**CLASSIFIERS**

1. RANDOM FOREST CLASSIFIER

A random forest is a machine learning technique for solving classification and regression problems. It makes use of ensemble learning, which is a technique for solving complicated problems by combining several classifiers.The random forest is a classification algorithm consisting of many decisions trees. **It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees** whose prediction by committee is more accurate than that of any individual tree.

It determines the outcome based on the decision trees' predictions. It forecasts by averaging the output of various trees. The precision of the result improves as the number of trees grows. A random forest method overcomes the drawbacks of a decision tree algorithm. It reduces dataset overfitting and improves precision. It generates forecasts without requiring a large number of package setups (like scikit-learn).

 Built up areas, Natural forest, Natural water bodies, artificial water bodies, Transportation, Bare areas, Crop land, Grasslands and shrubs were chosen as the eight classes to train each classifier by displaying the raw image on the computer screen and them randomly from the image. Each class was given two training sets, each of which had more than 40 points. Training samples and reflectance data for bands 1 through 7 were used to train the classifier. This resulted in a total of above 350 trees.

REFERENCES

Liaw, A., Wiener, M. (2002): Classification and Regression by random Forest. Forest, 23.

Pal., M. (2003): Random Forest Classifier for Remote Sensing Classification. In: International Journal of Remote Sensing, Volume 26, Issue 1. Pages 217-222.

https://blogs.fu-berlin.de/reseda/random-forest/

1. SUPPORT VECTOR MACHINE CLASSIFIER

The Support Vector Machine (SVM) is a classification and regression prediction tool that employs machine learning theory to maximize predicted accuracy while avoiding data overfitting. Support Vector machines are systems that use the hypothesis space of linear functions in a high-dimensional feature space and are trained with an optimization theory learning algorithm that incorporates a learning bias derived from statistical learning theory. Support Vectors Classifier tries to find the best hyperplane to separate the different classes by maximizing the distance between sample points and the hyperplane.

SVM algorithms use a set of mathematical functions that are defined as the kernel which selects the type of hyperplane to be used to separate the data and takes data as input and transform it into the required form, for example from 1D to 2D. In this case Radial basis function was used because it has localized and finite response along the entire x-axis and returns the inner product between two points in a suitable feature space. The gamma parameter specifies how far a single training example's influence extends, with low values indicating "far" and large values indicating "near." The inverse of the radius of influence of samples chosen as support vectors by the model can be seen in the gamma parameters. Technically speaking, large gamma leads to high bias and low variance models and smaller gamma leads to low bias and high variance models. The gamma value selected in this case is 1 because it exactly fits the training dataset and if the gamma is increased it will lead to overfitting as the classifier tries to perfectly fit the training data.

REFERENCES

Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks" (PDF). Machine Learning. 20 (3): 273–297. CiteSeerX 10.1.1.15.9362.

Cortes, Corinna; Vladimir Vapnik (1995). "Support-Vector Networks". Machine Learning. 20 (3): 273–297.

Ben-Hur, Asa; Horn, David; Siegelmann, Hava; Vapnik, Vladimir N. ""Support vector clustering" (2001);" Journal of Machine Learning Research. 2: 125–137.

Joachims, Thorsten; "Transductive Inference for Text Classification using Support Vector Machines", Proceedings of the 1999 International Conference on Machine Learning (ICML 1999), pp. 200–209.

Drucker, Harris; Burges, Christ. C.; Kaufman, Linda; Smola, Alexander J.; and Vapnik, Vladimir N. (1997); "Support Vector Regression Machines", in Advances in Neural Information Processing Systems 9, NIPS 1996, 155–161, MIT Press.

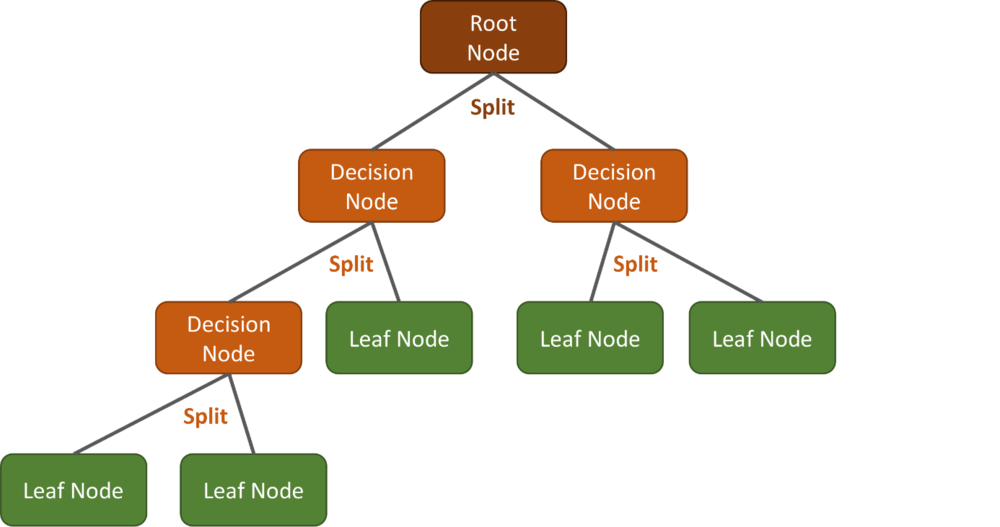
Suykens, Johan A. K.; Vandewalle, Joos P. L.; "Least squares support vector machine classifiers", Neural Processing Letters, vol. 9, no. 3, Jun. 1999, pp. 293–300.

1. CLASSIFICATION AND REGRESSION TREE CLASSIFIER

A Classification and Regression Tree (CART) is a machine learning predictive technique. It is a decision tree in which each fork is separated into a predictor variable and each node at the end includes a prediction for the target variable, and illustrates how the values of a target variable can be anticipated based on other values. When splitting data, the CART algorithm seeks out the independent variable that produces the most homogeneous group. This is determined for a classification issue with a categorical response variable by computing the information obtained based on the entropy arising from the split. Homogeneity is quantified in numeric responses using statistics like standard deviation and variance.

A decision tree is a relatively simple structure that consists of three types of elements: one Root Node, which is the starting point for all training samples, multiple Decision Nodes, where we split our data using simple if-else decision rules, and multiple Terminal Nodes, where we assign classes for our classification purpose.

This classifier process can be summarized in the diagram below;



REFERENCES

Letham, Ben; Rudin, Cynthia; McCormick, Tyler; Madigan, David (2015). "Interpretable Classifiers Using Rules and Bayesian Analysis: Building a Better Stroke Prediction Model". Annals of Applied Statistics. 9 (3): 1350–1371.

Wang, Fulton; Rudin, Cynthia (2015). "Falling Rule Lists" (PDF). Journal of Machine Learning Research.

Barros, R. C.; Cerri, R.; Jaskowiak, P. A.; Carvalho, A. C. P. L. F. (2011). "A bottom-up oblique decision tree induction algorithm". Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA 2011). pp. 450–456.