Forest cover type prediction

Daniel Lemes Gribel, PUC-Rio

December 1, 2015

Contents

1 Introduction **2**

2 Dataset **2**

3 Problem statement **3**

4 Baselines **3**

5 Models **4**

5.1 Decision forest based models 4

5.2 Decision tree based models 5

5.3 Neural networks 5

5.4 Features engineering 5

5.5 Blend 6

6 Experiments **6**

7 Results **7**

8 Conclusion **8**

9 References **9**

1 Introduction

Natural resource managers responsible for developing ecosystem management strategies require basic descriptive information including inventory data for forested lands to support their decision-making processes. However, managers generally do not have this type of data for neighboring lands that are outside their jurisdiction. One method of obtaining this information is through the use of predictive models [1].

In this report, the methodology and the main results for predictive models applied to the forest cover problem will be presented. In this work, the focus will be given especially on the models chose to predict this multi-class classification problem and the results achieved in terms of accuracy.

The task consists basically in predicting the forest cover type from descriptive variables. The studied area includes four wilderness areas located in the Roosevelt National Forest in Colorado, where each observation (30 x 30 meter cell) was determined by the US Forest Service. These areas represent forests with minimal human-caused disturbances, so forest covers are more a result of ecological processes than forest management practices [2].

2 Dataset

The complete dataset is composed by 581,012 instances, where each observation (instance) corresponds to a 30 x 30 meters cell. For each observation, 12 measures (attributes) are given. However, two categorical properties (wilderness area and soil type) were binarized, in order to have a dataset composed only by numerical attributes. So, the final dataset is composed by 10 quantitative variables, 4 binary wilderness areas and 40 binary soil type variables, given a total of 54 columns of data. Below, some description about these attributes is given:

- **Elevation**: Elevation in meters.
- **Aspect**: Aspect in degrees azimuth.
- **Slope**: Slope 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 9am**: Hillshade index at 9am, summer solstice (0 to 255).
- **Hillshade Noon**: Hillshade index at noon, summer soltice (0 to 255).
- **Hillshade 3pm**: Hillshade index at 3pm, summer solstice (0 to 255).
- **Horizontal distance to fire points**: Horizontal distance to nearest wildfire ignition points.
- **Wilderness area**: Wilderness area designation. 4 binary columns, 0 (absence) or 1 (presence).
- **Soil type**: Soil type designation. 40 binary columns, 0 (absence) or 1 (presence).
- **Cover Type**: Forest cover type designation. This is the attribute to be predicted (1 to 7).

The dataset was divided in the following way: 15,120 observations (2.6%) compose the training set and 565,892 observations (97.4%) compose the test set. In order to evaluate the accuracy on trainig set, it was selected 75% for training and 25% for validation.

Regarding the Cover Type attribute, the following distribution is observed in th whole dataset:

Table 1: Cover type distribution on whole dataset

Cover Type	Number of observations
1 Spruce-Fir	211,840
2 Lodgepole Pine	283,301
3 Ponderosa Pine	35,754
4 Cottonwood/Willow	2,747
5 Aspen	9,493
6 Douglas-fir	17,367
7 Krummholz	20,510
Total records	581,012

3 Problem statement

The input of the problem is composed by N forest cells (x_1, x_2, $\ldots$, x_N), where x_i corresponds to a 54-dimensional vector. Each element in vector x_i represents an attribute value.

The output of the problem is a function y which will receive an instance x as input and will classify it as one of the 7 cover types. Hence:

$$y(x) = \{1, 2, 3, 4, 5, 6, 7\}$$

Finally, the model evaluation is measured by the accuracy:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

where TP is the number of true positive classifications, TN is the number of true negative classifications, FP is the number of false positive classifications and FN is the number of false negative classifications.

4 Baselines

Table 2 shows some known baselines for the Cover type problem:

Table 2: Baseline (published studies)

Model	Author(s)	Year	Accuracy
Multi-class SVM	Crain; Davis [3]	2014	78.64%
Feed-forward artificial neural network	Blackard; Denis [4]	1999	70.58%
Linear discriminant analysis	Blackard; Denis [4]	1999	58.38%
Statistical baseline (most frequent class)	-	-	48.76%

Apart the above results that correspond to published studies, it will also be reported a comparison to results achieved on a Kaggle competition for this problem, which happened in 2015. Below, the top-15 teams accuracies on Kaggle competition:

Table 3: Baseline (Kaggle competition)

