


Mercedes-Benz Greener Masking Challenge Masking Challenge–1st Place Winner's Interview

December 28, 2017 · by Edwin Chen ·  kaggle.com (/r/mercedes-benz-greener-masking-challenge-masking-challenge1st-place-winners-interview/)

ARTICLES (/articles/)

GO TO PAGE (/r/mercedes-benz-greener-masking-challenge-masking-challenge1st-place-winners-interview/)

^ 0





(/r/mercedes-benz-greener-masking-challenge-masking-challenge1st-place-winners-interview/)

To ensure the safety and reliability of each and every unique car configuration before they hit the road, Daimler's engineers have developed a robust testing system. But, optimizing the speed of their testing system for so many possible feature combinations is complex and time-consuming without a powerful algorithmic approach. In this competition launched earlier this year, Daimler challenged Kagglers to tackle the curse of dimensionality and reduce the time that cars spend on the test bench. Competitors worked with a dataset representing different permutations of Mercedes-Benz car features to predict the time it takes to pass testing. Winning algorithms would contribute to speedier testing, resulting in lower carbon dioxide emissions without reducing Daimler's standards. The dataset contained an anonymized set of variables (8 categorical and 368 binary features), labeled X0, X1, X2..., each representing a custom feature in a Mercedes car. For example, a variable could be 4WD, added air suspension, or a head-up display. The dependent variable was the time (in seconds) that the car took to pass testing for each variable. Train and test sets had 4209 rows each. In this interview, first place winner, gmobaz, shares how he used an approach that proposed important interactions. Basics What was your backgrounds prior to entering this challenge? I studied at UNAM in Mexico to become an Actuary and hold a Master in Statistics and Operations

Research from IIMAS-UNAM. I've been involved in statistics for several years; worked some years at IIMAS as a researcher in the Probability and Statistics Department and have worked since then for a long time in applied statistics, mainly as a statistical consultant in health sciences, market research, business processes and many other disciplines. How did you get started competing on Kaggle? After some years working in the oil industry, in a non-related field, I decided to go back to statistics but was aware that I had to refresh my mathematical, computational and statistical skills, reinvent myself and learn at least R well enough to get back. That's when I found Kaggle's website. It had the best ingredients for learning by doing: having fun, real problems, real data and a way to compare my progress. Since then, I've participated regularly on Kaggle, mainly to keep in shape and to be aware of recent advances. What made you decide to enter this competition? At a first glance, this competition seemed to have elements in common with the Bosch competition. Working with many binary and categorical features is a very interesting problem and good solutions are difficult to find. Before entering the competition, I had time to follow the discussions and read some splendid EDA's, particularly by SRK, Head or Tails and Marcel Spitzer that helped a lot in gaining insight to understand the manufacturing and modelling problems. Let's Get Technical What preprocessing and feature engineering did you do? Before doing any modelling or feature engineering, first thing I usually try to do is to get what I call a basic kit against ignorance: main concepts, bibliography and grab whatever helps to understand the problem from the sector/industry perspective. In this way there will be a guide to propose new features and a clearer understanding of datasets and measurement issues like missing values. With an anonymized set of features, what kind of new features would be interesting to explore? I imagined passing through the test bench as part of a manufacturing processes where some activities depend on previous ones. I set up some working hypotheses: A few 2- or 3-way interactions and a small set of variables could be relevant in the sense that test time changes could be attributable to a small set of variables and/or parts of few subprocesses. Lack of synchronization between manufacturing subprocesses could lead to time delays. The following are the features considered in the modelling process: I found that parameters for XGBoost in kernels, for example, by Chippy or anokas and findings in EDA's were consistent with the working hypotheses. So, how to explore interactions? Just two-way interactions of binary variables would lead to explore 67528 new variables, which sounded like a lot of time and effort, so the task was to identify quickly some interesting interactions. Search for them was done looking at patterns in preliminary XGBoost runs. Some pairs of individual variables appeared always "near" in the variable importance reports. With just three pairs of individual features, two-way interactions were included and, additionally, a three-way interaction. Thinking on the subprocesses, I imagined that the categorical features, were some sort of summary of parts of the manufacturing testing process. The holes in the sequencing of the binary feature names took me to define nine groups of binary variables, consistent with the eight categorical ones. Within these nine groups, cumulative sums of binary variables were thought as aids to catch some joint information of the process. Despite the burden of introducing quite a few artificial and unwanted dependencies, models based on decision trees can handle this situation. After some playing with the data, I decided to recode eleven of the levels of first categorical feature (trigger of the process?) One-hot encoding of categorical features was applied, that is, the original and the ones created for interaction variables. One-hot encoding variables were kept if sum of ones exceeded 50. Since this value looks reasonable, but arbitrary, it is subject to tests. To include or not ID was a question I tried to answer in preliminary runs. Discussions in the forum suggested that including ID was totally consistent with my thoughts on the Mercedes process. I detected very modest improvements in preliminary runs: it was included. It is known that decision tree algorithms can handle categorical features transformed

