Machine Learning Engineering Nanodegree

Capstone Proposal

Juzer Shakir

August 23rd, 2018

Proposal:

Forest Cover-type Data

Use cartographic variable to classify Forest Categories.

Supervised Learning, Classification.

Domain Background:

Given elevation, hydrologic, soil, and sunlight data can we predict what type of tree would be in a small patch of forest? This project attempts to predict the predominant type of tree in sections of wooded area. 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 and de/reforestation concerns. Forest cover type data is often collected by hand or computed using remote sensing techniques, e.g. satellite imagery. Such processes are both time and resource intensive. In this project, we aim to predict forest cover type using cartographic data and a variety of classification algorithms.

This study area includes 4 Wilderness Areas located in the *Roosevelt National Forest* of *Northern Colorado.* These area represent forests with minimal human-caused disturbances, so that existing forest cover types are more a result of ecological process rather than forest management practices.

Each observation is 30m x 30m forest cover type determined from US Forest Service (USFS) Region 2 Resource Information System (RIS) data. Independent variables were derived from the data originally obtained from US Geological Survey (USGS) and USFS data.

Dataset Link: Download

Downloaded From UCI: Visit Site

Research Paper Link: Visit Site

Original Owners of Database:

Jock A. Blackard and Dr. Denis J. Dean.

Remote Sensing and GIS Program

Department of Forest Sciences

College of Natural Resources

Colorado State University

Fort Collins, CO 80523

Donors of database:

1. Jock A. Blackard Email
2. Dr. Denis J. Dean Email
3. Dr. Charles W. Anderson Email

Relevant Papers:

- Blackard, Jock A. and Denis J. Dean. 2000. "Comparative Accuracies of Artificial Neural Networks and Discriminant Analysis in Predicting Forest Cover Types from Cartographic Variables." Computers and Electronics in Agriculture 24(3):131-151. Visit Site

- Blackard, Jock A. and Denis J. Dean. 1998. "Comparative Accuracies of Neural Networks and Discriminant Analysis in Predicting Forest Cover Types from Cartographic Variables." Second Southern Forestry GIS Conference. University of Georgia. Athens, GA. Pages 189-199.

- 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.

Papers that cite this dataset: Visit Site

Problem Statement:

We have been given a total of 54 attributes, these attributes contain Binary and Quantative attributes, and we need to predict which *Forest Cover-Type* is it from the given features.

Dataset and Inputs:

This dataset has been taken from UCI Machine Learning Repository. (Link above).

Dataset Info*:*

No. of Instances	581,012
No. of Attributes (Features)[1]	54
Associate Task	Classification
Dataset Characteristic	Multivariate
Attribute Characteristic	Categorical, Integer
Missing Value	None
Area	Life
Target Variable	Forest Cover Type

Our Dataset consists of more than half a million observation and has 54 attributes or features to help us predict type of Forest Cover which is our target or output variable. The target variable has 7 different classes hence making this a Multi-Class Classification problem.

[1] No. of Attributes consists of:

10 Quantative variable, 4 Binary Variable (*Wilderness Area*) and other 40 Binary Variable (*Soil Type*). Which makes a total of 54 Variable/Features/Attributes.

Attribute Info:

Given is the attribute name, attribute type, the measurement unit and a brief description.

Name	Data Type	Measurement	Description
Elevation	Quantative	Meters	Elevation in Meters
Aspect		Azimuth	Aspect in degrees Azimuth
Slope		Degrees	Slope in Degrees
Horizontal_distance_to_Hydrology		Meters	Horz distance to nearest surface
Vertical_distance_to_Hydrology			Vert distance to nearest surface
Horizontal_distance_to_Roadways			Horz distance to nearest roadway
Hillshade_9am		0 to 255 index	Hill shade index at 9am
Hillshade_Noon			Hill shade index at Noon
Hillshade_3pm			Hill shade index at 3pm
Horizontal_distance_to_Fire_Points		Meters	Horz distance to nearest Fire Point
Wilderness_Area (x4)	Binary	0-Absence or 1-Present	Wilderness area Designation
Soli_Type (x40)			Soil Type area Designation
Cover_Type	Integer	1 to 7	Forest Cover Type

Data is in raw from (not scaled).

As we can see from the above table, the *wilderness area* has 4 columns, and these columns are binary meaning it can only be present one for each observation. Here we have 4 types of Wilderness area, and it can have only one of these in each observation/instances that are given to us. Same goes for *Soil Type* feature. These are what we call in Machine Learning are '*one-hot encoded*' features for us.

We will explore this *Wilderness areas* in detail:

Wilderness_Area1	Rawah Wilderness Area
Wilderness_Area2	Neota Wilderness Area
Wilderness_Area3	Comanche Wilderness Area
Wilderness_Area4	Cache La Poudre Wilderness Area

Above is the table, on the left side of it are the column names in the dataset which gives us the information of wilderness and to the right of it are the names of the wilderness areas.

So for example if you have an observation that has a positive value '*1*' under the *Wilderness_Area2*, then that observation is taken from *Neota Wilderness Area*.

