Classification of Land Cover and Agricultural Land Using Google Earth Engine

Šumanovac, Luka

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:
Josip Juraj Strossmayer University of Osijek, Faculty of Agrobiotechnical Sciences Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet agrobiotehničkih znanosti Osijek

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:151:873314

JOSIP JURAJ STROSSMAYER UNIVERSITY OF OSIJEK

FACULTY OF AGROBIOTECHNICAL SCIENCES OSIJEK

Luka Šumanovac

Graduate Studies Digital Agriculture

Course GNSS and Sensors in Agriculture

CLASSIFICATION OF LAND COVER AND AGRICULTURAL LAND USING GOOGLE EARTH ENGINE

Graduate thesis

Osijek, 2024

JOSIP JURAJ STROSSMAYER UNIVERSITY OF OSIJEK

FACULTY OF AGROBIOTECHNICAL SCIENCES OSIJEK

Luka Šumanovac

Graduate Studies Digital Agriculture

Course GNSS and Sensors in Agriculture

CLASSIFICATION OF LAND COVER AND AGRICULTURAL LAND USING GOOGLE EARTH ENGINE

Graduate thesis

Reviewers:

1. Mladen Jurišić, PhD, Full Professor with Tenure, Chair
2. Dorijan Radočaj, PhD, Assistant, Mentor
3. Ivan Plaščak, PhD, Full Professor, Memeber

Osijek, 2024

Contents

TEMELJNA DOKUMENTACIJSKA KARTICA

1. Introduction

The most important indicator of surface change on Earth is land cover (Herold, 2009). Ongoing land cover changes have negative impacts on the Earth's terrestrial ecosystems, climate, biodiversity, climate, etc. (Salazar et al., 2012). To observe these changes, remote sensing data has been used as a primary data source for land cover monitoring and mapping over a period of time (Li et al., 2017) There are many data sources for land cover change studies such as LANDSAT data, Synthetic Aperture Radar (SAR), Satellite Pour observation de la Terre (SPOT), Moderate Resolution Imaging Spectroradiometer (MODIS), and Sentinel 2 satellite data (Phan et al., 2020). The European Space Agency (ESA) launched the first Sentinel satellites – Sentinel 1-A - in 2014 as part of the Copernicus program. Since then, several satellite missions have been launched as part of this program, including satellites 1, 2, 3, 4, 5 and 6 (ESA, 2024). The most important impact of the program was the launch of the Sentinel-2 satellites. The constellation of two polar-orbiting satellites, Sentinel 2A and Sentinel 2B, was launched on June 23, 2015 and March 7, 2017. Equipped with multispectral imaging instruments capable of recording 13 broadband bands, the main objective was to collect high-resolution satellite data for monitoring land cover and land use as well as climate change. Another important component of the Sentinel-2 satellites is their compatibility with the LANDSAT and SPOT programs, which is important for the continuous monitoring of land changes on Earth (Immitzer et al., 2016). Scientists face two major challenges when mapping land cover over a large study area: firstly, the enormous amount of data that needs to be processed, and secondly, the availability of cloud-free data. Traditional processing methods such as searching, filtering, downloading, cloud masking, atmospheric correction, etc. are labor-intensive at their core and require enormous storage capacity and access to high-performance computing for large amounts of data (Carrasco et al., 2019). The solution to these problems lies in the Google Earth Engine (GEE). GEE is a cloud computing platform released by Google in 2010 that can analyze and process global geospatial data (Amani et al., 2020). The database consists of satellite images from a variety of the above-mentioned satellite programs, which are updated daily, sub-daily, or weekly depending on the type of satellite. The first version contained only remote sensing data, but now various climate and weather layers, vector, social and demographic data, and digital elevation models have been added (Moore et al., 2011). The programming interface allows users to develop and run custom algorithms, parallelizing the analysis to involve multiple processors in the calculations, which greatly speeds up the process. This makes it easier to perform analysis on a global scale compared to

traditional desktop computing methods (Kumar et al., 2018). This thesis will be based on integrating GEE, QGIS software, and Sentinel data to make classification maps of Osijek-Baranja County while testing some of the GEE machine-learning algorithms.

1.2. Research Goal

The primary research objective of this thesis is the classification of land cover and agricultural land in Osijek-Baranja County as a study region. This approach shall determine the ratio between agricultural land and other types of classes on the map in the study period between 2020 and 2022. The second objective is to evaluate six classification methods and determine which of the algorithms provides the most accurate classification. The third objective is to perform five types of accuracy evaluations and metrics for the classification methods to determine the differences between them and which of the following methods is the most and least sufficient.

2. Literature Review

In 2018, authors Kumar and Mutanga examined the usage, trends, and potential of GEE since the open-source software was launched in 2010 (Kumar and Mutanga, 2018). After analyzing 485 articles of various types, they concluded that the usage of GEE increased from 2010 to 2018, with the authors citing numerous benefits, such as a powerful computing infrastructure, various sets of classifiers and algorithms, and computing capacity for data preparation. The authors also noted that the study showed variability in the geographic regions of GEE applications with the United States leading with 17% of published papers, followed by China and Brazil. Interestingly, only one paper from Croatia had been published by 2018. Most applications relate to forests, vegetation, land cover, ecosystems, and sustainability. The most commonly used satellite images were LANDSAT data, followed by MODIS (Kumar and Mutanga, 2018). A similar study to Kumar and Mutanga was published by Xiang Zhao and other authors who looked at advances and trends in GEE and Google Earth (GE) (Zhao et al., 2022). They analyzed the use of GEE and GE from 2006 to 2020 based on 530 articles. They found that there are several advantages of GEE, but also some of the limitations. Some of the merits are that the computer requirements of the user are low (Amani et al., 2020). Two programming languages are supported, Java Script and PythonAPI (Tamiminia et al., 2020). Google's computing capabilities are the reason for analyzing big geospatial data (Gorelick et al., 2017). Some of the limitations faced by users are the limited upload and download speed of the data. There are a limited number of training/validation examples for large-scale classifications, some of the deep learning algorithms are not available, some of the high-resolution satellite images are not supported, and there are also privacy issues (Amani et al., 2020). There are also some issues with the complexity of creating new tools (Liang et al., 2020). Despite all the advantages and disadvantages of GEE, the authors of the study state that GEE is a powerful analysis tool for remote sensing. A few scientists conducted research in 2022 to evaluate two types of land use classification platforms with different types of satellite image datasets, classifiers, and periods. The satellite image data are from LANDSAT, Sentinel 2, and Planet, and the machine learning algorithms are Support Vector Machine (SVM), Minimum Distance (MD), Random Forest (RF), and Correlation and Regression Tree (CART). The Kappa coefficient and the overall accuracy for the LULC maps created from 2017 to 2021 were used to evaluate the accuracy. They concluded that the SVM classifier achieved the best results for both platforms, with an overall accuracy of 87 for LANDSAT and 92% for Sentinel 2. From 2017 to 2021, the results show that 13,80% of forests were converted to non-forested

land and 14.10% to urban areas. In addition, 3.90% of land was converted to urban areas. These changes in land cover over the four years indicate a strong urbanization of the area (Ghyaour et al., 2021). A group of authors investigated land use and land cover change in North Korea from 2001 to 2018 using Landsat datasets and the RF machine learning algorithm to create LULC maps. Two metrics were used to evaluate accuracy: Overall accuracy and Kappa coefficient. They improved the sampling methods and classification accuracy within farmland forest cover, with an Overall accuracy of 98,2% and a corresponding Kappa coefficient of 0,959. They found that in some parts of North Korea, through land use recognition, forests are undergoing a restoration process (Piao et al., 2021). Author Farda published a research paper on machine learning algorithms in GEE and their accuracy for multi-temporal mapping of coastal wetlands in Segara Anakan lagoon. Farda used satellite data from LANDSAT (Landsat 5 TM (1991), Landsat 7 ETM+ (2001), and Landsat 8 OLI (2014) and applied 10 machine learning algorithms, such as Fast Naïve Bayes, RF, CART, GMO MAX Entropy, Perceptron, Winnow, Voting SVM, Margin SVM, Pegasos and IKPamir (Farda, 2017). The author concluded that GEE is very useful in multitemporal mapping of coastal wetlands, with the best result obtained with the CART machine learning algorithm of 96,98% Overall accuracy (Farda, 2017.). In 2018, a group of authors from Canada first proposed a method that combines GEE, artificial neural networks, machine learning algorithms, and Sentinel-2 satellite image datasets to produce object-based maps for the annual spatial agricultural inventory (Amani et al., 2018). They performed several metrics to evaluate accuracy, such as an Overall accuracy of 77% and a Kappa coefficient of 0,74, and through a total of 17 classes of cropland, they managed to obtain results of 79 for producer accuracy and 77 for user accuracy. The research shows that GEE was efficient in terms of time, calculation, cost, and automation with the given results (Amani et al., 2018). Manzanze, Pocas, and Cuncha published a case study on agricultural land cover changes using GEE, every three years between the periods of 2012 to 2018 (Manzanze et al., 2018). Using datasets from LANDSAT combined with vegetation indices and texture features and the GEE algorithm RF, the Overall accuracy with the RF classifier was 94% for 2012, 98% for 2015, and 89% for 2018. From these results, they concluded that the maps are reliable and that agriculture is the most important cause of land cover change in the study area in Mozambique (Manzanze et al., 2018).

The authors from India checked the classification in GEE for six classes. With the following results: 8 % water, 20.8 % horticulture, 39.3 % agriculture, 15.1 % cultivated area, and 7 % other classes using the maximum likelihood classification algorithm (Getha et al., 2018). After

obtaining an Overall accuracy of 94,8% and a Kappa coefficient of 0,86, they found that the use of geospatial information tools such as GEE and QGIS is much more effective and user-friendly in monitoring land cover and land change than other similar platforms (Getha et al., 2018). Zurqani and a group of authors investigated the mapping and quantification of agricultural irrigated areas for different crop types in South Carolina using the GEE classifier RF Zurqani et al., 2021). The main objectives of the study were to capture relevant vegetative indices, quantify irrigated areas and drylands for subsequent mapping of the study area, and track temporal and spatial changes in agricultural irrigation. After analyzing Sentinel-2 satellite images and applying the RF classifier, the maps we produced over three years (2016, 2017, and 2019) resulted in an overall accuracy of 83,73%, 86,18%, and 84,55%, respectively. The authors note that these maps provide valuable information for the future development of irrigation systems and can help optimize water use in the same area (Zurqani et al., 2021). The authors published a paper in 2023 in which they used GEE and Intensity Analysis (IA) to create LULC maps to track changes in agricultural land and monitor the process of urbanization by combining GEE and IA methods for the first time in Indonesia (Ganharum et al., 2022). They used two time periods to track the urbanization process, the first from 2003-2013 and the second from 2013-2020. Like many other researchers, they used the LANDSAT dataset as the basis and RF as the machine learning algorithm. The Overall accuracy for the years 2003, 2013, and 2020 is 88%, 87% and finally 88%. The IA outputs yielded results that are 2.3 higher in the first period analyzed, 2003-2013, than those in 2013-2020. Annually, this was the loss of 1,850 ha of agricultural land. With these results, the authors found that policymakers in Indonesia have valuable sources of information for urban and regional planning and development (Ganharum et al., 2022). Karishma et al. prepared a study to assess the LULC change in the Bhavani Basni study area using the LANDSAT 8 dataset, GE, and GIS (Karisham et al., 2022,). They used CART and RF machine learning algorithms for supervised classification. To evaluate the accuracy, the high-resolution maps and the confusion matrix were evaluated (30% of the training data for each class was used). Five LULC classes were selected: the first and most important is agriculture, followed by cultivated land, fallow land, forest, and water. They concluded that the most important land uses in the study area were agricultural land and fallow land, while urbanization caused a 1,92 change in the built-up class from 2014 to 2019 (Karisham et al., 2022). A group of authors from Thailand published a large study on land change and land use in Thailand over 30 years, focusing mainly on the lack of arable land, agricultural practices, and expansion (Kruasilp et al., 2023). Researchers used

