ADDITIONAL ACADEMIC SUPPORT APPLICATION PREDICTIONS: A MACHINE LEARNING PRACTICE

PHYS T480: Big Data Physics

December 3rd, 2018

Team B, Section 001:

Jacob Zeitzew (jjz45@drexel.edu)
Philip Stephenson (pbs44@drexel.edu)
Earl Hyatt (ejh92@drexel.edu)

Introduction

Kaggle was started to encourage the growth of the fields of information technology, machine learning, and data science. The open source competition and project focused platform combines individuals and teams with larger corporations. Kaggle holds competitions in which teams compete to offer solutions to large scale problems that can be solved with machine learning. These problems use technology to make predictions based on existing data, such as suggesting what a customer wants or estimating how much a group wishes to spend. Example data sets are provided by companies, such as raw data that must be fed into the process and test data to compare solutions against. This data is usually provided in common formats, such as CSV files, JSON files, or SQLite databases.

Programing on Kaggle is centered around the idea of the “kernel”. [3] A kernel is a programming environment held in a Docker container. This almost entirely removes the issue of mismatched library dependencies. When entering a competition, users will create a new kernel for their submitted solution. A kernel can be of 3 types: a Jupyter notebook using Markdown with Python or R, a script for Python or R, or an RMarkdown script. The correct manner to submit a solution depends on the competition. Users upload predictions from a file, or from the kernel that produced those predictions. Each submission is processed, scored, and given an entry on the competition’s “public leaderboard”, which uses the publicly available data. It is also given a score on the “private leaderboard”, which judges the model using data unknown to the user. It is the ranking on this private leaderboard that determines the competitions winner.

Submissions are evaluated on a company-specific basis. Submissions must be formatted correctly, and the how the final results and method qualities are weighted is subject to that particular company's requirements. Some competitions judge on the area under a receiving operating characteristic or ROC curve, while others require that a winner's overall prediction distribution be accurate with little or no variance. Each competition has requirements for when rules must be accepted and for when solutions must be submitted. Once theses dates have passed and the requirements have been met, a winner is selected.

For the purposes of civic crowdfunding, the nonprofit organization DonorsChoose.org was founded. 80% of public schools in the United States have had at least one teacher request for supplies that could range from crayons, books telescopes, or even field trips. [1] The website gives people the chance to make an impact by donating and funding projects. The organization has grown year after year and they are currently predicting that in 2019, they'll be receiving close to 500,000 projects. For the purposes of allocating volunteer man hours and for identifying projects that need additional support before approval, DonorsChoose.org make their datasets open through Kaggle.com. [2] The training data set held more than 180,000 applications, including a unique ID, a Teacher ID, the teacher's prefered prefix, the state where the school is located, the submission date and time, the project submission category, the project submission subcategory, the projects intended age range, the project title, 4 project essays, a project resource summary, and the number of previous projects that the teacher has requested funding for already. The intended output is a file containing the predicted probability of approval for each submission in the test set.

Process

The provided dataset contained long-form text, categorical, and numerical features. These must all be converted to numerical features in some way before any classifier or machine learning algorithm can be applied to the data. Different types of features will require different feature engineering steps to suitably convert them to numerical data.

The easiest data to process is the categorical information. The two most important categoricals are the teachers' prefix (Mr., Ms., Mrs., or Uknown) and the state the submission came from. One hot encoding is used to transform these features into numerical features. One hot encoding works by creating a new column for every category, and assigning a 1 in the column corresponding to the category the record belongs to, and setting the others to 0, creating a sparse matrix with one 1 per set of category columns. The original categorical columns can then be dropped, as their information is now encoded in a new format.

The long-form text consists of a set of essays the teacher submits. For submissions prior to May 17, 2016 there are 4 essays, while submissions afterwards have 2. The essays all answer prompts relating to why the teacher believes the funds can help their classroom and why their students deserve the funds. Rather than analyzing each essay individually according to the prompt, all of the submissions' essays were added together into one long document which could then be processed. A process called term frequency-inverse document frequency (TF-IDF) was used to find meaningful features in the documents. TF-IDF works by first finding the frequency with which terms show up in the entire set of documents, and then weighting each document relative to the inverse frequency they appear in that document. This means which words which are common in all essays will be provided a low weight, while words which are rare in general but common in a specific essay will have a higher score.

The final feature created involved the teachers' number of previous submissions to DonorsChoose.org. In general, there was a clear positive correlation between number of submissions and acceptance rate, but the data was very noisy, especially at higher submission numbers. To combat the noise, teachers were binned in to 1 of 5 experience levels corresponding to different numbers of prior submissions.

These features were used to train a random forest classifier, which works by constructing a “forest” of decision trees and arriving at a final conclusion by consensus. After training, the outcomes were predicted for the test set. After uploading to Kaggle to verify the results, the model had a ROC AUC of 0.70, placing in the top half of submissions.

Receiver Operating Characteristics (ROC) Curve

An ROC curve is a comparison between the probability of testing *positive* for a present state and the probability of testing *negative* for a present state, for predictions made by a model. In other words, an ROC curve is a plot of the true positive rate (TPR, or *sensitivity*) versus the false positive rate (FPR). [7] They are defined as

$$TPR = \frac{\#\ of\ correct\ predictions\ of\ a\ state's\ presence}{\#\ of\ tested\ items\ truly\ having\ that\ state}$$

$$FPR = = \frac{\#\ of\ correct\ predictions\ of\ a\ state's\ absence}{\#\ of\ tested\ items\ truly\ missing\ that\ state}$$

By calculating the area under an ROC curve (a.k.a. the AUC or ROCAUC), one can characterize the effectiveness of a model in discriminating between whether or not the predicted state is present. An AUC of 0.5 would mean that only half of a model’s predictions would be correct, making the model unreliable. An AUC of 1 would mean that all of the model’s predictions are correct, and an AUC of 0 would mean that *all* of the model’s predictions are incorrect.

For discrete data, the AUC can be approximated with a trapezoidal Riemann sum. [8] This can be written simply as

$$AUC = \sum_{i=1}^{n} (y_{\,i} + y_{\,i-1}) \div 2(x_{\,i} - x_{\,i-1})$$

for each of the discrete observations from *i = 0* to *n*.

One recent (*2018-03-20*) high-profile competition using ROC curves in the evaluation process was the “Toxic Comment Classification Challenge” by Google and Jigsaw, with a first-place prize of $18,000. It used the mean of the AUC’s of each type of predicted label. [4]

Conclusion

There are several kinds of competitions on Kaggle. The most consequential are the “featured” competitions, which are aimed at solutions of commercial interest. Each featured Kaggle competition has its own specific set of rules, which cover the restrictions of the solution, as well as the legal technicalities of how a company uses a solution. Donorschoose.org works to connect individuals with classrooms in need, and posted a competition designed to predict the outcome of teachers’ funding requests. By using various feature engineering techniques including one-hot encoding, TF-IDF, and manual feature creation, a machine learning pipeline was created which obtained a ROC AUC of 0.70 on the test dataset.

References

[1]:
https://www.charitynavigator.org/index.cfm?bay=search.summary&orgid=9284#.UwOH7kJdV5N
[2]:
https://www.kaggle.com/c/donorschoose-application-screening
[3]:
https://www.kaggle.com/docs
[4]:
https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
[5]:
https://medium.com/@contactsunny/label-encoder-vs-one-hot-encoder-in-machine-learning-3fc273365621
[6]:
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html
[7]:
https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html
[8]:
https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/One_ROC_Curve_and_Cutoff_Analysis.pdf