Generating Land Cover Maps Using Machine Learning Algorithms

Abdurrahman YAMAN
Department of Electrical and Electronics Engineering
University of Ankara Yildirim Beyazit University
Ankara, TURKEY
17050261002@ybu.edu.tr

***Abstract*— Climate change has become a serious threat to humanity. Global land cover maps serve as an important means to tackle this problem. ESA has produced Global Land Cover Map from Sentinel-2 data. In this paper 3 (three) methods are used to generate landcover maps for unseen data by using limited number of training samples. Gibraltar is selected as a study area as there is a wide variety of land cover types.**

Keywords—Dataset, Machine Learning, Regression, Decision tree, k Nearest Neighbors, Linear Regression, Land Cover Map

I. INTRODUCTION

The aim of this paper is to show that, how generation land cover maps with machine learning algorithms is done. There are three methods are used to generate maps. Performance of the algorithms are compared with their F1 score values; K Nearest Neighbors Classification Method: 0.33985, Decision Tree Classification Method: 0.32997, Gaussian Naïve Bayes: 0.34304.

II. DATASET

Firstly, the dataset is pre-processed. Id columns are dropped. To test the algorithm inside the compiler, input test values and sample submission values are combined.

Then NDVI and NDWI values computed. NDVI (Normalized Difference Vegetation Index) and NDWI (Normalized Difference Water Index) are commonly used remote sensing indices that are employed for analyzing satellite imagery. They are frequently used for detecting and monitoring vegetation, water bodies and other land cover features.

NDVI is computed by using the formula

(NIR - Red) / (NIR + Red)

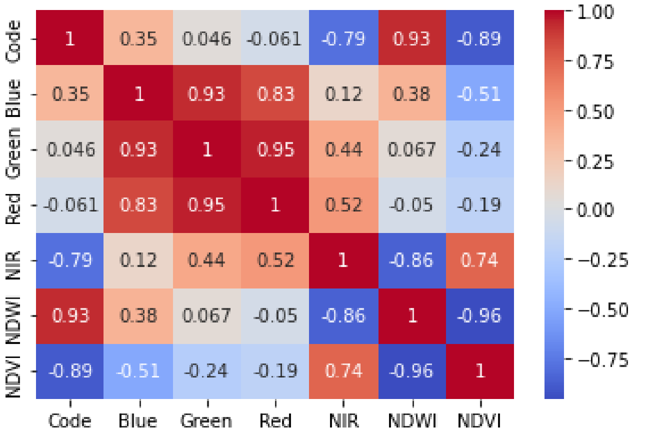
where NIR (Near Infrared) and Red are the values of the bands of the satellite image. NDVI ranges from -1 to 1, with values close to 1 indicating dense vegetation, and values close to -1 indicating water or bare soil. NDVI values close to zero suggest an absence of vegetation.

NDWI is computed by using the formula

(Green - NIR) / (Green + NIR)

where Green and NIR are the values of the bands of the satellite image. NDWI ranges between -1 and 1, with values close to 1 indicating the presence of water, and values close to -1 indicating the presence of vegetation or land. Both NDVI and NDWI use a ratio of two bands of a satellite image, with NDVI using Red and NIR bands and NDWI using Green and NIR bands. NDVI is primarily used for vegetation analysis and NDWI is used for water analysis. NDVI and NDWI are widely used and are extremely helpful in studying vegetation and water bodies.

After the calculation, correlation matrix is prepared. The correlation for all features are as follows:



Because the correlation of Red and Green is low, the features are dropped. Then, values of Blue and NIR values are divided by 1000 to normalize the data. After that, new dataset is used as train data.

III. LITERATURE REVIEW

There are 2 articles in total that have been studied on this subject.

A. *Investigation of the Effects of Kernel Functions in Satellite Image Classification Using Support Vector Machines (Harita Dergisi Temmuz 2010 Sayı 144-Taşkın KAVZOĞLU, İsmail ÇÖLKESEN) In this study, a detailed performance analysis was carried out for SVMs with regard to kernel function and chosen parameter values. In the analyses four particular kernel functions that have been most widely used in the literature were investigated with their effects on classification accuracy. It is found that radial basis function and Pearson VII function kernels produced the highest performance (>94%) for the data set considered in this study. Also, normalized polynomial kernel showed a poor performance with the lowest classification accuracy (91,78%) among all SVM models. On the other hand, it is found that all SVM models produced more accurate results compared to maximum likelihood classification. [1]*