several satellite data from Landsat 5, Landsat 8, Sentinel-1, and Sentinel-2 from 1990 to 2019 in Nan Province were used as the study area. As in many other studies before, the RF Classifier and GEE were used as the cloud computing platform in this study. The mean composition of the input data was used to create the datasets. The Overall accuracy for a total of 36 datasets was between 51.70 % and 96.95 %. The result of 96,95 % was the best, combining the Sentinel-1 dataset, the Modified Soil-Adjusted Vegetation Index (MSAVI), and some topographic variables. Interestingly, the combination of optical (e.g. LANDSAT8) and S1 Synthetic Aperture Radar (SAR) data provided better results than single S1 data. The forest class in the study decreased in the five consecutive time periods, while maize cover increased in the period 2010-2014. The authors concluded that the results strongly suggest that the RF classifier, GEE, and satellite datasets improve the accuracy of LULC classification in upland areas (Kruasilp et al., 2023). The authors Ashane, Fernando, and Senanayake published a longitudinal study over two decades on rice field map generation using GEE and RF machine learning classification methods (Ashane et al., 2023). Traditional segmentation and classification methods are often unable to distinguish different features between rice and other vegetation or crop species. By combining GEE and RF based on the LANDSAT dataset, they have managed to achieve an overall accuracy of 80 of the maps produced in the Sri Lankan study area. The maps we produced were also compared with the MODIS (MCD12Q19) land cover type while the following statistics showed the robustness of the proposed approach. The results with the RF classifier were obtained with only 200 trees in training, while the model was evaluated with 30 different vegetation and moisture indices, with Normalized Difference Water Index (NDWI), Enhanced Vegetation Index (EVI), and Soil Adjusted Vegetation Index (SAVI) providing the best results (Ashane et al., 2023).

3. Materials and Methods

Osijek-Baranja County (OBC) was selected as the study area to create the following classification code in GEE. OBC is located in the eastern part of Croatia, bordering Hungary to the north and Serbia to the east. The geographical position is 45° 32' North, 18° 44' East, the altitude is 90 m above sea level, and the area is 4152 km^2 (Wikipedia, 2024). Machine learning essentially means that computers learn to optimize performance criteria based on data samples or previous experience (Alpayadin, 2010). Machine learning is about learning the rules of an example, and the main goal is to learn the pattern of a given example and output a new example. The two main areas of machine learning are supervised and unsupervised classification. In unsupervised classification, computers are programmed to recognize unknown patterns in data sets without prior knowledge. Supervised classification is based on programming computers to predict classes or values for unobserved data points (test data) based on a classification model that has been tested on training data (Badillo et al., 2020). Supervised machine learning can be divided into 5 groups, which are listed with examples in Table 1. (Kotsiantis, 2007)

Table 1. Supervised Machine Learning Categories

Supervised Machine Learning	Examples
Logic-based algorithms	• Decision tree: CART (Classification and Regression Tree), Decision tree C4.5, RFs, GMO Max Entropy • Learning set of rules
Perceptron based technique	• Single layered perceptrons: Winnow
Statistical learning algorithms	• Fast Naive Bayes classifiers • Bayesian Networks
Instance-based learning	• Nearest neighbor algorithm • k-Nearest Neighbour (CNN)
Support vector machines	• Voting SVM (Support Vector Machines) • Margin SVM

	• Pegasos (Primal Estimated sub-GrAdient SOlver for Svm) • IKPamir (Intersection Kernel Passive Aggressive Method for Information Retrieval, SVM)

Source: Kotsiantis, 2007

Supervised classification is one of the most difficult machine learning techniques to master, and applications range from land cover generation to land cover change detection. The combination of the interactive nature of GEE and its ease of use at large scale is the perfect combination for remote sensing applications (Richards, 2013). In this paper, supervised classification algorithms such as Naïve Bayes, kNN, etc. are used to evaluate the class areas.

3. 1. Adding OBC Shapefile

The first step in creating the code was to add a shapefile, in this case, OBC, and add the variable "store" to form the region of interest for the study area, as shown in Figure 1. The shapefile of OBC was added via the "Add project" option and simply inserted into the code after uploading it to Cloud Assets.

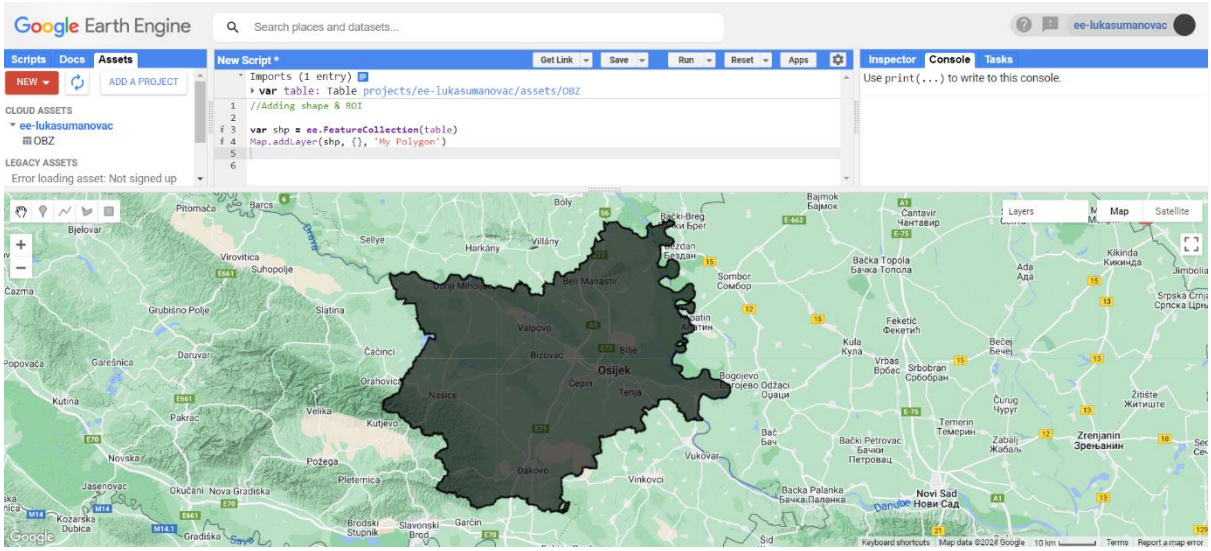

Figure 1. Adding shape and ROI

Source: Šumanovac, 2024

3. 2. Loading Sentinel Data Imagery

The second step of the code generation consisted of loading Sentinel satellite data from the image memory. The S2 multispectral instrument used to acquire Sentinel images samples 13 spectral bands with a global repetition rate of 5 days. The spectral resolutions are as follows:

(1) RGB and NIR - 10 meters
(2) Red edge and SWIR - 20 meters
(3) Atmospheric bands - 60 meters

This data is ideal for tracking the condition of and changes in vegetation, soil, and water cover (Earth Engine Catalog, 2024).

The first variable "image" was created, taking into account some special features. For example, the cloud percentage was lowered to 5, as it is one of the main problems in processing satellite imagery and producing valuable output data. The time frame in this case was from January 1 to December 31 of the year 2020. This was repeated for the years 2021 and 2022, as these years were also selected as study years. The second variable in the code was "visParmsTrue". This variable was created to insert RGB bands into the layer to get a true color composition for the study area. Figure 2. shows the loading of the sentinel data.

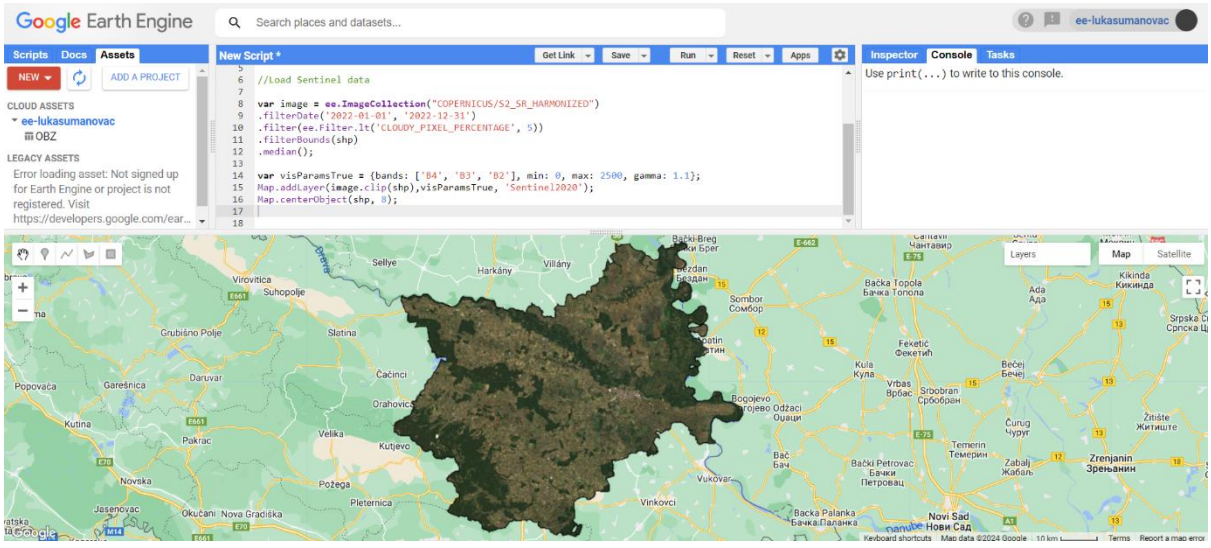

Figure 2. Loading Sentinel data

Source: Šumanovac, 2024

3. 3. Creating Classes for Classification

The third step in creating the code was the creation of classes for classification. Machine learning is composed of data and data sets. The dataset consists of multiple data points, in this case, classes, and each of these classes is an entity required for the analysis. To create datasets, multiple features need to be collected and measured. The features can be numeric, categorical, or ordinal, with each of these features representing a particular dimension within the feature space, with the value of the feature determining the position within that dimension. Combined, all characteristics form a characteristic vector. Several feature values and overall features determine its dimensionality (Badillo et al., 2020). Five classes were formed for the classification: water, arable land, wasteland, urban, and forest. The classes were numbered (assigned values) from 0 to 4 and assigned colors that correspond to the real characteristics of the class. Figure 3. shows how the water class is created.

