CIS 550: Advance Machine Learning

Hyperspectral Image Classification Report

Under Guidance of Dr. Ashokkumar Patel

Contributed by

Group 15

16) Dhyey Doshi

20) Keerthi Reddy

30) Deena Kurapati

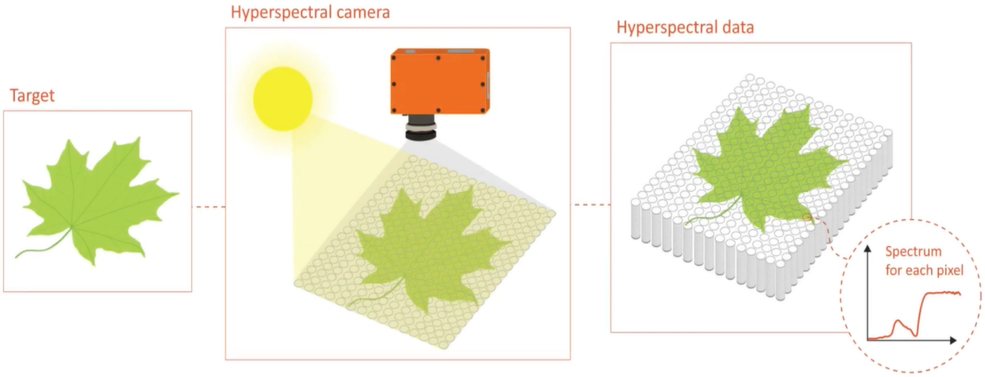
43) Rutu Patel

Abstract

Hyperspectral remote sensing technology is one of the advanced technologies for detailed land cover feature extraction. Hyperspectral datasets contain a large number of contiguous spectral bands with a narrow spectral bandwidth which enables the identification of peculiar absorption features for distinguishing different types of soils. The potential of Airborne Visible Infrared Imaging Spectrometer-Next Generation (AVIRIS-NG) data was tested for distinguishing different types of soils. The AVIRIS-NG data captured in 372 narrow contiguous bands (346–2505 nm) with a spectral sampling of 5 nm bandwidth and a 4m ground pixel size was used in this study. Optimal spectral bands from the reflectance data were selected on the basis of the different characteristics of various soils.

Hyperspectral Image

Hyperspectral image is a 3D data cube, which contains two-dimensional spatial information (image feature) and one-dimensional spectral information (spectral-bands). The spectral bands occupy very fine wavelengths, while the image features such as Land cover features and shape features disclose the disparity and association among adjacent pixels from different directions at a confident wavelength.



Rapid and accurate detection of the characteristics and changes of the Earth's surface is a necessity for better understanding of the relationships and interactions between natural and human phenomena, which provide a foundation for the management and judicious use of natural resources. Hyperspectral remote sensing imagery enables identification and classification of different land cover features including different types of soils, mainly due to the information contained in a large number of contiguous spectral bands with a narrow spectral bandwidth. The information present in the airborne hyperspectral dataset is coupled with improved spatial resolution which enables accurate identification of peculiar absorption features for distinguishing different land cover features. Airborne hyperspectral imagery not only gives abundance of spectral responses but also provides good ground pixel size that increases the classification accuracy.

To extract meaningful information from hyperspectral imagery, selection of optimum bands from the data is crucial for certain classes and also essential for better classification for that particular class. Band selection is one of the best ways to reduce storage and processing time for hyperspectral remote sensing data.

Dataset Description

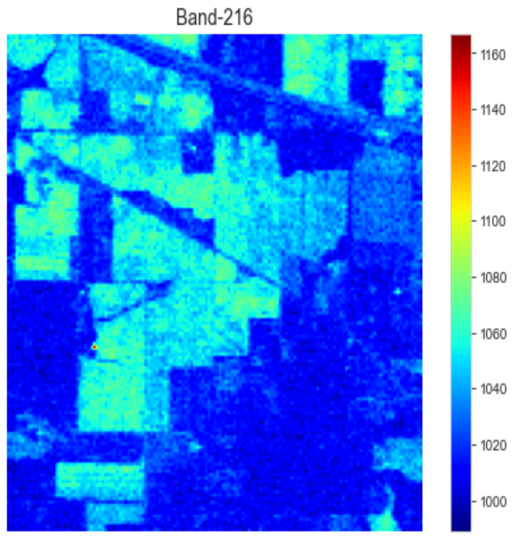
Data used was gathered by AVIRIS (Airborne Visible Infrared Imaging Spectrometer) sensor over the Indian Pines test site in North Western Indiana.

