Machine Learning Engineer Nanodegree

Capstone Proposal: Supervised Learning Applied to Kaggle Forest Cover Type Data Set

Jonathan J. Hull

June 14, 2017 (2nd re-submission after reviewer comments)

Domain Background

Kaggle competitions provide an ideal demonstration opportunity for the skills required of a machine learning engineer. In the Forest Cover Type competition, "you are asked to predict the forest cover type (the predominant kind of tree cover (7 classes are used)) from strictly cartographic variables (as opposed to remotely sensed data). The actual forest cover type for a given 30 x 30 meter cell was determined from US Forest Service (USFS) Region 2 Resource Information System data. Independent variables were then derived from data obtained from the US Geological Survey and USFS. The data is in raw form (not scaled) and contains binary columns of data for qualitative independent variables such as wilderness areas and soil type." [https://www.kaggle.com/c/forest-cover-type-prediction] This data has been used for comparing the accuracy of neural network and discriminant analysis classifiers [1].

Problem Statement

The problem is to "predict the forest cover type (the predominant kind of tree cover) from strictly cartographic variables (as opposed to remotely sensed data). The actual forest cover type for a given 30 x 30 meter cell was determined from US Forest Service (USFS) Region 2 Resource Information System data. Independent variables were then derived from data obtained from the US Geological Survey and USFS. The data is in raw form (not scaled) and contains binary columns of data for qualitative independent variables such as wilderness areas and soil type.

"This study area includes four wilderness areas located in the Roosevelt National Forest of northern Colorado. 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." [https://www.kaggle.com/c/forest-cover-type-prediction]

The inputs to our algorithm will be the 54 numeric features described below and the output (as this is a classification task) will be one of seven possible class labels that describe the predominant kind of tree cover in the 30x30 meter cell from which the corresponding features were extracted. We will model this as a supervised machine learning classification problem. We will apply at least three different machine learning algorithms and tune up their performance using the techniques learned during the course.

One technique we plan to use is a random forest classifier. It has been applied to many domains, is easily trained, and it's wicked fast in making predictions. Random forests worked reasonably well in the benchmark task (see the attached html file) and we expect that their performance will improve during this proposed project.

Datasets and Inputs

The Kaggle version of the Forest Cover data set will be used. "The training set (15120 observations) contains both features and class labels (Cover_Type). The test set contains only the features." [https://www.kaggle.com/c/forest-cover-type-prediction/data]

The 54 **features** associated with each observation follow. Besides basic geographical features, the remainder is from either the domain of forestry or geology.

1. Elevation - Elevation in meters
2. Aspect - Aspect in degrees azimuth
3. Slope - Slope in degrees
4. Horizontal_Distance_To_Hydrology - Horz Dist to nearest surface water features
5. Vertical_Distance_To_Hydrology - Vert Dist to nearest surface water features
6. Horizontal_Distance_To_Roadways - Horz Dist to nearest roadway
7. Hillshade_9am (0 to 255 index) - Hillshade index at 9am, summer solstice
8. Hillshade_Noon (0 to 255 index) - Hillshade index at noon, summer solstice
9. Hillshade_3pm (0 to 255 index) - Hillshade index at 3pm, summer solstice
10. Horizontal_Distance_To_Fire_Points - Horz Dist to nearest wildfire ignition points

Four different wilderness areas (binary columns):

11. Rawah Wilderness Area
12. Neota Wilderness Area
13. Comanche Peak Wilderness Area
14. Cache la Poudre Wilderness Area

40 different soil types (binary columns) and what they represent:

15. 1 Cathedral family - Rock outcrop complex, extremely stony.
16. 2 Vanet - Ratake families complex, very stony.
17. 3 Haploborolis - Rock outcrop complex, rubbly.
18. 4 Ratake family - Rock outcrop complex, rubbly.
19. 5 Vanet family - Rock outcrop complex complex, rubbly.
20. 6 Vanet - Wetmore families - Rock outcrop complex, stony.
21. 7 Gothic family.
22. 8 Supervisor - Limber families complex.
23. 9 Troutville family, very stony.
24. 10 Bullwark - Catamount families - Rock outcrop complex, rubbly.
25. 11 Bullwark - Catamount families - Rock land complex, rubbly.
26. 12 Legault family - Rock land complex, stony.
27. 13 Catamount family - Rock land - Bullwark family complex, rubbly.
28. 14 Pachic Argiborolis - Aquolis complex.
29. 15 unspecified in the USFS Soil and ELU Survey.

