Comparative Accuracy of Different Classification Algorithms for Forest Cover Type Prediction

Rahul R Kishore*, Shalvin S Narayan*, Sunil Lal† and Mahmood A Rashid*
*Computing, Information and Mathematical Sciences, The University of the South Pacific, Fiji
†Engineering & Advanced Technology, Massey University, New Zealand
Email: rahul_kishore@live.com, shalvin.narayan@tfl.com.fj, s.lal@massey.ac.nz, mahmood.rashid@usp.ac.fj

***Abstract*—Machine learning based classifiers used quite often for predicting forest cover types, are the Naïve Bayes classifier, the k-Nearest Neighbors classifier, and the Random forest classifier. This paper is directed towards examining all of these classifiers coupled with feature selection and attribute derivation in order to evaluate which one is best suited for forest cover type classification. Numerous training classifications were performed on each of the classifiers with different sets of features. Amongst the three classifiers evaluated in this work, the Random Forest classifier is exhibiting the best and highest accuracy over others. Feature selection also played a significant role in demonstrating the accuracy levels obtained in each of the classifiers.**

***Keywords*—Feature selection; Classification; Naïve Bayes; K-nearest neighbor, Random forest, Accuracy**

I. Introduction

Forest cover type classification has long been of interest in the United States [1], [2]. The US Forest Service and the US Geological Survey organization survey various forest areas throughout the United States in order to collect information and further analyze forestry data [3]. The collective goal of [1] and [4] is to accurately predict the forest cover for varying forests. Forest cover types are a predominant kind of tree cover which spans forest plots where many active approaches to accurate classification are currently implemented. More precisely, the supervised machine learning classification algorithms employed are: k-nearest neighbors classifier, Naïve Bayes classifier and more recently the random forests classifier. The goal of this paper is to take a critical look at all the three of these machine learning algorithms and evaluate the performance for forest cover type prediction, in terms of the percentage of instances classified correctly in a data set.

The study illustrated in [1] uses one of the most rudimentary machine learning algorithms, namely the k-nearest neighbors (kNN) classifier to predict forest cover types. The overall objective of the study was to use the kNN classifier to generate cover type maps in order to pass on information to the USDA Forest Services Forest Inventory and Analysis (FIA) monitoring systems such that forest planning and management could be better facilitated. Using the kNN classifier, the authors of the study were able to produce a useful map accuracy of 54.59% with the nearest neighbor (k) value of 1. Attempts to increase the accuracy produced using the kNN classifier was facilitated by the incorporation of Feature Selection. Feature Selection is a data preprocessing technique where by, attributes are chosen based on the notion of how much influence they impose on the final outcome of the instance to be classified (i.e. class label). However, results had later shown that feature selection for the data set in question did not achieve favorable accuracies for various k values.

Further to this, the work presented in [3] also made use of the kNN classifier for forest cover type prediction and production of cover type maps. However, a key difference between the previous study and the current study was the fact that Lopez *et al.* [1] had used accuracy measures for evaluation purposes and B. Wilson *et al.* [3] had used the root mean square error (RMSE) for measuring the accuracy of the model. In addition to this, this study had also incorporated Feature Selection. Feature Selection was implemented by means of a modified Fourier-based series transformation which had given favorable results by decreasing the number of variables by twenty-fold [3]. Since this study had not taken the approach of traditional accuracy, the aim of this work was to minimize the RMSE. Conclusively, the B. Wilson *et al.* [3] had achieved a RMSE of 0.97 with the k nearest neighbor as seven.

Secondly, the research outlined in [5] uses another fundamental supervised machine learning algorithm; the Naïve Bayes classifier. The Naïve Bayes classifier is different in essence from the kNN classifier. The kNN classifier is designed on the notion of similarity measures; this is in contrast to the Naïve Bayes classifier which is designed around the idea of probabilities. Hence the Naïve Bayes classifier is a probabilistic classifier. In comparison to the kNN classification approaches discussed earlier, this particular study is not aimed at generating forest cover type maps. The overall aim of this study was to examine the general probability estimation of Bayesian classifiers; particularly focused on the Naïve Bayes method. The study presented in [5] had measured accuracy levels with logarithmic values. When the Naïve Bayes classification algorithm was ran it was shown that it had reached an average accuracy of -16.03 (logarithmic value). A higher logarithmic value would have pointed to the fact that the Naïve Bayes classifier is a good prediction model for forest cover types, but this was not the case as it presented itself to be an inaccurate classifier. This is due to the fact that, the higher the logarithmic value the higher accuracy.

