Exploration of Various Machine Learning Techniques for Forest Cover Type Prediction

Chandra Prakash, Aditi Mittal and Vipasha Dhiman

***Abstract*— Machine learning classifiers like naive bayes classifier, k-nearest neighbors classifier, and random forest classifier are being widely used for prediction purpose. This paper compared the accuracy of naive bayes, random forest and neural Network for classification purpose along with comparing results of applying dimension reduction using principal component analysis and statistical data.**

***Index Terms*— Machine Learning, Naive Bayes, k-Nearest Neighbors, Random Forest, classifier.**

I. INTRODUCTION

Given elevation, soil, and sunlight data, the type of tree that would be in the patch of forest can be predicted. Understanding of forest lands is required by natural resources management to support their decision-making processes for developing strategies. The Forest Inventory and Analysis(FIA) surveThishe United States ifrests. The goal of FIA is to provide timely information for the development of policies for protection, management, and effective utilization of forest resources. o s a classification problem and variouys ts machine learning algorithms like Naive Bayes, k-Nearest Neighbors, Random Forest, support vector machines [1] are being widely used. Moreover, intensifying interest in forests and the environment have induced major changes in forest monitoring systems in the last few years. Moreover, intensifying interest in forests and the environment have induced major changes in forest monitoring systems in the last few years. This project attempts to predict the dominant type of tree in sections of wooded area using Naive Bayes, Neural Network and Random Forest with the preprocessing of data to reduce the number of features.
The dataset used includes four wilderness areas located in the Roosevelt National Forest of northern Colorado. The advantage of using this dataset if that these areas represent forests with minimal human caused disturbances. Thus, the algorithms trained will be best suited to predict forest cover types as they will give a result of ecological processes rather than forest management practices. This dataset has also been chosen because it presents a complex problem to solve as it contains 15,120 instances and 54 input variables. This means that network could be trained well and thus will be tested well.
The dominant forest cover type associated to each sample is one of the seven specific forest cover types which are spruce-fir, lodgepole pine, ponderosa pine, cottonwood/willow, aspen, douglas-fir and krummholz.

II. RELATED WORKS

This section describes about the related works done in the similar domain of machine learning.
The forest cover type data has been widely used to compare the accuracy of various multiple class classification problems. Daniel Gabriel [3] focused on decision forest based models, decision tree based models and neural networks to predict the forest cover type. Kaggle dataset was used in this work. Multi-class decision forest and One-vs-all decision forest in decision forest category were implemented whereas in decision tree based models boosted trees were used. Feature engineering including attributes extraction, linear combination and euclidean distance was carried out for the preprocessing of the data. Among the above three algorithms, the decision tree and forest resulted in the highest average accuracy of 80%.

Blackard and Dean [4] used different numbers of hidden nodes(6-300 nodes) for the 54 variables artificial neural network models. The mean accuracy was found out to be 70.52% and the optimal number of neurons was found out to be 120. The results were also compared with the linear discriminant analysis and quadratic discriminant analysis on the data.

Kishore and Narayan [5] applied different models like naive bayes, k- nearest neighbours and random forest to predict the forest cover type. Among all the techniques, k-nearest neighbour performed best with accyacy of 87%.

Lowd and Domingos[6] used naive bayes based models to predict the forest cover type. Naive Bayes, irrespective of its simplicity, showed the results that are comparable to Bayesian network, making it more attractive than Bayesian networks for prediction.

Crain and Davis[7] used forest cover type data to predict the forest cover type using multi class support vector machine and k-means clustering. The data was preprocessed using principal component analysis. SVM with 10 dimensional PCA-transformed data was ran which captured 99.997% of the variance and performed only minimally worse than using the entire data set with all 54 dimensions. Testing accuracy was found to be 78.24% using SVM.

Further, the section III of this paper deals with the processing and models used to predict the forest cover type. After this, in section IV, the results were obtained and comparison of different techniques like Neural Network, Random Forest and Naive Bayes are done. At the end, in section V, some conclusions were derived after observing the results.

III. METHODOLOGY

This section deals with the processing and models used to predict the forest cover type.

A. Data and Features

The data was collected from UCI machine learning repository[2] . The data set consisted of 15,120 samples of 30m x 30m patches of forest. Each data sample includes 54 attributes: elevation (in meters), slope, aspect (compass direction of slope face), vertical distance from water, horizontal distance from water, sunlight intensity at 9am, 12pm, and 3pm, 4 binary wilderness areas, and 40 binary soil types. Each sample was classified into one of seven forest cover types: Spruce/Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas Fir, or Krummholz. There are 2,160 samples for each of the seven cover types of forest.
Table I lists the attributes and the data type of various attributes present in the data-set.