Position	Team name	Accuracy
1	antgleb	100%
2	Ashish Singh	99.99%
3	ucbw207_2_forest	99.75%
4	Tian Zhou	99.53%
5	Doru Arfire	98.91%
6	TIEN VU	98.52%
7	Eugene Gritskevich	94.59%
8	Ivan Liu	94.27%
9	Celine Theeuws	91.21%
10	Deep Learner (Rahul Mohan)	90.92%
11	Audere Labs	90.39%
12	Gopal Joshi	89.60%
13	Sisyphus	89.25%
14	Hans	88.77%
15	6.867_RunpengLiu	87.44%

5 Models

In this section, the selected models, how they were configured and the features engineering applied in order to improve results are described. This work is mainly based on decision trees models, with some variants. Actually, 2 models are based on decision forests, 1 model is based on boosted trees and 1 model is defined by a neural network.

5.1 Decision forest based models

Two models used in this work are based on decision forests. The algorithm works by building multiple decision trees and then voting on the most popular output class. Voting is a form of aggregation, in which each tree outputs a frequency histogram of labels. The aggregation process sums these histograms and normalizes the result to get the "probabilities" for each label [5].

For this purpose, two similar models were tested: *Multi-class decision forest* and *One-vs-all decision forest*.

Multi-class decision forest: It follows the same idea of decision forests described above. However, it considers each class (in our case 7 classes) as a possible answer given by a leaf node (decision).

In this work, the following configuration was adopted for the *Multi-class decision forest* model:

```
Resampling method: Replicate
Number of decision trees: 70
Maximum depth of the decision trees: 90
Number of random splits per node: 256
Minimum number of samples per leaf node: 1
```

One-vs-all decision forest: It follows the same idea of decision forests described above. However, it considers only two classes at a time as a possible answer given by a leaf node (decision). Thus, a binary model is created for each of the multiple classes. Each of these binary models is assessed against its complement (all other classes in the model) as though it were a binary classification issue. Then, prediction is performed by applying the one-vs-all method to combine the results for all classes [5].

In this work, the following configuration was adopted for the *One-vs-all decision forest* model:

```
Resampling method: Replicate
Number of decision trees: 80
Maximum depth of the decision trees: 100
Number of random splits per node: 512
Minimum number of samples per leaf node: 1
```

5.2 Decision tree based models

Boosted trees: The basic idea involved in decision trees is to break up a complex decision into a union of several simpler decisions [6]. In addition, for boosting, the training events which were misclassified have their weights increased (boosted), and a new tree is formed. This procedure is then repeated for the new tree. In this way, many trees are built up [7].

In this work, the following configuration was adopted for the *Boosted trees* model:

```
Maximum number of leaves per tree: 40
Minimum number of samples per leaf node: 20
Learning rate: 0.1
Number of trees constructed: 1468
```

5.3 Neural networks

Multi-class neural networks: A neural network is a set of interconnected layers, in which the inputs lead to outputs by a series of weighted edges and nodes [8]. Thus, the target of neural networks lies on learning these weights on edges that lead to correct outputs.

In this work, the following configuration was adopted for the *Multi-class neural networks* model:

```
Number of hidden layers: 2
Number of nodes in each layer: 200
Learning rate: 0.1
Number of learning iterations: 1000
Initial learning weights diameter: 0.1
Momentum: 0
Type of normalizer: Binning
```

5.4 Features engineering

Feature engineering was applied in order to add more features (attributes) to the existing data. For now, features were generated as a consequence of manual analysis. For instance, data contain many features based on distance measures (e.g: elevation, vertical distance to hydrology, etc). Therefore, some features can be combined in order to generate new ones that express some relation between them. For this purpose, this work considers the generation of 11 new features, basically divided in 3 types of features engineering.

- **Attributes extraction**: Due to some relation between the Soil type attribute with climatic and geological zones, we can extract some extra properties from data, which are not available in the original dataset. These relations can be obtained on UCI Machine learning repository. Thus, this phase, which actually does not correspond to a feature engineering properly, generates 2 new attributes: Climatic zone and Geological zone.

