Machine Learning Engineer Nanodegree

Capstone Proposal

Bin luo
December 25th, 2016

Proposal

Domain Background

The United States Forest Service (USFS) is an agency of the U.S. Department of Agriculture that administers the nation's 154 national forests and 20 national grasslands, which encompass 193 million acres (780,000 km^2). Major divisions of the agency include the National Forest System, State and Private Forestry, Business Operations, and the Research and Development branch. Managing approximately 25% of federal lands, it is the only major national land agency that is outside the U.S. Department of the Interior.

In our project, we used the data/features was derived from the US Geological Survey and USFS, the forest types, our the targeted predication results, was determined by USFS.

Problem Statement

The problem this project is trying to solve 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 study area where the data derived from includes four wilderness areas located in the Roosevelt National Forest of northern Colorado. Each observation is a 30m x 30m patch. Our goal is to predict an integer classification for 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

This is a problem of supervised learning – and more specifically multi-categories classification problem, there are many methods could be explored, will be discussed in the below section. Being able to explore different algorithms is the major motivation I selected this competition as this will help me to establish a systematic view of the supervised learning techniques, and get some hands-on experience on the algorithms I was not able to try out during the courses.

Datasets and Inputs

In this study, we are using the publicly available dataset downloaded from Kaggle.

The training set (15120 observations) contains both features and the Cover_Type. The test set contains only the features. We must predict the Cover_Type for every row in the test set (565892 observations).

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. Below are the details about the metadata.

Data Fields:

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
11) **Wilderness_Area** (4 binary columns, 0 = absence or 1 = presence) - Wilderness area designation
12) **Soil_Type** (40 binary columns, 0 = absence or 1 = presence) - Soil Type designation
13) **Cover_Type** (7 types, integers 1 to 7) - Forest Cover Type designation

The wilderness areas are:

14) **Rawah Wilderness Area**
15) **Neota Wilderness Area**
16) **Comanche Peak Wilderness Area**
17) **Cache la Poudre Wilderness Area**

The soil types are:

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 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 - Typic Cryaquolls complex.
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.

Solution Statement

To tackle the problem described in Section 2, we will use supervised learning with feature selection to automatically learn evaluation functions. The data is pretty clean in terms of quality (e.g. no missing values), however, since many of the feature are not in same scale, so some data transformation/preprocessing need to be done before apply any algorithms. Also we will check the correlation among each of the features to see if there is opportunity to reduce features by using PCA (or other techniques) considering we have ~60 features. For the Evaluation, prediction model we will try different algorithms (linear, non-linear and bagging/boosting) and then chose the one with best performance based the evaluation metrics.

Benchmark Model

To benchmark our model, we will use Kaggle's benchmark model to check our predication accuracy. In The Public Leaderboard, the top 50^{th}'s performance is 0.81626, we will compares our final solution with the leaderboard to see where we stand. We will aim to have our final solution have an accuracy higher than 50^{th}.

Evaluation Metrics

The evaluation metric we will use for this competition is Mean F1-Score. The F1 score measures accuracy using the statistics precision p and recall r. Precision is the ratio of true positives (tp) to all predicted positives (tp + fp). Recall is the ratio of true positives to all actual positives (tp + fn).

The F1 metric weights recall and precision equally, and a good prediction algorithm will maximize both precision and recall simultaneously. Thus, moderately good performance on both will be favored over extremely good performance on one and poor performance on the other.

We will split the training set to be training and testing, will evaluate the performance of the algorithm we choose based on the F1 score of both training and testing set.

Project Design

Programming Language and Libraries

- Python 2.
- Scikit-learn. Open source machine learning library for Python.

Solution design summary

Step 1. Data preparation

1. Remove unnecessary columns by checking the statistic description of the data, e.g. if there are features with constants values, use *dataframe.describe()* function
2. Check if there is any skewness for any feature by using pandas *datafrme.skwe()* function
3. Use *MinMaxScaler()* function to standardize the non-categorical data

Step 2. Data exploration

1. Check if there is any correlation among the features by calculating the co-efficient, use *dataframe.corr()* function
2. If strong correlation observed among some feature pairs, will use PCA to reduce features.

Step 3. Feature Selection

1 We will explore the sk-learn feature selection algorithms to rank the features, the algorithms are planned be used are:
   a. Feature importance calculation by using ensembles of decision trees
      i. ExtraTreesClassifier
      ii. RandomForestClassifier
   b. Recursive feature elimination
   c. Univariate feature selection – SelectPercentile
2 Will rank the features based on the median of the performance/importance from all above algorithms and choose the final features.

Step 4 Training and Evaluating Models

1. Will evaluate the below predication model on training data (full set of feature and subset of feature based our feature selection from step 3)
   a. Linear
      i. LinearDiscriminatAnalysis
      ii. LogisticRegression
   b. Non-linear
      i. KNN
      ii. CART
      iii. SVM

    c. Bagging
        i. Random Forest
        ii. Extra Trees
    d. Neurons Network
    e. Boosting
        i. AdaBoost
        ii. Gradient Boosting
2. Evaluate the performance of above algorithm based on accuracy (F1 score) on training sets & testing set, also their training time will be another important consideration.
3. Select the best model and make predication on the testing set provided by Kaggle.

References

[1] http://scikit-learn.org/stable/modules/feature_selection.html

[2] https://en.wikipedia.org/wiki/Bootstrap_aggregating

[3] http://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html

[4] http://machinelearningmastery.com/feature-selection-in-python-with-scikit-learn/

[5] https://www.kaggle.com/c/bnp-paribas-cardif-claims-management/forums/t/19841/extra-trees-classifier-getting-worse-feature-selection

[6] http://machinelearningmastery.com/an-introduction-to-feature-selection/