Forest Cover Type Prediction

Ankush Agrawal, Apurv Garg, Sreekanth Krishnaiah, Subhankar Ghosh

December 20, 2017

Abstract

Understanding forest composition is a valuable aspect of managing the health and vitality of our wilderness areas. Classifying cover type can help further research regarding forest fire susceptibility, the spread of the Mountain Pine Beetle infestion, and de/reforestation concerns. In this project we predict forest cover type using cartographic data obtained by US Geological Survey (USGS) and US Forest Service (USFS). We have found that Random Forest predicts forest cover with 84.64% accuracy.

Introduction

What is forest cover? to do For any private, state, or federal land management agency it is essential to have proper data about our wilderness areas. Forest cover type is one of the most important information seeked by these agencies due to myriad applications and research studies. Classifying cover type can help further research regarding forest fire susceptibility, the spread of the Mountain Pine Beetle infestion, and de/reforestation concerns. Generally, such kind of data are directly recorded by field personnel and/or estimated from remotely sensed data. Both of the above mentioned techniques are time consuming and resource intensive. Predictive models are an alternative efficient technique to get such kind of data.

The study area includes four wilderness areas located in the Roosevelt National Forest of northern Colorado. Each observation is a 30m x 30m patch of forest land. We aim to predict the forest cover type. The seven types are:

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

Figure 1: Spruce/Fir

Figure 2: Lodgepole Pine

Figure 3: Ponderosa Pine

We have collected this dataset from Kaggle Forest Cover Type Prediction which contains 15120 observations.

Figure 4: Cottonwood/Willow

Figure 5: Aspen

Figure 6: Krummholz

Figure 7: Douglas-fir

The actual forest cover type for a given observation (30 x 30 meter cell) was determined from US Forest Service (USFS) Region 2 Resource Information System (RIS) data. Independent variables were derived from data originally obtained from US Geological Survey (USGS) and USFS data. Data is in raw form (not scaled) and contains binary (0 or 1) columns of data for qualitative independent variables (wilderness areas and soil types)

The predictors we are using give us information about

1. Elevation of the region in meters
2. Aspect in degrees, slope of the region in degrees
3. Horizontal distance to nearest surface water features
4. Vertical Distance to nearest surface water features
5. Horizontal Distance to nearest roadway
6. Hillshade index at 9 AM

7. Hillshade index at noon
8. Hillshade index at 3 PM
9. Horizontal distance to nearest wildfire ignition points
10. Wilderness of the area assigned (Rawah Wilderness Area, Neota Wilderness Area, Comanche Peak Wilderness Area, Cache la Poudre Wilderness Area)
11. Type of soil

We will be performing a multi-class classification task on forest cover type from this dataset. The response variable in the dataset is *Cover_Type*. In this study we will examine the ability of models like Random Forest, K-Nearest Neighbour, Elastic-Net and Extreme Gradient Boosting to predict cover type classes.

Materials and Models

We have collected this dataset from Kaggle Forest Cover Type Prediction which consists of the following features:

- Elevation - Elevation of the region in meters
- Aspect - Aspect in degrees azimuth
- Slope - Slope of the region in degrees
- Horizontal_Distance_To_Hydrology - Horizontal Distance to nearest surface water features
- Vertical_Distance_To_Hydrology - Vertical Distance to nearest surface water features
- Horizontal_Distance_To_Roadways - Horizontal Distance to nearest roadway
- Hillshade_at_9am (0 to 255 index) - Hillshade index at 9am, summer solstice
- Hillshade_at_Noon (0 to 255 index) - Hillshade index at noon, summer solstice
- Hillshade_at_3pm (0 to 255 index) - Hillshade index at 3pm, summer solstice
- Horizontal_Distance_To_Fire_Points - Horizontal Distance to nearest wildfire ignition points
- Wilderness_Area (4 binary columns, 0 = absence or 1 = presence) - Wilderness area designation
- Soil_Type (40 binary columns, 0 = absence or 1 = presence) - Soil Type designation
- Cover_Type (7 types, integers 1 to 7) - Forest Cover Type designation

Preprocessing

We first loaded the data in R

We randomly split the data into train-test datasets, with 70% data as trainset and 30% as testset

We checked for **missing values** in the training set. There was no missing values found so we do not implement an imputation method.

We noticed that the columns **soil_type_7 and soil_type_15** contain only 0 values so we removed them from both testset and trainset.

We converted the below mentioned columns from integers to factors since they are binary variables. * Wilderness_Area (4 binary columns, 0 = absence or 1 = presence) - Wilderness area designation - Wilderness_Area1 (0 = absence or 1 = presence) - Wilderness_Area2 (0 = absence or 1 = presence) - Wilderness_Area3 (0 = absence or 1 = presence) - Wilderness_Area4 (0 = absence or 1 = presence) * Soil_Type (40 binary columns, 0 = absence or 1 = presence) - Soil Type designation

Now once our preprocessing is done we move towards model fitting.

Evaluation Metric

We will use misclassification rate as our metric for evaluating the models.