In addition to these classifiers, [6] intimately examines the Random forests classifier. The overall aim of this work was to examine how effectively the Random forests classifier is able to predict overall land cover types. The study data in question was obtained from The Province of Granada [6] where the climate is mild and characterized by hot and dry summers as well as wet and cold winters. Similar to the study presented in [1], [6] had also performed data preprocessing in terms of Feature Selection. Feature Selection in this study was performed by

978-1-5090-5753-5/16 $31.00 © 2016 IEEE
DOI 10.1109/APWC.on.CSE.2016.27

Table I: Wilderness areas.

Number	Wilderness Area
1	Rawah
2	Neota
3	Comanche
4	Cache la Poudre

the Gini index and the out-of-bag (oob) subset. Thus coupling the Gini index with the oob subset, the authors were able to generate an acceptable accuracy classification of 92%.

All of these machine learning classifiers have attempted to accurately classify forest cover types. However, they have aimed to correctly classify forest cover types independently. The work of this paper will critically examine the kNN classifier, Naïve Bayes classifier and Random forests classifier. The respective results obtained from the experiment conducted will dictate which one of these classifiers is suitable candidate for predicting forest cover types.

II. Data

Data for this project was collected from Kaggle[1] Competition called Forest Cover Type Prediction [7] where the Roosevelt National Forest of Northern Colorado is the study area.

The actual forest cover type for a given observation (30×30 meter cell) was determined from US Forest Service (USFS) Region 2, Resource Information System (RIS) data. Independent variables were derived from data obtained from the US Geological Survey (USGS) and the USFS data. Data is in raw form and contains binary (0 or 1) columns for qualitative independent variables (wilderness areas and soil types).

The attributes [8] which were given in the data set was elevation in meters, aspect in degrees, slope in degrees, horizontal distance to hydrology, vertical distance to hydrology, horizontal distance to roadway, hill shade index at 9am for summer solstice, hill shade index at noon for summer solstice, hill shade index at 3pm for summer solstice, horizontal distance to nearest wildfire ignition points, wilderness area designation (had 4 binary columns), soil type designation (had 40 binary columns) and forest cover type designation as the class. Tables I–III show the different wilderness areas, soil types and forest cover types respectively.

This study area included four wilderness areas. These areas represent forests with minimal human caused disturbances, so that existing forest cover types are more a result of ecological processes rather than forest management practices.

The training set contained 15120 observations with both features and the cover type. The test set contained only the features with 565892 observations where the cover type was to be predicted.

III. Methods

A. Data Normalization and Derived Attribute

The dataset contained 4 binary columns for wilderness areas and 40 binary columns for soil types. To normalize the binary to categorical data the respective binary column data was transformed into categorical data. Hence, there was only one column for wilderness area and one column for soil type which contained the respective categorical data.

A derived attribute was also calculated for this study. The derived attribute was the Euclidean distance between the horizontal distance to hydrology and vertical distance to hydrology which was called Distance_To_Hydrology. After calculating the derived attribute, we came up with 2 sets of data where the

[1]Kaggle: your home for data science (https://www.kaggle.com/)

Table II: Soil types.

No.	Soil Type
1	Cathedral family - Rock outcrop complex, extremely stony
2	Vanet - Ratake families complex, very stony
3	Haploborolis - Rock outcrop complex, rubbly
4	Ratake family - Rock outcrop complex, rubbly
5	Vanet family - Rock outcrop complex, rubbly
6	Vanet - Wetmore families - Rock outcrop complex, stony
7	Gothic family
8	Supervisor - Limber families complex
9	Troutville family, very stony
10	Bullwark - Catamount families - Rock outcrop complex, rubbly
11	Bullwark - Catamount families - Rock land complex, rubbly
12	Legault family - Rock land complex, stony
13	Catamount family - Rock land - Bullwark family complex, rubbly
14	Pachic Argiborolis - Aquolis complex
15	unspecified in the USFS Soil and ELU Survey
16	Cryaquolis - Cryoborolis complex
17	Gateview family - Cryaquolis complex
18	Rogert family, very stony
19	Typic Cryaquolis - Borohemists complex
20	Typic Cryaquepts - Typic Cryaquolls complex
21	Typic Cryaquolls - Leighcan family, till substratum complex
22	Leighcan family, till substratum, extremely bouldery
23	Leighcan family, till substratum - Typic Cryaquolls complex
24	Leighcan family, extremely stony
25	Leighcan family, warm, extremely stony
26	Granile - Catamount families complex, very stony
27	Leighcan family, warm - Rock outcrop complex, extremely stony
28	Leighcan family - Rock outcrop complex, extremely stony
29	Como - Legault families complex, extremely stony
30	Como family - Rock land - Legault family complex, extremely stony
31	Leighcan - Catamount families complex, extremely stony
32	Catamount family - Rock outcrop - Leighcan family complex extremely stony
33	Leighcan - Catamount families - Rock outcrop complex extremely stony
34	Cryorthents - Rock land complex, extremely stony
35	Cryumbrepts - Rock outcrop - Cryaquepts complex
36	Bross family - Rock land - Cryumbrepts complex, extremely stony
37	Rock outcrop - Cryumbrepts - Cryorthents complex, extremely stony
38	Leighcan - Moran families - Cryaquolls complex, extremely stony
39	Moran family - Cryorthents - Leighcan family complex extremely stony
40	Moran family - Cryorthents - Rock land complex, extremely stony

Table III: Forest Cover Type.

