "Walk down" the input nodes (left if current feature value smaller than threshold, right if larger);
    end
    Store predicted classes of the terminal node (most frequent amount of samples that were assigned to it) as the vote for this tree;
end
Finally, count votes of each tree;
The prediction with most votes becomes the final prediction (majority voting);
```

1.4.2 Neural networks and deep learning

The theory of backpropagation for deep neural networks is more thoroughly explained in the report for project 2 (Olsen et al., 2020). However, we will be using additional methods to evaluate the data that were not previously discussed. Leaky ReLU was discussed in Olsen et al. (2020) and is a adaptation of the rectified linear unit where some of the values $x \leq 0$ are let trough by a value α, to not completely cutoff information.

$$\text{LeakyReLU}(x) = \begin{cases} \alpha x & x \leq 0 \\ x & 0 \leq x \end{cases} \tag{3}$$

Also for the neural network classification we will be using the softmax function as the last activation function to evalute the class, given by

$$(x) = \frac{e^{x_j}}{\sum_{c=0}^{C} e^{x_c}} \tag{4}$$

To combat overfitting in Olsen et al. (2020) both $l1$ and $l2$ regularization were used. Here, the weights, analogous to estimators, are assigned a penalty term with either $l1$ or $l2$ norm. $l2$ norm is the euclidean space and is given by $\sqrt{l^2}$, whereas $l1$ norm is just the absolute value and can be analogous to taxicab geometry (Black, 2019). Another way to combat overfitting, including regularization which is also previously discusses, is to implement dropout in the layers. The "Dropout rate" assigns the weights in a layer a percentage chance to not be changed during back-propagation minimization.

Adamax is a variant of the Adam optimizer (Kingma and Ba, 2017) (see also Olsen et al. (2020) for our discussion of the Adam optimizer). As the Adam optimizer calculates the rolling average (biased first momentum estimate) and divides it by the rolling variance (second raw moment estimate), Adamax divides the rolling average with the exponentially weighted infinity norm (Kingma and Ba, 2017). Finally, the cost function we use is "categorical cross entropy", since we are doing classification (see Olsen et al., 2020).

1.4.3 Permutation feature importance

Permutation feature importance is the idea of shuffling the order of the explanatory variables/features to estimate what features are more important in the model, i.e. having the most impact on the prediction. To calculate the importance of a value, a shuffled feature is predicted upon many times and averaged upon. The importance is then calculated by taking the score of the unshuffled prediction and subtracting the mean of the shuffled prediction Breiman (2001).

The benefit of using this method in contrast to Gini or entropy purity is that this method is not biased as it does not favor data with low degree of possible outputs. If some of the features are binary, i.e. having two possible outcomes 1 or 0, the Gini or entropy purity would be imprecise. Permutating the features instead and only evaluating the outcome would give us a higher degree of importance to this feature (Breiman (2001)). A caveat of this implementation arises when some of the features are highly correlated. Testing the permuted feature importance for both of these features will give a low importance because if one of them are permuted, the other one will be non-permuted and still feed the model the information. And thus it might appear that both features are "un-important" even though they are crucial for the model performance (Breiman, 2001).

2 Methods

2.1 Data

2.1.1 Vegetation distribution: AR18x18

The vegetation type distribution data stems from a ground truth wall-to-wall field survey with the name "AR18x18". It contains sample plots across the Norwegian mainland on an 18x18 km spatial grid (Figure 2) and is presented in detail by Strand (2013) and Bryn et al. (2018). Each grid cell intersection hereby constitutes a sample plot location (Figure 2).

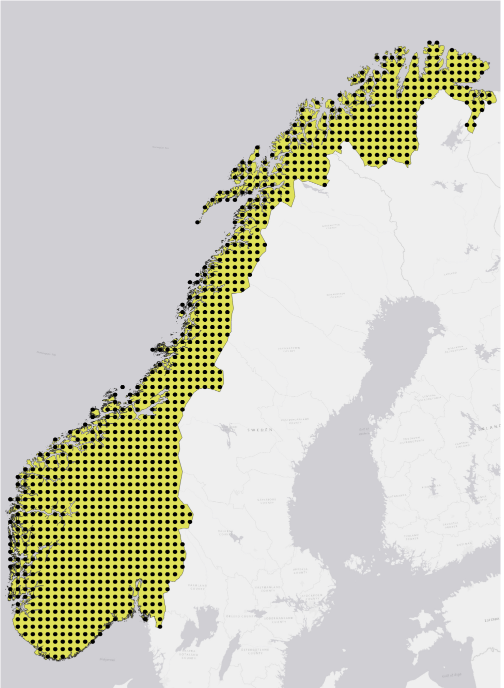

Figure 2: *The spatial design of the AR18x18 land cover survey. The locations of the plots (black circles) form a 18km grid across mainland Norway.*

Each sample plot represents a total area of 0.9 km^2 (1500x600 m; Figure 3) in which a group of experts classified the dominating land cover into a set of homogenous polygons (min. polygon size: 0.5 ha). Accordingly, each polygon represents one of the 54 land cover classes presented in subsection 1.3. The majority of the data was collected between 2004 and 2014. Subsequently, one random point was generated within each polygon that is finally used as the target class and location in the feature matrix. The points have a buffer to the polygon borders to avoid delineation problems during data collection and only one point is generated per polygon to minimize redundancies in the data caused by spatial correlation. This method is detailed in Horvath et al. (2019). In total, this resulted in $n = 22,154$ observations of vegetation types.

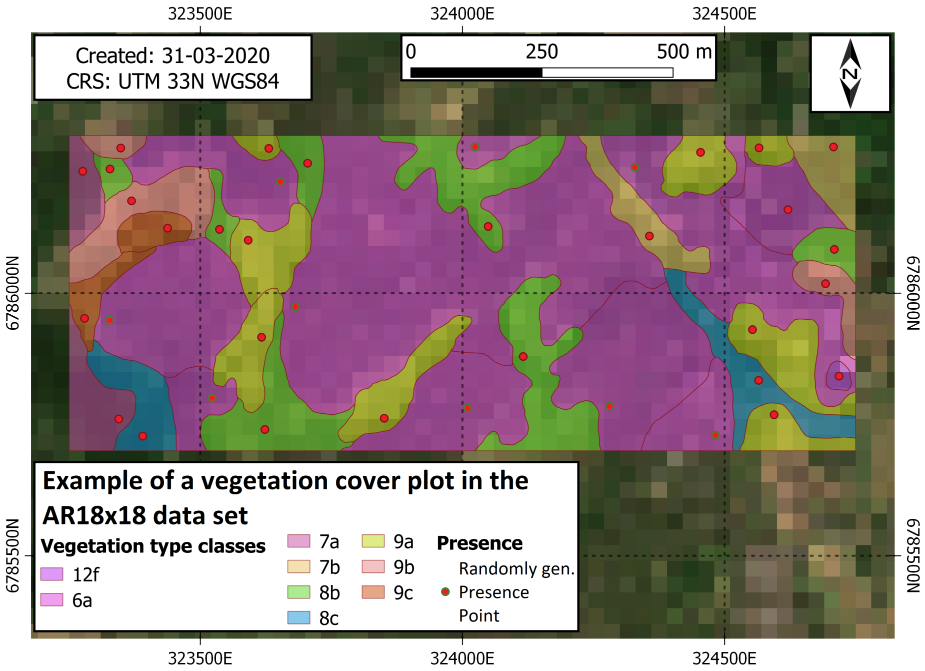

Figure 3: *Map of a randomly chosen sample plot (size: 0.9 ha) within the AR18x18 data set (total: 1,018 plots). The polygons of different colors depict field observations of coherent vegetation types as determined by experts. The red points are randomly generated and used for extracting values from the environmental rasters for the feature matrix ($n = 22,154$). The background pixels are an RGB composite of SENTINEL-2A imagery.*

2.1.2 Environmental predictors

Most of the environmental predictors used in this study were adopted from Horvath et al. (2019). Describing the data sources and provenance extensively is out of the scope of this project. Please refer to the original reference, the references therein, and supplementary materials for details. The predictors were available in a geographic raster format (e.g. GeoTIFF) and cover the entire extent of Norway (”wall-to-wall”). If necessary, the rasters were transformed to the Cartesian UTM33N WGS84 coordinate reference system.