Data Source: http://lesun.weebly.com/hyperspectral-data-set.html

- Wavelength: 0.4 - 2.5 * 10^(-6) metres
- Height: 145 pixels
- Width: 145 pixels
- Spectral band: 220
- Number of Classes: 16

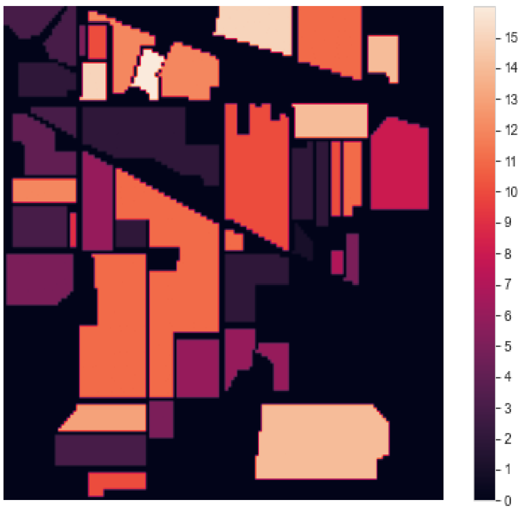
Target Classes:

1. background
2. alfalfa
3. corn-notill
4. corn-min
5. corn
6. grass/pasture
7. grass/trees
8. grass/pasture-mowed
9. hay-windrowed
10. oats
11. soybeans-notill
12. soybeans-min
13. soybean-clean
14. wheat
15. woods
16. bldg-grass-tree-drives



--(1)

fig(1) - Visualizing Random Sample band



--(2)

fig(2) – Ground Truth of the dataset

Methodology

The supervised classification takes the advantage of rich spectral information and has explored many applications including urban development, the monitoring of land changes, target detection, and resource management. In supervised classification only labeled data is used to train the classifier. A large number of supervised classification methods have been discussed in the literature, some of the prominent methods are nearest neighbor classifier, decision trees, random forest, support vector machines (SVMs), etc.

Before model training and testing, data has to be preprocessed. For pre-processing, outliers have to be detected and dealt with. Interquartile Range(IQR) is used to detect the outliers in the dataset. The interquartile range is a measure of where the "middle fifty" is in a data set. Where a range is a measure of where the beginning and end are in a set, an interquartile range is a measure of where the bulk of the values lie.

Interquartile range = Upper Quartile – Lower Quartile

$Q_2 = Q_3 - Q_1$

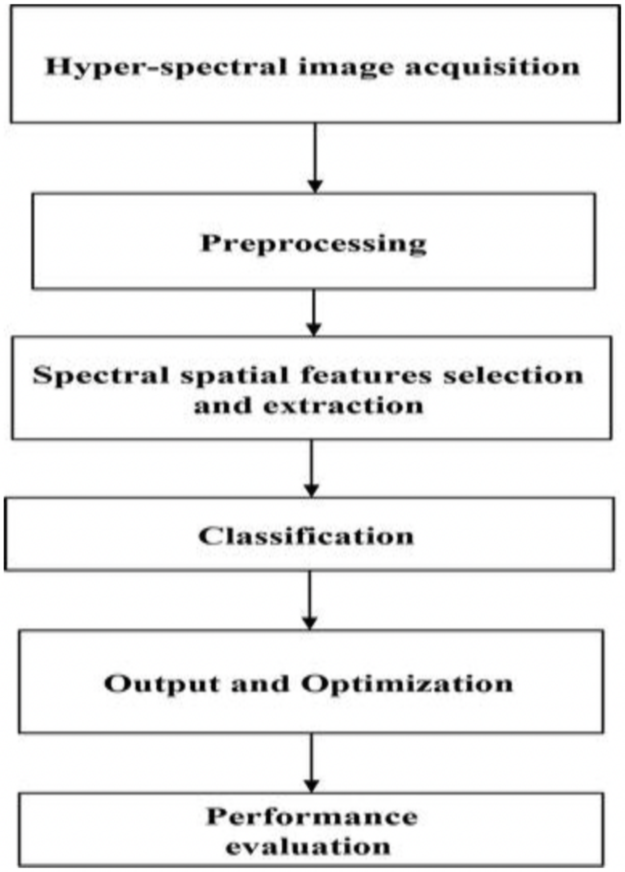
where,

IQR = Interquartile range (IQR = Q_2)

$Q_1 = (1/4)[(n + 1)]^{th}$ term)

$Q_3 = (3/4)[(n + 1)]^{th}$ term)

n = number of data points


Hyper-spectral image acquisition
Preprocessing
Spectral spatial features selection and extraction
Classification
Output and Optimization
Performance evaluation

1) XGBoost:

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

Mathematically, we can write our model in the form

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \epsilon \mathcal{F}$$

where, K is the number of trees, f is the functional space of F, F is the set of possible CARTs. The objective function for the above model is given by

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where, first term is the loss function and the second is the regularization parameter.