Number	Forest Cover Type
1	Spruce/Fir
2	Lodgepole Pine
3	Ponderosa Pine
4	Cottonwood/Willow
5	Aspen
6	Douglas-fir
7	Krummholz

first set contained 12 attributes and the second set contained 11 attributes, second data set having the derived Euclidean distance. For the purpose of this paper, we will call the first dataset as dataset I and the second dataset as dataset II.

B. Feature Selection

The aim of feature selection is to choose a subset of features for improving prediction accuracy or decreasing the size of the structure without significantly decreasing prediction accuracy of the classifier built using only the selected features [9].

In many applications, the size of a dataset is so large that learning might not work as well before removing these unwanted features hence reducing the number of irrelevant and redundant features drastically reduces the running time of a learning algorithm and yields a more general concept [10].

Information gain is frequently employed as a feature selection method in the field of machine learning [11], [12], [13]. In this paper, we have employed information gain as our feature selection method. It measures the number of bits of information obtained for a category prediction by knowing the presence or absence of a term in a given document [13]. The numerical dataset was discretized. Information gain method was used in conjunction with entropy measure to rank the individual attributes in Waikato Environment for Knowledge Analysis (WEKA[2]) with the most important attribute being on the top.

Experiments were carried out using Naïve Bayes, K-Nearest Neighbor and Random Forest classification method by eliminating each attribute one by one from the bottom of the ranked list and the results for predicting the training data using 10 fold cross validation were noted. In N fold cross validation each instance of the data have a chance to participate in the training as well as testing of the data [14]. To save time for the experiments, a batch job using WEKA Java libraries was created for all the different cases of the experiment and ran on multiple set of computers.

C. Classification Algorithms

1) Naïve Bayes Classification Algorithm: Naïve Bayes is one of the machine learning supervised classification algorithm which is also called Naïve Bayes learner. It is one of the Bayesian learning method for constructing classification models which assign class model to problem instance. The Naïve Bayes classifier is based on the assumption that attribute values are independent of each other [15]. Although independence is generally a poor assumption, in practice Naïve Bayes often competes well with more sophisticated classifiers [15]. Naïve Bayes has proven effective in many practical applications, including text classification, medical diagnosis, and systems performance management [16]. The approach used by the Naïve Bayes classifier is given by:

$$\hat{y} = argmax_{k\{1..k\}}\ p(C_k)\prod_{i=1}^{n} p(x_i|C_k) \qquad (1)$$

where $\hat{y}$ denotes the resulting Naïve Bayes classifier. The objective of the Equation (1), is to find the maximum probability of the product between the given hypothesis probability of the class $p(C_k)$ and the probability of the attribute values x_i given the hypothesis of the class C_k given by $p(x_i|C_k)$. The result of the maximum probability from Equation(1), which has the corresponding hypothesis of the class C_k determines the class of the instance.

[2]WEKA: http://www.cs.waikato.ac.nz/ml/weka/

Algorithm 1: Naïve Bayes Classification Setup

```
Get the preprocessed ranked training dataset I and II from feature selection;
Iterate through dataset I and II for Naïve Bayes classification;
foreach feature set repeat until termination do
    Load dataset I into WEKA and perform Naïve Bayes classification with 10 fold cross validation;
    Load dataset II into WEKA and perform Naïve Bayes classification with 10 fold cross validation;
    Remove one bottom most feature from the dataset I and II without replacement;
end
```

The work presented in [17] depicts that Naïve Bayes classifier performs very well given a very large set of independent attributes with the ability to handle missing data in the dataset. However, Naïve Bayes classifier works on the strong assumption that the features in the given dataset is independent of each other which is clearly dishonored in the natural world where various types of features are related to each other [18]. It was highlighted in the work done by [19] that doing feature selection can improve the accuracy of Naïve Bayes classification.