As some data sources were originally provided in varying spatial resolutions, common interpolation and resampling techniques were applied so that all rasters are aligned with a cell size of 100x100 m^2.

The full set comprises 10 categorical and 45 continuous variables (total: n=55). Transforming the categorical variables into binary format (one-hot-encoding) results in a total of 84 input features. The environmental variables can be thematically grouped into the following domains (abbreviation code and examples in brackets): geology (GEO; e.g. nutrient content in bedrock), land cover (LC; e.g. "Forest" from the AR50 survey), topography/topology (TOPO; e.g. elevation above sea level / DEM), hydrology (HYD; e.g. proximity to rivers), snow (SNOW; e.g. snow water equivalent in April), and climatic (CLIM; e.g. mean annual temperature). Refer to Table 4 in the appendix for a full list of the variables.

Moreover, a new set of remotely sensed variables was added to the environmental predictors. The spectral data originates from Copernicus Sentinel-2A satellite imagery (Drusch et al., 2012). The satellites have a revisit time of 5 days and collect the spectral reflectance of 13 different electromagnetic bands. Furthermore, the product is preprocessed to account for atmospheric disturbances and, thus, provides readily usable surface reflectance values. The data was obtained and further processed using the cloud computing software Google Earth Engine (Gorelick et al., 2017). The final output values were produced as follows:

1. Retrieve an image collection containing all images between the available period, i.e. 2018-2019
2. Use a cloud filtering algorithm to remove images with too high cloud cover
3. Use remaining images to produce a "Greenest pixel" composite (see https://developers.google.com/earth-engine/tutorials/tutorial_api_06), i.e. generate one final pixel value for each 10x10m pixel during the period
4. Resample to 30x30m to reduce file size
5. Export data and transform to the UTM33N WGS84 coordinate reference system to match the other input data
6. Extract values at the vegetation type occurrence points (randomly generated in an assigned polygon of AR18x18 data set)

In addition to the raw spectral reflectance data, we calculated a selection of vegetation and spectral indices. For example, Tsai et al. (2018) showed that including vegetation indices in vegetation distribution modelling can improve classification accuracy. Table 1 shows the chosen indices and how they were calculated from the spectral reflectance values of different bands. Refer to (Xue and Su, 2017) for detailed explanations and evaluations.

Table 1: *Vegetation indices used as input features that were calculated from the SENTINEL-2A ”greenest pixel” composite.*

Index	Long name	Formula
NDVI	Normalized Difference Vegetation Index	$NDVI = (B8–B4)/(B8 + B4)$
EVI	Enhanced Vegetation Index	$EVI = 2.5 \times (B8 - B4)/(B8 + 6 \times B4 - 7.5 \times B2 + 1)$
EVI2	Enhanced Vegetation Index (alternative form)	$EVI2 = 2.5 \times (B8 - B4)/(B8 + 2.4 \times B4 + 1)$
SAVI	Soil Advanced Vegetation Index	$SAVI = 1.5 \times (B8–B4)/(B8+B4+0.5)$
NDII	Normalized Difference 819/1600 [IR]	$NDII = (B8 - B11)/(B8 + B11)$
TCW	Tasseled Cap index of wetness	$TCW = 0.1509 \times B2 + 0.1973 \times B3 + 0.3279 \times B4 + 0.3406 \times B8 + 0.7112 \times B11 + 0.4572 \times B12$

2.2 Study setup

This section details the experimental setups for the different objectives of the report. It further presents the preprocessing and resampling techniques we used for each setup. Finally, we briefly introduce and motivate the evaluation metrics we chose to analyze and compare the classification results.

2.2.1 Experiments

The following paragraph summarizes the experimental setups. We assign a name to each experiment (**bold heading**) and the corresponding *acronyms* are used to further refer to them throughout the report.

A. Custom Random Forest Implementation - (RF_{custom})

For the first part of the project, we wrote customized code that implements the multi-class classification

Random Forest algorithm. To test its functionality, we use a common benchmark data set: the Iris flower data (Fisher, 1936; Anderson, 1936). It contains a set of n=150 observations of three different species of a flower genus (Iris setosa, Iris versicolor, and Iris virginica), along with 4 simple continuous trait measurements (Sepal length, Sepal width, Petal length, and Petal width). Therefore, the random forest will use multi-class classification to predict the species from the plant traits.

The performance of the custom implementation is compared to the "RandomForestClassifier" in the scikit-learn package (Pedregosa et al., 2011). It should be noted that due to time constraints, we could only implement basic features for our use case. However, the implementation includes the adjustable hyper-parameters *mtry* (number of random features used for growing the individual trees) and *max_depth* (limits the depth or levels of trees, may avoid overfitting). The CART decision trees (based on Gini coefficient) are a separate class. Hence, they may also be used as a separate classification algorithm. The implementation only supports bootstrap sub-sampling for fitting the data. We also included a function to calculate the out-of-bag (OOB) score: each tree stores a list with the indices of samples that were used during the bootstrap fitting. Then, the OOB-score is calculated by predicting each sample of the data used for model fitting with all trees that were not trained with this specific sample. The metric is representing the resulting classification accuracy. Therefore, the OOB-score servers as a proxy for the model bias.

Moreover, in a first step, we split the data into 80% train and 20% test sets for fitting the custom model. The results are subsequently evaluated using the chosen multi-class classification metrics and the confusion matrix. Finally, we compare the implementation to the scikit-learn classifier: we perform 10-fold cross-validation (both models get trained and tested on exactly the same subsets of the data) and create box-plots of the resulting OOB-scores, accuracies, and the weighted averages of the F1-scores to validate the robustness of the results. We use the macro average, i.e. unweighted mean average, to calculate precision, recall, and f1-scores.