B. *Kernel Function Selection for the Solution of Classification Problems via Support Vector Machines (ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ İİBF DERGİSİ, NİSAN 2014, 9(1), 175- 198 - Sevgi AYHAN, Şenol ERDOĞMUŞ) One of the most important machine learning algorithms developed for to accomplish classification task of data mining is Support Vector Machines. In the literature, Support Vector Machines has been shown to outperform many other techniques. Kernel function selection and parameter optimization play important role in implementation of Support Vector Machines. In this study, Kernel function selection process was ground on the randomized block experimental design. Univariate ANOVA was utilized for kernel function selection. As a result, the research proved that radial based Kernel function was the most successful Kernel function was proved. [2]*

IV. METHODS

Three learning method is used in this study. Which are;

(i) K Nearest Neighbors Classification Method,

(ii) Decision Tree Classification Method, and,

(iii) Gaussian Naive Bayes.

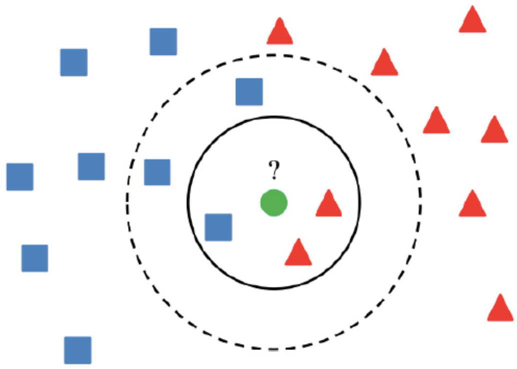
A. *K Nearest Neighbors Classification Method,*

K-Nearest Neighbors (KNN) is a classification method that involves finding the k-number of closest training examples in feature space and determining the class of a new sample based on the majority of the classes of those examples.

The main hyperparameter in KNN is "k", which determines the number of nearest neighbors to consider when making a decision.

Another hyperparameter is the distance metric used to calculate the distance between a sample and its nearest neighbors. Common distance metrics include Euclidean, Manhattan, and Minkowski distance.

KNN also has a weighting scheme, which assigns a weight to each of the k nearest neighbors based on their distance from the sample. Finally, data preprocessing may be necessary before applying KNN, such as normalizing, standardizing or scaling the data.



Some of the KNN's characteristics are as follows:

Non-parametric: KNN is a non-parametric approach since it makes no assumptions about the distribution of the underlying data.

KNN is regarded as a "lazy" learning algorithm because it does not actively learn a model. This indicates that it does not need a lot of training time, but compared to other algorithms, it can need longer time to make predictions.

Sensitive to scale: Before employing KNN, it is essential to scale the data because KNN is sensitive to the scale of the features.

Easy to use: KNN is a straightforward approach that is simple to use, especially for small datasets.

KNNs frequently exhibit high bias and low variance, which indicates that overfitting may be less likely.

When utilizing the KNN algorithm, selecting the value of k is crucial because it can impact the model's performance. When deciding on the value of k, take into account the following:

Smaller values of k will increase the model's sensitivity to data noise, which may cause it to overfit.

Larger values of k will reduce the model's sensitivity to data noise, but they may also provide models with high bias and oversimplification.

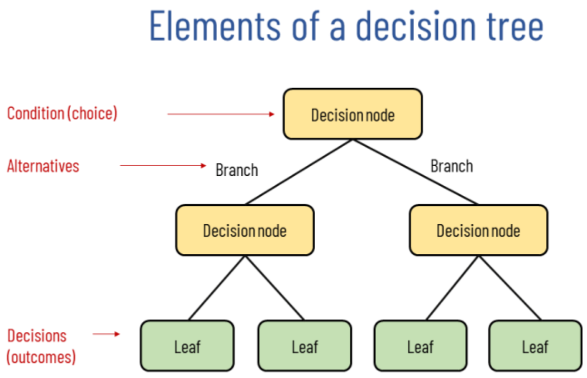
Selecting a number for k that is both small enough to capture the complexity of the data and large enough to smooth out the noise is generally advised.

When determining the value of k, it's necessary to take the dataset's size into account as well. A smaller value of k might be more suited for small datasets, whereas a greater value of k might be better suited for large datasets.

K value in this project is taken as 29 with respect to grid search.