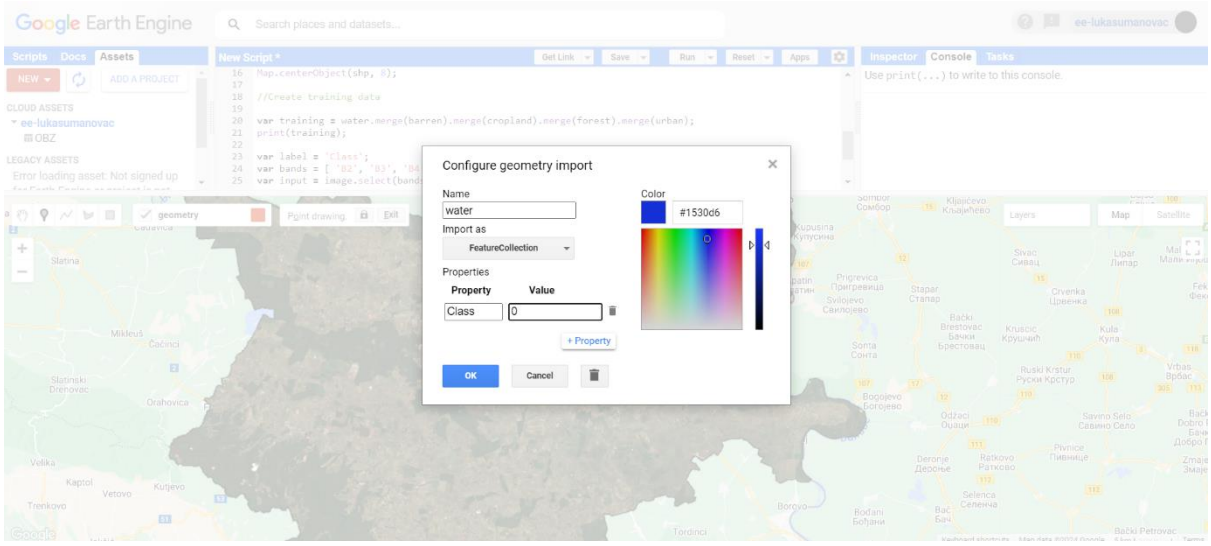

Figure 3. Creating water class

Source: Šumanovac, 2024

3. 4. Creating Training Data

The fourth step is one of the most important steps of this code, which is to create, merge, and training data. There are three main phases in classification: the training phase, the testing phase, and the validation phase. The model is trained using input values, in this case, five of the classes, and this phase is the training phase. In this phase, the parameters of the model are

adjusted and the training error measures how well the trained model fits the selected classes in the training data. The main goal of the training phase is to learn an algorithm for recognizing class labels for unseen data that occurs in the test phase. The test error cannot be tested because the output data is unknown. For this reason, the validation phase is used to test the performance of the trained model (Tharwat, 2018). For each of the five classes, 80 training samples were created using point drawings at different points in the study area. It was of great importance to be precise in the manual definition of each class, as this provides a guideline for the creation of future maps. Figure 4. shows how the point system works for arable land (yellow in this case). A variable training was created to summarize all five classes.

Figure 4. Cropland training data

Source: Šumanovac, 2024

3. 5. Overlaying Points on the Imagery

The fifth step was to overlay the points on the image and train the training set with the points for each class as shown in Figure 5. and the code that executes the functions.

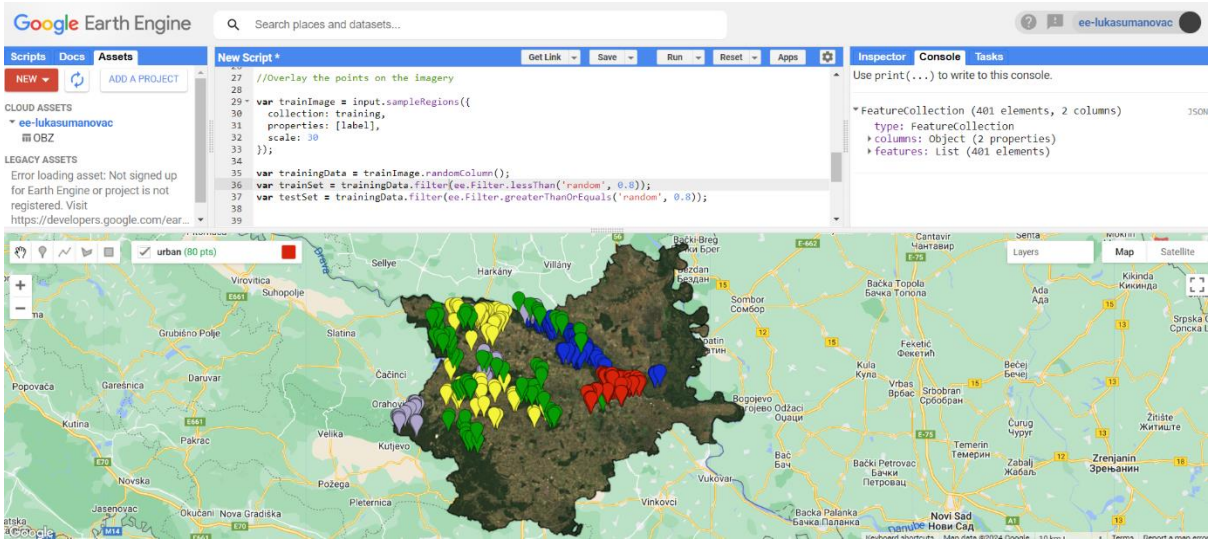

Figure 5. Overlaying points on the imagery

Source: Šumanovac, 2024

After overlaying points on the imagery, it was important to define color parameters according to the HTML Google parameters and apply the classification model to the training set. For example, the ‘color code "253494" was used for the water class or "EC340C" for the wasteland class, as can be seen in Figure 6. The same color palette was later used to create LULC maps in the QGIS software (HTML Color Codes, 2024). The first classification model used in the code was CART.

```
//Classification model

var classifier = ee.Classifier.smileCart().train(trainSet, label, bands);

//Classify image

var classified = input.classify(classifier);

//Define color parameters

var landcoverPalette = [
  '253494', //water (0)
  '969696', //barren (1)
  'FF8000', //cropland (2)
  '006837', //forest (3)
  'EC340C', //urban (4)
  ];

Map.addLayer(classified.clip(shp), {palette: landcoverPalette, min: 0, max:4}, 'classification CART');
```

Figure 6. Classification model and color parameters

Source: Šumanovac, 2024

The results of the classification with the added color parameters can be seen in Figure 7. The new layer labeled "classification CART" will be used for later accuracy assessments and for testing various GEE metrics.

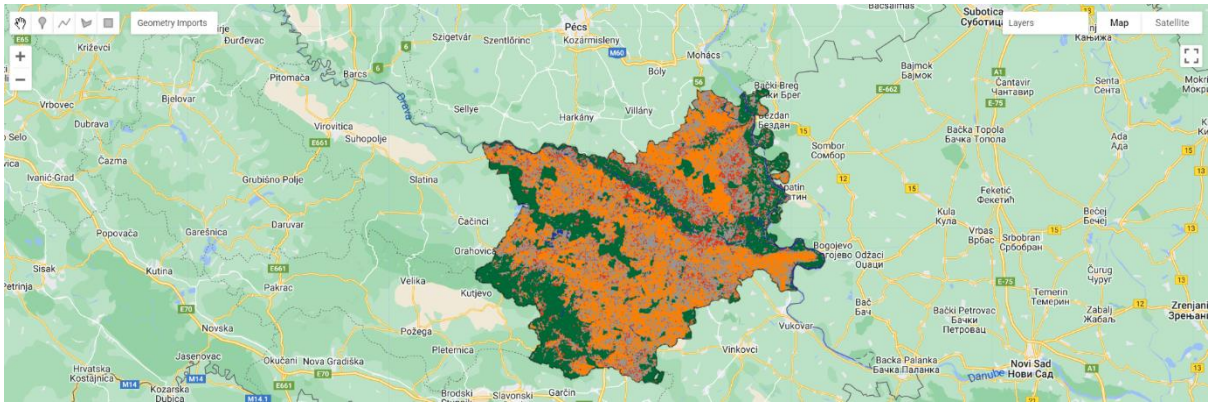

Figure 7. Classification CART layer

Source: Šumanovac, 2024

3. 6. Machine Learning Algorithms for Classification and Accuracy Assessment

In classification models, the training data from the classification model is used to predict a class label for an unknown sample. These new results must be evaluated and analyzed to determine the performance of the various classification algorithms. These results can be of different types, such as discrete tree types or continuous types (Tharwat, 2018). There are mainly two problems in multi-class classification. The binary classification model, where there are only two classes, and the multi-class classification, where there are more than two classes. In binary classification, for example, there can be two classes P and N. P is assigned to the positive classes and N to the negative classes. If classifying some unknown samples into P or N, the trained model will classify the true classes of the later unknown classes in the training phase. This model will produce continuous or discrete outputs. The discrete outputs of the model represent the predicted discrete class label of an unknown sample, and the continuous outputs represent the probability of class membership. Figure 8. illustrates the confusion matrix for the binary classification problem. There are four possible outputs of a 2x2 confusion matrix. The green box represents correct predictions of the unknown samples, the pink box represents incorrect predictions. For example, if the sample is positive and is recognized as positive, then it is counted as a true positive. If the sample is negative but is classified as positive, it is a false negative or a type II error. Conversely, if the sample is negative and is recognized as negative,

it is considered a true negative. If the sample is negative but is categorized as positive, it is considered a false negative or type I error (Tharwat, 2018).

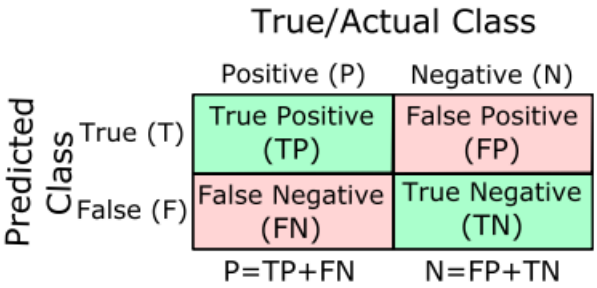

Figure 8. 2x2 Confusion Matrix

Source: Tharwat, 2018

This classification model is based on multi-class classification, in that case, Figure 9. shows a confusion matrix for a multi-class problem (three classes: A, B, C). TP_A in the green box is showing true positive for class A, and E_{AB} and E_{AC} are class A that were incorrectly classified as classes B or C. To calculate these errors, or in this case the False Negatives of A - FN_A, it is the sum of E_{AB} and E_{AC}. The sum of all classes A, that were incorrectly classified as classes B or C:

$$FN_A = E_{AB} + E_{AC}$$

M x m confusion matrix there can be m correct classifications and m^2-m (Srinivasan, 1999).

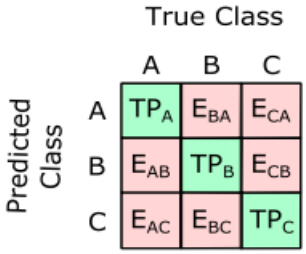

Figure 9. Multi-class Confusion Matrix

Source: Tharwat, 2018

To evaluate the accuracy of the aforementioned classification model, the confusion matrix

function was tested. The function will give an insight into the overall accuracy of the model, with corresponding outputs. Figure 10. shows this part of the code. Later, six classification methods and five confusion matrix methods were tested. The methodology tested six classification models:

```
//Accuracy assesment

var confusionMatrix = ee.ConfusionMatrix(testSet.classify(classifier)
.errorMatrix({
  actual: 'Class',
  predicted: 'classification'
}));

print('Confusion Matrix:', confusionMatrix);
print('Overall accuracy', confusionMatrix.accuracy());
```

Figure 10. Confusion matrix assessment

Source: Šumanovac, 2024

- ee.Classifier. smile art
- ee.Classifier.smileNaiveBayes
- ee.Classifier.smileRandomForest
- ee.Classifier. libsvm
- ee.Classifier.kNN
- ee.Classifier.MinimumDistance

The methods used to test the Confusion Matrix accuracy are:

- Training Error Matrix
- Training Overall Accuracy
- Consumer's Accuracy
- Producer's Accuracy
- Kappa statistics

For successful classification, a sufficiently efficient classifier must be chosen for a small number of training samples (Foody et al., 2004) This paper tests how well six of the above classifiers perform. Machine learning algorithms, such as the one used in GEE, are useful for finding patterns in complex spatial data while mitigating dimensionality problems (Richards, 1999).