B. Random Forest for vegetation distribution - (RF_{veg})

For the classification of the vegetation type distribution, we used the RandomForestClassifier() from scikit-learn (Pedregosa et al., 2011) due to its optimized computation time. The chosen hyper parameters for the final results were: n_estimators (i.e. decision trees in forest) = 500, split criterion = Gini index, max_depth = 15, max_features = square root of number of input features, and bootstrap sampling = True. These values are close to standard values in the literature (see e.g. Valavi et al., 2020) and it should be noted that time constraints prohibited a sophisticated hyper-parameter tuning process. Subsequently, we used 5-fold cross-validation (i.e. each test fold contains 20% of the data) to quantify model robustness. The resulting metrics are compared to the neural network classification.

C. Deep neural network for vegetation distribution - (NN_{veg})

We use the ”keras” Python library (Chollet et al., 2015) to create a sequential feed forward artificial neural network (NN), and scikit-learn (Pedregosa et al., 2011) for necessary preprocessing of the data. Firstly, the categorical variables in the feature matrix were transformed into binary variables for each category (”one-hot-encoding”) and the continuous variables were scaled using the sklearn StandardScaler(). Moreover, the target variable was also transformed into binary variables using the ”LabelBinarizer” functionality. We then designed a fixed architecture for the NN based on running tests, literature research, and knowledge from the course. The final setup is depicted in Figure 4. The goal was to develop a sophisticated architecture with multiple hidden layers and state-of-the-art hyper-parametrization to achieve high classification accuracy while simultaneously minimizing overfitting.

It is important to note that time and hardware constraints prevented us from running sophisticated hyper-parameter tuning procedures. Hence, we use standard values for most parameters (e.g. learning rate of 0.1 for the LeakyRelU function). We then used the same 5-fold cross-validation procedure as described for RF_{veg} to evaluate the model performance. The results are finally compared to the Random Forest output.

2.2.2 Evaluation metrics

Accuracy is the most common metrics to evaluate classification. It is best used in a case of trying to get as many correct classifications as possible, but in the case of imbalanced class distribution the less represented classes in the data set may have very wrong classifications with the metrics still being high (Grandini et al., 2020).

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{5}$$

Precision is a fraction of the true positives divided by all the positives given per class, and is therefore a measure of how trustworthy a ”positive” is, given by the model (Grandini et al., 2020).

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{6}$$

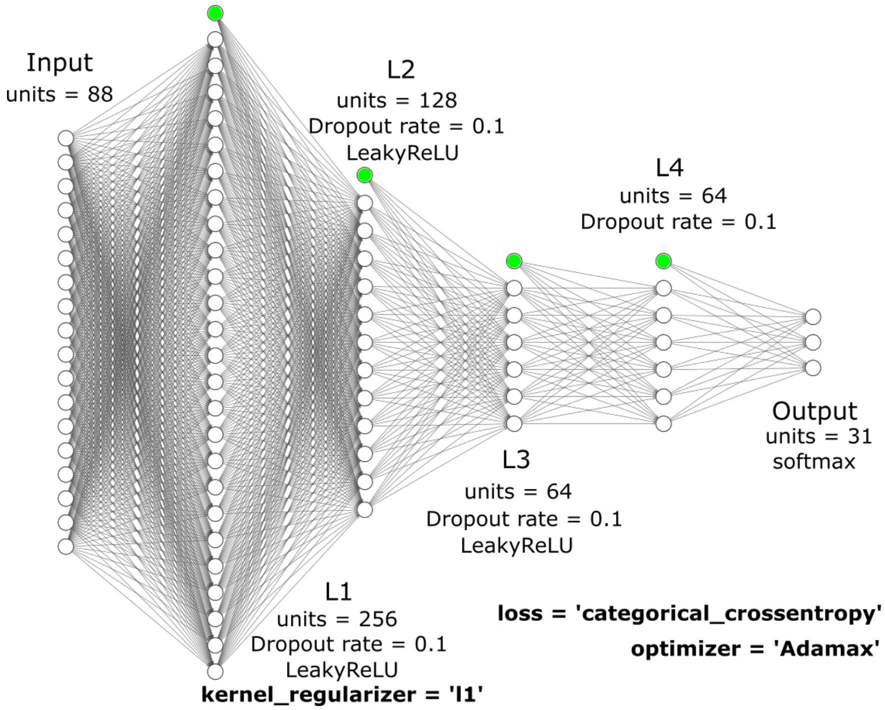

Figure 4: *Architecture of the deep neural network. The scheme depicts presences of bias units (green circles), number of units per layer (white circles), possible dropout rates, activation functions (alpha=0.1), possible regularization (lambda=0.1), the chosen loss functions in relevant layers, and the optimizer. Note that for simplicity, the amount of depicted units n is reduced (n/10, adjusted downward).*

Recall is a fraction of the true positives divided by both the true positives and the false negatives, and measures how effective the model is at finding a positive (Grandini et al., 2020).

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7}$$

Both of precision and recall can be viewed as column and row of 5, respectively.

F1-score is the harmonic mean of the precision and the recall, times 2. Unlike accuracy, the F1-score tends to give larger credit to smaller represented classes (if imbalanced class distribution), and favors more or less equal precision and recall (Grandini et al., 2020).

$$\mathrm{F}_1 = \frac{2}{R^{-1} + P^{-1}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{8}$$

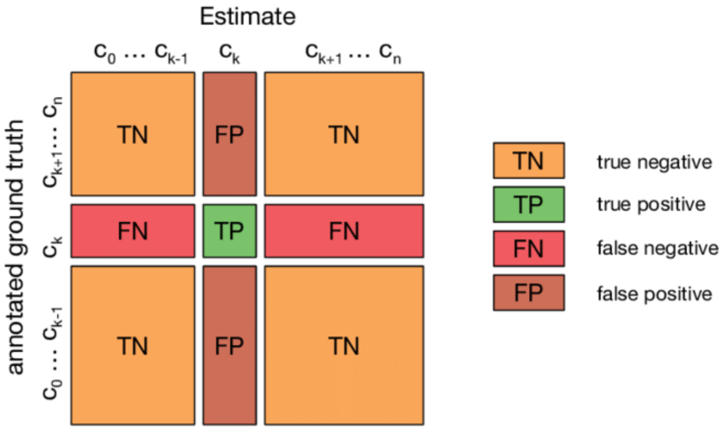

Figure 5: *Schematic representation of a confusion matrix for multi-class classification. Obtained from Krüger (2016).*

2.2.3 Permutation feature importance

As presented in the theory section, one common issue with calculating permutation feature importance is that highly correlated variables can severely distort the importance of individual features. We have, therefore, decided to adapt the approach described in Scikit-learn (2020). According to the Spearman's rank correlation coefficient, there is high correlation between certain groups of input features (Figure 6). Especially some remote sensing variables (Figure 6, lower right corner) and some climatic variables (Figure 6, center and top-left) are strongly interrelated.

To account for correlations and to generate more meaningful importance values for the different groups of features, we performed hierarchical clustering based on the Spearman rank order. Hierarchical clustering is, for instance, explained in Nielsen (2016). Here, we used a threshold value of "1" and "ward distance" as cluster forming criterion in the "fcluster()" function contained in scipy's "hierarchy" module (see code on github for details). Subsequently, we chose only the first variable within each cluster to train the Random Forest and neural network models. To ensure that the performance is still comparable to the models trained on all features, we split the data in 80% train and 20% test and evaluated the classification accuracy.