For purpose of this study, as presented in Algorithm 1, we took dataset I and dataset II for training data separately and carried out the feature selection using information gain and ranked the attributes. After raking the feature, the classification test was carried out in WEKA using 10 fold cross validation technique. The first iteration of the classification experiment took all of the features and the results were noted. The second iteration was carried out by eliminating the bottom most feature from the ranked features without replacing the feature to the ranked set of features in each cycle of the experiment. This procedure was carried out until 3 features were left which was the termination condition to do the experiment. All the results were collected and tabulated to find the best model according to the classification accuracy.

2) K-Nearest Neighbor Classification Algorithm: The K-Nearest Neighbor (kNN) is another basic supervised machine learning algorithm. The kNN classifier is an elementary classification algorithm [1]. The kNN classification algorithm is based around the notion of how near certain instance(s) of points within a data set are to one another. The classifier determines how near an instance of data is to another instance, hereby referred to as the neighbor by a similarity metric (e.g. a function used to compute distance) [3]. A few of the common distance functions used with respect to the kNN classifier are: Euclidean, Cosine and Manhattan. In this particular study, the Euclidean distance metric was used to compute the similarity between instances as shown in Equation (2).

$$d(p,q) = d(q,p) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \qquad (2)$$

Algorithm 2: k-Nearest Neighbors Classification Setup

```
Get the preprocessed ranked training dataset I and II from feature selection;
Iterate through dataset I and II for k-Nearest Neighbors classification;
foreach feature set repeat until termination do
    foreach k until termination do
        Load dataset I and dataset II separately into WEKA and perform kNN classification with 10 fold cross validation;
    end
    Remove one bottom most feature from the dataset I and II without replacement;
end
```

where $d(p, q)$ denotes the resulting distance of the instance being calculated. The objective of the formula is to find a distance between a test instance and a training instance which usually has an accompanying class label associated with it. Further, the test instance is then classified by a majority vote of its surrounding neighbors. The majority vote aspect of the classifier is as follows; after the distances have been computed for a test instance the class label that is appearing most commonly amongst its k nearest neighbors is then applied to the test instance. Depending on the k value chosen, the appropriate majority is casted and the resulting label is applied [1]. Due to the fact that the kNN classifier implements a voting mechanism, it is for this reason that k values are usually chosen to be odd numbered values [20]. Odd numbers are chosen mainly to allow contingencies, in the event that a tie is resulting in the voting aspect.

As the Algorithm 2 presents, for this this study the training dataset I and training dataset II were extracted from Feature Selection and then loaded into WEKA. The k values chosen for this study ranged from 1 to 21. The odd numbered value of 21 was chosen due to the fact that there may be a tie when voting happens [20]. The range was also chosen because studies have shown that after k reaches 10, the accuracy usually starts to either be stable or decrease [20]. All of the results were then collected and subsequently graphed to find the best nearest neighbor which yielded the best accuracy for the kNN classifier.

3) Random Forest Classification Algorithm: Random forest is an ensemble machine learning classification method which consists of a collection of small decision trees where each decision tree predicts a class and the maximum frequency of predicted class is the class of the instance to be classified [6]. For instance, given the instance of the dataset as the input vector x as shown in Equation (3).

$$\hat{C}_{rf}^{B} = majorityvote\{\hat{C}_b(x)\}_1^B \tag{3}$$

where, $\hat{C}_b(x)$ is the class prediction of the bth random forest tree. As presented by [6], random forest has many advantages in the field of remote sensing such as, it runs efficiently on large data set, it has the ability to handle large amount of features and is robust to outliers and noise.

Random forest is developed by making smaller decision trees. Smaller decision tress is developed by selecting a random subset of instances from the given dataset with replacement. The subset is then divided into 3 parts where two third of the subset is used to construct the decision and the other one third which is called the out of the bag subset is used to test

Algorithm 3: Random Forest Classification Setup

```
Get the preprocessed ranked training dataset I and II from feature selection;
Iterate through dataset I and II for random forest classification;
foreach feature set repeat until termination do
    foreach k until termination do
        foreach m until termination do
            Load dataset I into WEKA and perform random forest classification with 10 fold cross validation;
            Load dataset II into WEKA and perform random forest classification with 10 fold cross validation;
        end
    end
    Remove one bottom most feature from the dataset I and II without replacement
end
```

the newly made decision tree [21]. The output error given by the out of the bag subset is called out of the bag error. The lower the out of the bag error, the better the decision tree is in predicting the class for the out of the bag subset [21]. In this way of random selection of subset and out of the bag test, every instance gets a chance to participate in building and testing of the decision tree [6].

The random forest classifier only needs two parameters to create the prediction model–the number of classification decision trees k and number of prediction variables m [6] where m is less that the total number of features in the dataset. While growing the trees, k and m has to be kept constant throughout the creation of prediction model. According to [22], it was highlighted that increasing k, the generalization error increases, decreasing the random forests prediction accuracy small and on the other hand decreasing m, decreases the correlation between individual decision trees and increasing the overall accuracy of the random forests prediction model.