Model Fitting

Random Forest centred and scaled

```
set.seed(1337)
rf_grid = expand.grid(mtry = 21)
rf_with_preprocessing = train(form = Cover_Type ~. , data = train_data,
                      method = "rf",
                      trControl = trainControl(method = "oob"),
                      preProcess = c("center", "scale"),
                      importance = TRUE,
                      tuneGrid = rf_grid)
```

Our best model had an mtry value of 21. We found the train and test errors and stored them for comparison later.

Random Forest without centering and scaling

```
set.seed(1337)

rf_without_preprocess = train(form = Cover_Type ~. , data = train_data,
                      method = "rf",
                      trControl = trainControl(method = "oob"),
                      tuneGrid = rf_grid)
```

We found the train and test errors and stored them for comparison later.

Elastic net

```
set.seed(1337)
train_model_en = train(form = Cover_Type ~. , data = train_data,
                  method = "glmnet",
                  trControl = trainControl(method = "cv", number = 5),
                  tuneLength = 10)
```

We found the train and test errors and stored them for comparison later.

KNN without pre scaling

```
train_model_knn = train(form = Cover_Type ~. , data = train_data,
                    method = "knn",
                    trControl = trainControl(method = "cv", number = 5))
```

We found the train and test errors and stored them for comparison later.

KNN with pre scaling

```
train_model_knn_2 = train(form = Cover_Type ~. , data = train_data,
                    method = "knn",
                    trControl = trainControl(method = "cv", number = 5),
                    preProcess = c("center", "scale"))
```

We found the train and test errors and stored them for comparison later.

Extreme Gradient Boosting

We found the train and test errors and stored them for comparison later.

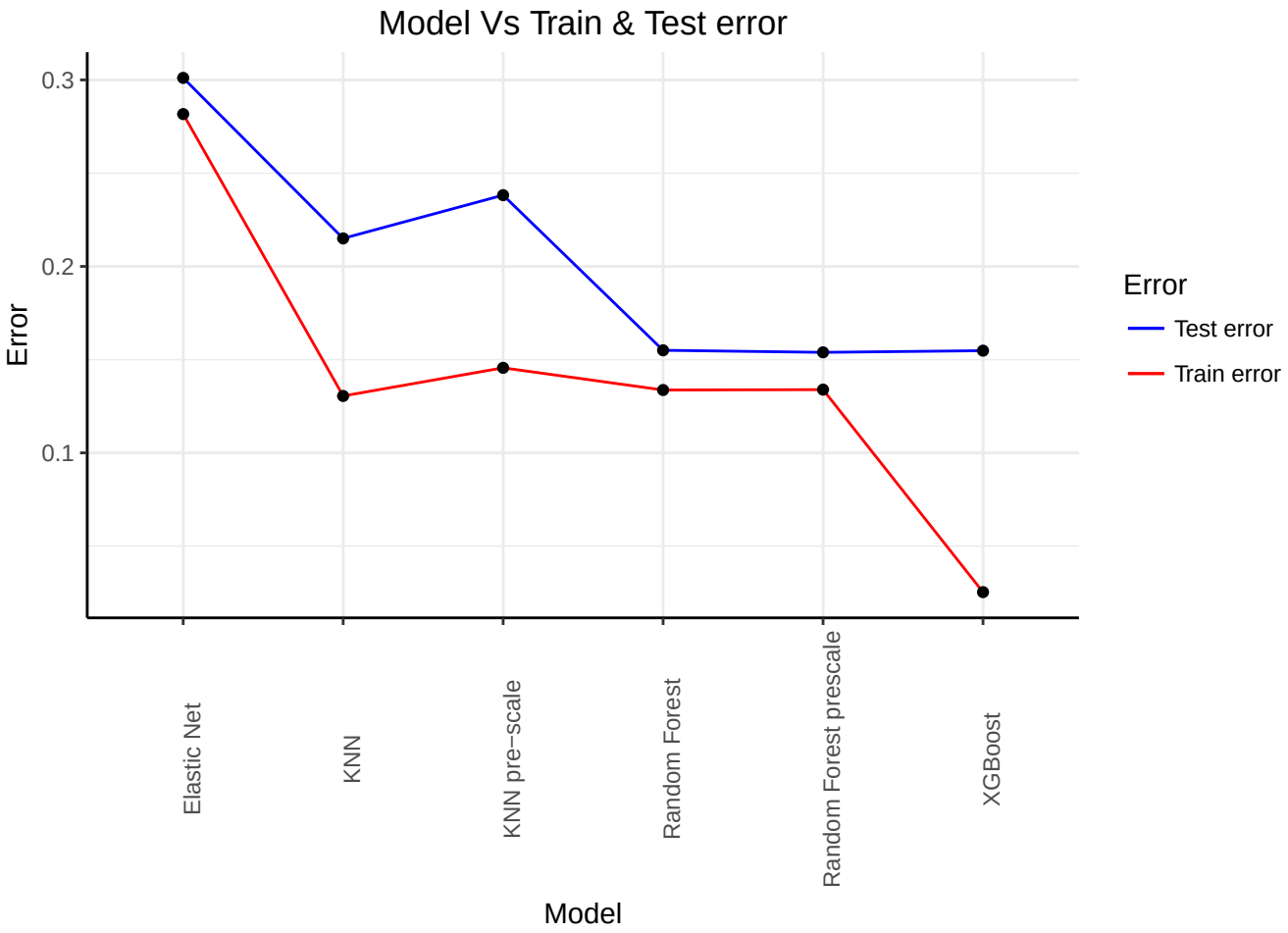

Figure 8: Comparative study between models

Results

The variable importance we obtained from random forest that we had trained.