3. 7. CART Classification Model

CART is a classification algorithm used in GEE that is based on an entropy structure, or simply put, a decision tree. Decision trees have become one of the most powerful tools in machine learning. The operating principle of the algorithm is based on the classification of unknown samples based on the features present in the training data. By growing the decision tree and giving it more information to learn in the form of the IF-THEN function in object classification, the model can be built by piecewise approximation (Li et al., 2011). One of the main problems of the algorithm is its high sensitivity to changes in the training data (Bishop, 2006). Figure 11. shows the basic operating principle of the decision tree. The code for the CART classification model has already been shown in Figure 6.

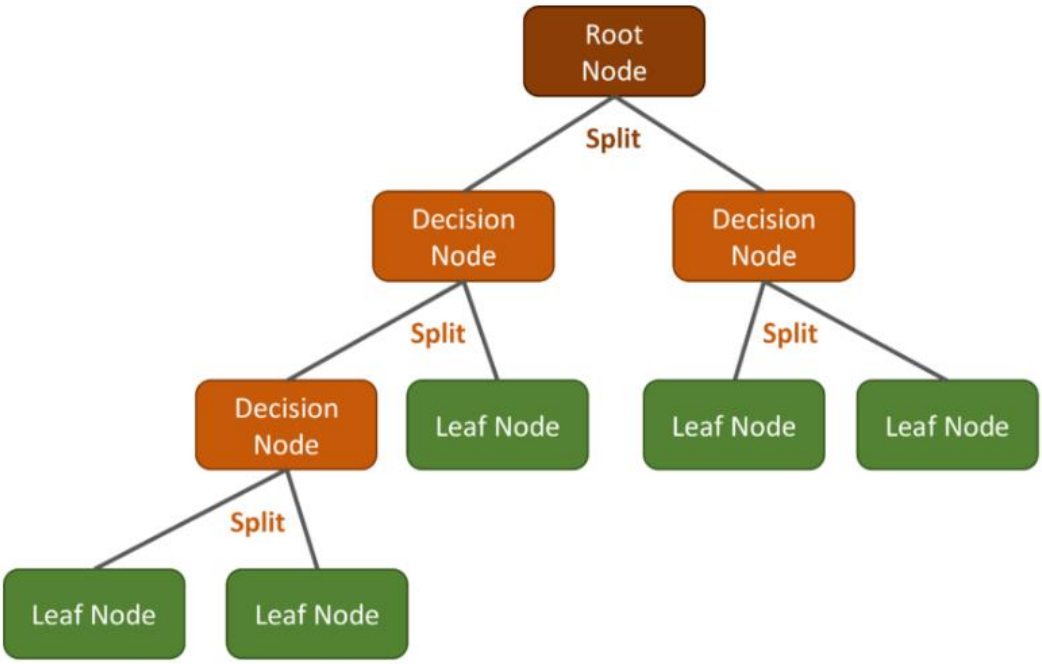

Figure 11. Decision tree Example

Source: Quelle: FU Berlin, RESEDA 2024

3. 8. Naïve Bayes Classification Model

The Naïve Bayes Classifier is a classification method that assumes that each feature depends only on the class itself, as shown in Figure 12. This would mean that a parent is assigned to a feature (Domingos et al., 1997).

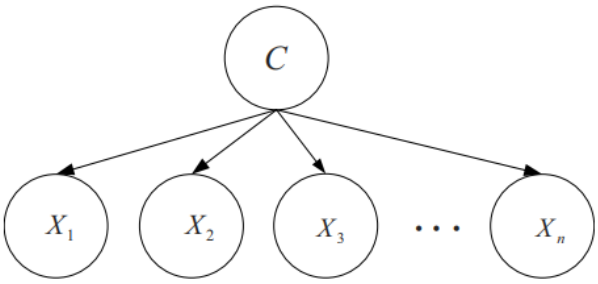

Figure 12. Naive Bayes Classifier

Source: Taheri et al., 2021

The main advantage of this classification method is that the classification process requires little training data to determine the range of parameters, as it assumes independence between variables and focuses only on the individual variable variations within each class without considering the entire covariance matrix (Paas et al., 2017). Naïve Bayes incorporates probability analysis, interactive mapping, and spatial pattern analysis when applied to digital databases, as explained by author Kadirhodjaev (Kadirhodjaev, 2018). The classification algorithm is shown in Figure 13. below.

```
//Classification Naive Bays
var classifierNB =  ee.Classifier.smileNaiveBayes().train(trainSet, label, bands);

// // Classify image

var classifiedNB = input.classify(classifierNB);

Map.addLayer(classifiedNB.clip(shp), {palette: landcoverPalette, min: 0, max: 4}, 'classification Naive Bayes');

// Get information about the trained classifier.

print('Results of trained classifier', classifierNB.explain());

// Get a confusion matrix and overall accuracy for the training sample.

var trainAccuracy1 = classifierNB.confusionMatrix();

print('Training error matrix for Naive Bayes', trainAccuracy1);
print('Training overall accuracy for Naive Bayes', trainAccuracy1.accuracy());
print("Consumer's accuracy for Naive Bayes ", trainAccuracy1.consumersAccuracy());
print("Producer's accuracy for Naive Bayes", trainAccuracy1.producersAccuracy());
print('Kappa statistic for Naive Bayes', trainAccuracy1.kappa());
```

Figure 13. Naive Bayes Classification model

Source: Šumanovac, 2024

3. 9. Random Forest Classification Model

The third algorithm that was tested is RF. RF is one of the most commonly used algorithms for land use classification maps, showing high efficiency (Rodriguez-Galliano et al., 2012). There

are several reasons for the great popularity of this algorithm. The first reason is that it provides fast analysis and efficiency, as stated in Shabani's work (2021) (Shabani, 2021). It is insensitive to outliners, noise, and overtraining and can treat different features of the same variable together (Naghibi et al., 2016). Similar to the CART algorithm, the RF also consists of several unconnected and independent decision trees, which are shown in Figure 14. and represent the RF Prediction Scheme. The number of entries was set to 100 in this work.

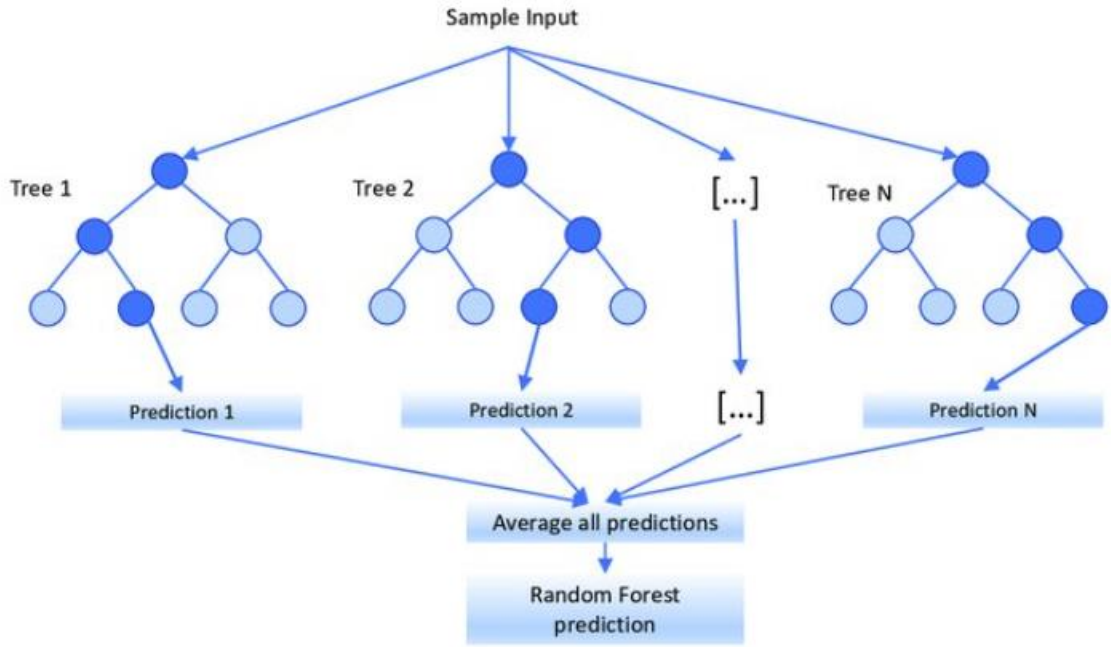

Figure 14. RF prediction scheme

Source: Segure et al., 2022

To test the RF classification model, several trees are set to 100 and the seed is set to 0. The code is shown in Figure 15. with all metrics. Most authors who have conducted similar research achieved the best results with RF classifiers, as already mentioned in the text.

```
// Classification model using Random Forest

var classifierRF = ee.Classifier.smileRandomForest({
  numberOfTrees: 100, // You can adjust the number of trees as needed
  seed: 0
}).train(trainSet, label, bands);

// // Classify image Random Forest

var classifiedRF = input.classify(classifierRF);

Map.addLayer(classifiedRF.clip(shp), {palette: landcoverPalette, min: 0, max: 4}, 'classification Random Forest');

// Get information about the trained classifier.

print('Results of trained classifier', classifierRF.explain());

var trainAccuracy = classifierRF.confusionMatrix();

print('Training error matrixRandom Forest', trainAccuracy);
print('Training overall accuracyRandom Forest', trainAccuracy.accuracy());
print("Consumer's accuracy Random Forest", trainAccuracy.consumersAccuracy());
print("Producer's accuracy Random Forest", trainAccuracy.producersAccuracy());
print('Kappa statistic for Random Forest', trainAccuracy.kappa());
```

Figure 15. Classification Model: RF

Source: Šumanovac, 2024

3. 10. SVM Classification Model

Another widely used machine learning algorithm for remote sensing and LULC is SVM. It has gained importance because it achieves high classification quality with a limited number of training examples (Mantero et al., 2005). SVM is a linear binary classification method in which the working principle is based on the concept that training samples that are close to the class boundaries discriminate a class better than other training samples. SVM is based on finding an optimal hyperplane that separates different classes in given training samples. In Figure 16., it can be seen that the samples that are close to the classes and have the smallest distance to the hyperplane are used as support vectors. These support vectors are then used for training (Shetty, 2019). The most important parameters for determining the support vectors are the kernel function, the gamma, and the cost parameters (Murtaza et al., 2014). SVM is used for solving complex multi-class problems with two solutions. The first solution is referred to as one against all, where all classes are taken together and generate a number of n classes. The second solution is referred to as one against one, where the method s $(n(n-1))/2$ generates pairwise classifiers for all possible combinations from the given input classes (Pal et al., 2005).

