KDD DATA PROJECT

Predicting ‘Exciting’ DonorsChoose.org projects

Abstract

Data from the KDD Cup 2014 Challenge is used to correlate “exciting” DonorsChoose.org projects to US states and school poverty levels. A score indicating how exciting a project is relative to the poverty level of a school is also calculated.

Ashish Naik

ashishsnaik@yahoo.com

Table of Contents

KDD Data Project

Summary

A total of 239 school projects are identified as among the Top 5 exciting projects in each of 51 US States, across 3 school poverty levels - highest, high, and moderate.

The correlation and ranking of the "exciting" projects is done based on the average number of underprivileged students that could benefit from the project. A project score was calculated as the ratio of 'number of underprivileged students served by an exciting project' to 'total number of underprivileged students served by all exciting projects in the state'. A project with a higher score, got a higher rank.

The ranking is done such, because it focuses on the (possible) number of underprivileged students reached, rather than merely considering the poverty levels of schools or the total number of students reached.

As an example, in the State of Iowa (IA), with only two top ranking projects, a project from a school with high poverty level is ranked higher than one from a school with highest poverty level. Similar can be observed for the State of New Hampshire (NH) where two projects from school(s) with moderate poverty level are ranked higher than one from a school with highest poverty level.

To predict whether a project is exciting or otherwise, an ensemble of 15 Generalized Boosted Regression Models (GBM) was used.

Please see the attached KDDTopProjects.csv file to view the correlation and ranking results.

Introduction

This data project uses data from the KDD Cup 2014 Challenge and attempts to correlate the predicted "exciting" DonorsChoose.org projects, in the test set, to US states and school poverty levels identifying the top 2-5 projects per state that would benefit poorer schools. It also attempts to compute a score indicating how exciting a project is relative to the poverty level of a school.

The KDD data sets contain information about: the projects itself (projects.csv), donations received by projects (donations.csv), project text posted by the teachers (essays.csv), the resources requested for a project (resources.csv), and outcomes of the projects - whether exciting or otherwise - (outcomes.csv). Projects posted on or after 2014-01-01 are in the test set and the data sets do not contain information about the donations to or outcomes of these projects.

Steps Used to Arrive at the Results

No exciting projects were reported before 2010-04-14. This could be because DonorsChoose.org tracked the projects differently or the criteria for exciting projects changed around that time. So, projects posted before April 2010 are not considered for analysis.

Projects posted in 2010 (April and later), 2011, and 2012 are used for model training and those posted in 2013 are used for model evaluation (validation and testing).

Data Cleaning and Transformations

Source Files

- kdd_clean_data.py
- kddData.R
- kddGBM.R (GBM specific transformations)

Performed cleaning of the 'text' features from donations.csv, essays.csv, and resources.csv using a python script (kdd_clean_data.py). This ensured that all records where read by the R code. Did a find-replace to remove '\r\n' character-tuple from the text in resources.csv, before running python script.

Handling Missing Values

- Boolean values were replaced by 'f' (or numeric zero for outcome data).
- Numeric values were replaced by column median value (or numeric zero for outcome data).
- Non-numeric values were replaced by '#NA#' value.

The missing values for School NCESID, Secondary Focus Subject, and Secondary Focus Area were given special treatment.

- School NCESID - missing values replaced by the numeric factor-level of the corresponding *schoolid*, in order to keep the ncesid same for a school and different across schools.
- Secondary Focus Subject - missing values replaced by Primary Focus Subject, as available.
- Secondary Focus Area - missing values replaced by Primary Focus Area, as available.

Specifically for GBM Model:

- Boolean ‘t’ (true) and ‘f’ (false) values were converted to numeric 1 and 0 respectively.
- Non-numeric features were replaced by their numeric factor-levels.

Feature extraction and selection

Source Files

- kddData.R
- kddGBM.R

The following features were extracted from the data.

Essay data

Word counts of the *title, short_description*, *need_statement*, and *essay* were calculated. Missing values received a word count of zero.

Resource Data

For each project the number of resources requested, number of vendors used, total item quantities, total cost of resources, and average cost per item were calculated.