Finally, the permutation importance for both models is calculated by randomly permuting each remain-

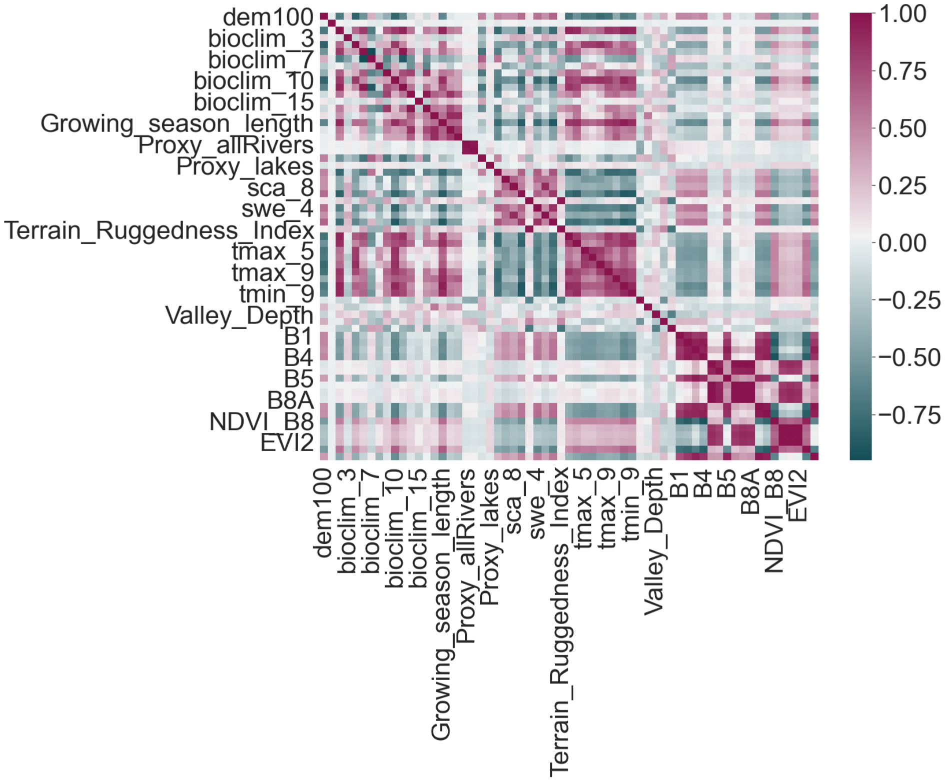

Figure 6: *Heatmap showing the spearman correlation between the input features. Note that most variable name labels are omitted for visibility, but that adjacent features often form thematic groups.*

ing feature 5 times, each time making new predictions on the test set. We then present the 5 most important features for each method based on the mean and standard deviation of the calculated permutation importance.

3 Results

3.1 Custom Random Forest

Table 2 summarizes important performance metrics on the Iris flower data using the custom Random Forest implementation. Under the given setup (randomly split 80% train and 20%test), there was a total of 2 misclassifications. Specifically, two actual *Iris versicolour* were predicted as *Iris virginica*. As there were $n = 30$ samples in the test set, the resulting classification accuracy was 0.93.

Table 2: *Confusion matrix with chosen metrics for the classification results (test set: 20% of the data).*

		Predicted							
		Setosa	Versicolour	Virginica	**Sum**	**Acc.**	**Precision**	**Recall**	**F_1-score**
Actual	Setosa	9	0	0	9	-	1.00	1.00	1.00
	Versicolour	0	9	2	11	-	1.00	0.82	0.90
	Virginica	0	0	10	10	-	0.83	1.00	0.91
	Sum / Total	9	9	12	30	0.93	0.94*	0.93*	0.93*

*: weighted average.

The metrics of the custom RF implementation compared to scikit-learn's RandomForestClassifier() using the same hyperparameter settings and 10-fold cross-validation are depicted in Figure 7. Both algorithms yield high, almost constant accuracy on the training data as measured by OOB-score (approx. 0.95). The accuracy and f1-score of both implementations on the test sets is also very similar and high (median approx. around 0.95), with slightly higher median values for the custom implementation, but more values closer to 1 for scikit-learn. Generally, there is higher spread among the test prediction metrics compared to the OOB-score.

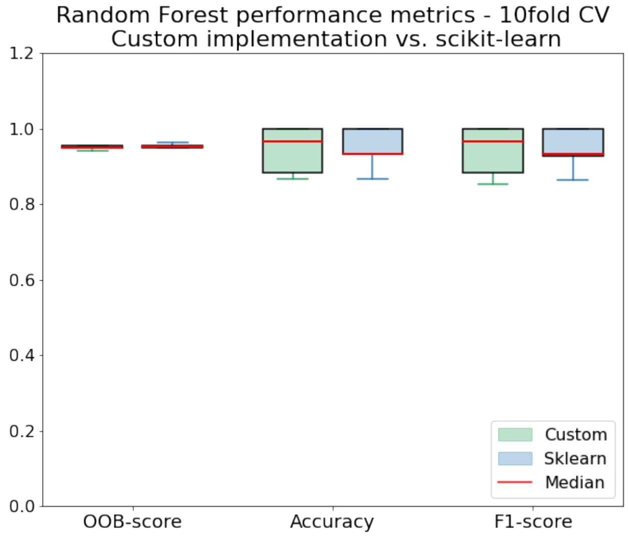

Figure 7: *Comparison of performance metrics (OOB-Score, Accuracy, F_1-score) between the custom Random Forest implementation and scikit-learn's RandomForestClassifier. The results are based on randomly shuffled 10-fold cross-validation. Both implementations used identical samples for training and test data.*

3.2 Vegetation type distribution

3.2.1 Model performances

Comparisons of model performances, scikit learns random forest and keras deep neural network with the use of 5 fold cross validation is shown in figure 8 as a boxplot. The neural network scored both a higher accuracy and a F1-score compared to the random forest. However, we can see that the precision is ≈ 50% higher for the neural network than for the random forest, and that the recall is ≈ 30% lower.

Figure 9 depicts the first two levels of nodes of one randomly chosen decision tree. It is a way to visualize that during the first split, for instance, spatial reflectance from band 1 with a threshold value of 0.03 was the variable that split the groups the best way in terms of lowest resulting Gini coefficients. Further, it shows that most observations in this bootstrap sample (n=11,184) belonged to vegetation type class ”2ef”.