Table 1: Variable importance table

	1	2	3	4	5	6	7
Id	19.33632	17.018787	33.11384	27.60869	31.75618	36.48499	25.14081
Elevation	95.35731	44.300995	33.10269	43.70145	100.00000	50.48976	74.20911
Aspect	15.60093	10.831309	23.44590	14.16295	23.55149	19.67960	16.50134
Slope	12.83520	8.945049	20.40845	13.97005	21.68949	22.69449	17.57575
Horizontal_Distance_To_Hydrology	16.81302	23.625361	26.35279	44.23429	31.35069	37.46526	22.09561
Vertical_Distance_To_Hydrology	18.36509	11.755047	23.59005	18.06045	30.04286	26.68130	18.39814
Horizontal_Distance_To_Roadways	19.49713	22.140600	27.31528	26.81088	44.55286	32.19693	24.15635
Hillshade_9am	18.51342	12.747510	28.32246	17.58527	24.88094	29.45820	21.36871
Hillshade_Noon	15.49941	20.879121	20.45218	16.63590	26.69221	24.76618	19.40359
Hillshade_3pm	17.61415	13.026468	19.05445	13.22670	21.52449	21.65955	20.70812

The misclassification rate across all the above models are shown in Fig @ref(fig:plot-comparison). We can say from the figure that Random Forest with preprocessing performs the best among all the models with a test

accuracy of 84.60%.

Table 2: Model Comparison

Model	TrainError	TestError
Random Forest prescale	0.1339	0.1539
Random Forest	0.1337	0.1550
Elastic Net	0.2817	0.3011
KNN	0.1305	0.2150
KNN pre-scale	0.1456	0.2382
XGBoost	0.0253	0.1548

The same result can be seen from Table @ref(tab:tab-comparison).

Discussion

Random Forest without preprocessing works the best as we can see from Table @ref(tab:tab-comparison) and Figure @ref(fig:plot-comparison). The results of Random forest with preprocessing and Extreme Gradient Boosting are very close to our best model. Our best model Randome Forest has gives us the best results with 500 trees and 21 as the value of mtry which is approximately $m/3$ where m is the number of predictors we have used.

Our dataset consists of both qualitative and quantitative predictors, and most of them are not normally distributed that is why LDA or QDA was not considered for modelling. It is not possible to fit a particular distribution to the predictors and also it is not possible to find a clear boundary between the classes therefore Generative models were not used. All the models under consideration are Discriminative.

We have tried to fit a parametric Elastic Net model but it gave high misclassification rate compared to the other non-parametric models we used like k-Nearest Neighbour, Random Forest and Extreme Gradient Boosting. Inspite of their high interpretability and high speed the Parametric models fail to give high accuracy due to their limited complexity. The non parametric models although they run the risk of overfitting are capable of giving high accuracy with their high flexibility because they make no assumptions about the underlying data. In our case this is exactly what had happened since the response variable did not have any visible pattern with respect to the predictors so a highly flexible model like Random Forest and XGBoost was expected to perform better.

Except the Elevation feature no other features(after/before transformation) showed linear relationship with the response variable so it was expected that a non-linear model would perform better compared to a linear one. This is precisely what we see in the results where non linear models like Random Forest and XGBoost performed much better that the linear Elastic net model.

Due to high dimensionality of the dataset k-Nearest Neighbour performed poorly since k-NN is a spacial model and with increase in the number of dimensions the distance between points increase, also the non significant features contribute the same as the most important features. From Fig @ref(fig:var-imp) we can see that the 11 most important features are Elevation, Horizontal distance to roadways, Horizontal distance to fire points, Horizontal distance to hydrology, Vertical distance to hydrology, Hillshade at 9AM, Wilderness area 4, Aspect, Hillshade at noon, Hillshade at 3PM and slope so if we can get the above features we can pretty well predict the forest cover even though if we do not know the soil types.

Conclusion

Understanding forest composition is a valuable aspect of managing the health and vitality of our wilderness areas. Classifying cover type can help further research regarding forest fire susceptibility, the spread of

the Mountain Pine Beetle infestion, and de/reforestation concerns. In this project we predict forest cover type using cartographic data obtained by US Geological Survey (USGS) and US Forest Service (USFS). We analyzed the data, inspected for missing value and 0 variation columns. After conversion of categorical columns to factors we compared a number of models like k-Nearest Neighbour, Random Forest and Extreme Gradient boosting and found out that Random Forest performs the best in predicting forest cover type giving us a test accuracy of 84.60%. Our future work would encompass experimenting with other modelling techniques and going into the Artificial Neural Networks and deep-learning to predict forest cover type.