Some Background Information for these 4 *Wilderness Area*:

Neota probably has the highest mean elevation values of the 4 Wilderness Areas. *Rawah* and *Comanche* would have a lower mean elevation value, while *Cache la Poudre* would have the lowest mean elevation value. (More on Page 9)

Now let's explore *Soil Type* Feature:

Soil Type feature has 40 columns of it, meaning there are 40 types of Soils collected from 4 Wilderness Areas in the *Roosevelt National Forest.*

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, extremely bouldery
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

Forest Cover Type (The variable for our prediction):

This is the variable which we are going to predict and it has only one column which represents integer values from 1 to 7, where these digits represent type of forest cover for the observations. This is the variable which is not *one-hot encoded* like *Soil Type* and *Wilderness Area*, that's why it doesn't have 7 columns to represent each class for the observations.

Now let's look at the names of these types of Forest Cover:

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

So an observation which has '*5*' integer value in a *Forest Cover Type* column, it means that observation has *Aspen Forest Cover Type* for that observation.

After taking a look at this, More Background Information for 4 *Wilderness Areas:*

As for primary major tree species in these areas, *Neota* would have *Spruce/Fir* (type 1), while *Rawah* and *Comanche* would probably have *Lodgepole Pine* (type 2) as their primary species, followed by *Spruce/Fir*

and *Aspen* (type 5). *Cache la Poudre* would tend to have *Ponderosa Pine* (type 3), *Douglas-fir* (type 6), and *Cottonwood/Willow* (type 4).

The *Rawah* and *Comanche* areas would tend to be more typical of the overall dataset than either the *Neota* or *Cache la Poudre*, due to their assortment of tree species and range of predictive variable values (elevation, etc.) *Cache la Poudre* would probably be more unique than the others, due to its relatively low elevation range and species composition.

Now let's have a look at the Class Distribution of Forest Type in this Huge Dataset:

No. of records of Spruce / Fir	211 840
No. of records of Lodgepole Pine	283 301
No. of records of Ponderosa Pine	35 754
No. of records of Cottonwood / Pillow	2 747
No. of records of Aspen	9 493
No. of records of Douglas-Fir	17 367
No. of records of Krummholz	20 510
Total Records	581 012

The distribution of these classes is not equal. Spruce and Lodgepole have the most while Cottonwood has the least records. Having such distribution will not give us appropriate results because of unequal amount of distribution. We will take more detail look and come up with a solution in the project. Also to note here that every observation is

assigned to some class of forest type and no observation is empty with that info, so this is good for us.

Basic Summary Statistics:

Now let's see what the statistics has to say for the features we have been provided: (*Thanks to Phil Rennert for the Summary Values*)

Feature Name	Mean	Std dev
Elevation	2959.36	279.98
Aspect	155.65	111.91
Slope	14.10	7.49
Horizontal_distance_to_Hydrology	269.43	212.55
Vertical_distance_to_Hydrology	46.42	58.30
Horizontal_distance_to_Roadways	2350.15	1559.25
Hillshade_9am	212.15	26.77
Hillshade_Noon	223.32	19.77
Hillshade_3pm	142.53	38.27
Horizontal_distance_to_Fire_Points	1980.29	1324.19

Solution Statement:

Having 54 features including 44 Boolean feature values and having half a million dataset, training this on most models would take a lot of time and also would over-fit the data.

Shrinking the data would also give us good performance and avoid overfitting it.

What I would like to do is remove the Boolean features and first train only on the remaining 10 features and then second train on all of the 54 features. By Comparing scores and runtime (not much important) we will choose whether or not we need keep Boolean features or not. If our score on second training data is less or quite equal to the first training score this means those features don't quite give any additional info to predict and perform well so we then remove the Boolean features.

Having less features would yes decrease our training time but does it give better performance than keeping all features? We will find this out in the project.

Models that I would like to choose for this dataset which I think would be interesting to look results at are:

1. Random Forest Classifier (RFC)
2. Stochastic Gradient Descent (SGD) Classifier
3. Support Vector Machine (SVM)
4. K-Nearest Neighbor

Out of these 4 models, RFC will give us better results than any but it would be tough call against SGD too, since both these models can perform very well handling high dimensional data. SGD requires lots of hyper-parameter tuning given its flexibility and also sensitive to feature scaling. All models have its pros and cons, with all that said my final solution model will be **Random Forest.**

As we had seen in the *Dataset and Input* section, the number of observation for each class is unequal and vary by very large numbers. *Fir* and *Lodgepole* have the highest number of observation over 200k+ while *Cottonwood* with least of just 2k+ followed by *Aspen* 9k+. Such variation can cause the algorithm to perform poor on the classes with less observation and perform really well on classes with good amount of observation. Hence, affecting the accuracy score.

To avoid this, we use boosting algorithm. The boosting algorithm converts weak learners to strong learners. More deep information related to this can be found here.

I have chosen **AdaBoost (Adaptive Boosting)** as my boosting algorithm. I will be using AdaBoost and set base estimator to all the model I chose above.