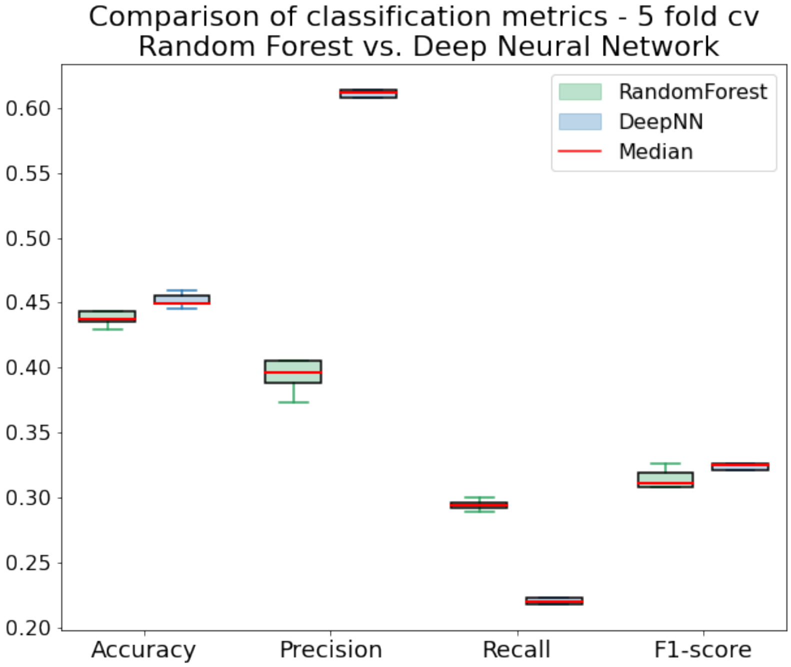

Figure 8: *Box plot comparisons of random forest and Neural Network with 5-fold cross validation. The box plot show the* $25\% - 75\%$ *quartiles and the whiskers are the outliers.*

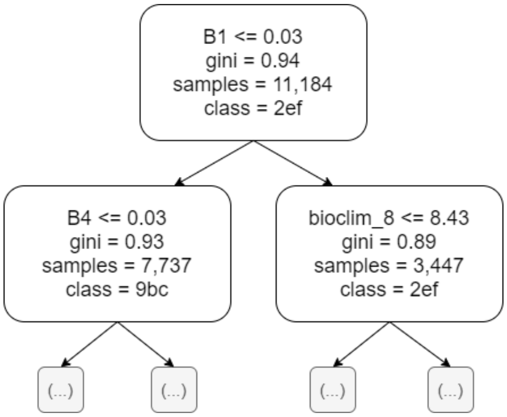

Figure 9: *First two levels of nodes for one exemplary decision tree out of the total 500 trees within the fitted random forest.*

3.2.2 Variable importance

The variable importance for both the random forest and the deep neural network is shown in table Table 3. The features have been stripped for highly correlated features, and have test accuracies (untouched 20% of the data set) displayed. The deep neural network scored a .02 higher test accuracy than the random forest. The five most important features are ranked with mean importance for each of the models. Both models share the $NDVI\_B8$ as the most important feature, but the importance for the deep neural networks is an order of magnitude larger.

Table 3: *The five most important variables according to permutation importance with n=5 random permutations per variable. Both models were trained on an identical random 80% subset of the data, the accuracy is referring to predictions on the remaining 20% test set. Note that highly correlated input features were removed before fitting.*

	Random Forest Classifier		**Multi Layer Perceptron**	
Accuracy	0.41		0.43	
Rank	Variable name	Mean imp. (±SD)	Variable name	Mean imp. (±SD)
1	NDVI_B8	0.069 (±0.003)	NDVI_B8	0.159 (±0.007)
2	B3	0.015 (±0.003)	B8	0.073 (±0.004)
3	SAVI	0.012 (±0.003)	dem100	0.045 (±0.004)
4	B8	0.011 (±0.003)	Growing_season_length	0.04 (±0.001)
5	ar50_veg1	0.004 (±0.002)	B3	0.029 (±0.003)

4 Discussion

4.1 Custom implementation

First, it should be stated that the custom implementation of the Random Forest algorithm works and behaves as expected with the given data input formats. The feature matrix shows that it predicts the correct *Iris* species with high confidence, which is reflected in all of the chosen metrics. However, in the sampled data, there seems to be an overlap in the feature value distributions of *Iris Versicolour* and *Iris Virginica* and, without additional data, it is unlikely that this can be resolved by a different classification approach without inducing low variance.

Moreover, the custom implementation performs comparably well on the Iris data as the scikit-learn RandomForestClassifier(). Especially the high similarity between the narrow OOB-score distributions gives us confidence that we implemented the algorithm correctly. The differences between the distributions even though they were fit to the same input data might be explained by differences in the random sampling of input features and the internal bootstraps created during fitting. One important detail to note is that our implementation, however, did not have the same computational efficiency as the benchmark model. The RandomForestClassifier() implements parallel threading and apparently more efficient looping or matrix operations, too. Hence, we chose to use it for the much bigger vegetation type data set in the following steps.

4.2 Vegetation type distribution

The first difference between the Random Forest and the artificial neural network classifiers is the way the input data must be processed. While the RF can treat categorical input features without one-hot-encoding, this steps need to be added to create the NN feature matrix, adding feature matrix dimensionality. In addition, for NN, it is recommended to scale the input data as it may enhance the classification accuracy (Hastie et al., 2017). For RF, scaling is usually not required (Breiman, 2001). This leads to arguably easier handling of the RF classifier. Also, its working principle is arguably more intuative, which might facilitate the interpretability of results for target groups with limited backgrounds in ML approaches. If the chosen number of trees is low, the possibility to create plots for each tree including threshold values (Figure 1) is a good way to shed light on the black-box nature of ML algorithms (Cutler et al., 2007). This is arguably more difficult in the case of NN.

In terms of performance, the NN has achieved slightly higher overall classification accuracy. However, the differences are small and the distributions partly overlap (Figure 7). Generally, we must concede that the model performance in both cases is rather low and cannot be used to produce maps with high confidence. Other studies with similar goals, but other spatial domains and target variables, have been considerably more successful (e.g. Bricher et al., 2013). On the other hand, when putting it into perspective with a base line accuracy (commonly a classifier only predicting the most frequent class; in this case base line accuracy = 0.136), the model performs significantly better. This shows that the approach works to some extent and that this report is a step towards the direction of automatically classifying vegetation cover in Norway. Additionally, the amount of classes relatively high compared to other classification systems - thus, ecological delineation is not always straight-forward and the same experts may even assign different classes to the same ground truth observation (Bryn et al., 2018). The spatial domain and large extent of the study area is one of the main challenging factors (see e.g. Strand, 2013). One problem is that one needs to rely on consistent and accurate data sets for the whole domain to obtain robust models. In general, there are large uncertainties and temporal mismatches associated with the input feature data collection (Horvath et al., 2019). One next step could thus be to test the performance of Bayesian Machine Learning approaches, which take uncertainties in inputs more into account.

One main distinction between the two models are the differences in the "Precision" and "Recall" values they produce. For one, it stresses how important it is to include these metrics as based on the commonly used classification accuracy alone, it is not possible to withdraw this valuable information. Depending on the use case, there might be trade-offs between a desired model behavior that can, thus, influence the choice of a final model. In our case, it could further be an indication for problems with the considerable class imbalances that are present in the data set (see tables in Horvath et al., 2019) that lead to differences in the model behavior. However, to analyze this in detail, it would be necessary to analyze the confusion matrices in detail, which we did not have time to do. Generally, training the NN took more time, mainly due to the sophisticated chosen architecture. It should be validated if a simpler design would yield the same performance.