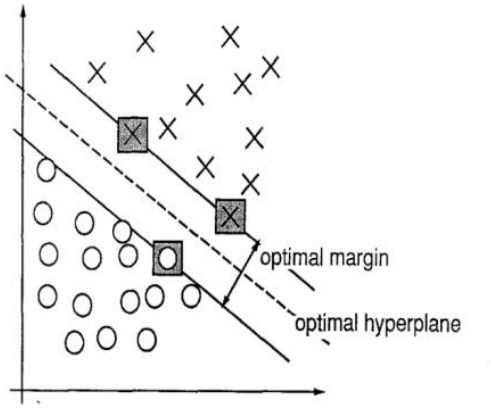

Figure 16. Hyperplane and Supporting vector

Source: Cortes et al., 1995

In Figure 16. there're classes in 2-dimensional space that are linearly separable, and this is the reason why the support vectors are on the decision boundary. It is also important to mention that this is not an ordinary example. Figure 17. shows the code for the LIBSVM classifier, using the same principle as in the previous examples.

```
// Classification LIBSVM

var classifierLIBSVM = ee.Classifier.libsvm().train(trainSet, label, bands);

// Classify image

var classfiedLIBSVM = input.classify(classifierLIBSVM);

Map.addLayer(classfiedLIBSVM.clip(shp), {palette: landcoverPalette, min: 0, max: 4}, 'Classification LIBSVM');

// Get information about the trained classifier.

print('Results of trained classifier', classifierLIBSVM.explain());

// Get a confusion matrix and overall accuracy for the training sample.

var trainAccuracy2= classifierLIBSVM.confusionMatrix();

print('Training error matrix for LIBSVM', trainAccuracy2);
print('Training overall accuracy for LIBSVM', trainAccuracy2.accuracy());
print("Consumer's accuracy for LIBSVM ", trainAccuracy2.consumersAccuracy());
print("Producer's accuracy for LIBSVM", trainAccuracy2.producersAccuracy());
print('Kappa statistic for LIBSVM', trainAccuracy2.kappa());
```

Figure 17. Classification model: LIBSVM

Source: Šumanovac, 2024

3. 11. kNN Classification Model

The fifth classification model used in this study is well known. As shown in Table 1., kNN is an instance-based learning classification model. Among the various machine learning algorithms, kNN is one of the simplest and is widely used in all types of classification models due to its adaptability and simplicity (Mahesh, 2020). This classifier predicts the unknown sample based on the features of the given training data. For this purpose, the nearest neighbor that is closest to the samples to be tested is determined from the training data. Then the majority rule is applied to decide which classification should be selected or finalized (Zhang et al., 2017). There are some advantages and disadvantages of using this algorithm. Classic kNN has some problems as it is unbiased for all classification-dependent neighbors, moreover, some unnecessary data features are included in the classification model, etc. (Bhatia et al., 2010). The positive sides are that different data points can be weighted, k-parameters can be optimized, etc. (Lamba et al., 2016). The classical kNN classifier consists of a variable parameter called k. Figure 18. illustrates the working principle of the classical kNN. Since k is 3 for Querry B, it finds 3 nearest neighbors in the vicinity. In this case, there are 2 class 1 and 1 class 0. After applying the majority rule, the final classification will result in class 1. The same is true for query A, 3 class 0, and 2 class 1. Since there is 1 more class 0 than class 1, the majority rule will determine that it is class 0. Even though this is an example of classical kNN, there are several kNN variants. These include Adaptive KNN, Fuzzy KNN, Weight Adjusted KNN, Mutual KNN, K-means clustering-based KNN, etc. (Udin et al., 2022).

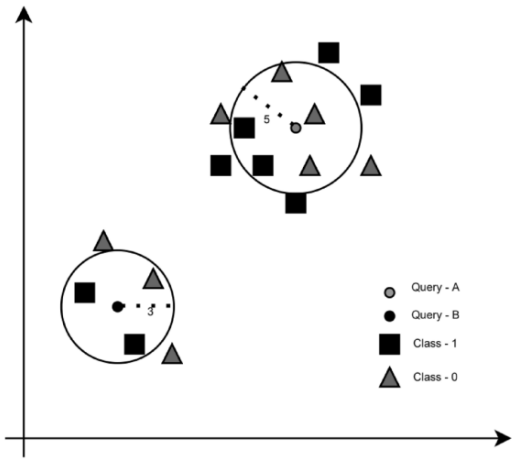

Figure 18. Working principle of kNN algorithm

Source: Udin et al., 2022

The same instance and methodology were used to create a line of code for the kNN classification model. The code is shown in Figure 19.

```
//Classification kNN

var classifiedkNN = ee.Classifier.smileKNN(5).train(trainSet, label, bands);

// Classify image

var classfiedkNN = input.classify(classifiedkNN);

Map.addLayer(classfiedkNN.clip(shp), {palette: landcoverPalette, min: 0, max: 4}, 'Classification kNN');

// Get information about the trained classifier.

print('Results of trained classifier', classifiedkNN.explain());

// Get a confusion matrix and overall accuracy for the training sample.

var trainAccuracy3= classifiedkNN.confusionMatrix();

print('Training error matrix for kNN', trainAccuracy3);
print('Training overall accuracy for kNN', trainAccuracy3.accuracy());
print("Consumer's accuracy for kNN", trainAccuracy3.consumersAccuracy());
print("Producer's accuracyn for kNN", trainAccuracy3.producersAccuracy());
print('Kappa statistic for kNN', trainAccuracy3.kappa());
```

Figure 19. Classification Model: kNN

Source: Šumanovac, 2024

3. 12. Minimum Distance Classification Model

The last classifier used in this study is MD. The first step in the working principle of the MD machine learning algorithm is to calculate mean vectors and draw decision boundaries. The second step is to assign the pixels to the closest class according to the previously drawn decision boundary (Torabi et al., 2019). As illustrated in Figure 20.

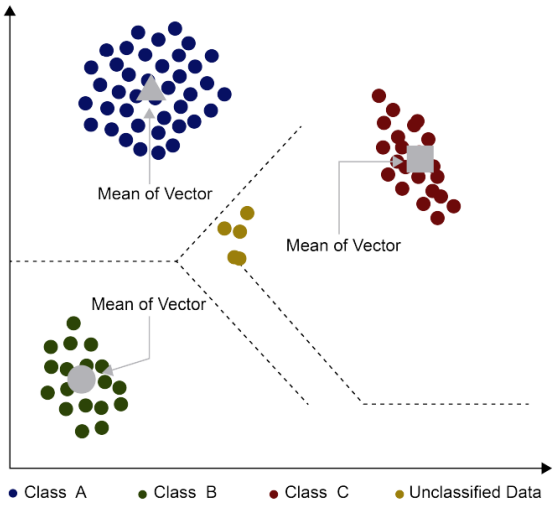

Figure 20. MD working principle

Source: Murtaza et al., 2014

Like other classification methods, this algorithm also uses two bands to evaluate the training data. The formula to calculate the Euclidean distance for each pixel in the image is as follows:

$$\mathcal{D}_i(x) = \sqrt{(X - m_i)^T (x - m_i)}$$

, where D = Euclidian distance, i = the ith class, x = n-dimensional data (where n is the number of bands), and mi = mean vector of a class (Murtaza et al., 2014). Figure 21. MD Classification Model in the GEE.

```
//Classification Minimum Distance

var classifiedMD = ee.Classifier.minimumDistance('mahalanobis').train(trainSet, label, bands);

// Classify image

var classfiedMD = input.classify(classifiedMD);

Map.addLayer(classfiedMD.clip(shp), {palette: landcoverPalette, min: 0, max: 4}, 'Classification Minimum distance');

// Get information about the trained classifier.

print('Results of trained classifier', classifiedMD.explain());

// Get a confusion matrix and overall accuracy for the training sample.

var trainAccuracy4= classifiedMD.confusionMatrix();

print('Training error matrix for Minimum Distance', trainAccuracy4);
print('Training overall accuracy for Minimum Distance', trainAccuracy4.accuracy());
print("Consumer's accuracy for Minimum Distance", trainAccuracy4.consumersAccuracy());
print("Producer's accuracyn for Minimum Distance", trainAccuracy4.producersAccuracy());
print('Kappa statistic for Minimum Distance', trainAccuracy4.kappa());
```

Figure 21. Classification Model: MD

Source: Šumanovac, 2024

Accuracy evaluation is used to evaluate the performance of the classification methods and also the training sample design methods. In this study, 80 training samples were purposively selected for each of the five classes throughout the study area. It was of great importance to select the samples correctly and to be as accurate as possible, as this determines the overall accuracy of the classification. There are several available performance evaluation metrics in

the GEE software, such as Overall accuracy, Error matrix, Kappa statistic, Producer's accuracy, and Consumer's accuracy, which we use as metrics to perform evaluations based on the confusion matrix. The confusion matrix has already been explained in the text. Overall accuracy (OA) is one of the most commonly used metrics because it is simple and easy to interpret. OA expresses the percentage of data correctly classified by the classification method, as explained in the following formula: (Plourde, 2003).

$$\textbf{Overall Accuracy (OA)} = \frac{\text{Number of Correctly Classified Samples}}{\text{Number of Total Samples}} \%$$

The second metric used for evaluation was Consumer accuracy (CO). Consumer fidelity tests the reliability of the confusion matrix, which was defined as correct for each row of the matrices (GEE Reference, 2024). The formula according to Nasiri et al. for the calculation is as follows (Nasiri et al., 2022) :

$$\textbf{Consumers Accuracy (CO)} = \frac{\text{Number of Correctly Classified Samples in each Class}}{\text{Number of Samples Classified to that Class}}$$

Cohen's kappa, denoted by the Greek letter κ, is a robust statistical measure for assessing the reliability of ratings or classifications, regardless of whether they were made by different raters (interrater reliability) or by the same rater at different times (interrater reliability). It has similarities with correlation coefficients and ranges from -1 to +1. In this range, 0 means the degree of agreement that is to be expected purely by chance, while 1 stands for perfect agreement between the raters. Like other correlation statistics, kappa is standardized and can be interpreted consistently across different studies (McHugh, 2012). Cohen proposed the following interpretation of the kappa results, which are shown in Table 2.

Table 2. Cohen's interpretation of Kappa results

Value of Kappa	Level of Agreement	% Reliable Data
0-.20	None	0-4%
.21-.39	Minimal	4-15%
.40-.59	Weak	15-35%
.60-.79	Moderate	35-63%

.80-.90	Strong	64-81%
Above .90	Almost Perfect	82-100%

Source: McHugh, 2012.

And the formula for calculating the Kappa Coefficient according to Nasiri et al., goes as follows (Nasiri et al., 2022) :

$$\textbf{Kappa} = \frac{\text{Overall Accuracy} - \text{Estimated Chance Agreement}}{1 - \text{Estimated Chance Agreement}}$$

The error matrix has already been explained earlier in the text, along with the problem of the confusion matrix in classification study areas. The last of the matrices used for the analysis was the Producers Accuracy. According to the GEE reference database, Producers Accuracy (PO) calculates the accuracy of the confusion matrix as correct for each column (GEE Reference, 2024). The formula according to Nasiri et al. is as follows (Nasiri et al., 2022).

$$\textbf{Producers Accuracy (PO)} = \frac{\text{Number of Correctly Classified Samples in Each Class}}{\text{Number of Samples From Reference Data in Each Class}}$$

PO or Precision Score is the probability that the pixel was correctly classified in the given class, while CO or Recall is the probability that the pixel that was correctly classified on the map also corresponds to the class on the ground (Jain et al., 2016).