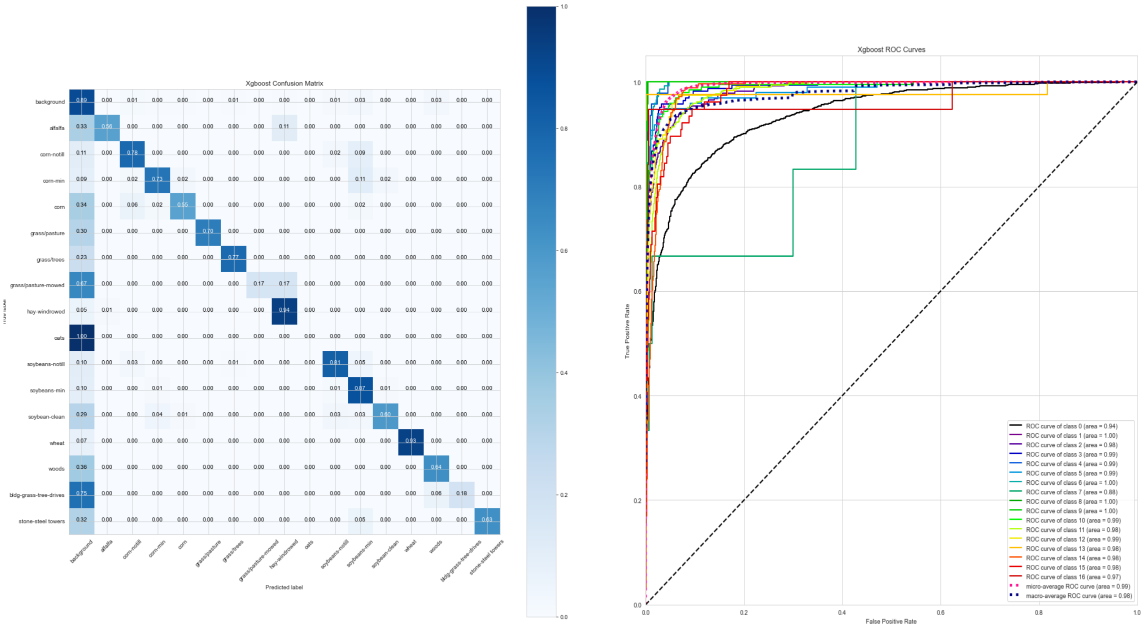
Algorithm Features:

- **Sparse Aware** implementation with automatic handling of missing data values.
- **Block Structure** to support the parallelization of tree construction.
- **Continued Training** so that you can further boost an already fitted model on new data.

Results:

Accuracy: 84.20%

Confusion Matrix and ROC Curve:



2) K-Nearest Neighbor Classifier:

The k-nearest neighbor algorithm is a supervised learning classifier that is non-parametric and uses proximity to classify or predict how a particular data point will be grouped. It can be applied to classification or regression problems, but it is most frequently used as a classification algorithm because it relies on the idea that similar points can be found close to one another.

In order to determine which data points are closest to a given query point, the distance between the query point and the other data points will need to be calculated.

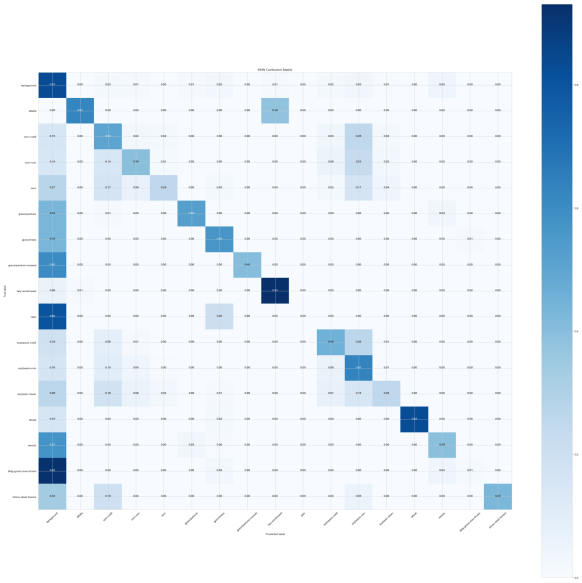
Euclidean distance formula:

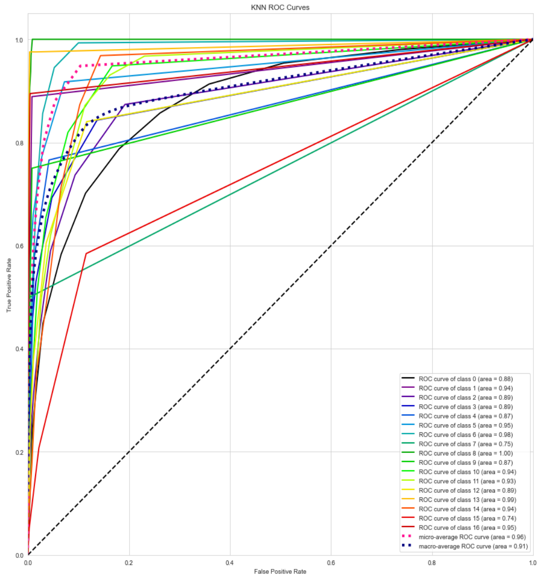
$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

<u>Results</u>:

Accuracy: 67.01%

Confusion Matrix and ROC Curve:





3) Random Forest Classifier:

Random Forest combines the simplicity of Decision Trees with flexibility with vast improvement in accuracy. There are three main hyperparameters for random forest algorithms that must be set prior to training. These include node size, the number of trees, and the number of features sampled. The random forest classifier can then be used to address classification or regression issues.

<u>Gini Index</u>:

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

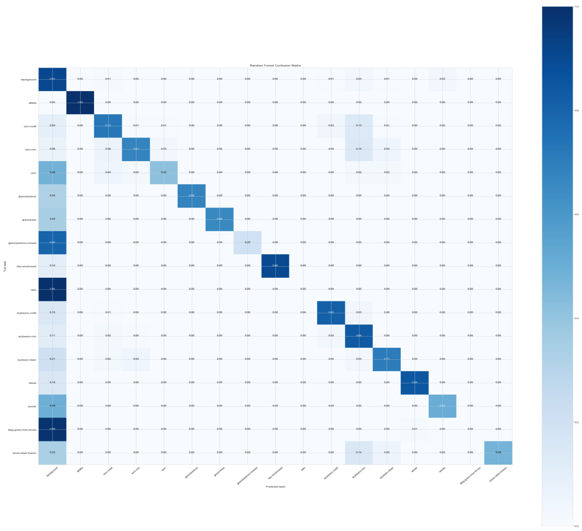
<u>Entropy</u>:

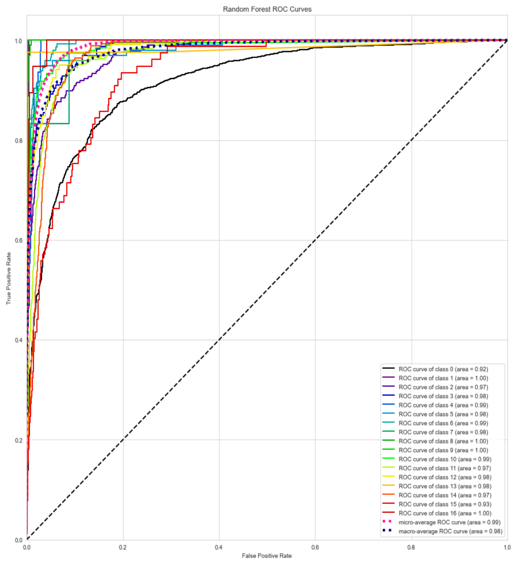
$$Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

Results:

Accuracy: 80.38%

Confusion Matrix and ROC Curve:





4) Multilayer Perceptron:

A multilayer perceptron (MLP) is a fully connected class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to mean any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons.

If a multilayer perceptron has a linear active function in all neurons, that is, a linear function that maps the weighted inputs to the output of each neuron, then linear algebra shows that any number of layers can be reduced to a two-layer input-output model.

The computations taking place at every neuron in the output and hidden layer are as follows,