Regarding the feature importance, both models list the NDVI using Band 4 and 8 ($NDVI\_B8$) as the variable that most heavily influenced classification accuracy. Generally, many RS variables are included in the table of most important variables (Table 3). This demonstrates that including satellite data in producing vegetation type maps for Norway proved essential, which was hypothesized, but has not been previously shown (Horvath et al., 2019; Tsai et al., 2018). We therefore argue to incorporate additional remotely sensed information from different sensors, such as MODIS, which provide longer time series than the SENTINEL-2 mission. New tools such as the Google Earth Engine make this once enormous task more feasible today. However, there are also climatic variables high in the list, e.g. growing season length and DEM100 (elevation is a proxy for climatic conditions), which stresses the need to combine satellite data with ancillary information for best results.

Furthermore, the magnitudes between the feature importance values between RF and NN considerably differ. Therefore, the NN relies more heavily on individual features while the RF model is more generalizeable. This is probably connected to the correlations between the input features even after removing some clustered variables: the RF might use the "whole range of information" while the NN model might focus more on the features that contain "the highest amount of information" for the task as all variables are always included when fitting (unlike with the RF). However, this is speculation and remains to be tested. In any case, these findings show the benefit of using multiple models for the same classification task for improved interpretability.

Lastly, we suggest future work to include additional hyper-parameter tuning and in-depth analysis of the individual class performance metrics. The former can be computationally expensive, especially for NN, and was thus out of the scope of our report. Existing techniques to handle class imbalances should be considered. Moreover, spatial resampling should be incorporated to quantify potential overfitting caused by spatial correlations.

5 Conclusions

The custom implementation of the RF was benchmarked with the *Iris* test data and showed a good comparison with the sci-kit learn RF implementation with high similarity of the OOB-scores. However, our custom implementation was inferior with regards to the computationally efficiency, which is why we ended up using the sci-kit learn implementation for the analysis of the bigger data set.

The findings in this report show that the NN provided slightly better results than the RF, but that the accuracy distributions from the k-fold cross validation overlap. The accuracy ranges below 0.5, which results in a poor model performance not suitable for creating maps with high degree of confidence. It should be noted that the classification of the vegetation carries a lot of ambiguity, even when predicted by experts. We also find that the the models' feature importance for vegetation classification shares features, but that the the NN weighs fewer features more heavily than the RF. This highlights the value of using multiple models for prediction, as they have offer different perspectives for evaluating the data. At the same time, we evaluate RF to be easier to use and interpret. It was also faster in our use case.

Going forward, interesting investigations not regarded due to time constraints are furthermore tuning of hyperparameters, especially regarding the size of the model architechture, overfit estimation due to spatial autocorrelation and more thourough analysis of class performance metrics. In addition, the ecological meaning of (mis)classifications should be investigated further to identify potential problems. In general, the work presented here yields valuable insights for additional research.

References

Anderson, E. (1936). The species problem in iris. *Annals of the Missouri Botanical Garden*, 23(3):457–509.

Araújo, M. B., Anderson, R. P., Márcia Barbosa, A., Beale, C. M., Dormann, C. F., Early, R., Garcia, R. A., Guisan, A., Maiorano, L., Naimi, B., O'Hara, R. B., Zimmermann, N. E., and Rahbek, C. (2019). Standards for distribution models in biodiversity assessments. *Science Advances*, 5(1).

Barrett, B., Raab, C., Cawkwell, F., and Green, S. (2016). Upland vegetation mapping using random forests with optical and radar satellite data. *Remote Sensing in Ecology and Conservation*, 2.

Black, P. E. (2019). Manhattan distance.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Bricher, P. K., Lucieer, A., Shaw, J., Terauds, A., and Bergstrom, D. M. (2013). Mapping sub-antarctic cushion plants using random forests to combine very high resolution satellite imagery and terrain modelling. *PLoS One*, 8(8):e72093.

Bryn, A., Strand, G.-H., Angeloff, M., and Rekdal, Y. (2018). Land cover in norway based on an area frame survey of vegetation types. *Norsk Geografisk Tidsskrift - Norwegian Journal of Geography*, 72:131–145.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Cutler, D. R., Edwards Jr., T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., and Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, 88(11):2783–2792.

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., Meygret, A., Spoto, F., Sy, O., Marchese, F., and Bargellini, P. (2012). Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote Sensing of Environment*, 120:25–36.

Duveiller, G., Hooker, J., and Cescatti, A. (2018). The mark of vegetation change on earth's surface energy balance. *Nature Communications*, 9(1):679.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals Eugenics*, 7:179–188.

Forkel, M., Drüke, M., Thurner, M., Dorigo, W., Schaphoff, S., Thonicke, K., von Bloh, W., and Carvalhais, N. (2019). Constraining modelled global vegetation dynamics and carbon turnover using multiple satellite observations. *Scientific Reports*, 9(1):18757.

Franklin, J., Davis, F. W., Ikegami, M., Syphard, A. D., Flint, L. E., Flint, A. L., and Hannah, L. (2013). Modeling plant species distributions under future climates: how fine scale do climate projections need to be? *Global Change Biology*, 19(2):473–483.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., and Moore, R. (2017). Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*.

Grandini, M., Bagli, E., and Visani, G. (2020). Metrics for multi-class classification: an overview.

Guisan, A. and Zimmermann, N. (2000). Guisan a, zimmermann ne. predictive habitat distribution models in ecology. ecological modeling. *Ecologial Modelling*, 135:147–186.

Hastie, T., Tibshirani, R., and Friedman, J. (2017). *The Elements of Statistical Learning The Elements of Statistical Learning, vol. 27.* Springer.

Henderson, E. B., Ohmann, J. L., Gregory, M. J., Roberts, H. M., and Zald, H. (2014). Species distribution modelling for plant communities: stacked single species or multivariate modelling approaches? *Applied Vegetation Science*, 17(3):516–527.

Hengl, T., Walsh, M. G., Sanderman, J., Wheeler, I., Harrison, S. P., and Prentice, I. C. (2018). Global mapping of potential natural vegetation: an assessment of machine learning algorithms for estimating land potential. *PeerJ*, 6:e5457.

Horvath, P., Halvorsen, R., Stordal, F., Tallaksen, L. M., Tang, H., Bryn, A., and Feilhauer, H. (2019). Distribution modelling of vegetation types based on area frame survey data. *Applied Vegetation Science*, 22(4):547–560.

Jinru, X. and Su, B. (2017). Significant remote sensing vegetation indices: A review of developments and applications. *Journal of Sensors*, 2017:1–17.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Krüger, F. (2016). *Activity, Context, and Plan Recognition with Computational Causal Behaviour Models.* PhD thesis.

Lian, X., Piao, S., Huntingford, C., Li, Y., Zeng, Z., Wang, X., Ciais, P., McVicar, T., Peng, S., Ottle, C., Yang, H., Yang, Y., Zhang, Y., and Wang, T. (2018). Partitioning global land evapotranspiration using cmip5 models constrained by observations. *Nature Climate Change*, 8:640–646.

Müller, C., Stehfest, E., van Minnen, J. G., Strengers, B., von Bloh, W., Beusen, A. H. W., Schaphoff, S., Kram, T., and Lucht, W. (2016). Drivers and patterns of land biosphere carbon balance reversal. *Environmental Research Letters*, 11(4):044002.

Nielsen, F. (2016). *Hierarchical Clustering*, pages 195–211.