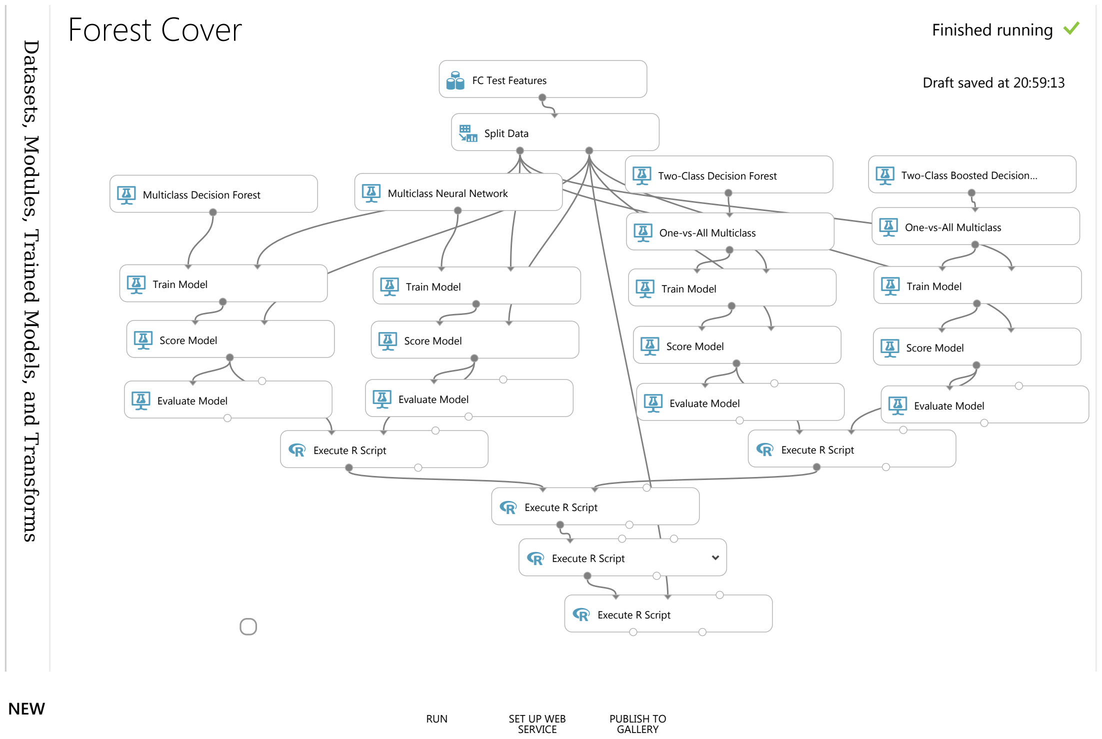
- **Linear combination**: The sum and difference between two pair of attributes. The following pairs of measures are considered, generating, therefore, 2x4 = 8 new attributes: (Elevation, Vertical distance to Hydrology), (Horizontal distance to Hydrology, Horizontal distance to Fire Points), (Horizontal distance to Hydrology, Horizontal distance to Roadways) and (Horizontal distance to Fire Points, Horizontal distance to Roadways).
- **Euclidean distance**: The euclidean distance applied to attributes Horizontal distance to Hydrology and Vertical distance to Hydrology, generating 1 new attribute.

5.5 Blend

Finally, after models selection and features engeneering, this work performed a blend procedure, by selecting the most popular (mode) prediction along all models in each instance being tested.

6 Experiments

Experiments were performed in Azure Machine Learning Studio. The figure below illustrates the experiment, which contains the 4 models described previously and some R scripts that perform the blend phase. An external R script performed the features engineering phase, in such a way that the dataset loaded to Azure is the dataset with the new features.



7 Results

The next two tables indicate the performance of each model (as well as blend procedure) on training and test sets, remembering that the evaluation is made on the basis of accuracy:

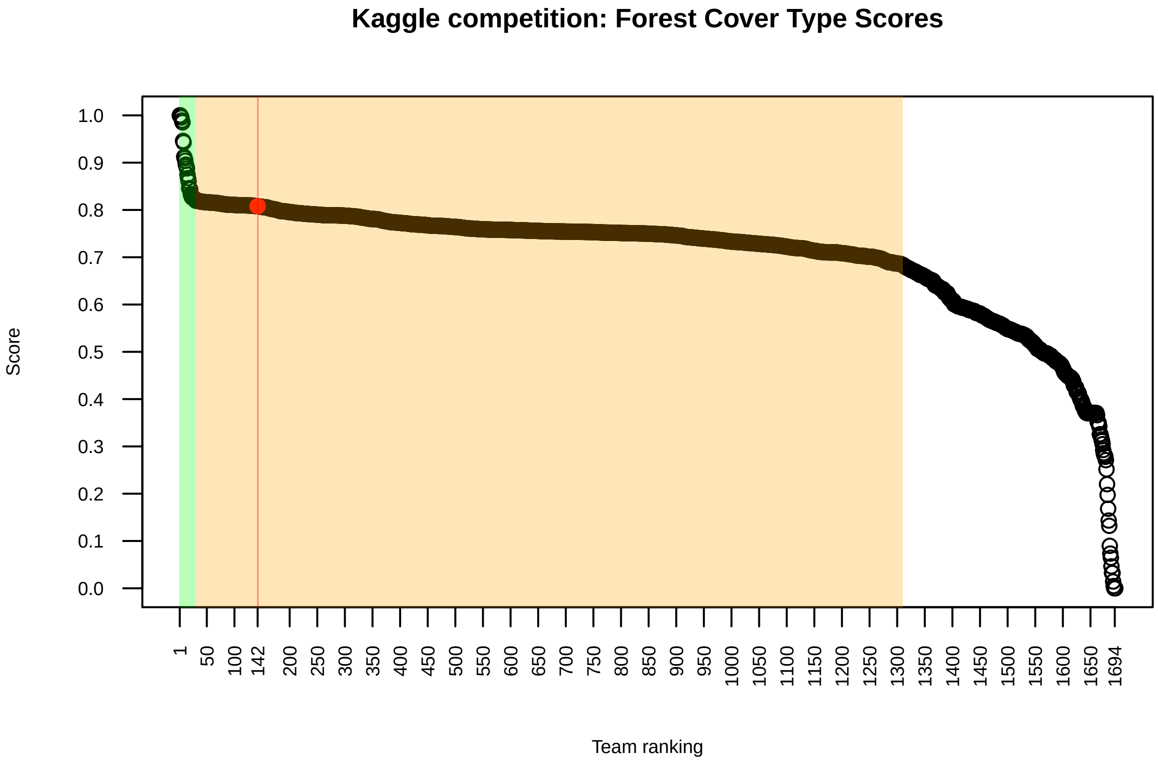
Table 4: Models accuracy on training set

Position	Team name	Accuracy
1	Boosted tree	89.25%
2	Blend	89.12%
3	One-vs-all Decision Forest	88.57%
4	Multi-class Decision Forest	87.96%
5	Neural networks	86.08%

Table 5: Models accuracy on test set

Position	Team name	Accuracy
1	Blend	80.80%
2	One-vs-all Decision Forest	80.18%
3	Boosted tree	80.01%
4	Multi-class Decision Forest	79.27%
5	Neural networks	74.30%

The best result obtained on test set (80.80%), would give this approach the position nº 142 on Kaggle leaderboard (in a total of 1694 teams participating), as illustrated by the figure below:



8 Conclusion

For the Forest Cover Type prediction task, models based on trees/forests had a good accuracy, achieving 80% of corrected predictions on test set. Possibly, as data are descriptive and present some relation between attributes, tree-based models performed well. They are efficient in computation and memory usage during training and prediction. Tree-based models run approximatelly 4-5 times faster then the neural network with 2 hidden layers. For future work, the investigation of convolution neural networks and a automated mechanism to generate better features is addressed.

9 References

[1] Blackard, Jock A. 1998. **Comparison of Neural Networks and Discriminant Analysis in Predicting Forest Cover Types**. Ph.D. dissertation. Department of Forest Sciences. Colorado State University. Fort Collins, Colorado. 165 pages.

[2] The Forest Cover Type dataset. Donors of database: Blackard, J., Dean, D., Anderson, C.

[3] Crain, K., Davis, G. **Classifying Forest Cover Type using Cartographic Features**. Stanford University, December 2014.

[4] Blackard, Jock A., Dean, Denis J. **Comparative accuracies of artifical neural networks and discriminant analysis in predicting foorest cover types from cartographic variables**. Computers and Electronics in Agriculture, 1999.

[5] Microsoft Azure Documentation. **Multiclass Decision Forest**.

[6] Safavian, S., Landgrebe, D. **A Survey of Decision Tree Classifier Methodology**. IEEE Transactions on Systems, Man, and Cybernetics. May, 1991.

[7] Roe, B., Yang, H., Zhu, J. **Boosted decision trees, a powerful event classifier**.

[8] Microsoft Azure Documentation. **Multiclass Neural network**.