Project Data

The number of underprivileged students that could possibly benefit from the project and the month and weekday the project was posted.

Assumption: Underprivileged students in a school are from the average poverty-level percentage within the school's *poverty_level* range. The students benefited *(students_reached)* from a project contain that much percent of students that are underprivileged.

School and Teacher Credibility (Project, Donation, and Outcome Data)

The 'exciting projects' criteria can be influenced (at least in part) by teachers' capabilities, for example, to gather at least one teacher-acquired donor. The criteria can also be influenced by schools' abilities to create an environment that promotes externally funded projects, attract project funding, and build a reputation. To accommodate for this, the following credibility features for both, schools and teachers, were extracted:

- number of projects posted
- number of projects that were fully funded
- number of projects that were exciting
- number of projects that had a great chat
- number of projects that received donations from a thoughtful donor
- average number of donations received
- average number of green donations received
- average number of teacher referred donors
- average number of donors that donated $100 plus
- average number of non-teacher referred donors that donated $100 plus

Features that showed near-zero variance in the training data were not considered for analysis.

In order to ease the modeling process, the target variable *is_exciting* was tagged with 'y_' and the features were tagged with 'x_'.

Modeling Process

Source Files

- kddGBM.R

Handling Class Imbalance

The training data showed about 12:1 class imbalance between the non-exciting and exciting projects. To handle this, an ensemble learning approach with EasyEnsemble Informed Under-Sampling [1] was used.

GBM Model

Model Training and Parameter Selection

Area under the ROC Curve (AUROC) was used as the performance metric.

A 3-way data split was used for model training, parameter selection, and performance estimation as mentioned below.

1. The data training data were split into training, validation, and test sets.
    - Training Set: Years 2010 (April and later), 2011, and 2012 data.
    - Validation and Test Sets: Year 2013 data. The data was randomly sampled into two equal and class-balanced sets for validation (parameter tuning) and testing (performance estimation).
    - Training set: 289360 samples
    - Validation set: 65665 balanced samples
    - Test set: 65664 balanced samples

2. 5-fold cross validation was used to tune parameters.

- Training set: 5 resamples of balanced training data, each with 22432 minor + 22432 random major samples.
- Validation set: 5 resamples of 60% of the validation set (39399 random samples each).
- Tuning parameters grid:
  - interaction.depth: 7, 8, 9
  - n.trees: 2000 to 4000 , with 500 step size
  - shrinkage: 0.1, 0.01
  - n.minobsinnode: 10, 50

The GBMParameterTuning.txt file in the submission shows the parameter tuning output. Below *Figure 1: GBM Parameter Tuning* shows the parameter tuning plot.

