# Prediction of Forest Cover using Ensemble Classification Techniques

Shreshtha Kulkarni: [skulkar6@mail.depaul.edu](mailto:skulkar6@mail.depaul.edu)

CDM,DePaul University: CSC-529 Advanced Data Mining, January 2017

## Introduction

The goal of this project is to predict predominant type of tree cover in small sections of forest on the basis of some cartographic variables like Elevation, Soil type, Hillshade, etc. Information regarding the forest land is highly required for developing ecosystem management strategies which will facilitate the decision making process. Classifying forest cover type can also help further research in areas like forest fire susceptibility, deforestation concerns etc.

We would analyze the prediction accuracy improvements that can be achieved by using ensemble classification techniques over the traditional Decision Trees.

## Data Description

This data is picked up from Kaggle although it is originally hosted by UCI Machine Learning repository. This dataset is generated from a study conducted by US Geological Survey and US Forest Service in four wilderness areas of Roosevelt National Forest of northern Colorado. The dataset comprises of 10 numeric variables and two qualitative variables originally obtained from US Geological Survey and USFS data. Description for each of the variables if given below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data Type** | **Measurement** | **Description** |
| Elevation | Quantitative | Meters | Elevation |
| Aspect | Quantitative | Degrees Azimuth | Aspect |
| Slope | Quantitative | Degrees | Slope |
| Horizontal\_Distance\_To\_Hydrology | Quantitative | Meters | Horz. Dist. to nearest surface water features |
| Vertical\_Distance\_To\_Hydrology | Quantitative | Meters | Vert. Dist. to nearest surface water features |
| Horizontal\_Distance\_To\_Roadways | Quantitative | Meters | Horz. Dist. To nearest roadway |
| Hillshade\_9am | Quantitative | 0 to 255 index | Hill shade index at 9am, summer solstice |
| Hillshade\_Noon | Quantitative | 0 to 255 index | Hill shade index at noon, summer solstice |
| Hillshade\_3pm | Quantitative | 0 to 255 index | Hill shade index at 3pm, summer solstice |
| Horizontal\_Distance\_To\_Fire\_Points | Quantitative | Meters | Horz. Dist. To nearest wildfire ignition points. |
| Wilderness Area (4 binary columns) | Qualitative | 0 (absence), 1(presence) | Wilderness Area designation |
| Soil Type (40 binary columns) | Qualitative | 0 (absence), 1(presence) | Soil Type designation |
| Cover Type | Class Variable | 1 to 7 | Forest Cover Type designation |

Table1: Description of features of Forest Cover Type dataset

The four wilderness areas referred here are **Rawah (area 1)**, **Neota (area 2)**, **Comanche Peak (area 3)** and **Cache La Poudre (area 4)**. The seven tree cover types referred here are

|  |  |
| --- | --- |
| **Class Number** | **Description** |
| 1 | Spruce/Fir |
| 2 | Lodgepole Pine |
| 3 | Ponderosa Pine |
| 4 | Cottonwood/Willow |
| 5 | Aspen |
| 6 | Douglas |
| 7 | Krummholz |

## Data Cleaning and Preprocessing

After reviewing the summary of data, I could see that data is pretty clean in the sense there were no missing values and the distribution of various numeric variables looked fine. However, the class variable was converted to factor and the numeric variables were standardized because they all were expressed on different scales and units.

However, due to computational power restrictions we will take a shorter sample of about 20% of original size of the data stratified to keep the original class representations. This was achieved by the *sample.split* method of library *caTools* in R.

*ind = sample.split(forestCovDS$Cover\_Type,SplitRatio = 0.2)*

*forestCovShort <- forestCovDS[ind,]*

This dataset was then divided into training and test sets in the ratio of 70% and 30% respectively.