$$\mathrm{o(x) = G(b(2) + W(2)h(x))} \tag{1}$$

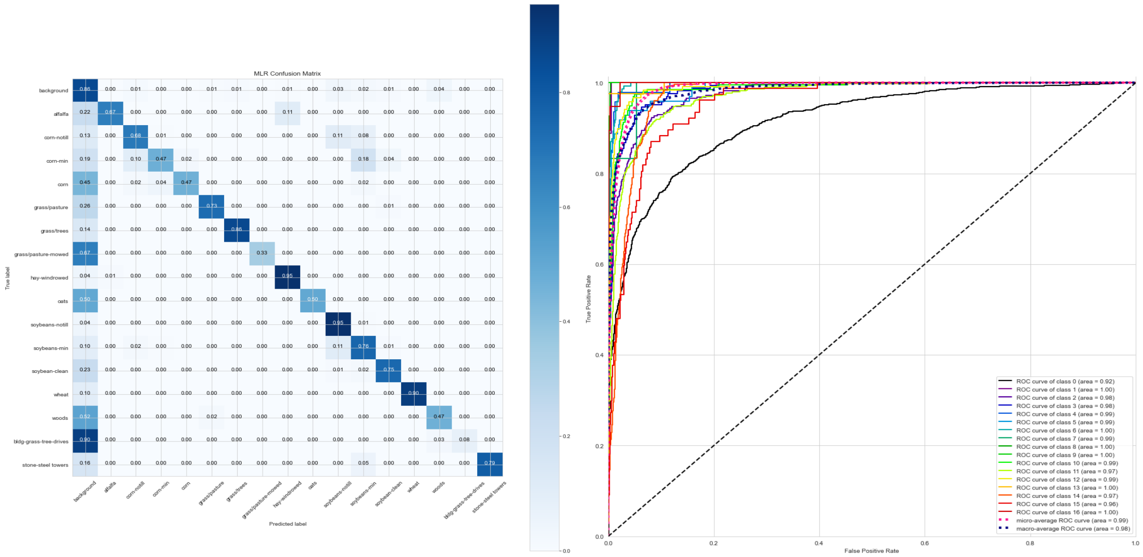
$$\mathrm{h(x) = \Phi(x) = s(b(1) + W(1)x)} \tag{2}$$

with bias vectors b(1), b(2); weight matrices W(1), W(2) and activation functions G and s. The set of parameters to learn is the set θ = {W(1), b(1), W(2), b(2)}. Typical choices for s include tanh function or the logistic sigmoid function

Results:

Accuracy: 76.12%

Confusion Matrix and ROC Curve:



5) Gradient Boosting:

Gradient Boosting gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

Gradient Boosting Algorithm can be used in regression as well as classification problems. In regression problems, the cost function is MSE whereas, in classification problems, the cost function is Log-Loss.

Consider a GB algorithm with M stages, at each stage m suppose some imperfect model Fm

$$F_{m+1}(x_i) = F_m(x_i) + h_m(x_i) = y_i$$

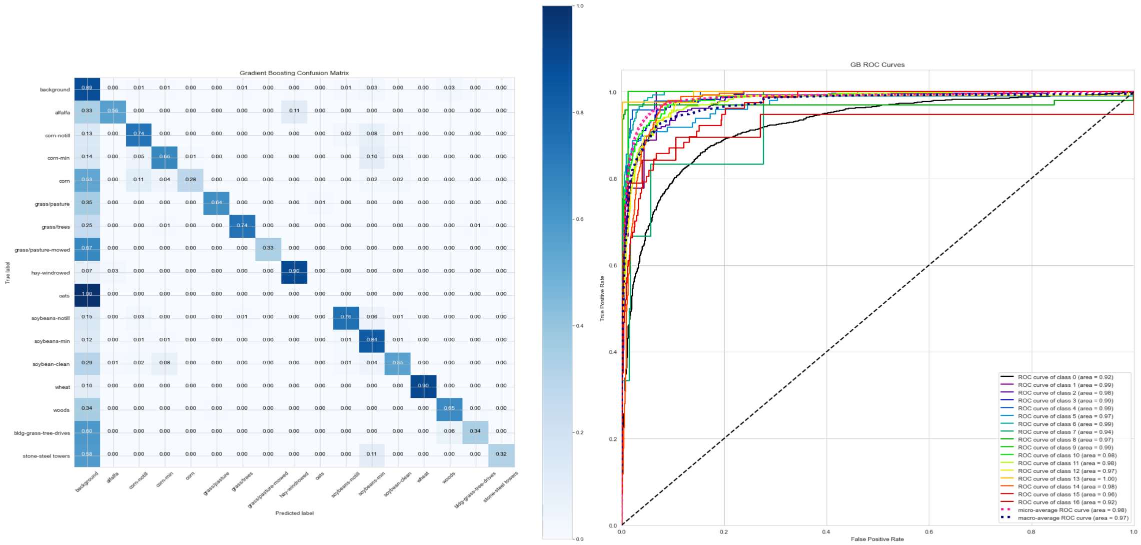
$$h_m(x_i) = y_i - F_m(x_i)$$

$$-\frac{\partial L_{\mathrm{MSE}}}{\partial F(x_i)} = \frac{2}{n}(y_i - F(x_i)) = \frac{2}{n}h_m(x_i)$$

Results:

Accuracy: 81.28%

Confusion Matrix and ROC Curve:



6) RBF SVM:

SVM algorithm attempts to find a hyperplane that separates these two classes with the highest possible margin. If classes are fully linearly separable, a hard-margin can be used. Otherwise, it requires a soft-margin.