Nieto-Lugilde, D., Maguire, K. C., Blois, J. L., Williams, J. W., Fitzpatrick, M. C., and Peres-Neto, P. (2018). Multiresponse algorithms for community-level modelling: Review of theory, applications, and comparison to species distribution models. *Methods in Ecology and Evolution*, 9(4):834–848.

Olsen, H. B., Dombrovksi, E., and Keetz, L. T. (2020). Project 2: Classification methods with neural networks and logistic regression. `https://github.com/haakools/FYS-STK4155AUTUMN/blob/main/Project%202/Project2FYSSTK4155_aut2020.pdf`.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Phillips, S. J., Dudík, M., Elith, J., Graham, C. H., Lehmann, A., Leathwick, J., and Ferrier, S. (2009). Sample selection bias and presence-only distribution models: implications for background and pseudo-absence data. *Ecological Applications*, 19(1):181–197.

Protopapas, P., Rader, K., Dave, R., and Levine, M. (2017). Lecture 15: Regression trees random forests. `https://harvard-iacs.github.io/2017-CS109A/lectures/lecture15/presentation/lecture15_RandomForest.pdf`.

Rekdal, Y. and Larsson, J. Y. (2005). Veiledning i vegetasjonskartlegging - m 1:20 000 – 50 000.

Richardson, A. D., Keenan, T. F., Migliavacca, M., Ryu, Y., Sonnentag, O., and Toomey, M. (2013). Climate change, phenology, and phenological control of vegetation feedbacks to the climate system. *Agricultural and Forest Meteorology*, 169:156–173.

Scikit-learn (2020). *Permutation Importance with Multicollinear or Correlated Features.* Available at https://bit.ly/3mh2kRm. Accessed: 10-12-2020.

Strand, G.-H. (2013). The norwegian area frame survey of land cover and outfield land resources. *Norsk Geografisk Tidsskrift - Norwegian Journal of Geography*, 67(1):24–35.

Tsai, Y. H., Stow, D., Chen, H. L., Lewison, R., An, L., and Shi, L. (2018). Mapping vegetation and land use types in fanjingshan national nature reserve using google earth engine. *Remote Sensing*, 10(6):927.

Valavi, R., Elith, J., Lahoz-Monfort, J. J., and Guillera-Arroita, G. (2020). Modelling species presence-only data with random forests. *bioRxiv*.

Van Rossum, G. and Drake Jr, F. L. (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Xue, J. and Su, B. (2017). Significant remote sensing vegetation indices: A review of developments and applications. *Journal of Sensors*, 2017:1353691.

Appendices

A Tables

Table 4: *Wall-to-wall environmental predictors used in the feature matrix. For additional details refer to Horvath et al. (2019).*

	Category	Abreviation	Description
0	GEO	losmasse_1	Soil/bedrock under water
1	GEO	losmasse_10	Till
2	GEO	losmasse_11	Till, continuous cover
3	GEO	losmasse_12	Till, discontinuous
4	GEO	losmasse_13	Boulder clay
5	GEO	losmasse_14	Melt-out-till (Ablation till)
6	GEO	losmasse_15	Marginal moraine/zone of marginal moraines
7	GEO	losmasse_16	Drumlin
8	GEO	losmasse_17	Rogen moraine
9	GEO	losmasse_20	Glacio-fluvial deposit
10	GEO	losmasse_21	Glaciofluvial and fluvial deposit
11	GEO	losmasse_22	Esker
12	GEO	losmasse_30	Glacio-lacustrine or subglacial lake deposit
13	GEO	losmasse_31	Glaciofluvial and glacio-lacustrine/subglacial lake deposit
14	GEO	losmasse_35	Lacustrine deposit
15	GEO	losmasse_36	Glacio-lacustrine/subglacial lake and lacustrine deposit
16	GEO	losmasse_37	Beach deposit lacustrine/glacilacustrine
17	GEO	losmasse_40	Marine fine-grained deposit
18	GEO	losmasse_41	Marine fine-grained deposit, great thickness prevalent
19	GEO	losmasse_42	Marine beach deposit, continuous cover
20	GEO	losmasse_43	Marine fine-grained deposit, discontinuous

Table 5: *Wall-to-wall environmental predictors used in the feature matrix (CONTINUED).*

	Category	Abreviation	Description
21	GEO	losmasse_50	Fluvial deposit
22	GEO	losmasse_53	Flood deposit
23	GEO	losmasse_54	Flood deposit, continuous
24	GEO	losmasse_55	Flood deposit, discontinuous
25	GEO	losmasse_60	Eolian deposit
26	GEO	losmasse_70	Weathered material
27	GEO	losmasse_71	Weathered material, continuous cover
28	GEO	losmasse_72	Weathered material, continuous cover, great thickness prevalent
29	GEO	losmasse_73	Weathered material (boulder field)
30	GEO	losmasse_80	Colluvium (slide material)
31	GEO	losmasse_81	Colluvium (slide material), continuous cover
32	GEO	losmasse_82	Colluvium (slide material), discontinuous
33	GEO	losmasse_88	Rock glacier deposit
34	GEO	losmasse_90	Peat and bog (organic material)
35	GEO	losmasse_100	Humus cover/thin peat cover
36	GEO	losmasse_101	Fluvial erosion scarp
37	GEO	losmasse_102	Abandoned fluvial channel
38	GEO	losmasse_120	Fill material (anthropogenic material)
39	GEO	losmasse_121	Waste rock dump
40	GEO	losmasse_122	Anthropogenic material
41	GEO	losmasse_130	Exposed bedrock
42	GEO	losmasse_301	Debris flow deposit, continuous
43	GEO	losmasse_302	Debris flow deposit, discontinuous
44	GEO	losmasse_303	Clay-slide deposit, continuous
45	GEO	losmasse_304	Clay-slide deposit, discontinuous
46	GEO	losmasse_305	Rockslide deposit, continuous
47	GEO	losmasse_307	Rockfall deposit, continuous
48	GEO	losmasse_308	Rockfall deposit, discontinuous
49	GEO	losmasse_309	Snow avalanche deposit, continuous
50	GEO	losmasse_310	Snow avalanche deposit, discontinuous

Table 6: *Wall-to-wall environmental predictors used in the feature matrix (CONTINUED).*