## Preliminary Data Analysis

Data analysis was performed mostly visually and center and spread were inspected for individual variables along with their relationship with other variables. Some of the results are presented below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Variable** | **Min** | **Q1** | **Median** | **Mean** | **Q3** | **Max** |
| Elevation | 1859 | 2809 | 2996 | 2959 | 3163 | 3858 |
| Aspect | 0.0 | 58.0 | 127 | 155.7 | 260 | 360 |
| Slope | 0.0 | 9.0 | 13.0 | 14.1 | 18.0 | 66.0 |
| Horizontal\_Distance\_To\_Hydrology | 0.0 | 108.0 | 218.0 | 269.4 | 384.0 | 1397.0 |
| Vertical\_Distance\_To\_Hydrology | -173.00 | 7.00 | 30.00 | 46.42 | 69.00 | 601.00 |
| Horizontal\_Distance\_To\_Roadways | 0 | 1106 | 1997 | 2350 | 3328 | 7117 |
| Hillshade\_9am | 0.0 | 198.0 | 218.0 | 212.1 | 231.0 | 254.0 |
| Hillshade\_Noon | 0.0 | 213.0 | 226.0 | 223.3 | 237.0 | 254.0 |
| Hillshade\_3pm | 0.0 | 119.0 | 143.0 | 142.5 | 168.0 | 254.0 |
| Horizontal\_Distance\_To\_Fire\_Points | 0 | 1024 | 1710 | 1980 | 2550 | 7173 |

Table2: Five Number summary for quantitative variables

From Table1 we can see that quantitative variables are pretty well distributed. Figure2 shows the correlation between quantitative variables. Notice class variable was not converted to factor till now hence it appears in the correlation diagram. We can notice that it has highest correlation with Elevation which we will see in other graphs as well as when we observe importance of various predictors.

However, the class variable itself is heavily skewed with major percentage of data being available for only type 1 and 2. This might unfold itself as a major challenge as we build our models.

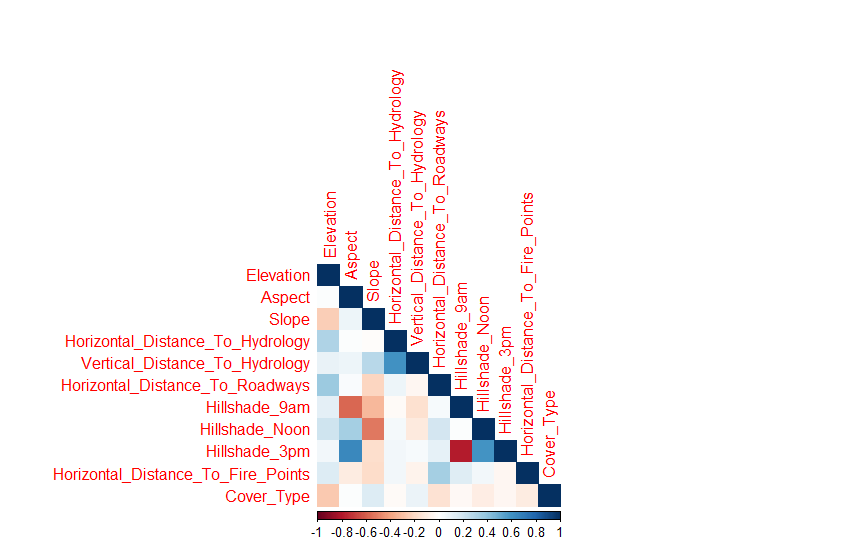
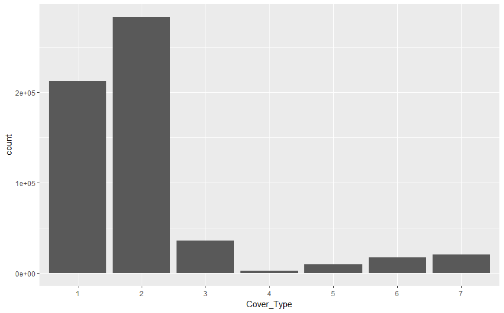