A kernel is a function that takes the original non-linear problem and transforms it into a linear one within the higher-dimensional space.

We can add a third dimension by squaring the existing dimensions. Using this three-dimensional space with x, y, and z coordinates, we can now draw a hyperplane (flat 2D surface) to separate red and black points. Hence, the SVM classification algorithm can now be used.

RBF is the default kernel used within the sklearn's SVM classification algorithm and can be described with the following formula:

$$K(x, x') = e^{-\gamma||x-x'||^2}$$

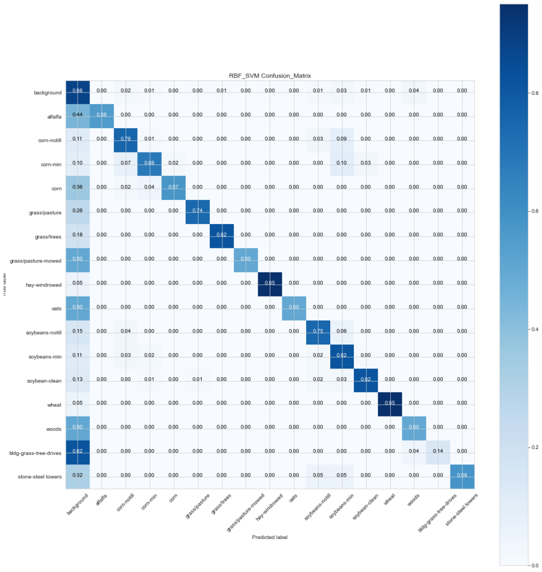
where gamma can be set manually and has to be >0. The default value for gamma in sklearn's SVM classification algorithm is:

$$\gamma = \frac{1}{n\, features * \sigma^2}$$

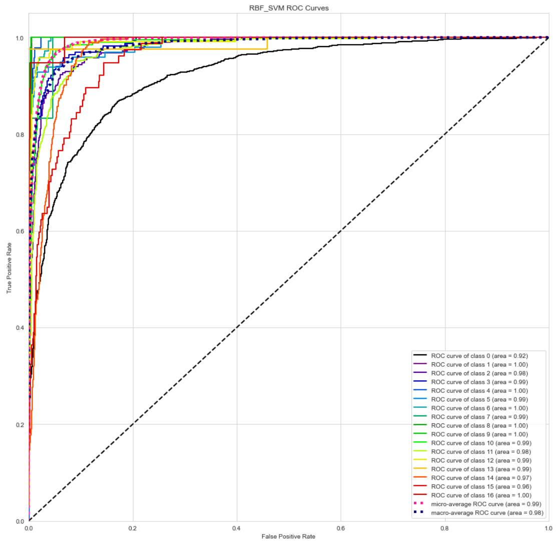
<u>Results</u>:

Accuracy: 79.76%

Confusion Matrix and ROC Curve:


RBF_SVM Confusion_Matrix

	background	alfalfa	corn-notill	corn-min	corn	grass/pasture	grass/trees	grass/pasture-mowed	hay-windrowed	oats	soybeans-notill	soybeans-min	soybean-clean	wheat	woods	bldg-grass-tree-drives	stone-steel towers
background	0.88	0.00	0.02	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.03	0.01	0.00	0.04	0.00	0.00
alfalfa	0.44	0.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
corn-notill	0.11	0.00	0.76	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.09	0.00	0.00	0.00	0.00	0.00
corn-min	0.10	0.00	0.07	0.68	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.03	0.00	0.00	0.00	0.00
corn	0.36	0.00	0.02	0.04	0.57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
grass/pasture	0.26	0.00	0.00	0.00	0.00	0.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
grass/trees	0.18	0.00	0.00	0.00	0.00	0.00	0.82	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
grass/pasture-mowed	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
hay-windrowed	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
oats	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00
soybeans-notill	0.15	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.06	0.00	0.00	0.00	0.00	0.00
soybeans-min	0.11	0.00	0.03	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.82	0.00	0.00	0.00	0.00	0.00
soybean-clean	0.13	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.03	0.82	0.00	0.00	0.00	0.00
wheat	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.00	0.00
woods	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00
bldg-grass-tree-drives	0.82	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.14	0.00
stone-steel towers	0.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.05	0.00	0.00	0.00	0.00	0.58



7) Poly SVM:

In machine learning, the polynomial kernel is a kernel function commonly used with support vector machines (SVMs) and other kernelized models, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.