As presented in Algorithm 3, for this experiment, the training dataset I and II which we got from the feature selection using information gain ranked attributes was loaded into WEKA to perform the random forest classification. The range of the k values used in the experiment was from 50 to 150. The reason for choosing this range was to explore the performance of the random forest prediction as highlighted by [22] which stated that increasing k value decreases the performance of the random forest prediction accuracy. For the dataset I, the minimum m value was chosen to be 1 and maximum m value was chosen to be 12 and for dataset II, the minimum m value was chosen to be 1 and maximum m value was chosen to be 11 where m was always less than or equal to the number of feature left in the dataset after the removal of the bottom most ranked feature one by one without replacement in each cycle until the dataset set reached to a limit of 3 features only. The reason for choosing 1 as the minimum m value for dataset I and II was to explore the idea presented by [6] that decreasing m, decreases the correlation between individual decision trees and increasing the overall accuracy of the random forests prediction model. The experimental setup was carried out until there was only 3 features left in the dataset and the k value reached from 50 to 150. All the results were collected, tabulated and graphed to find the best combination of k, m and number of features in the dataset which gave best accuracy for the random forest prediction model.

4) Test Data Classification Prediction: For the classification of the test data provided which had 565892 observations, the appropriate classification model and feature combination was chosen by analyzing all the results and getting the best model and parameters. The test data and training data was loaded into WEKA and classification was performed to get the class for the test data. The data collected from WEKA after classifying the test data was loaded into Kaggle to get the accuracy of the predicted class of the test data.

IV. Results and Discussion

A. Feature Selection

As given in Table IV, the ranking score was obtained from the feature selection method for training dataset I. For dataset II, the ranked scores are given in Table V. The ranking scores shown in Table IV-V imply the importance of the feature in determining the class. The lower the ranking score, the less important that feature is in determining the class and the higher the ranking score, the more important that feature is in determining the class.

Table IV: Training Dataset-I Ranked Features.

Ranking Score	Feature
1.4638	Elevation
1.3222	Soil_Type_Categorical
0.7377	Wilderness_Categorical
0.4335	Horizontal_Distance_To_Roadways
0.3012	Horizontal_Distance_To_Fire_Points
0.1871	Horizontal_Distance_To_Hydrology
0.1564	Hillshade_9am
0.123	Vertical_Distance_To_Hydrology
0.1034	Aspect
0.099	Hillshade_3pm
0.0977	Slope
0.0653	Hillshade_Noon

Table V: Training Dataset-II Ranked Features.

Ranking Score	Feature
1.4638	Elevation
1.3222	Soil_Type_Categorical
0.7377	Wilderness_Categorical
0.4335	Horizontal_Distance_To_Roadways
0.3012	Horizontal_Distance_To_Fire_Points
0.1868	Distance_To_Hydrology
0.1564	Hillshade_9am
0.1034	Aspect
0.099	Hillshade_3pm
0.0977	Slope
0.0653	Hillshade_Noon

The combination of features which was derived from the ranking scores for the training dataset I and dataset II, used to carry out all the classification experiments for Naïve Bayes, K-nearest neighbor and random forest are given in Table VI-VII. The feature set size was decreased by taking out the least ranked attribute one by one from the ranked feature set list.

B. Training Classification Accuracies

1) Naïve Bayes Training Classification Accuracy: The operations regarding the Naïve Bayes classifier allowed testing to be performed on preselected data and non-preselected data. Thus, the Naïve Bayes classifier was applied to both dataset I and dataset II.

The results will firstly look at the accuracy levels from dataset I. Using dataset I, the Naïve Bayes classifier was able to achieve an accuracy level of 67.6024% with 7 attributes. The Table VIII highlights the remaining accuracy levels accompanied by the number of attributes chosen using Feature Selection. Thus using feature selection, using 7 attributes yielded the highest accuracy level.

In comparison to the dataset I, dataset II had achieved a slightly better accuracy level. Running the Naïve Bayes classifier on 7 attributes had resulted in an accuracy level of 67.3413%. In Table IX, the rest of the results are shown with the number of attributes and their corresponding accuracy levels.

The poor accuracy levels actually show an intrinsic disadvantage of the Naïve Bayes algorithm, specifically for Forest Cover type prediction. The Naïve Bayes classifier is a machine learning algorithm which is based on probabilities [18], it also assumes that attributes are independent of each other. This notion of independence may be a reason why the accuracy level for the Naïve Bayes classifier yielded poor results, in many cases it is highly probable that the attributes regarding a forest cover type are related. Thus the independence brought on by the Naïve Bayes classifier may be the reason why the accuracy levels were poor.