Figure 2: Correlation between quantitative variables

Figure 1:Distribution of class variable

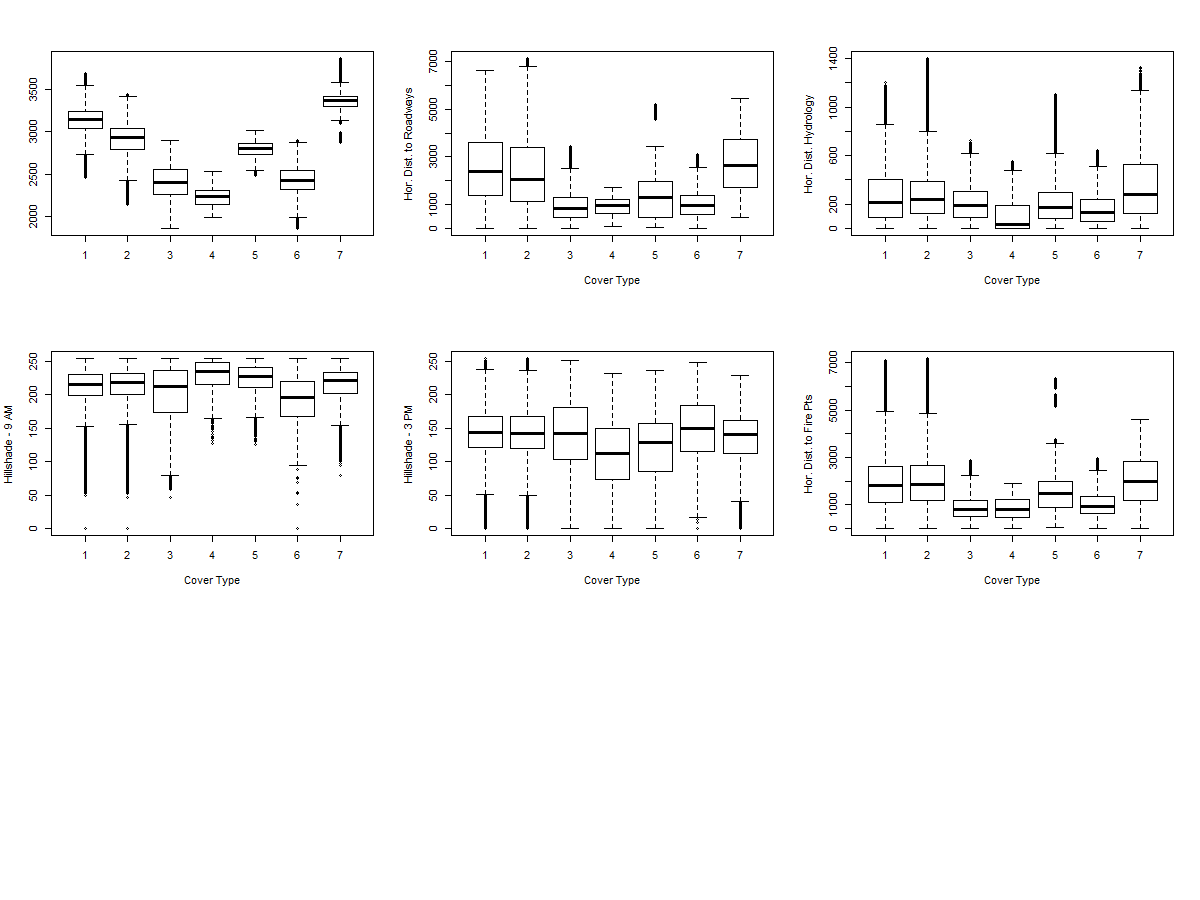


Figure 3:Variation of Cover Type with quantitative Predictors

Figure 3 show us the distribution of Cover Type with few numeric variables visually. Again notice that Elevation shows maximum variation per cover type.