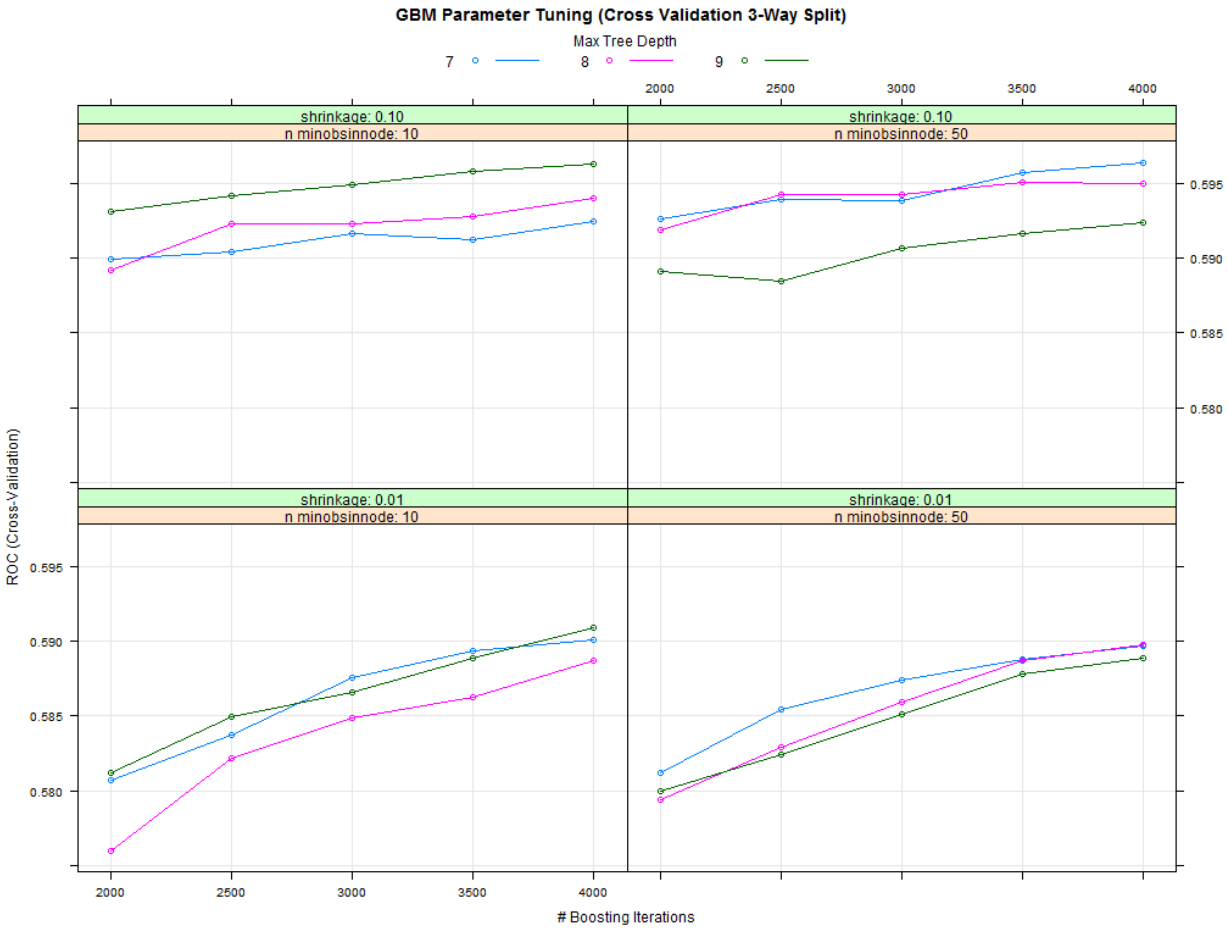

Figure 1: GBM Parameter Tuning

3. Performance of the final model was accessed on the test set using the below tuned parameters.
   - Final model parameters:
     - interaction.depth: 7
     - n.trees: 4000
     - shrinkage: 0.1
     - n.minobsinnode: 50

   The Test Set AUROC was 0.67.

Figure 2: GBM ROC Curve - Test-set below shows the ROC curve.