2) K-Nearest Neighbor Training Classification Accuracy: Similar to the Naïve Bayes classifier, the kNN classifier was also applied on both dataset I and dataset II. The datasets were dataset I and dataset II and the k values chosen for the experiment ranged from 1 to 21. The number 21 was chosen due to the fact that there may arise a situation in the voting aspect where a tie may occur. Hence an odd number is needed to break the contest [1].

Further; using dataset I the kNN classifier was able to achieve an accuracy level of 86.713% with 6 features. The k value associated with this accuracy level was 1. Table X highlights the resulting accuracies with the associated k values and number of features.

Table XI illustrates the resulting accuracies with the associated k values and the number of features chosen. In contrast to dataset I, dataset II had achieved similar results. With 6 attributes and the k value chosen as 1, the resulting accuracy was 86.5873%. The difference from both of these accuracies was deemed irrelevant as the accuracy levels did not differ in terms of acceptable magnitudes i.e. the accuracy did not reach a higher level when the derived attribute Distance_to_Hydrology was used.

3) Random Forest Training Classification Accuracy: The best accuracies for each feature set size are given in Table XII and XIII. The highest of the all accuracies is highlighted in bold. The k value represents the number of trees and the m value represents the number of random split variables that were used to build random forest classification model for the respective feature set of the training dataset I and dataset II. The highest accuracy with dataset I was 87.6786% with feature set size of 9 and k = 112 and m = 3. Highest accuracy from Table XIII

Table VI: SET OF FEATURES FOR DATASET-I.

Feature Set Size	Features Used
12	Elevation,Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Horizontal_Distance_To_Hydrology, Hillshade_9am_Vertical_Distance_To_Hydrology, Aspect_Hillshade_3pm_Slope_Hillshade_Noon
11	Elevation,Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Horizontal_Distance_To_Hydrology, Hillshade_9am, Vertical_Distance_To_Hydrology, Aspect, Hillshade_3pm, Slope
10	Elevation,Soil_Type_Categorical Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Horizontal_Distance_To_Hydrology, Hillshade_9am, Vertical_Distance_To_Hydrology, Aspect, Hillshade_3pm
9	Elevation,Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Horizontal_Distance_To_Hydrology, Hillshade_9am, Vertical_Distance_To_Hydrology, Vertical_Distance_To_Hydrology, Aspect
8	Elevation,Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Horizontal_Distance_To_Hydrology, Hillshade_9am, Vertical_Distance_To_Hydrology
7	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Horizontal_Distance_To_Hydrology, Hillshade_9am
6	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Horizontal_Distance_To_Hydrology
5	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points
4	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points
3	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways

Table VII: SET OF FEATURES FOR DATASET-II.

Feature Set Size	Features Used
11	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Distance_To_Hydrology, Hillshade_9am, Aspect, Hillshade_3pm, Slope, Hillshade_Noon
10	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Distance_To_Hydrology, Hillshade_9am, Aspect, Hillshade_3pm, Slope
9	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Distance_To_Hydrology, Hillshade_9am, Aspect, Hillshade_3pm
8	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Distance_To_Hydrology, Hillshade_9am, Aspect
7	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Distance_To_Hydrology, Hillshade_9am
6	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points, Distance_To_Hydrology
5	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Fire_Points
4	Elevation, Soil_Type_Categorical, Wilderness_Categorical, Horizontal_Distance_To_Roadways
3	Elevation, Soil_Type_Categorical, Wilderness_Categorical,

Table VIII: NAIVE BAYES TRAINING CLASSIFICATION ACCURACIES FOR DATASET-I.

Feature Set Size	Accuracy Level
3	64.3386%
4	65.9524%
5	65.2447%
6	65.3704%
7	**67.6024%**
8	66.9048%
9	66.6071%
10	65.8796%
11	65.8664%
12	66.2037%

for dataset II was higher than the highest accuracy in Table XII for dataset I.

The highest accuracy for dataset II was 87.6582% with feature set size of 8 and $k = 143$ and $m = 3$. As per the result, it is clearly highlighted that feature selection improves the accuracy of the classification. On the other hand, decreasing the feature set size below 8 started to decrease the accuracy again. This is due to the dependency between the features where features are related to each other in determining the class of the instance. Dataset II contained the derived attribute Distance_To_Hydrology which was the Euclidean distance between horizontal distance to hydrology and vertical distance

Table IX: NAIVE BAYES TRAINING CLASSIFICATION ACCURACIES FOR DATASET-II.