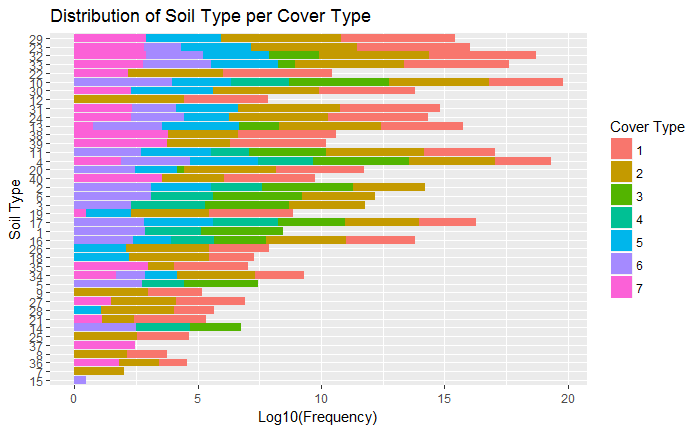


Figure 5: Dist. Soil Type per Cover Type

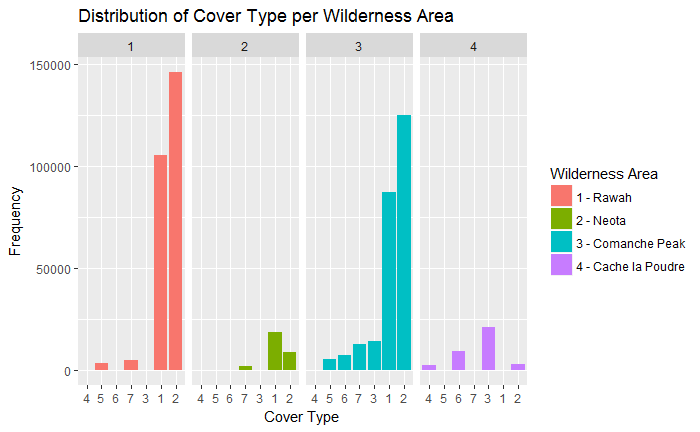


Figure 4: Dist. Cover Type per Wilderness Area

Figure 4 and 5 attempt to show the distribution of qualitative variables wilderness area and soil type per cover type. We can see that wilderness area 2 has least amount of tree cover and while cover type 1 and 2 are found in almost all wilderness areas and soil type cover type 4,5 and 6 take on very specific soil types and wilderness areas with cover type 4 is found only in wilderness area 4. These would prove to be significant observations as we will see when we inspect our models.

## Experimental Results

### Decision Tree (RPART)

To start our analysis, let’s first see the prediction power of a simple Decision Tree model. The metric used here is Accuracy, precision and recall. As mentioned above, we divide our representative sample into train and test sets in 70-30 ratio and build our model. Below are the results of evaluation on test set.

|  |  |
| --- | --- |
| *Overall Statistics*  *Accuracy : 0.6732*  *95% CI : (0.6682, 0.6781)*  *Kappa : 0.4514* | *Statistics by Class:*    *Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6 Class: 7*  *Sensitivity 0.7361 0.7371 0.71855 0.000000 0.0000 0.00000 0.00000*  *Specificity 0.7545 0.7283 0.96568 1.000000 1.0000 1.00000 1.00000*  *Pos Pred Value 0.6300 0.7243 0.57666 NaN NaN NaN NaN*  *Neg Pred Value 0.8343 0.7409 0.98139 0.994989 0.9834 0.97068 0.96623*  *Prevalence 0.3622 0.4920 0.06110 0.005011 0.0166 0.02932 0.03377*  *Detection Rate 0.2666 0.3626 0.04390 0.000000 0.0000 0.00000 0.00000*  *Detection Prevalence 0.4232 0.5006 0.07613 0.000000 0.0000 0.00000 0.00000*  *Balanced Accuracy 0.7453 0.7327 0.84211 0.500000 0.5000 0.50000 0.50000* |

These results show that DT performs poorly on this data with some of the individual class accuracies no better than a random guess. This gives us ample room for improvement using the ensemble technique. We will evaluate Bagging next in this analysis.

### Bagged CART

We first use a simple bagged CART model using the train function from caret and method *treebag*.

*model.bag.forest.short <- train(Cover\_Type ~.,data=trainForestCov,method="treebag")*

This gives us the following metrics

|  |  |
| --- | --- |
| *Overall Statistics*  *Accuracy : 0.8989*  *95% CI : (0.8957, 0.902)*  *Kappa : 0.836* | *Statistics by Class:*  *Class: X1 Class: X2 Class: X3 Class: X4 Class: X5 Class: X6 Class: X7*  *Sensitivity 0.8835 0.9318 0.89469 0.772727 0.593482 0.76117 0.88027*  *Specificity 0.9493 0.9020 0.99181 0.998999 0.997655 0.99449 0.99729*  *Pos Pred Value 0.9082 0.9020 0.87671 0.795322 0.810304 0.80658 0.91901*  *Neg Pred Value 0.9349 0.9318 0.99314 0.998856 0.993170 0.99280 0.99582*  *Prevalence 0.3622 0.4920 0.06110 0.005011 0.016598 0.02932 0.03377*  *Detection Rate 0.3200 0.4584 0.05466 0.003872 0.009851 0.02232 0.02972*  *Detection Prevalence 0.3524 0.5082 0.06235 0.004868 0.012157 0.02767 0.03234*  *Balanced Accuracy 0.9164 0.9169 0.94325 0.885863 0.795569 0.87783 0.93878* |

This gives a major boost on Accuracy but we can still see that some of the individual class accuracies are not that great. This looks like a problem of class imbalance. Since, treebag doesn’t provide any tuning parameters we will try to minimize the imbalance by repeated bootstrap sampling without replacement. After trying several values for number of folds and number of repeats, it was found that following model gives the maximum accuracies across classes.