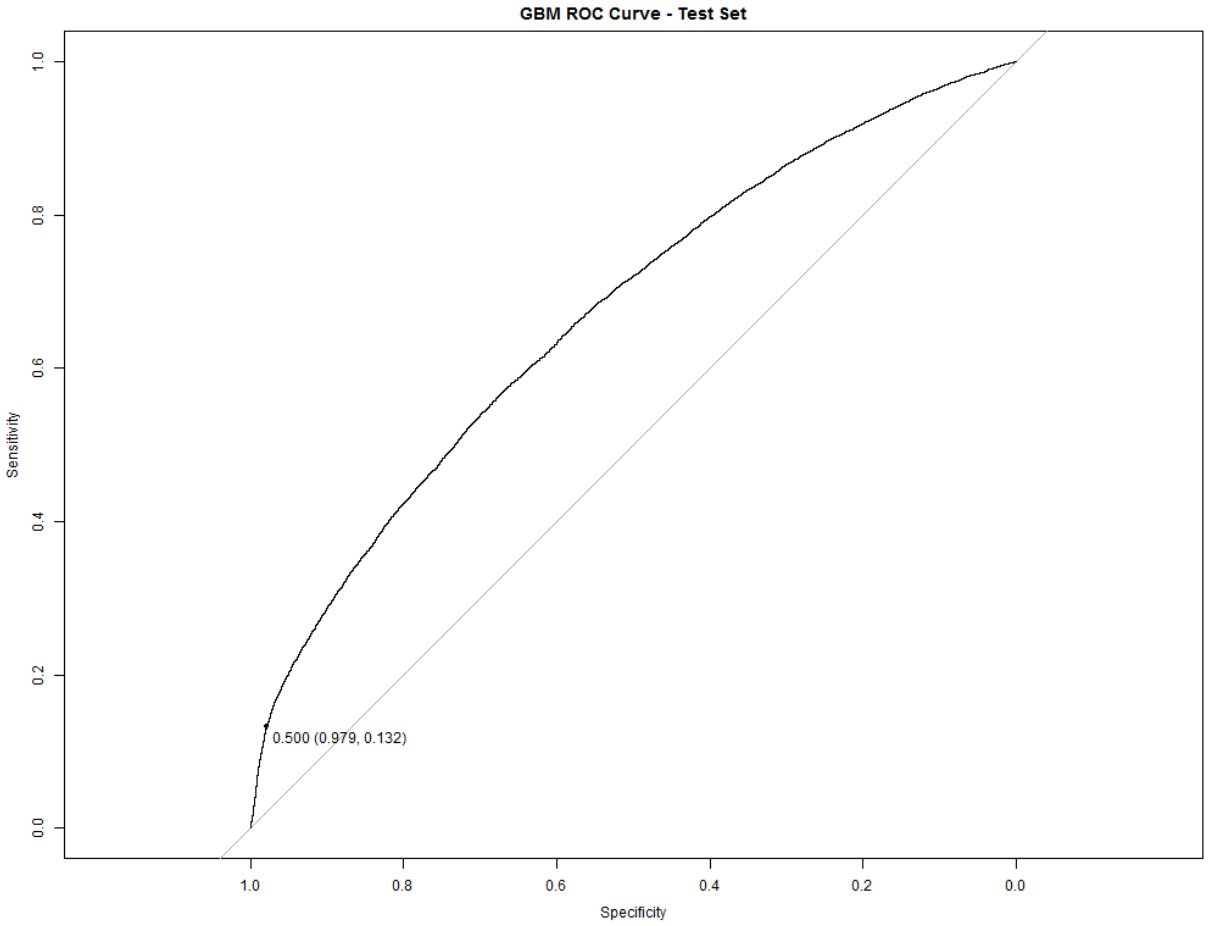

Figure 2: GBM ROC Curve - Test-set

Test Data Predictions

Source Files

- kddGBM.R
- kddMySql.R

Predictions on the test data were made using an ensemble of 15 GBM models trained on balanced training data samples with the tuned parameters. The kddMySql.R code was used to create a MySQL 'kddcupdb' database and populate the 'predictions' table.

The gbmPredictions.csv.gz file in the submission contains the test set prediction results.

Correlating and Ranking Exciting Projects

Source Files

- kddmysql.sql

The above MySQL script was used to create a 'view' of the exciting projects from the 'kddcupdb .predictions' table and generate the correlation and ranking.

Performance on Kaggle Leaderboard

'White Noise' is my Kaggle.com screen name.