3. 13. Exporting data to the Cloud

The final puzzle of the code was exporting the data to the cloud in the form of Excel or TIF data. Figure 22. shows the last few lines of code. The first task is to export the image to Google Drive, with a maximum of 1e13 pixels, and then export the training data in the form of an Excel spreadsheet.

```
//Export classified map to Google drive

Export.image.toDrive({
  image: classified,
  description: "Sentinel_2_CART",
  scale: 10,
  region: shp,
  maxPixels: 1e13,
});

// Export training data

Export.table.toAsset({
  collection: training,
  description: 'Sentinel_2022',
  assetId: 'Sentinel_2022'
});

Export.table.toDrive({
  collection: training,
  description: 'Sentinel_2022',
  fileFormat: 'SHP'
});
```

Figure 22. Exporting data

Source: Šumanovac, 2024

4. Results

Visualization of exported TIF data from GEE processed in QGIS software. QGIS is a free and open-source software for analyzing and editing geodata (Menke et al., 2016). Some of the steps involved in processing the data are importing raster data (each of the classification methods per year) and vector data (OB County shapefile), clipping the rasters using a mask layer, and classifying the classes into palletized/unique values. The same HTML values that were in the code are also used for the visualization of the five classes. The final step was to create two mappings, each of which contains three classification methods, as shown in Figure 23.

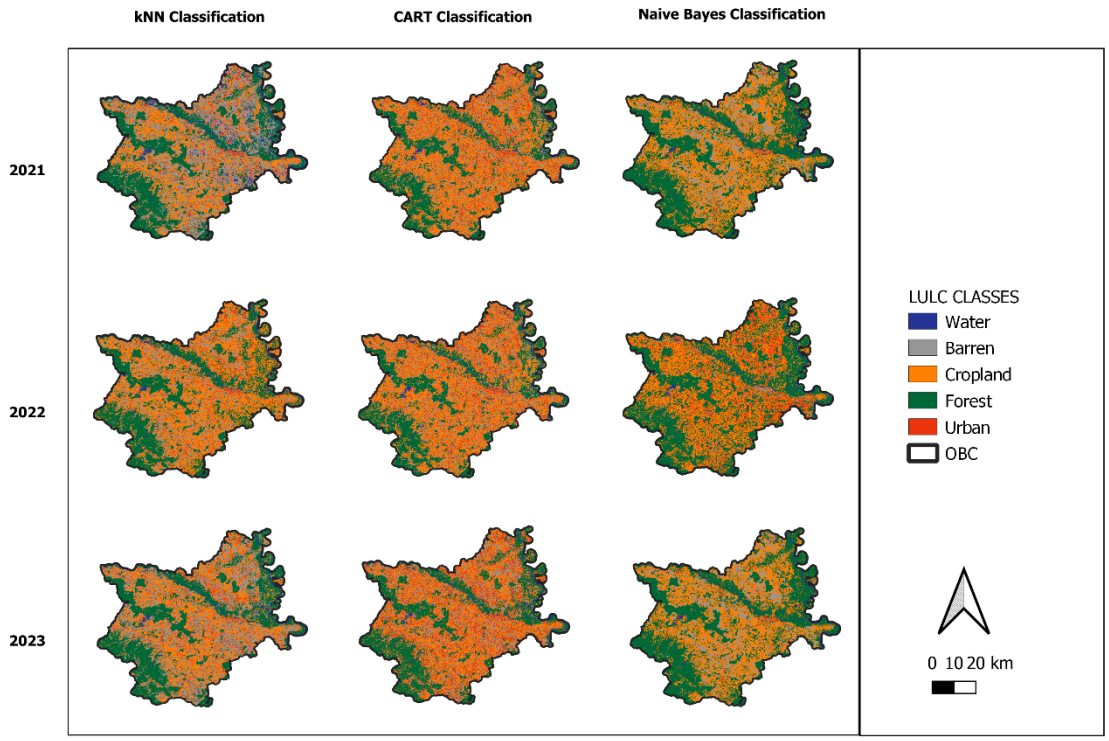

Figure 23. LULC Classification map for kNN, CART, and Naive Bayes Classifier

Source: Šumanovac, 2024

Figure 23. shows LULC classification maps for kNN, CART, and Naive Bayes classifiers for the years 2021, 2022, and 2023. Figure 24. shows the LULC classification maps for RF, MD, and libsvm classifiers for the same years as in the previous figure.

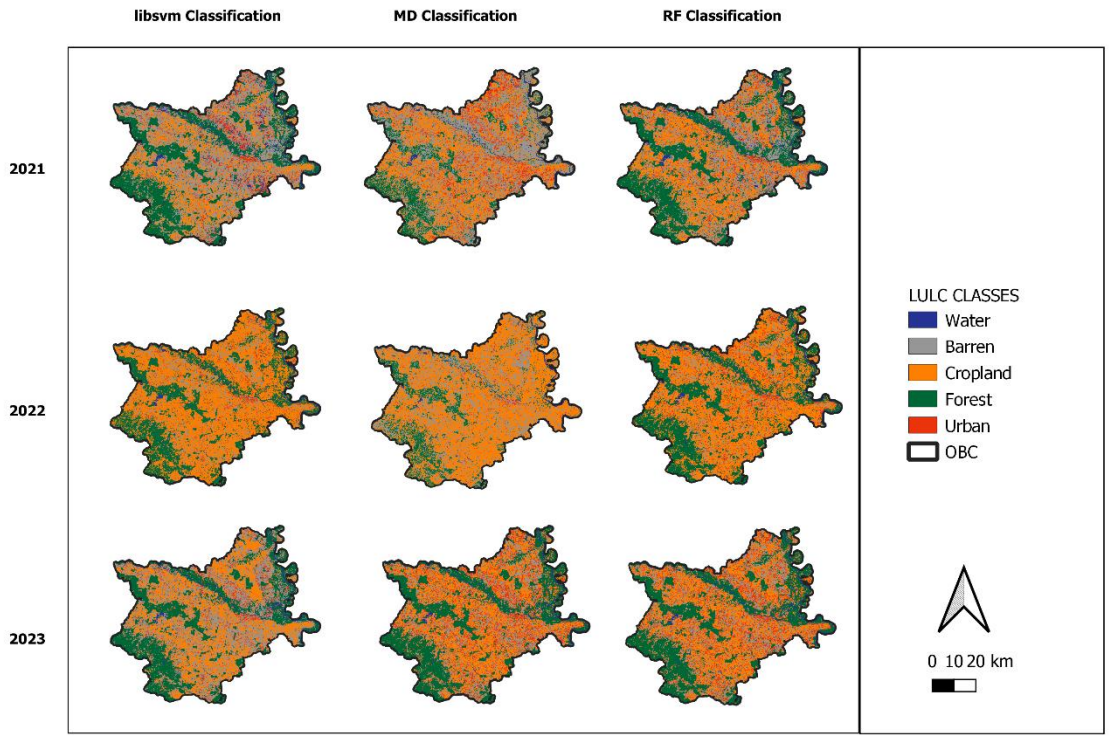

Figure 24. LULC Classification map for libsvm, MD and RF Classifier

Source: Šumanovac, 2024

When comparing these two figures, some differences between the six classifiers can be seen. The kNN classifier classified part of the area as infertile in 2021, while this year it was classified as arable land and forest, and in 2023, as in 2021, a larger part of the infertile area was recognized. In the CART classifier, more and more of the forest class was classified with advancing age. If one compares the Naïve Bayes classification method with CART and kNN, for example, there are far more forest and arable land classes and fewer barren, urban classes in the latter classification methods. In the libsvm classification, the year 2022 has more arable land and forest classes than the years 2023 and 2021, while the percentage of infertile classes in the figure is significantly higher. The MD classification method has much fewer forest classes compared to RF and CART classifiers, for example, especially in 2021 and 2022. The MD classifier has forest classes that are incorrectly classified as barren classes in 2021, while 2022 has many forest classes that are incorrectly classified as cropland. For the RF classification method, most arable land classes are detected in 2022, so a continuous trend can be seen for all classification methods during the three years. To evaluate the accuracy of the

methods, the average OA, PO, CO, and Kappa were determined for each year. The table below shows the results of the accuracy assessment for 2021:

Table 3**.** Accuracy Assessment for 2021

	Overall Accuracy	**Producer's Accuracy**	**Kappa Coefficient**	**Consumer's Accuracy**
CART	0,81	0,82	0,77	0,82
Naïve Bayes	0,64	0,62	0,54	0,70
Random Forest	0,99	0,99	0,99	0,99
libsvm	0,79	0,78	0,74	0,78
kNN	0,86	0,86	0,83	0,88
MD	0,76	0,76	0,70	0,79

Source: Šumanovac, 2024

After averaging the 4 metrics of the confusion matrix, it can be seen that the RF classification method provides by far the best results with an average accuracy of 99%. The second best method used was kNN with an average accuracy of 85%, closely followed by libsvm, and the worst classifier was Naive Bayes with an OA of 64%. The kappa coefficient of the RF classifier has almost perfectly reliable data, while Naive Bayes has between 15-25% reliable data. Table 3. shows the results for the year 2022:

Table 4. Accuracy Assessments for the year 2022

	Overall Accuracy	**Producer's Accuracy**	**Kappa Coefficient**	**Consumer's Accuracy**
CART	0,76	0,78	0,74	0,79
Naïve Bayes	0,74	0,76	0,68	0,74
RF	0,99	0,99	0,99	0,99
libsvm	0,80	0,78	0,75	0,84
kNN	0,85	0,84	0,81	0,85
MD	0,78	0,78	0,73	0,79

Source: Šumanovac, 2024

According to the average calculations for 2022, the trends are similar to those for 2021. The best classification method is the RF Classifier with almost perfectly reliable data and an average value of 99%. In second place is again the kNN classification method with similar results as in 2021. The Naive Bayes classification method achieved significantly better results in 2022 with an overall accuracy of 72% compared to 64% in 2021. The kappa coefficient increased from 54% in 2021 to 68% in 2022. The MD classification method also showed an improvement of 2-3% in each metric, while the CART classification method showed a lower accuracy of 3% in OA, PO, and kappa coefficient. The kappa coefficient again shows the best results for the RF classifier, while the kNN classifier shows a high level of agreement and 64-81% reliable data. Table 4. shows the results for the year 2023:

Table 5. Accuracy Assessment for the year 2023

	Overall Accuracy	**Producer's Accuracy**	**Kappa Coefficient**	**Consumer's Accuracy**
CART	0,75	0,72	0,64	0,72
Naïve Bayes	0,62	0,61	0,53	0,62
RF	0,99	0,99	0,99	0,99
libsvm	0,75	0,74	0,69	0,78
kNN	0,84	0,83	0,79	0,70
MD	0,66	0,66	0,57	0,76

Source: Šumanovac, 2024

According to the calculations for 2023, there is a large difference between the results of the MD classifier in 2022 and 2023. For 2023, the OA is 66%, while in 2022 it was 78%; the same applies to the PO and the other two metrics. The kappa coefficient is only 57% in 2023, which means that the level of agreement is low and only 14-35% of the data is reliable. Again, the RF Classifier provides the best results, and similar trends can be seen in 2021. Naïve Bayes has a lower percentage of overall accuracy, dropping from 74% in 2022 to 62% in 2023. The CART classification method has almost the same results, while the libsvm classifier has 5% lower results than the previous year in some of the metrics.