TABLE I
NAME AND DATA TYPES OF ATTRIBUTES

Name	Data Type
Elevation	quantitative
Aspect	quantitative
Slope	quantitative
Horizontal_Distance_To_Hydrology	quantitative
Vertical_Distance_To_Hydrology	quantitative
Horizontal_Distance_To_Roadways	quantitative
Hillshade_9am	quantitative
Hillshade_Noon	quantitative
Hillshade_3pm	quantitative
Horizontal_Distance_To_Fire_Points	quantitative
Wilderness_Area (4 binary columns)	qualitative
Soil_Type (40 binary columns)	qualitative
Cover_Type (7 types)	integer

B. Data Preprocessing

Data preprocessing is a very critical step in any machine learning process as the accuracy of models is directly dependent on how the data is prepared for the input to the classifier or other machine learning models.
There are 54 dimensions of data and reducing them is a necessary step for improve the over accuracy and the computational speed of the classifiers. If the scales of features are not same, re-scaling and standardization of data may be necessary for some algorithms. This work applied and compared two techniques for data reduction:

- Using statistical measures
- Principal Component Analysis

1) Using statistical measures: In this method, statistical measures like standard deviation, mean, variance, minimum value, maximum value and count are calculated. In case of high number of dimensions, variables are dropped which have low variance compared to others because these variables will not explain the variation in target variable. If the count of variables is less than other variables i.e. there are some missing values, reason should be identified and then missing values need to be computed or variables are dropped using the appropriate methods. If the value of variable is constant for every record, that variable should be dropped as it will not be contributing to the model.

2) Principal Component Analysis: Principal component analysis (PCA) is a dimension reduction or data compression method that uses an orthogonal transformation to convert a dataset of possibly correlated features into a set of values of linearly uncorrelated variables called principal components.[8] PCA reduces attribute space from a larger number of variables to a smaller number of dimensions. It determines the minimum number of factors that will have the maximum variance from the data in a specific multivariate analysis. It is started by normalizing the data by subtracting the mean from each data point. It is important to normalize as original variables can be on the different scale and can contribute significantly towards variance. Then the covariance matrix for the data is calculated using the given formula which would measure how two variables changes together.

$$cov_{(x,y)} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

Eigen values and Eigen vectors of the above matrix are calculated. This helps in finding underlying patterns in the data.

$$|A - \lambda I| = 0$$

where A is the covariance matrix, I is the identity matrix and λ gives the eigen values

$$(A - \lambda I)V = 0$$

where V is the eigen vector for a particular eigen value of same number of rows as that of the covariance matrix and number of columns as 1. The eigen values are arranged in descending order and top k(no. of required components) eigen values and their associated eigen values are selected. Then the complete dataset is multiplied with the eigen vectors to get k principal components.

C. Models

1) Naive Bayes: Naive Bayes is a supervised machine learning technique[6]. It works on the famous Bayes Theorem and uses the mathematical concepts of probability like Apriori, posteriori and conditional probability. Despite of its simplicity, It works well for some classification problems and particularly those which are with high dimensional inputs. Following is the formula showing the Bayes Theorem,

$$P(\ A|B) = \frac{P(\ A)\ P(\ B|A)}{P(\ A)\ P(\ B|A) + P(A')P(B|A')}$$

According to Bayes Theorem, the probability of the presence of A given B can be calculated as, the probability of A times the probability of B given A up on the total probability of B in A and A complement.

In Naive Bayes,the probability of a given set of inputs present in each class was calculated and then they were compared to the probabilities of every class to find most likely class for that set of inputs.

2) Neural Network: Artificial neural network(ANN) is a computing network that works in similar way as the neuron in human biological brain.[9] It is composed of a number of highly interconnected processing elements called neurons working together to solve a specific problem. They can be used to extract patterns that are difficult to be noticed by humans or other techniques. In ANN, the edges at a connection between artificial neurons have weights that is a real number, and the output of each neuron is computed using non-linear function of the sum of its inputs. Weight are adjusted as learning proceeds. The weights increases or decreases the strength of the signal at a connection. The learning process takes the inputs and the desired outputs and updates its weights accordingly, so the calculated output gets close to the desired output. The prediction process takes input and generate, using the internal state, the most likely output according to its past experience. Backpropagation [10] is used for supervised learning of neural network using gradient descent. The method calculates the gradient of the error function with respect to the weights. An error is computed at the output and transferred backwards through every layer. Learning rate is the parameter that guides your stochastic gradient descent i.e. how the weights changes in every iteration. For neural network algorithm, the dataset needs to be normalized.

TABLE II
CONFIGURATION OF VARIOUS PARAMETERS IN NEURAL NETWORK

Parameter	**With PCA**	**Without PCA**
Number of folds	10	10
Learning rate	0.1	0.01
Number of epochs	20	25
Number of hidden neurons	15	10
Number of output units	8	8
Number of inputs	10	12
Activation Function	sigmoid	sigmoid

3) Random Forest: Random Forest is a supervised learning algorithm and is easy to use It can produce a good result even without hyper-parameter tuning and can be used for both classification and regression problem.[11][12] Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. It adds additional randomness to the model while growing the trees - instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. Thus, the steps involved in random forest can be summarised as follows:

a) Randomly select k features from total m features
b) Among k features, calculate node d using the best split point.
c) Split the node into daughter nodes.
d) Repeat 1 to 3 steps until all k features are used.
e) Build forest by repeating steps 1 to 4 for n number times to create n number of trees.