30. 16 Cryaquolis - Cryoborolis complex.
31. 17 Gateview family - Cryaquolis complex.
32. 18 Rogert family, very stony.
33. 19 Typic Cryaquolis - Borohemists complex.
34. 20 Typic Cryaquepts - Typic Cryaquolls complex.
35. 21 Typic Cryaquolls - Leighcan family, till substratum complex.
36. 22 Leighcan family, till substratum, extremely bouldery.
37. 23 Leighcan family, till substratum - Typic Cryaquolls complex.
38. 24 Leighcan family, extremely stony.
39. 25 Leighcan family, warm, extremely stony.
40. 26 Granile - Catamount families complex, very stony.
41. 27 Leighcan family, warm - Rock outcrop complex, extremely stony.
42. 28 Leighcan family - Rock outcrop complex, extremely stony.
43. 29 Como - Legault families complex, extremely stony.
44. 30 Como family - Rock land - Legault family complex, extremely stony.
45. 31 Leighcan - Catamount families complex, extremely stony.
46. 32 Catamount family - Rock outcrop - Leighcan family complex, extremely stony.
47. 33 Leighcan - Catamount families - Rock outcrop complex, extremely stony.
48. 34 Cryorthents - Rock land complex, extremely stony.
49. 35 Cryumbrepts - Rock outcrop - Cryaquepts complex.
50. 36 Bross family - Rock land - Cryumbrepts complex, extremely stony.
51. 37 Rock outcrop - Cryumbrepts - Cryorthents complex, extremely stony.
52. 38 Leighcan - Moran families - Cryaquolls complex, extremely stony.
53. 39 Moran family - Cryorthents - Leighcan family complex, extremely stony.
54. 40 Moran family - Cryorthents - Rock land complex, extremely stony.

The **class label** is the Cover_Type. It can be one of the following seven different values:

class label	represents	class label	represents	class label	represents
1	Spruce/Fir	4	Cottonwood/Willow	7	Krummholz
2	Lodgepole Pine	5	Aspen		
3	Ponderosa Pine	6	Douglas Fir		

A sample of a **feature vector** follows (read left to right, feature name: feature value):

Elevation	2596	Aspect	51	Slope	3
Horizontal_Distance_To_Hydrology		258	Vertical_Distance_To_Hydrology		0
Horizontal_Distance_To_Roadways		510	Horizontal_Distance_To_Fire_Points		6279
Hillshade_9am		221	Hillshade_Noon		232
Hillshade_3pm		148			

Wilderness_Area1	1	Wilderness_Area2	0	Wilderness_Area3	0	Wilderness_Area4	0
Soil_Type1	0	Soil_Type2	0	Soil_Type3	0	Soil_Type4	0
Soil_Type5	0	Soil_Type6	0	Soil_Type7	0	Soil_Type8	0
Soil_Type9	0	Soil_Type10	0	Soil_Type11	0	Soil_Type12	0
Soil_Type13	0	Soil_Type14	0	Soil_Type15	0	Soil_Type16	0
Soil_Type17	0	Soil_Type18	0	Soil_Type19	0	Soil_Type20	0
Soil_Type21	0	Soil_Type22	0	Soil_Type23	0	Soil_Type24	0
Soil_Type25	0	Soil_Type26	0	Soil_Type27	0	Soil_Type28	0
Soil_Type29	1	Soil_Type30	0	Soil_Type31	0	Soil_Type32	0
Soil_Type33	0	Soil_Type34	0	Soil_Type35	0	Soil_Type36	0
Soil_Type37	0	Soil_Type38	0	Soil_Type39	0	Soil_Type40	0
Cover_Type	5						

Solution Statement

At least three machine classifiers will be developed and applied to the Forest Cover data set, bound by the rules of the Kaggle competition. The forest cover training data set will be partitioned into training and validation sets (80% and 20%) and the used to develop the classifiers. Starting from a baseline implementation provided by a benchmark model, the performance of our solution will be improved by judicious application of the skills required by a well-trained machine learning engineer. These will include feature visualization, feature correlation, data transformation (as appropriate), normalization, preprocessing, feature evaluation, a training and testing pipeline, model tuning, feature importance and subset selection, and ensemble selection. As substantial improvements are achieved, the performance on the kaggle test set will be determined by submitting my results to the kaggle leaderboard.

The primary objective will be to illustrate that classification performance can be substantially improved above the baseline. A secondary objective will be to demonstrate the techniques that were used in a way that can be shared on my github site as part of my machine learning engineer portfolio.

Benchmark Model

The benchmark model is described in JonathanHull_capstone_proposal_benchmark_061417.html (copy attached). A number of supervised classifiers (DecisionTree, LogisticRegression, SGD, Bagging, AdaBoost, Gradient Boosting and Random Forest) were applied to the kaggle training set. The Random Forest classifier provided nearly the highest accuracy (82.011%). It was applied to the kaggle test set and a leaderboard score of 0.70538 was obtained (see Fig. 1).