*trainCtrlBag <- trainControl(method="repeatedcv",number=10,repeats = 5,classProbs = TRUE,returnResamp = "all")*

*model.bag.forest.short <- train(Cover\_Type ~.,data=trainForestCov,method="treebag",trControl=trainCtrlBag)*

|  |  |
| --- | --- |
| *Overall Statistics*  *Accuracy : 0.914*  *95% CI : (0.911, 0.9169)*  *Kappa : 0.8604* | *Statistics by Class:*    *Class: X1 Class: X2 Class: X3 Class: X4 Class: X5 Class: X6 Class: X7*  *Sensitivity 0.8970 0.9457 0.91146 0.818182 0.62264 0.80291 0.89376*  *Specificity 0.9591 0.9120 0.99381 0.999170 0.99809 0.99534 0.99776*  *Pos Pred Value 0.9257 0.9123 0.90556 0.832370 0.84615 0.83874 0.93310*  *Neg Pred Value 0.9425 0.9454 0.99424 0.999084 0.99366 0.99405 0.99629*  *Prevalence 0.3622 0.4920 0.06110 0.005011 0.01660 0.02932 0.03377*  *Detection Rate 0.3249 0.4653 0.05569 0.004100 0.01033 0.02354 0.03018*  *Detection Prevalence 0.3510 0.5100 0.06149 0.004925 0.01221 0.02807 0.03234*  *Balanced Accuracy 0.9280 0.9288 0.95264 0.908676 0.81037 0.89912 0.94576* |

### Bagged AdaBoost

This algorithm gives us the tuning parameters maxdepth (max tree depth) and mfinal (#trees). One model kept mfinal constant at 50 and evaluated maxdepth between 5 to 10. Metric evaluated was Accuracy again. Resampling procedure used was 10-fold Cross Validation repeated 3 times. Final metric were

Bagged AdaBoost

*maxdepth Accuracy Kappa*

*5 0.7091825 0.5169502*

*6 0.7264430 0.5488901*

*7 0.7401654 0.5687697*

*8 0.7580244 0.5997831*

*9 0.7746044 0.6283187*

*10 0.7912705 0.6558812*

Another iteration of this algorithm was tried with maxdepth constant at 10 and mfinal values evaluated were 50,80,100,200 and 300. This tree finally converged at 200. However, the overall accuracy didn’t see much of an increase.

When we evaluate the confusion matrix for this model, we can see that recall values for under-represented classes are very low.

|  |  |
| --- | --- |
| *Overall Statistics*    *Accuracy : 0.7931*  *95% CI : (0.7888, 0.7973)*  *Kappa : 0.6594* | *Statistics by Class:*    *Class: X1 Class: X2 Class: X3 Class: X4 Class: X5 Class: X6 Class: X7*  *Sensitivity 0.7554 0.8664 0.83494 0.518987 0.209790 0.45767 0.69141*  *Specificity 0.8904 0.7782 0.98515 0.999424 0.999301 0.99462 0.99680*  *Pos Pred Value 0.7981 0.7876 0.78888 0.803922 0.833333 0.72997 0.88474*  *Neg Pred Value 0.8638 0.8599 0.98899 0.997816 0.986995 0.98297 0.98911*  *Prevalence 0.3646 0.4870 0.06232 0.004527 0.016390 0.03080 0.03436*  *Detection Rate 0.2754 0.4220 0.05204 0.002350 0.003438 0.01410 0.02375*  *Detection Prevalence 0.3451 0.5357 0.06596 0.002923 0.004126 0.01931 0.02685*  *Balanced Accuracy 0.8229 0.8223 0.91005 0.759206 0.604546 0.72615 0.84410* |