TABLE III
CONFIGURATION OF VARIOUS PARAMETERS IN NEURAL NETWORK

Parameter	**Value**
Number of folds	5
Number of trees	1,5,10
Impurity measure	Gini Index
Number of features	square root of number of columns-1

IV. RESULTS

This section shows the results obtained after the implementation of algorithms like Neural Network, Random Forest and Naive Bayes and shows the comparison of these techniques.

A. Using statistical measures and general preprocessing

Standard deviation of all the columns was calculated. Standard deviation of Soil_Type7 and Soil_Type15 was found as 0 and dropped. Soil_Type8 and Soil_Type25 had least number of 1 as compared to 0. Thus these two column were dropped. Soil Types and Wilderness Areas columns were one hot encoded, and combined into one column each. Total number of features got narrowed down to 12.

B. Random Forest

Table IV lists the accuracy of Random forest using sklearn and implementing algorithm from scratch. It also compares accuracy for data after PCA and without PCA.

TABLE IV
ACCURACY FOR RANDOM FOREST

Random Forest						
	Without sklearn			With sklearn		
	1 tree	5 trees	10 trees	1 tree	5 trees	10 trees
Without PCA	77	81.64	82.78	74.93	80.15	81.28
With PCA	68.16	73.73	75.82	69.67	72.33	74.67

C. Neural Network

Table V lists the accuracy of Random forest using sklearn and implementing algorithm from scratch. It also compares accuracy for data after PCA and without PCA.

TABLE V
ACCURACY FOR NEURAL NETWORK

Neural Network		
	Without sklearn	With sklearn
With PCA	73.630	75
Without PCA	50.34	58

D. Naive Bayes

Table VI lists the accuracy of Naive Bayes with different amount of training and testing data.

TABLE VI
ACCURACY FOR NAIVE BAYES

Naive Bayes	
Training Data	Accuracy
25%	70.06
50%	70.23
75%	71.05
80%	70.50
95%	71.825

Table VII lists the accuracy of Naive Bayes with different feature set size.

TABLE VII
ACCURACY FOR NAIVE BAYES

Naive Bayes		
Feature set size	Deleted Feature	Accuracy
12	NONE	71.06
10	Wilderness Area & Soil Type	65.22
7	Hillshade	66.21
5	Slope & Aspect	62.16
4	Horizontal Distance To Hydrology	61.26

V. DISCUSSION

For the prediction of forest cover type, Random Forest and Naive Bayes performed way more better than Neural Networks. Random Forest without principal component analysis gives about 80% accuracy and Naive Bayes gave 71%. The reason can be the implementation of random forest. Random forest takes random subset of features and then build the decision tree using impurity index to select the best split. Both algorithms are more accurate and faster than Neural Network in forest cover type prediction. It is evident from the results that most of the features in determining the forest cover type are important and can affect the accuracy if omitted. For future work, the multivariate support vector machine algorithm can be implemented and checked against Random forest and Naive Bayes.

REFERENCES

[1] Lotte, Fabien, Marco Congedo, Anatole Lcuyer, Fabrice Lamarche, and Bruno Arnaldi. ”A review of classification algorithms for EEG-based braincomputer interfaces.” Journal of neural engineering 4, no. 2 (2007): R1.

[2] Bache, K. Lichman, M. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science (2013).

[3] Gribel, Daniel Lemes. ”Forest cover type prediction.” (2015).

[4] Blackard, Jock A., and Denis J. Dean. ”Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables.” Computers and electronics in agriculture 24, no. 3 (1999): 131-151.

[5] Kishore, Rahul R., Shalvin S. Narayan, Sunil Lal and Mahmood A. Rashid. Comparative Accuracy of Different Classification Algorithms for Forest Cover Type Prediction. 2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE) (2016): 116-123.

[6] Lowd, Daniel, and Pedro Domingos. ”Naive Bayes models for probability estimation.” In Proceedings of the 22nd international conference on Machine learning, pp. 529-536. ACM, 2005

[7] Crain, K., and G. Davis. ”Classifying forest cover type using cartographic features.” Published report (2014).

[8] Wold, Svante, Kim Esbensen, and Paul Geladi. ”Principal component analysis.” Chemometrics and intelligent laboratory systems 2, no. 1-3 (1987): 37-52.

[9] Aleksander, Igor, and Helen Morton. An introduction to neural computing. Vol. 3. London: Chapman & Hall, 1990.

[10] Hecht-Nielsen, Robert. ”Theory of the backpropagation neural network.” In Neural networks for perception, pp. 65-93. 1992..

[11] Rodriguez-Galiano, Victor Francisco, Bardan Ghimire, John Rogan, Mario Chica-Olmo, and Juan Pedro Rigol-Sanchez. ”An assessment of the effectiveness of a random forest classifier for land-cover classification.” ISPRS Journal of Photogrammetry and Remote Sensing 67 (2012): 93-104.

[12] Svetnik, Vladimir, Andy Liaw, Christopher Tong, J. Christopher Culberson, Robert P. Sheridan, and Bradley P. Feuston. ”Random forest: a classification and regression tool for compound classification and QSAR modeling.” Journal of chemical information and computer sciences 43, no. 6 (2003): 1947-1958.