Feature Set Size	Accuracy Level
3	64.3386%
4	65.9524%
5	65.2447%
6	65.4563%
7	**67.3413%**
8	67.0437%
9	66.3228%
10	66.0185%
11	66.4352%

Table X: K-NEAREST NEIGHBOR TRAINING CLASSIFICATION BEST ACCURACIES FOR DATASET I.

Feature Set Size	Best Accuracy Level	K-Value
3	70.6349%	18
4	78.287%	5
5	85.4167%	1
6	**86.713%**	1
7	86.1772%	1
8	86.455%	1
9	85.3704%	1
10	84.2328%	1
11	83.2474%	1
12	82.5595%	1

Table XI: K-NEAREST NEIGHBOR TRAINING CLASSIFICATION BEST ACCURACIES FOR DATASET II.

Feature Set Size	Best Accuracy Level	K-Value
3	70.6349%	18
4	78.287%	5
5	85.4167%	1
6	**86.5873%**	1
7	86.1839%	1
8	84.9868%	1
9	84.0278%	1
10	82.6852%	1
11	82.1495%	1

Table XII: RANDOM FOREST TRAINING CLASSIFICATION BEST ACCURACIES FOR DATASET-I.

Feature Set Size	Best Accuracy	k	m
3	65.2976%	103	1
4	78.6971%	105	1
5	84.8876%	96	1
6	87.0238%	72	2
7	87.5132%	80	3
8	87.5926%	141	2
9	**87.6786%**	112	3
10	87.586%	135	5
11	87.4735%	90	4
12	87.2354%	94	5

Table XIII: RANDOM FOREST TRAINING CLASSIFICATION BEST ACCURACIES FOR DATASET II.

Feature Set Size	Best Accuracy	k	m
3	65.2976%	103	1
4	78.6971%	105	1
5	84.8876%	96	1
6	87.0238%	61	1
7	87.3413%	148	3
8	**87.6582%**	143	3
9	87.4603%	114	5
10	87.4735%	144	4
11	87.3254%	123	3

to hydrology had the highest accuracy from both Table XII and Table XIII. This exercise of deriving one feature from two features resulted in the classification accuracy being highest as an attempt to reduce the feature set size without affecting the dependency, but the highest accuracy of dataset I and dataset II was very close. Furthermore, according to Table XII and Table XIII, it can be inferred that most of the highest accuracies which had the size of the trees k lied in the upper half of the range 50 to 150 and the m values was smaller than its respective feature set size which was used in the experiment. Thus, as according to [6], increasing the value of k and decreasing the value of m gives a better accuracy for classification.

4) Test Data Classification Accuracy: The prediction classification accuracy of the test dataset obtained from Kaggle by uploading the predicted classes was 76.802% where out of 565892 test data instances, 434616 instances were classified correctly. The model used for predicting the test data classes was random forest classification with 6 features from training dataset II as training data with test data having the same type and number of features where parameters for random forest classification was $k = 143$ and $m = 3$ as these set of parameters gave the highest training classification accuracy.

C. Discussion

Using the Naïve Bayes classifier on dataset I with feature set size 7, the resulting accuracy level was 67.6024%. In an attempt to improve the accuracy level of this classifier a derived attribute was generated in dataset II and obtained an accuracy level of 67.3413% with a feature set size of 7.

In addition to the Naïve Bayes classifier the training data was also used to build a model for classification using the kNN classifier. Using dataset I with a feature set size of 6 attributes and a k value of 1 the kNN classifier had reached an accuracy level of 86.713%. Similar to the Naïve Bayes classification method, another attempt at increasing the accuracy was made and found an accuracy level of 86.5873% with a feature set size of 6 and a k value of 1.

Moreover, the training data was also used to build another model for classification using the Random forests classifier. With dataset I, feature set size of 9, k value of 112 and m value as 3, the accuracy level obtained was 87.6786%. In another attempt to achieve higher accuracy, with the derived attribute the resulting accuracy from dataset II with a feature set size of 8, k value of 143 and m value of 3, was 87.6852%.

In comparison, the accuracy levels obtained from the Naïve Bayes classifier and Random forests classifier is significantly different. By looking at the accuracy levels alone it is evident that the Random forest classifier is superior to the Naïve Bayes classifier. There is a strong reason as to why the Naïve Bayes classifier does not perform as well as the Random forest classifier. According to [23] the Naïve Bayes classifier assumes that all of the features in a dataset is independent hence it is called Naïve. The Naïve Bayes classifier just calculates the probabilities of each individual attribute. Due to this the Naïve Bayes classifier ignores; or is more strictly put, blind to any dependences in the dataset.