	Category	Abreviation	Description
51	GEO	losmasse_311	Rockslide/rockfall deposit, continuous
52	GEO	losmasse_312	Rockslide/rockfall deposit, discontinuous
53	GEO	losmasse_313	Snow avalanche and debris flow deposit, cont.
54	GEO	losmasse_314	Snow avalanche and debris flow deposit, disc.
55	GEO	losmasse_315	Debris flow and rockfall deposit, continuous
56	GEO	losmasse_316	Debris flow and rockfall deposit, discontinuous
57	GEO	losmasse_321	Stone rich solifluction material on steep slopes
58	LC	AR50_type10	Developed area
59	LC	AR50_type20	Agricultural area
60	LC	AR50_type30	Forest
61	LC	AR50_type50	Barren land
62	LC	AR50_type60	Bog and Fen
63	LC	AR50_type70	Glacier
64	LC	AR50_type81	Freshwater
65	LC	AR50_type82	Ocean
66	LC	AR50_type99	Not mapped
67	GEO	geo_norge1	Nutrient-poor bedrock
68	GEO	geo_norge2	Nutrient-average bedrock
69	GEO	geo_norge3	Nutrient-rich bedrock
70	TOPO	Total_insolation	Potential Incoming Solar Radiation
71	TOPO	Valley_Depth	Valley depth
72	TOPO	Vertical_Distance_to_Channel_Network	Vertical distance to channel network
73	TOPO	Visible_Sky	Visible Sky
74	TOPO	Aspect	Aspect
75	CLIM	bioclim_1	Annual Mean Temperature
76	CLIM	bioclim_3	Isothermality (BIO2/BIO7) (* 100)
77	CLIM	bioclim_5	Max Temperature of Warmest Month
78	CLIM	bioclim_6	Min Temperature of Coldest Month
79	CLIM	bioclim_7	Temperature Annual Range (BIO5-BIO6)
80	CLIM	bioclim_8	Mean Temperature of Wettest Quarter

Table 7: *Wall-to-wall environmental predictors used in the feature matrix (CONTINUED).*

	Category	Abreviation	Description
81	CLIM	bioclim_9	Mean Temperature of Driest Quarter
82	CLIM	bioclim_10	Mean Temperature of Warmest Quarter
83	CLIM	bioclim_11	Mean Temperature of Coldest Quarter
84	CLIM	bioclim_12	Annual Precipitation
85	CLIM	bioclim_15	Precipitation Seasonality (Coefficient of Vari...
86	CLIM	bioclim_17	Precipitation of Driest Quarter
87	CLIM	bioclim_18	Precipitation of Warmest Quarter
88	TOPO	Curvature_arcMap	Standard Curvature (profile and planform)
89	TOPO	dem100	Digital Elevation Model
90	CLIM	Growing_season_length	Growing season length
91	CLIM	precip_5	Average Precipitation in May
92	CLIM	precip_6	Average Precipitation in June
93	HYD	Proxy_allRivers	Proximity to Rivers
94	HYD	Proxy_ALLwater	Proximity to all water bodies
95	HYD	Proxy_coast	Proximity to Coast
96	HYD	Proxy_lakes	Proximity to Lakes
97	SNOW	sca_2	Snow covered area in February
98	SNOW	sca_7	Snow covered area in July
99	SNOW	sca_8	Snow covered area in August
100	SNOW	sca_9	Snow covered area in September
101	TOPO	Slope	Slope
102	SNOW	swe_4	Snow water equivalent in April
103	SNOW	swe_8	Snow water equivalent in August
104	SNOW	swe_10	Snow water equivalent in October
105	TOPO	Terrain_Ruggedness_Index	Terrain ruggedness index
106	CLIM	tmax_1	Maximum temperature in January
107	CLIM	tmax_2	Maximum temperature in February
108	CLIM	tmax_5	Maximum temperature in May
109	CLIM	tmax_6	Maximum temperature in June
110	CLIM	tmax_8	Maximum temperature in August

Table 8: *Wall-to-wall environmental predictors used in the feature matrix (CONTINUED).*

	Category	Abreviation	Description
111	CLIM	tmax_9	Maximum temperature in September
112	CLIM	tmax_10	Maximum temperature in October
113	CLIM	tmin_5	Minimum temperature in May
114	CLIM	tmin_9	Minimum temperature in September
115	TOPO	Topographic_Wetness_Index	Topographic wetness index
116	RS	NDVI	Normalized Difference Vegetation Index, Band 8
117	RS	SAVI	Soil Advanced Vegetation Index
118	RS	EVI	Enhanced Vegetation Index
119	RS	EVI2	Enhanced Vegetation Index 2
120	RS	TCW	Tassled Cap Wetness Index
121	RS	NDII	Normalized Difference 819/1600
122	RS	B1	Sentinel-2A Band 1 Spectral Reflectance
123	RS	B2	Sentinel-2A Band 2 Spectral Reflectance
124	RS	B3	Sentinel-2A Band 3 Spectral Reflectance
125	RS	B4	Sentinel-2A Band 4 Spectral Reflectance
126	RS	B5	Sentinel-2A Band 5 Spectral Reflectance
127	RS	B6	Sentinel-2A Band 6 Spectral Reflectance
128	RS	B7	Sentinel-2A Band 7 Spectral Reflectance
129	RS	B8	Sentinel-2A Band 8 Spectral Reflectance
130	RS	B8A	Sentinel-2A Band 8A Spectral Reflectance
131	RS	B9	Sentinel-2A Band 9 Spectral Reflectance
132	RS	B11	Sentinel-2A Band 11 Spectral Reflectance
133	RS	B12	Sentinel-2A Band 12 Spectral Reflectance

Table 9: *List of all land cover classes contained in the AR18x18 data set.*

Code	Land-cover class	Code	Land-cover class	Code	Land-cover class
1a (*m*)	Moss snowbed	5a (*m*)	Poor broadleaf deciduous forest	10b (*m*)	Coastal calluna heath
1b (*m*)	Sedge and grass snowbed	5b (*m*)	Rich broadleaf deciduous forest	10c	Damp heath
1c	Frozen ground, leeward	6a	Lichen and heather pine forest	10d (*r*)	Crags and thicket
2a	Frozen ground, ridge	6b	Bilberry pine forest	10e (*r*)	Moist and shore meadows
2b	Dry grass heath	6c (*r*)	Meadow pine forest	10f (*r*)	Sand dunes and gravel beaches
2c	Lichen heath	6d (*r*)	Pine forest on lime soils	10g (*r*)	Pioneer alluvial vegetation
2d	Mountain avens heath	7a	Lichen and heather spruce forest	11a (*r*)	Cultivated land
2e (*m*)	Dwarf shrub heath	7b	Bilberry spruce forest	11b	Pastures
2f (*m*)	Alpine calluna heath	7c	Meadow spruce forest	12a (*r*)	Barren land
2g	Alpine damp heath	8a	Damp forest	12b	Boulder field
3a (*m*)	Low herb meadow	8b	Bog forest	12c	Exposed bedrock
3b (*m*)	Low forb meadow	8c (*m*)	Poor swamp forest	12d (*r*)	Built-up areas
4a	Lichen and heather birch forest	8d (*m*)	Rich swamp forest	12e (*r*)	Scattered housing
4b	Bilberry birch forest	9a (*m*)	Bog	12f (*r*)	Artificial impediment
4c	Meadow birch forest	9b (*m*)	Deer-grass fen	12g (*r*)	Glaciers and perpetual snow
4d (*r*)	Birch forest on lime soils	9c (*m*)	Fen	13a (*r*)	Water courses (fresh)
4e	Alder forest	9d (*m*)	Mud-bottom fen and bog	13b (*r*)	Water bodies (fresh)
4f (*r*)	Flood-plain shrubs	9e	Sedge marsh	13c (*r*)	Estuaries
4g	Pasture land forest	10a (*m*)	Coastal heath	13d (*r*)	Sea and ocean

(*m*): Land cover class merged with similar class due to rare occurrence or challenges regarding ecological delineation.

(*r*): Land cover class excluded due to rare occurrence or irrelevance for the project.