5. Discussion

An open-source online software was used to create a code that generates six classification methods on Sentinel-2 satellite images in the study area of Osijek-Baranja County in the period between 2021, and 2023. The code consisted of 200 lines written in GEE by JavaScript programming language, while the GEE is also compatible with PythonAPI programming language (Tamiminia et al., 2020). The code consisted of six classification methods, to see which of the machine learning algorithms brings the best results and which of the algorithms was less likely to perform well. The six classification methods that we're used are kNN, MD, RF, CART, libsvm, and Naïve Bayes. Most of the researches are using only two or three classifiers, like Manzanze that only used RF classifier. (Manzanze et al., 2018). Exceptions are found in Ghyaour et al. research, in which authors used CART, RF, MD, and SVM classification methods and in Ferda's paper with 10 machine learning algorithms (Ghyaour et al., 2021, Farda, 2017). Each of these machine learning algorithms generated tiff data that was later used to create LULC classification maps in the QGIS software. QGIS as a tool used for visualisation was reliable and appropriate for monitoring land, just like in Getha's paper (Getha et al., 2018). Two maps were created, each containing three classification methods for the years 2021, 2022 and 2023. After the LULC maps were created, an accuracy assessment of the confusion matrix was performed for each of the classification methods. The metrics that we're for performance evaluation are Overall Accuracy, Producer Accuracy, Consumer Accuracy and Kappa Coefficient. Other authors are most commonly use Overall Accuracy for evaluation (Ganharum et al., 2022). The code also generated a training error matrix for each of the classification algorithms, but this was not used in the study. The analysis showed that many authors were most successful with the RF algorithm, just like in Piao's research with 98,2% of Overall Accuracy and 0,95 Kappa result, and this study also achieved good results with this classifier (Piao et al., 2021). For all metrics aand in every year, the results for RF were between 99% and 100 %. The second best classifier was kNN with an overall accuracy of 85% in 2022 and 86% in 2021. The Kappa Coefficient calculated for kNN was 75%, 81% and 79% in the years 2021 to 2023, which means that the agreement was mostly moderate and the data was 35-63% reliable according to the Cohen table. The third best classifier was libsvm, but similar results were also obtained with the CART classification method. CART classification method had lower performce, but authors like Farda had great results with 96,98% Overall accuracy, using Landsat datasets (Farda, 2017). The best results of libsvm we're in 2022 with 80% Overall accuracy, but also similar results we're in 2021, 84% of consumer accuracy was the

highest metric in 2022. For the CART classification algorithm, the producers was 82%, 78% and 72%, as shown, there is a trend that decreases every year. There is also a decreasing percentage for each metric for the CART classification method. The lowest results were obtained with the MD classification algorithm, the overall accuracy was 78%, 76% and 66%, and the lowest Kappa result in the entire study was only 57% in 2023. This means that the agreement was only moderate. Some authors had the best result with SVM classifier, while in this study SVM classifier was slightly inferior to others (Ghyaour et al., 2021). After analyzing all of the outputs, results that accompanied trends similar to other researchers with the quality of the RF classification method like in Manzazne's research for year 2015 with 98% Overall accuracy. Further development of the code is required. The visual representation of the classification algorithm was done in QGIS, a tool that has proven to be a good resource for analyzing geospatial data. LULC changes in the classes were also performed in QGIS using the SCP machine learning plugin, but a small time frame with a difference of only three years provided inappropriate results for display in the study. A longer time frame would probably provide better results, and most likely a different satellite image dataset. GEE as software performed well, was easy to use and reliable. The time to explore the tiff images was long in some cases, the problem that was also stated in Amani’s paper (Amani et al., 2020).

6. Conclusion

The performance of GEE was evaluated as an open-source tool for analyzing remote sensing data. QGIS was used as a geospatial data analysis, creation, and visualization software to create LULC maps for the data imported from GEE. The primary objective of the study was to create a classification of land and agricultural land in Osijek-Baranja County as a study area. The code was written to use the Sentinel 2 satellite image dataset and six classification methods for 5 classes in the training data. The classification methods that were used are CART, RF, MD, kNN, libsvm and Naïve Bayes. Six metrics were used to evaluate the accuracy of the classification methods: Overall Accuracy, Consumer Accuracy, Producer Accuracy and Kappa Coefficient. For the LULC maps created in QGIS, there were three classification methods per year (2021, 2022, 2023), which made the differences in the algorithms visible. The best algorithm was RF with an overall accuracy of 99% in each year. Other algorithms also performed well, but with lower efficiency. The least efficient algorithm was Naïve Bayes.

7. References

1. Alpaydin, E. (2020). *Introduction to machine learning*. MIT Press.
2. Amani, M., Ghorbanian, A., Ahmadi, S. A., Kakooei, M., Moghimi, A., Mirmazloumi, S. M., ... Brisco, B. (2020). Google Earth Engine cloud computing platform for remote sensing big data applications: A comprehensive review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 5326-5350.
3. Amani, M., Ghorbanian, A., Ahmadi, S. A., Kakooei, M., Moghimi, A., Mirmazloumi, S. M., ... Brisco, B. (2020). Google Earth Engine cloud computing platform for remote sensing big data applications: A comprehensive review. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, 5326-5350.
4. Amani, M., Ghorbanian, A., Ahmadi, S. A., Kakooei, M., Moghimi, A., Mirmazloumi, S. M., ... Brisco, B. (2020). Google Earth Engine cloud computing platform for remote sensing big data applications: A comprehensive review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, 5326-5350.*
5. Amani, M., Kakooei, M., Moghimi, A., Ghorbanian, A., Ranjgar, B., Mahdavi, S., ... Mohammadzadeh, A. (2020). Application of Google Earth Engine cloud computing platform, sentinel imagery, and neural networks for crop mapping in Canada. *Remote Sensing*, *12*(21), 3561.
6. Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., ... Zhang, J. D. (2020). An introduction to machine learning. *Clinical pharmacology therapeutics*, *107*(4), 871-885.
7. Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., ... Zhang, J. D. (2020). An introduction to machine learning. *Clinical pharmacology therapeutics*, *107*(4), 871-885.
8. Bhatia, N. (2010). Survey of nearest neighbor techniques. *arXiv bpreprint arXiv:1007.0085*.
9. Bishop, C. M., Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). New York: Springer.
10. Carrasco, L., O'Neil, A. W., Morton, R. D., Rowland, C. S. (2019). Evaluating combinations of temporally aggregated Sentinel-1, Sentinel-2, and Landsat 8 for land cover mapping with Google Earth Engine. *Remote Sensing*, *11*(3), 288.

11. Domingos, P., Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, *29*, 103-130.
12. Elmahdy, S., Mohamed, M., Ali, T. (2020). Land use/land cover changes impact on groundwater level and quality in the northern part of the United Arab Emirates. *Remote Sensing*, *12*(11), 1715.
13. European Space Agency (ESA). (n.d.). *Sentinel-2 mission*. Sentinel Online. Retrieved [February 22, 2024], from https://sentinel.esa.int/web/sentinel/missions/sentinel-2
14. Farda, N. M. (2017, December). Multi-temporal land use mapping of coastal wetlands area using machine learning in Google Earth engine. In IOP Conference Series: Earth and Environmental Science (Vol. 98, No. 1, p. 012042). IOP Publishing.
15. Fernando, W. A. M., Senanayake, I. P. (2023). Developing a two-decadal time-record of rice field maps using Landsat-derived multi-index image collections with a random forest classifier: A Google Earth Engine-based approach. *Information Processing in Agriculture*.
16. Foody, G. M., Mathur, A. (2004). Toward intelligent training of supervised image classifications: directing training data acquisition for SVM classification. *Remote Sensing of Environment*, *93*(1-2), 107-117.
17. Freie Universität Berlin. (n.d.). *Random Forest*. FU Blog Service. Retrieved [March 5, 2024], from https://blogs.fu-berlin.de/reseda/random-forest
18. Gandharum, L., Hartono, D. M., Karsidi, A., Ahmad, M. (2022). Monitoring urban expansion and loss of agriculture on the north coast of west Java province, Indonesia, using Google Earth Engine and intensity analysis. The Scientific World Journal, 2022.
19. Geetha, M., Karegowda, A., Sudhira, H. S. (2019). Land use and land cover mapping of Davangere using Google Earth Engine. International Journal of Recent Technology and Engineering (IJRTE), 3, 474.
20. Ghayour, L., Neshat, A., Paryani, S., Shahabi, H., Shirzadi, A., Chen, W., ... Ahmad, A. (2021). Performance evaluation of sentinel-2 and landsat 8 OLI data for land cover/use classification using a comparison between machine learning algorithms. Remote Sensing, 13(7), 1349.
21. Google for Developers. (n.d.). *Sentinel-2 datasets*. Retrieved [March 1, 2024], from https://developers.google.com/earth-engine/datasets/catalog/sentinel-2
22. Gorelick, N. Google Earth Engine. In EGU General Assembly Conference Abstracts; American Geophysical Union: Vienna, Austria, 2013; p. 11997.

23. Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. Remote sensing of Environment, 202, 18-27.
24. Herold, M., Woodcock, C., Cihlar, J., Wulder, M., Arino, O., Aachard, F., Sessa, R. (2009). Assessment of the Status of the Development of the Standards for the Terrestrial Essential Climate Variables: T9 Land Cover.
25. HTML Color Codes. (n.d.). *HTML Color Codes*. Retrieved [March 4, 2024], from https://htmlcolorcodes.com/
26. Immitzer, M., Vuolo, F., Atzberger, C. (2016). First experience with Sentinel-2 data for crop and tree species classifications in central Europe. *Remote sensing*, *8*(3), 166.
27. Jain, M., Dawa, D., Mehta, R., Dimri, A. P., Pandit, M. K. (2016). Monitoring land use change and its drivers in Delhi, India using multi-temporal satellite data. *Modeling earth systems and environment*, *2*, 1-14.
28. Jiang, L., Wang, W., Yang, X., Xie, N., Cheng, Y. (2011). Classification methods of remote sensing image based on decision tree technologies. In *Computer and Computing Technologies in Agriculture IV: 4th IFIP TC 12 Conference, CCTA 2010, Nanchang, China, October 22-25, 2010, Selected Papers, Part I 4* (pp. 353-358). Springer Berlin Heidelberg.
29. Kadirhodjaev, A., Kadavi, P. R., Lee, C. W., Lee, S. (2018). Analysis of the relationships between topographic factors and landslide occurrence and their application to landslide susceptibility mapping: a case study of Mingchukur, Uzbekistan. *Geosciences journal*, *22*, 1053-1067.
30. Karishma, C. G., Kannan, B., Nagarajan, K., Panneerselvam, S., Pazhanivelan, S. (2022). Land use land cover change detection in the lower Bhavani basin, Tamil Nadu, using geospatial techniques. Journal of Applied and Natural Science, 14(SI), 58-64.
31. Kotsiantis, S. B., Zaharakis, I., Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), 3-24.
32. Kumar, L., Mutanga, O. (2018). Google Earth Engine applications since inception: Usage, trends, and potential. *Remote sensing*, *10*(10), 1509.
33. Kurt Menke, G. I. S. P., Smith Jr, R., Pirelli, L., John Van Hoesen, G. I. S. P. (2016). *Mastering QGIS*. Packt Publishing Ltd.