B. *Decision Tree Regression Method.*

A supervised learning algorithm known as decision tree regression is used to forecast a continuous target variable. It functions by building a tree-like model of decisions based on the characteristics of the data, with each leaf node representing the target variable and each interior node indicating a "test" on a characteristic (prediction). Up until the leaves are pure, the tree is constructed by continuously dividing the data depending on the most important attribute (contain data points belonging to only one class). By tracing the branch across the tree that matches to the characteristics of the new data point, the finished tree can then be used to forecast new data points. Decision tree classification is a quick and efficient technique to create predictions, but if the tree is not pruned appropriately, it can be prone to overfitting.



When using decision tree classification, the following factors should be considered:

Decision trees are susceptible to overfitting, especially if they are allowed to grow deeply. By choosing a maximum depth or a minimum number of samples needed to divide a node, you can prune the tree to prevent overfitting.

Any form of data can be used with decision trees, however some properties may be more crucial than others for making predictions. It's crucial to think about which elements are most pertinent to the work at hand and to only include those aspects in the model.

Missing values can be handled using decision trees, albeit they could perform less well if there are a lot of missing values in the dataset. You can try imputing the missing values or employing algorithms that are resilient to missing values to handle missing values.

Decision trees are capable of handling classes that are unevenly distributed, but they may perform less well in cases where the imbalance is severe. You can attempt undersampling the majority class or oversampling the minority class to deal with imbalanced classes, or you can employ methods made expressly for imbalanced data.

C. Gaussian Naive Bayes Method

Gaussian Naive Bayes is a method for classifying data based on the Bayes theorem and the assumption that features are independent of one another. Despite the "naive" assumption, the algorithm often still performs well in practice.

It models the probability of a class using the Gaussian distribution, and estimates the mean and variance of each feature for each class. Then it uses these estimates to determine the likelihood of a new set of features belonging to each class, and the class with the highest likelihood is chosen as the prediction.

It doesn't have any hyperparameters, but some techniques to improve the performance, like Laplace smoothing, which is used to avoid probabilities to be zero.

V. PERFORMANCE EVALUATION

A machine learning system's performance is evaluated by analyzing the system's accuracy, dependability, and efficiency in fulfilling its objectives. It is a crucial phase in the design and implementation of any machine learning system since it makes sure the system is operating as intended and identifies any potential improvement areas.

Evaluation criteria used in this study:

(i) F1 score(macro): The F1 score is a metric that evaluates the performance of a binary classification model by combining precision and recall. Precision measures the proportion of correct positive predictions made by the model, while recall measures the proportion of actual positive instances that were correctly identified. The F1 score is calculated by taking the harmonic mean of precision and recall. A score closer to 1 indicates a better balance between precision and recall, while a score closer to 0 indicates a poor balance. It's particularly useful when dealing with imbalanced datasets or when trying to find a balance between precision and recall.

The F1 score is calculated as:

F1 = 2 * (precision * recall) / (precision + recall)

The macro-F1 score is a variation of the F1 score that is used to evaluate the performance of multi-class classification models. Instead of computing a single F1 score for the whole dataset, it calculates the F1 score for each class individually and then takes the average of these scores. This approach gives equal weight to all classes, regardless of their distribution or size, and is not affected by class imbalance. However, it may not be the best metric to use in cases where the class distribution is highly imbalanced or when misclassifying certain classes has a high cost.

The macro-F1 score is a straightforward and equitable method to evaluate the performance of multi-class classification models. It treats all classes equally, regardless of their size or distribution, and is not affected by class imbalance. However, it may not be the most appropriate metric when the class distribution is highly imbalanced or when misclassifying certain classes has a high cost.

Macro-F1 score is calculated as:

Macro-F1 = (F1_class1 + F1_class2 + ... + F1_classn) / n

Where F1_class1 is the F1 score for the ith class and n is the number of classes.

Method	**F1 Score**
KNN	0.33985
Decision Tree	0.32997
Gaussian Naive Bayes	0.34304

VI. CONCLUSION

Although the KNN and Gaussian Naive Bayes algorithms produced similar results, KNN is slightly better for this dataset.
Data preprocessing and feature engineering is vital for some datasets.
To try as many as possible learning algorithms provides more reliable results.

References

[1] Investigation of the Effects of Kernel Functions in Satellite Image Classification Using Support Vector Machines (Harita Dergisi Temmuz 2010 Sayı 144-Taşkın KAVZOĞLU, İsmail ÇÖLKESEN)

[2] Kernel Function Selection for the Solution of Classification Problems via Support Vector Machines (ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ İİBF DERGİSİ, NİSAN 2014, 9(1), 175- 198 - Sevgi AYHAN, Şenol ERDOĞMUŞ