### 5.4 Random Forest

I used the parallel Random Forest algorithm from caret package for this analysis. The final results were comparable to bagged CPART trees. It only gave marginal increase in the overall accuracy and precision and recall values of individual classes. Model was tuned for no. of predictors and the final model selected 41 predictors to be giving maximum efficiency. Final metric is given below:

|  |  |
| --- | --- |
| *Overall Statistics*  *Accuracy : 0.9213*  *95% CI : (0.9184, 0.9241)*  *Kappa : 0.8728* | *Statistics by Class:*  *Class: X1 Class: X2 Class: X3 Class: X4 Class: X5 Class: X6 Class: X7*  *Sensitivity 0.9055 0.9512 0.93333 0.746835 0.64860 0.81674 0.88991*  *Specificity 0.9632 0.9190 0.99254 0.999482 0.99869 0.99681 0.99804*  *Pos Pred Value 0.9339 0.9176 0.89270 0.867647 0.89183 0.89047 0.94175*  *Neg Pred Value 0.9467 0.9520 0.99556 0.998849 0.99417 0.99419 0.99609*  *Prevalence 0.3646 0.4870 0.06232 0.004527 0.01639 0.03080 0.03436*  *Detection Rate 0.3301 0.4632 0.05817 0.003381 0.01063 0.02516 0.03057*  *Detection Prevalence 0.3535 0.5048 0.06516 0.003897 0.01192 0.02825 0.03247*  *Balanced Accuracy 0.9343 0.9351 0.96294 0.873159 0.82365 0.90678 0.94397* |

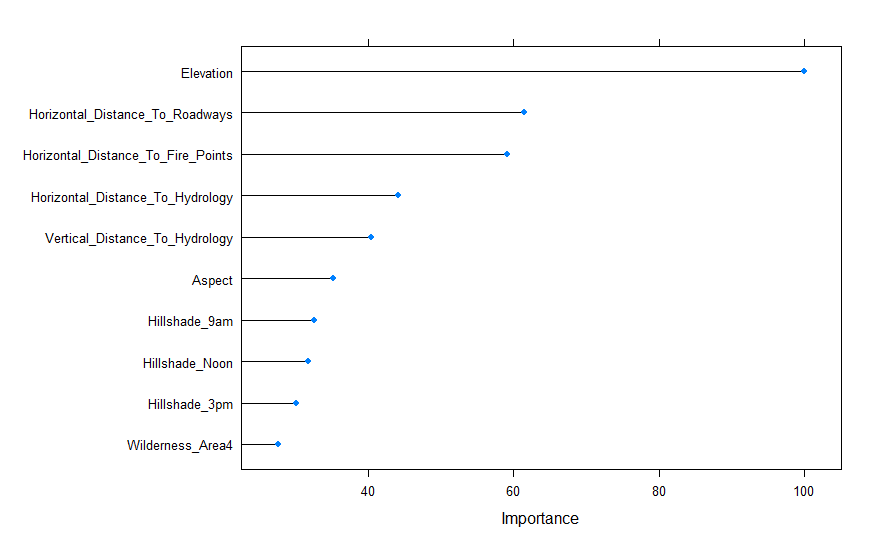
## Experimental Analysis

We can see that out of three types of algorithms tried only bagged CPART trees and randomForest gave us the maximum accuracy. However, baggedCPART was chosen as our final modeling technique because of its simplicity and time taken to build the model was much less than other techniques.

The final model was fitted on complete forest cover dataset (≈600,000 rows) by creating an ensemble of 10 models with 10% of training data each and then arriving at the final predictions by majority voting. This gave out a final accuracy of approx. 90% and Kappa of ≈85% with average precision and recall values of 91% and 70% across classes. We have also used repeated 10-fold CV which repeats 3 times to prevent overfitting the results.

We can also see that resampling and bagging have addressed imbalanced class ratios to a great extent which was a major concern at the start of this analysis and a reason why traditional Decision Trees failed miserably over this data.

The next step was to evaluate the important features as considered by the model. This was done with the help function ***varImp*** , output of which gives first 10 features in order of their importance:



We can see that **Elevation** is considered to be the most important feature followed by the **Horizontal\_distance\_to\_Roadways.** This was evident from the initial visual exploratory analysis that is provided in section 3.

## Conclusion

In this study, we found that ensemble techniques really proved efficient in accurately predicting the predominant cover type of the forest which was our main goal. However, we do realize that the recall value for couple of classes were not up to a satisfactory level. Further studies would be directed over trying out different classification techniques e.g multi-class SVM and clustering to see if we can improve the accuracy across all the classes.