Intuitively, the polynomial kernel looks not only at the given features of input samples to determine their similarity, but also combinations of these. In the context of regression analysis, such combinations are known as interaction features. The (implicit) feature space of a polynomial kernel is equivalent to that of polynomial regression, but without the combinatorial blowup in the number of parameters to be learned. When the input features are binary-valued (booleans), then the features correspond to logical conjunctions of input features.

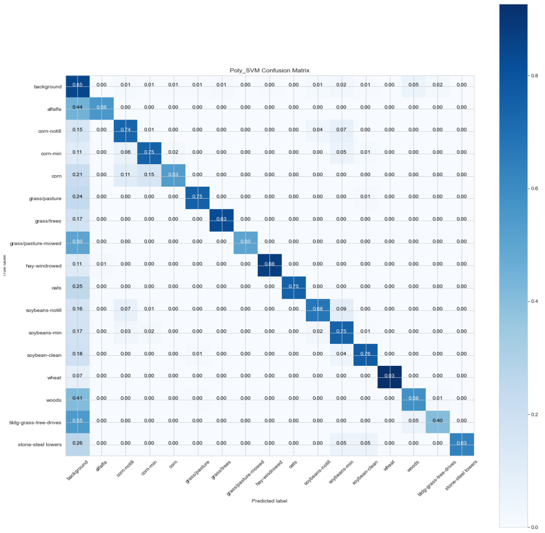
For a degree d polynomial, the polynomial kernel is defined as:

$$K(x, y) = (x^\mathsf{T} y + c)^d$$

Results:

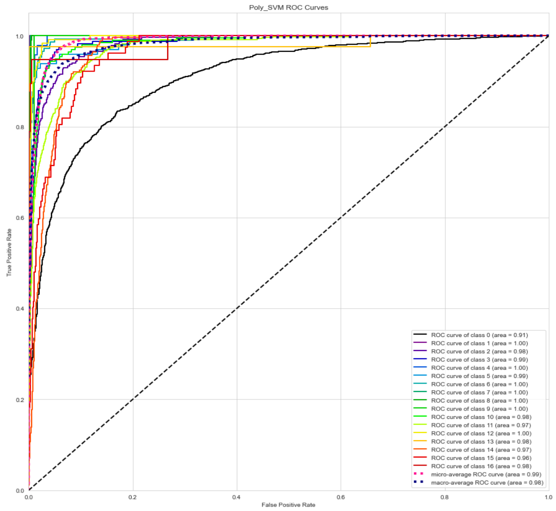
Accuracy: 78.04%

Confusion Matrix and ROC Curve:



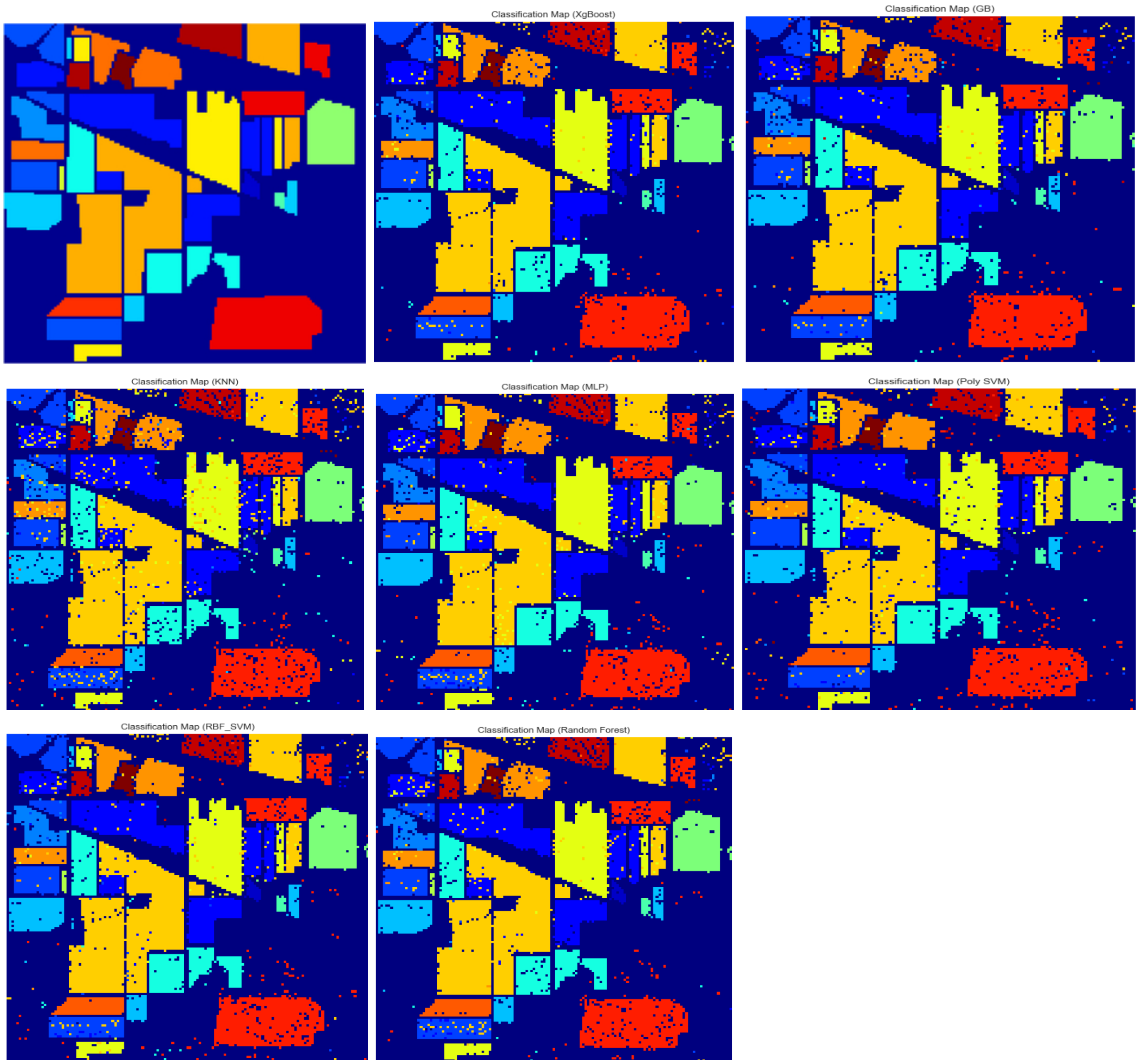
Poly_SVM Confusion Matrix

	background	alfalfa	corn-notill	corn-min	corn	grass/pasture	grass/trees	grass/pasture-mowed	hay-windrowed	oats	soybeans-notill	soybeans-min	soybean-clean	wheat	woods	bldg-grass-tree-drives	stone-steel towers
background	0.85	0.00	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.05	0.02	0.00
alfalfa	0.44	0.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
corn-notill	0.15	0.00	0.74	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.07	0.00	0.00	0.00	0.00	0.00
corn-min	0.11	0.00	0.06	0.75	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.01	0.00	0.00	0.00	0.00
corn	0.21	0.00	0.11	0.15	0.53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
grass/pasture	0.24	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
grass/trees	0.17	0.00	0.00	0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
grass/pasture-mowed	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
hay-windrowed	0.11	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
oats	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00
soybeans-notill	0.16	0.00	0.07	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.68	0.09	0.00	0.00	0.00	0.00	0.00
soybeans-min	0.17	0.00	0.03	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.75	0.01	0.00	0.00	0.00	0.00
soybean-clean	0.18	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.04	0.76	0.00	0.00	0.00	0.00
wheat	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.93	0.00	0.00	0.00
woods	0.41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.58	0.01	0.00
bldg-grass-tree-drives	0.55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.40	0.00
stone-steel towers	0.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.05	0.00	0.00	0.00	0.63



Results

Classification Methods	**Accuracy**
K-Nearest Neighbor	67.01%
Random Forest	83.38%
Gradient Boosting	81.28%
Multilayer Perceptron	76.12%
RBF Support Vector Machine	79.76%
Poly Support Vector Machine	78.04%
XGBoost	84.2%


Classification Map (XgBoost)
Classification Map (GB)
Classification Map (KNN)
Classification Map (MLP)
Classification Map (Poly SVM)
Classification Map (RBF_SVM)
Classification Map (Random Forest)

Conclusion

Hyperspectral Image Classification continues to draw interest in the blooming field of hyperspectral remote sensing due to its accrued benefits in many applications.

This study was aimed at accurate identification and classification of various types of classes using airborne hyperspectral remote sensing data(AVIRIS-NG) employing various approaches. Through experiments, several important parameters that affect the accuracy of the model are discussed and tuned on the dataset of Indian Pines. The study revealed the improved performance of the XgBoost classifier in comparison to other classifiers for hyperspectral data classification.

References

https://nbviewer.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/neural_network/multilayer_perceptron_demo.ipynb

https://scikit-learn.org/stable/modules/ensemble.html#gradient-tree-boosting

https://youtu.be/ayp7hP0Xr8Q

https://www.intechopen.com/chapters/70188

https://a-a-r-s.org/proceeding/ACRS2017/ID_753_1652/1605.pdf