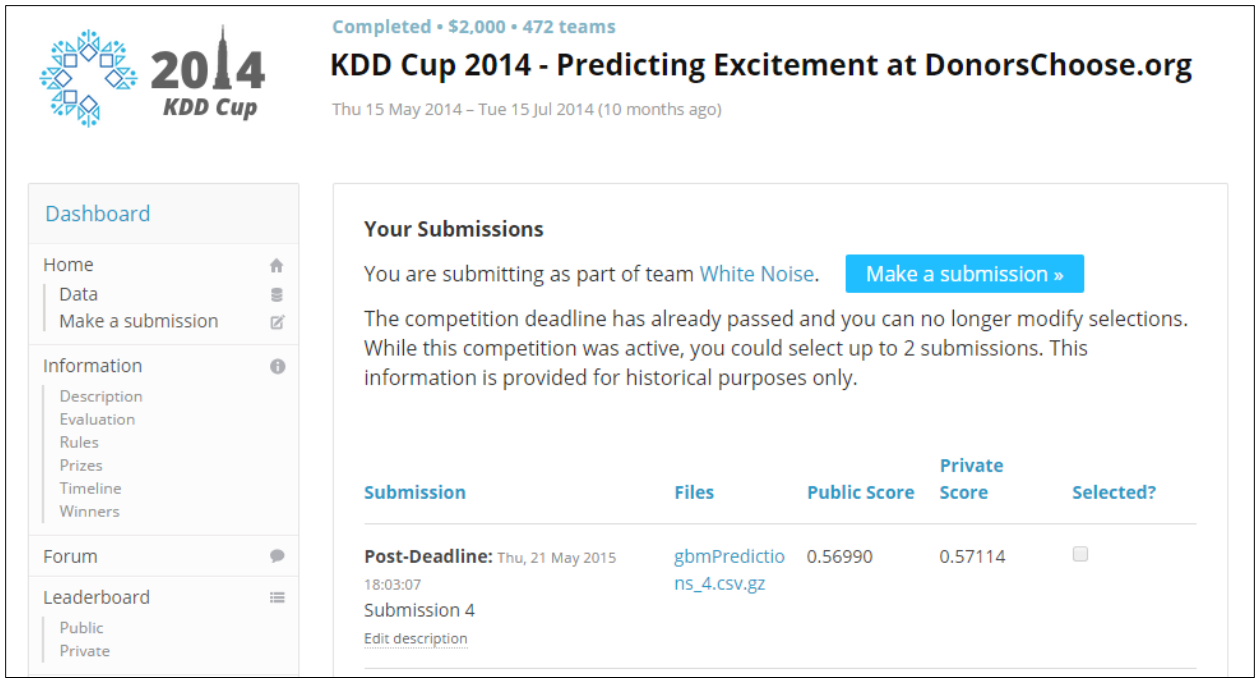

Figure 3: Kaggle public and private scores

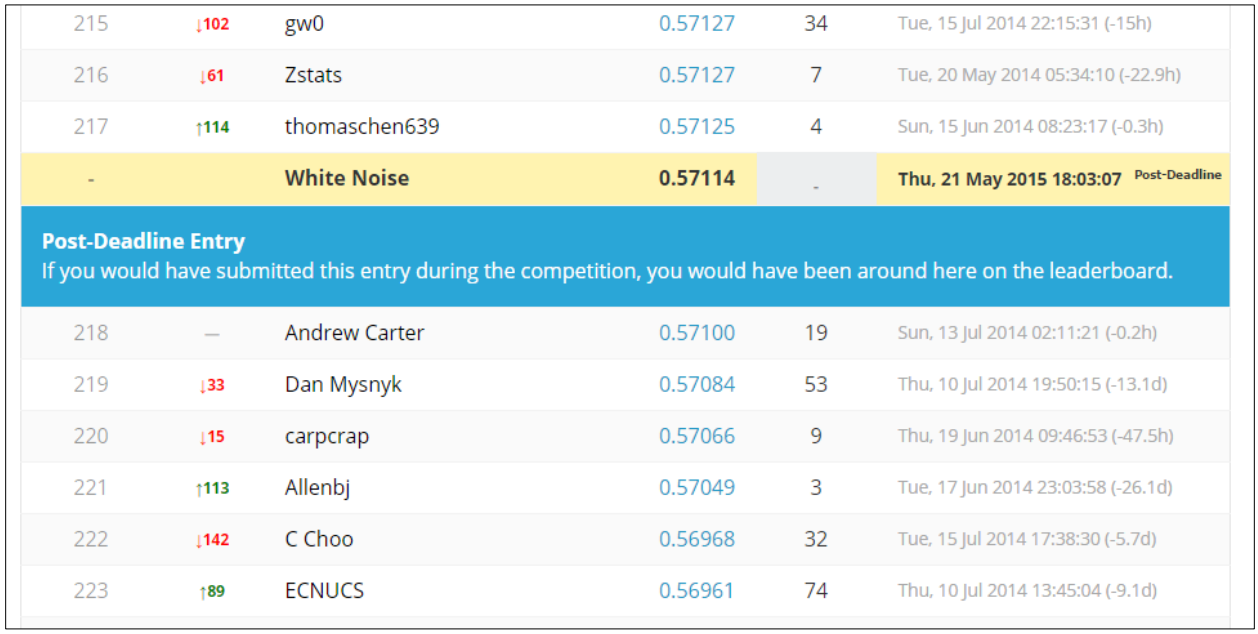

Figure 4: Kaggle leaderboard ranking

Additional Comments

Due to time constraints, the following, which would help increase the prediction accuracy, was not incorporated in the solution.

- NLP based analysis.
- Additional features and/or weights, especially to accommodate the fact that the test data is only from months January through May, while the training data is from January through December.
- Testing other types of models such as Random Forest and SVM and using higher number of trees.

Project Environment

- R Version 3.1.3 (Packages: caret, gbm, pROC, qdap, plyr), RStudio Version 0.98.507
- Python 2.7.5 (Packages: pandas, shutil, sys, os, re, gc), PyCharm 2.7.3
- MySQL 5.7, MySQL Workbench 6.0 CE
- Microsoft Windows 7 Professional
- Intel(R) Core(TM) i7-2640M CPU @ 2.80GHz, 2801 MHz, 2 Core(s), SSD, 12GB RAM

Executing the code

To execute the code, following requirements need to be satisfied:

- Directory structure as mentioned below.

  <working_directory>

  - R, Python, and MySql Source Code Files *(mandatory)*
  - /data/data_files *(mandatory, should contain kdd input data files)*
  - /data/input *(optional, created by python script to copy cleaned data files)*
  - /data/rdata *(optional, created by kddCommon.R)*
  - /data/rmodels *(optional, created by kddCommon.R)*
  - /output *(optional, created by kddCommon.R)*
- Copy R, Python, MySQL source files to the <working_directory>
- Change line 3 in kddCommon.R to set the working directory.
- In kddMySql.R source file, change lines 25 and 30 (dbConnect) to provide the appropriate MySQL username, password, and host.
- The required Python and R packages mentioned in the Project Environment must be installed.

Sequence of execution:

Remove '\r\n' character-tuple from the text in resources.csv. Execute the source files in the following sequence:

1) kdd_clean_data.py
2) kddmain.R
3) kddmysql.sql

The kddMySql.R code (executed through kddmain.R) will create a 'kddcupdb' database, if it does not exist, and also a create kddcupdb.predictions table. Approximate execution time with the aforementioned project environment is about 24 hours. The kddmysql.sql will show the correlation and ranking result.

NOTE: If kddcupdb.predictions already exists, IT WILL BE OVERWRITTEN.

Executing R code using R Console

```
> setwd("<working directory>")
> con <- file("kddmain_output.log")
> sink(con, append=TRUE)
> sink(con, append=TRUE, type="message")
> source("kddmain.R")
```

References

1. Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou, "Exploratory Undersampling for Class-Imbalance Learning," IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS – PART B, 2008

2. Kaggle KDD Cup 2014 Competition Forum, https://www.kaggle.com/c/kdd-cup-2014-predicting-excitement-at-donors-choose/forums

3. http://www.stackoverflow.com

4. http://www.stackexchange.com

Appendix

List of Files

1. KDDDataProject_AshishNaik.pdf – This document.
2. KDDTopProjects.csv – Correlation and ranking results.
3. gbmPredictions.csv.gz – Test set prediction results.
4. kdd_clean_data.py – Python code used to clean the KDD data.
5. kddCommon.R – R common definitions.
6. kddData.R – R code used to read KDD data, extract and transform features.
7. kddGBM.R – R code used for GBM modeling and predictions.
8. kddmain.R – Main R file used to run the R code.
9. kddMySql.R – R code used to create the KDD database and predictions table.
10. kddmysql.sql – MySQL script used to generate exciting projects correlation and ranking.
11. GBMParameterTuning.txt – GBM parameter tuning output from R.