34. Lamba, A., Kumar, D. (2016). Survey on KNN and its variants. *Int. J. Adv. Res. Comput. Commun. Eng*, *5*(5), 430-435.
35. Li, C., Gong, P., Wang, J., Zhu, Z., Biging, G. S., Yuan, C., ... Clinton, N. (2017). The first all-season sample set for mapping global land cover with Landsat-8 data. *Science Bulletin*, *62*(7), 508-515.
36. Mahesh, B. (2020). Machine learning algorithms review. *International Journal of Science and Research (IJSR).[Internet]*, *9*(1), 381-386.
37. Mananze, S., Pôças, I., Cunha, M. (2020). Mapping and assessing the dynamics of shifting agricultural landscapes using Google Earth Engine cloud computing, a case study in Mozambique. Remote Sensing, 12(8), 1279.
38. Mantero, P., Moser, G., Serpico, S. B. (2005). Partially supervised classification of remote sensing images through SVM-based probability density estimation. *IEEE Transactions on geoscience and remote sensing*, *43*(3), 559-570.
39. McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia Medica*, *22*(3), 276-282.
40. Moore, R., Parsons, E. (2011). Beyond SDI, bridging the power of cloud-based computing resources to manage global environment issues. In *INSPIRE Conference 2011, June 27–July 1, 2011, Edinburgh*.
41. Murtaza, K. O., Romshoo, S. A. (2014). Determining the suitability and accuracy of various statistical algorithms for satellite data classification. *International journal of geomatics and geosciences*, *4*(4), 585-599.
42. Murtaza, K. O., Romshoo, S. A. (2014). Determining the suitability and accuracy of various statistical algorithms for satellite data classification. *International journal of geomatics and geosciences*, *4*(4), 585-599.
43. Naghibi, S. A., Pourghasemi, H. R., Dixon, B. (2016). GIS-based groundwater potential mapping using boosted regression tree, classification and regression tree, and random forest machine learning models in Iran. *Environmental monitoring and assessment*, *188*, 1-27.
44. Paas, W., Groot, J. C. (2017). Creating adaptive farm typologies using Naive Bayesian classification. *Information Processing in Agriculture*, *4*(3), 220-227.
45. Pal, M., Mather, P. M. (2005). Support vector machines for classification in remote sensing. *International journal of remote sensing*, *26*(5), 1007-1011.

46. Richards J.A. (2013) Supervised Classification Techniques. In: Remote Sensing Digital Image Analysis. Springer, Berlin, Heidelberg. pp 247-318
47. Richards, J. A., Richards, J. A. (2022). *Remote sensing digital image analysis* (Vol. 5). Berlin/Heidelberg, Germany: Springer.
48. Salazar, A., Baldi, G., Hirota, M., Syktus, J., McAlpine, C. (2015). Land use and land cover change impacts on the regional climate of non-Amazonian South America: A review. *Global and Planetary Change*, *128*, 103-119.
49. Segura, D., Khatib, E. J., Barco, R. (2022). Dynamic packet duplication for industrial URLLC. *Sensors*, *22*(2), 587.
50. Shabani, S., Jaafari, A., Bettinger, P. (2021). Spatial modeling of forest stands for susceptibility to logging operations. *Environmental Impact Assessment Review*, *89*, 106601.
51. Shetty, S. (2019). *Analysis of machine learning classifiers for LULC classification on Google Earth Engine* (Master's thesis, University of Twente).
52. Tamiminia, H., Salehi, B., Mahdianpari, M., Quackenbush, L., Adeli, S., Brisco, B. (2020). Google Earth Engine for geo-big data applications: A meta-analysis and systematic review. ISPRS journal of photogrammetry and remote sensing, 164, 152-170.
53. Tharwat, A. (2020). Classification assessment methods. *Applied computing and informatics*, *17*(1), 168-192.
54. Torabi, M., Hashemi, S., Saybani, M. R., Shamshirband, S., Mousavi, A. (2019). A Hybrid clustering and classification technique for forecasting short-term energy consumption. *Environmental progress sustainable energy*, *38*(1), 66-76.
55. Uddin, S., Haque, I., Lu, H., Moni, M. A., Gide, E. (2022). Comparative performance analysis of K-nearest neighbor (KNN) algorithm and its different variants for disease prediction. *Scientific Reports*, *12*(1), 6256.
56. Wikipedia contributors. (n.d.). *Osječko-baranjska županija*. Wikipedia, The Free Encyclopedia. Retrieved [February 25, 2024], from https://hr.wikipedia.org/wiki/Osje%C4%8Dko-baranjska_%C5%BEupanija
57. Zhang, S., Li, X., Zong, M., Zhu, X., Cheng, D. (2017). Learning k for known classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(3), 1-19.

58. Zhao, Q., Yu, L., Li, X., Peng, D., Zhang, Y., Gong, P. (2021). Progress and trends in the application of Google Earth and Google Earth Engine. *Remote Sensing*, *13*(18), 3778.
59. Zurqani, H. A., Allen, J. S., Post, C. J., Pellett, C. A., Walker, T. C. (2021). Mapping and quantifying agricultural irrigation in heterogeneous landscapes using Google Earth Engine. Remote Sensing Applications: Society and Environment, 23, 100590.

8. Summary

Research was carried out to classify the land and agricultural land in Osijek-Baranja County. Six classification methods were tested to evaluate the efficiency of these machine learning algorithms, identifying the most efficient and the least efficient algorithms. LULC maps were created in the QGIS software, each containing three classification methods for the years 2021, 2022, and 2023. The first LULC map contained the classification algorithms kNN, CART, and Naïve Bayes, and the second LULC map contained the classification algorithms MD, RF, and libsvm. QGIS proved to be a valuable tool for visualizing the data exported from GEE. The RF was the most efficient classification method after evaluation with 99% Overall Accuracy in each study year, along with other metrics such as Producer's Accuracy, Consumer's Accuracy, and Kappa Coefficient. The least efficient algorithm was Naïve Bayes.

Key words: Google Earth Engine, remote sensing, machine learning, algorithms, classification

9. Sažetak

Istraživanje provedeno je u svrhu klasifikacije zemljišta i poljoprivrednog zemljišta u Osječko-baranjskoj županiji. Testirano je šest metoda klasifikacije kako bi se procijenila učinkovitost algoritama strojnog učenja, identificirajući najučinkovitije i najmanje učinkovite algoritme. LULC karte izrađene su u softveru QGIS, a svaka sadrži tri metode klasifikacije za godine 2021., 2022. i 2023. Prva LULC karta sadržavala je klasifikacijske algoritme *kNN*, *CART* i *Naïve Bayes*, a druga *LULC* karta sadržavala je klasifikacijske algoritme *MD* , *RF* i l*ibsvm*. QGIS se pokazao kao vrijedan alat za vizualizaciju podataka eksportiranih iz GEE-a. *RF* je bio najučinkovitija metoda klasifikacije nakon evaluacije s 99% ukupne točnosti u svakoj studijskoj godini, zajedno s drugim metrikama kao što su *Producer's Accuracy, Consumer's Accuracy,* i *Kappa Coefficient*.. Najmanje učinkovit algoritam bio je Naïve Bayes.

Ključne riječi: Google Earth Engine, daljinska istraživanja, strojno učenje, algoritmi, klasifikacija

10. List of Tables

11. List of Figures

BASIC DOCUMENTATION CARD

Josip Juraj Strossmayer University of Osijek **Graduate thesis**
Faculty of Agrobiotechnical Sciences Osijek
University Graduate Studies, Digital Agriculture, Plant production

Classification of Land Cover and Agricultural Land Using Google Earth Engine

Luka Šumanovac

Abstract

Research was carried out to classify the land and agricultural land in Osijek-Baranja County. Six classification methods were tested to evaluate the efficiency of these machine learning algorithms, identifying the most efficient and the least efficient algorithms. LULC maps were created in the QGIS software, each containing three classification methods for the years 2021, 2022, and 2023. The first LULC map contained the classification algorithms kNN, CART, and Naïve Bayes, and the second LULC map contained the classification algorithms MD, RF, and libsvm. QGIS proved to be a valuable tool for visualizing the data exported from GEE. The RF was the most efficient classification method after evaluation with 99% Overall Accuracy in each study year, along with other metrics such as Producer's Accuracy, Consumer Accuracy, and Kappa Coefficient. The least efficient algorithm was Naïve Bayes.

Thesis performed at: Faculty of Agrobiotechnical Sciences Osijek
Mentor: PhD Dorijan Radočaj, Assistant Professor

Number of pages: 42
Number of figures: 24
Number of tables: 5
Number of references: 59
Number of appendices: -
Original in: English

Key words: Google Earth Engine, remote sensing, machine learning, algorithms, classification

Thesis defended on date: x

Reviewers:

1. PhD Mladen Jurišić, Full Professor with Tenure, chair
2. PhD Dorijan Radočaj, Assistant Professor, mentor
3. PhD Ivan Plaščak, Full Professor, member

Thesis deposited at: Library, Faculty of Agrobiotehnical Sciences, Josip Juraj Strossmayer University ofOsijek, Vladimira Preloga 1

TEMELJNA DOKUMENTACIJSKA KARTICA

Sveučilište Josipa Jurja Strossmayera u Osijeku **Diplomski rad**
Fakultet agrobiotehničkih znanosti Osijek
Sveučilišni diplomski studij, Digitalna poljoprivreda, Biljna proizvodnja

Klasifikacija zemljišta I poljoprivrednog zemljišta pomoću Google Earth Enginea

Luka Šumanovac

Sažetak

Istraživanje je provedeno u svrhu klasifikacije zemljišta i poljoprivrednog zemljišta u Osječko-baranjskoj županiji. Testirano je šest metoda klasifikacije kako bi se procijenila učinkovitost ovih algoritama strojnog učenja, identificirajući najučinkovitije i najmanje učinkovite algoritme. LULC karte izrađene su u softveru QGIS, a svaka sadrži tri metode klasifikacije za godine 2021., 2022. i 2023. Prva LULC karta sadržavala je klasifikacijske algoritme *kNN, CART* i *Naïve Bayes*, a druga *LULC* karta sadržavala je klasifikacijske algoritme *MD , RF* i l*ibsvm*. QGIS se pokazao kao vrijedan alat za vizualizaciju podataka eksportiranih iz GEE-a. *RF* je bio najučinkovitija metoda klasifikacije nakon evaluacije s 99% ukupne točnosti u svakoj studijskoj godini, zajedno s drugim metrikama kao što su *Producer's Accuracy, Consumer's Accuracy*, i *Kappa Coefficient*.. Najmanje učinkovit algoritam bio je Naïve Bayes.

Rad je izrađen pri: Fakultet agrobiotehničkih znanosti Osijek
Mentor: doc. dr. sc. Dorijan Radočaj

Broj stranica: 42
Broj grafikona i slika: 24
Broj tablica: 5
Broj literaturnih navoda: 59
Broj priloga: -
Jezik izvornika: engleski

Ključne riječi: Google Earth Engine, daljinska istraživanja, strojno učenje, algoritmi, klasifikacija

Datum obrane: x

Stručno povjerenstvo za obranu:

1. prof. dr. sc. Mladen Jurišić, predsjednik
2. doc. dr. sc. Dorijan Radočaj, mentor
3. izv. prof. dr. sc. Ivan Plaščak, član

Rad je pohranjen u: Knjižnici Fakulteta agrobiotehničkih znanosti Osijek, Sveučilište Josipa JurjaStrossmayera u Osijeku, Vladimira Preloga 1