I will be first applying models with AdaBoost and then I will apply AdaBoost to each model. Doing this we will get to see how much improvement the models make in terms of accuracy.

Benchmark Model:

Let's discuss the model used that were used by the authors of this dataset:

They implemented two models for this study, a feedforward neural network and a statistical model based on discriminate analysis. The overall objectives of this research were to first construct these two predictive models, and second to compare and evaluate their respective classification accuracies predicting forest cover type from given features.

They had used all the features given in the data, 12, as independent variable to the model and *Forest Cover Types* as dependent variable. Several subsets of these variables were examined to determine the best overall predictive model.

For each subset of cartographic variables examined in this study, relative classification accuracies indicate the neural network approach outperformed the traditional discriminant analysis method in predicting forest cover types. The final neural network model had a higher absolute classification accuracy (70.58%) than the final corresponding linear discriminant analysis model (58.38%). In support of these classification results, thirty additional networks with randomly selected initial weights were derived. From these networks, the overall mean absolute classification accuracy for the neural network method was 70.52%, with a 95% confidence interval of 70.26% to 70.80%.

- First 11,340 records were used for training data.
- Next 3,780 records were used for validation.
- Last 565,892 records used for testing data.
- Final performance by Neural Network was 70%.
- Final performance by Linear Discriminate Analysis was 58%.

Discussing My Benchmark Model:

The Benchmark model I will be choosing for this particular dataset is Naïve Bayes.

Naïve Bayes are algorithm based on Bayes theorem with the 'naïve' assumption of independence between every pair of features. They work quite well in real-world situation such as spam filtering and document classification, work extremely fast and also require small amount of training data.

Naïve Bayes is a decent classifier but it is known to be bad estimator and the probability outputs are not taken too seriously.

All that said, I think Naïve Bayes would give approximately 50-65% accuracy score and that's my assumption.

Evaluation Metric:

The evaluation metric I would like to choose is Accuracy.

Accuracy mean how many data points/observations are predicted correctly out of all number of observation.

In a Confusion Matrix point of View Accuracy is calculated by:

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

Where,

TP	True Positives
TN	True Negatives
FP	False Positive
FN	False Negative

True Positives and True Negatives are the ones which the classifier correctly predicts the class of an observation.

Let's say that we have an observation where the actual class is 'A' and the class predicted by the classifier is also 'A', this is called as True Positives, where 'True', in this context, stands for the classifier predicting the right class and 'Positive' stands for the classifier predicting it as Positive 'A' class.

We have other observation where the actual class is not 'A' and class predicted by the classifier is also not 'A', this is called as True Negatives, because the predicted class is equal to actual class (True) and 'Negative' because classifier predicts as not 'A'.

Same idea goes for FP and FN.

So adding TP and TN gives us all the correct values predicted by the classifier and dividing by the sum all possible states is the same as dividing by total number of observations. This gives us an output value ranging

from 0 to 1 where 0 means the classifier has 0% accuracy meaning it has classified all classes wrong and 1 means the classifier has 100% accuracy classifying all classes correctly.

Given the data, accuracy is what we want to know, to know how well it classifies all the classes since we have uneven amount of observation in each class.

Project Design:

First I will extract all the required and statistical information that we can get from it such as:

- Shape of the dataset
- Datatype of each feature
- Min and Max Value of each feature
- Percentile of each feature
- Mean and Median value of each feature
- Class Distribution

(I will show visualizations of these wherever necessary.)

- Visualizing Correlation between features.
- Visualizing highly correlated features with scatter plot.
- Visualizing every feature with violin plot against Cover Type.
- Visualizing Forest Class distinction in each wilderness area.
- Visualizing Forest Class distinction in each Soil Type area.

- Then, I will do Data Cleaning process, in which I will find any missing values if present in the dataset (Information given in UCI states that there are no missing values but still I would like to crosscheck it) and delete that observation.

- I will also look for if there're any duplicate copies of an observation and delete it since we do not want multiple entries of a single observation being fed to an algorithm.

- Categorical features such as Soil and Wilderness have a value of either 0 or 1 for each observation, there might be a case where some features might have 0 value throughout the data which means that no observation has that feature which means we should get rid of that feature.

- Since our data has value ranging from 4000 to negative values and has 44 boolean features we will apply feature scaling to it so that we have a specific range of values between 0 to 1 for all features. I will be using MinMaxScalar().

- Split the data for train-test split as 75%-25% respectively.

- Run the benchmark model Naïve Bayes on train set and see its performance.

- Will use 10 K-Fold CV in train set to train the models.

- Want to train on two types of data, first will all features (X1), second with removing all categorical features (X2). (Soil and Wilderness)

- Train the SVM, RF, SGDC and KNN on train set as k-fold CV on X1 data.

- Train the SVM, RF, SGDC and KNN on train set as k-fold CV on X2 data.

- Compare X1 and X2 accuracy result. Then choosing the best model with highest accuracy and also the data it trained on.

- Testing the model on the data (X1 or X2). If X2 is the dataset we chose then I will remove categorical features from the 25% test set that we had split.

- Evaluating final results.