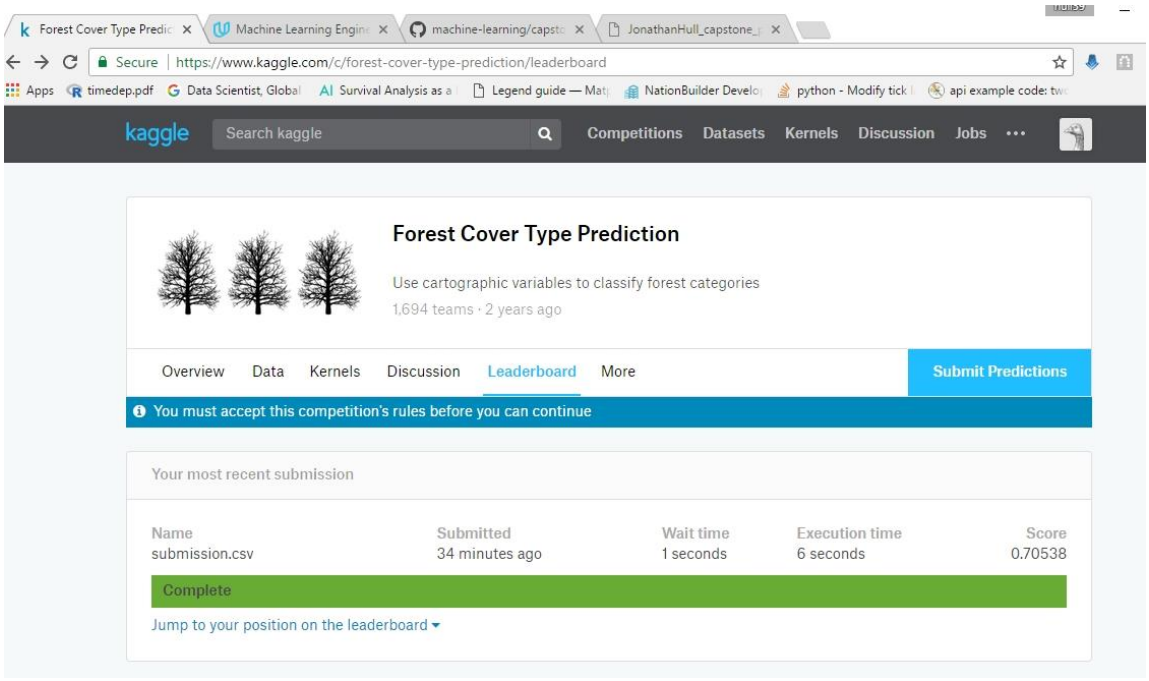

Fig. 1. Benchmark performance of random forest

Evaluation Metric

We will quantify our performance with our *accuracy* on the kaggle test set, which is the same as the kaggle leaderboard score (0.70538 in Fig. 1 is equal to 70.538% accuracy) and compare it to the value that was obtained while preparing this proposal (0.70538, as shown in Fig. 1). Accuracy is calculated as a percentage as 100.0 * num_correct / num_testing_samples where num_correct is the number of samples in the test set that are correctly classified and num_testing_samples is the number of samples in the test set. A sample is correctly classified if the decision of the classifier is the same as the sample's classification.

We will abide by the kaggle rules throughout the project and will identify the individual contributions of the techniques that were used as we developed our implementation.

Project Design

The project workflow will include the following steps.

1. Data exploration – confirm basic understanding of the composition of the data set.

2. Feature visualization – visualize distributions of categorical and numeric features.

3. Feature correlation – calculate correlations between features and determine redundancies.

4. Data transformation – based on the results of feature visualization, transform the features (e.g., log transformation), as appropriate.

5. Feature normalization – scale numeric features if appropriate.

6. Feature preprocessing – apply one-hot and dummy variable encoding to categorical features.

7. Feature evaluation – calculate Pearson correlations between features to give us some insight into the features that will be most useful for classifying an unknown sample and the type of classifier that's appropriate for the data set. We will investigate the use of SelectKBest() from sklearn for this task.

8. Training and testing pipeline -- create a training and testing pipeline that allows us to quickly train models using various sizes of training data and perform predictions on the testing data.

9. Model tuning – use GridSearchCV to tune the chosen classifiers to improve their performance.

10. Feature importance and subset selection – use the feature importance value from one of the best classifiers to select a subset of features, train the classifiers on the subset and compare that to the best case results. The objective will be to see if we can use fewer features with a minimal degradation in performance.

11. Ensemble evaluation – we will evaluate the possibility of combining independently developed classifiers using various techniques such as majority vote, etc. Following the reviewer's suggestion (thanks very much!!!), we will consult the Kaggle Ensemblig Guide [https://mlwave.com/kaggle-ensembling-guide/] for suggestions about how to perform this step. One ensembling method we will consider is stacking. In this case we will save all of our model outputs and use them to train a "combiner" (e.g., logistic regression).

Conclusions

This project will illustrate the skills required of a machine learning engineer. Starting from a data set and classifiers that provide marginal performance, we expect to achieve substantial improvement in accuracy by application of appropriate feature engineering and model tuning techniques. The result will be a github site and jupyter notebook that can be shared with potential employers and used as a starting point for technical discussions during interviews.

References

1. J.A. Blackard and D.J. Dean , “Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables,” Computers and Electronics in Agriculture, vol.24, pages 131-151, 1999.