The Naïve Bayes classifier did not perform as well compared to kNN classifier. According to [24] these two approaches attempt to achieve different goals as the kNN classifier determines the class label of a test instance based on how near one instance is from another. Thus the resulting class label from the nearest instances is then applied to the test instance [25]. In contrast, the Naïve Bayes classifier is based upon evaluating entropy. Entropy takes a look into how much useful information is possible to extract. For this, to work correctly, the number of attributes must be large then only will the classifier be able to correctly classify instances [24].

Furthermore, for the Naïve Bayes classifier and the kNN classifier the accuracy levels were also inferior to that of the Random forest classifier. According to [26], the amount of noise in a dataset may force the accuracy level of the model to decrease because class labels may be incorrectly classified. The only way to combat this is to use feature selection, which this study did implement, however, the resulting accuracy was still not higher than the accuracy from the Random forest classifier. Hence, if the dataset which will be used for the training model has noise and an uneven distribution the kNN classifier may result in a poor accuracy level.

V. CONCLUSION

In this paper, we evaluated three different classification methods—Naïve Bayes classification, K-nearest neighbor, and random forest—for forest cover type prediction. Feature selection and attribute derivation was also carried out to increase the accuracy and remove the dependency between the features. Several training classification experiment was performed with different set of parameters on the different feature sets on dataset I & II with each classification method. It was found out that random forest with feature selection and attribute derivation prevailed with the highest accuracy for forest cover type prediction, followed by K-nearest neighbor classification method and the least accuracy was shown by the Naïve Bayes classification algorithm. Thus, it can be concluded that random forest classification is a better approach to use while predicting forest cover type as the features are dependent on each other. In addition to feature selection, attribute derivation also plays a critical role in the classification accuracy. Therefore, it becomes important to find the best type(s) and number of feature(s) to use for classification.

REFERENCES

[1] H. Franco-Lopez, A. R. Ek, and M. E. Bauer, "Estimation and mapping of forest stand density, volume, and cover type using the k-nearest neighbors method," *Remote Sensing of Environment*, vol. 77, no. 3, pp. 251 – 274, 2001.

[2] T. E. Avery and H. E. Burkhart, *Forest measurements*. Waveland Press, 2015.

[3] B. T. Wilson, A. J. Lister, and R. I. Riemann, "A nearest-neighbor imputation approach to mapping tree species over large areas using forest inventory plots and moderate resolution raster data," *Forest Ecology and Management*, vol. 271, pp. 182 – 198, 2012.

[4] A. K. Gjertsen, "Accuracy of forest mapping based on landsat TM data and a knn-based method," *Remote Sensing of Environment*, vol. 110, no. 4, pp. 420 – 430, 2007.

[5] D. Lowd and P. Domingos, "Naive bayes models for probability estimation," in *Proceedings of the 22Nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 529–536.

[6] V. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 93 – 104, 2012.

[7] D. N. A. Asuncion, "UCI machine learning repository," 2007.

[8] I. D. Moore, P. Gessler, G. Nielsen, and G. Peterson, "Soil attribute prediction using terrain analysis," *Soil Science Society of America Journal*, vol. 57, no. 2, pp. 443–452, 1993.

[9] D. Koller and M. Sahami, "Toward optimal feature selection," 1996.

[10] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 3, pp. 131–156, 1997.

[11] T. M. Mitchell, "Machine learning," 1997.

[12] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[13] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[14] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2, 1995, pp. 1137–1145.

[15] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22. IBM New York, 2001, pp. 41–46.

[16] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2-3, pp. 103–130, 1997.

[17] H. Zhang, "The optimality of naive bayes," *AA*, vol. 1, no. 2, p. 3, 2004.

[18] K.-M. Schneider, "Techniques for improving the performance of naive bayes for text classification," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2005, pp. 682–693.

[19] K. Torkkola, "Linear discriminant analysis in document classification," in *IEEE ICDM Workshop on Text Mining*. Citeseer, 2001, pp. 800–806.

[20] W. A. Chaovalitwongse, Y.-J. Fan, and R. C. Sachdeo, "On the time series k-nearest neighbor classification of abnormal brain activity," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 6, pp. 1005–1016, 2007.

[21] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[22] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[23] M. J. Islam, Q. J. Wu, M. Ahmadi, and M. A. Sid-Ahmed, "Investigating the performance of naive-bayes classifiers and k-nearest neighbor classifiers," in *Convergence Information Technology, 2007. International Conference on*. IEEE, 2007, pp. 1541–1546.

[24] S. Gianvecchio, M. Xie, Z. Wu, and H. Wang, "Humans and bots in internet chat: measurement, analysis, and automated classification," *IEEE/ACM Transactions on Networking (TON)*, vol. 19, no. 5, pp. 1557–1571, 2011.

[25] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of workshop on learning from imbalanced datasets*, 2003.

[26] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2014.