to numerical, something that makes no sense in other models. These features were also included, which completed the initial set of features considered. So, starting with 377 features (8 categorical, 368 binary and ID), I ended with 900 features; awful! And a relatively small dataset... Can you introduce your solution briefly? Two models were trained with XGBoost, named hereafter Model A and Model B. Both were built in a sequence of feature selection steps, like backward elimination. Model B uses a stacked predictor formed in a step of Model A. Any decision point in this sequence is preceded by a 30-fold cross validation (CV) to find the best rounds. The steps are very simple: Preliminary model with all features included, Model A, 900 features and Model B, 900+1, the stacked predictor. Feature selection. Keep the variables used by XGBoost as seen on variable importance reports (229 in Model A, 208 in Model B). Feature selection. Include features with gains above a cut value in the models; 0.1%, in percentage, was the cut value used, 53 in Model A, 47 in Model B. Both models use XGBoost and a 30-fold CV through all the model building process. The rationale for a 30-fold validation was to use it in a 30-fold stacking as input for Model B. The stacked predictor might damp the influence of important variables and highlight new candidates to look for some more interesting interactions. The most important features As can be seen from the graph below, interactions played the most important role in the models proposed (anonymized) features. By far, pair (X314, X315), jointly and pair levels 3-way interaction (X118, X314, X315) X314 (X118, X314, X315), levels (1,1,0) Individual features: X279, X232, X261, X29 Two levels of X0 recoded and X0 recoded Sum of X122 to X128 X127 Notably in the discussions, besides one kernel by Head and Tails dealing specifically with interactions, I found no other reference to any 2 or n-way interactions, different from the ones I used. How long did it take to train your model? During the contest, work was done in R Version 3.4.0, Windows version. After the contest, Version 3.4.1 was used. For common data in both models, initial data management took less than 4 seconds. For steps 1-3 in training method, Model A needed approximately 3.4 minutes, Model B took around 4.3 minutes on a desktop I7-3770 @3.40 GHz, 8 cores, 16 MB RAM. Starting from loading packages to submissions delivery for both models, the code took circa 8 minutes. Loading packages and preparing Model A took 4.5 seconds. To generate predictions for 4209 observations from test set took around 2.3 seconds. The winning solution was a simple average of both models. Individually each one outperformed the results of the 2nd place winner. The good news is that Model B does not really add value; stacking is therefore not necessary and a simpler model, model A, is advisable. What was the most important trick you used? I think the competition was on trapping individual variables and propose important interactions. The way I selected interactions was a shortcut for finding some of them. Trapping individual variables was mainly the goal of the stacking phase, without apparent success. The shortcut for identifying interactions looks attractive and I have used it before with good results. I was afraid on using cumulative sums of binary variables due the dependencies between them. Given the results, I would try shorter sequences around some promising variables. Words of wisdom What have you taken away from this competition? Any competition allows you to learn new things. After the competition, making tests, cleaning code, documenting and presenting results was an enriching experience. Do you have any advice for those just getting started in data science? 1. Identify your strengths and weaknesses: mathematics, your own profession, statistics, computer science. With the need to know from all, balance is needed and black holes in knowledge will appear almost surely. I found a quote in Slideshare from a data scientist, Anastasiia Kornilova, who summarizes my view very well (graph adapted with my personal bias): "It's the mixture that matters". There is always a chance to fill some black holes and don't worry: it will never end. 2. Learn from others with no distinction of titles, fame, etc. The real richness of Kaggle is the diversity of



approaches, cultures, experience, problems, professions, ... 3. If you compete in Kaggle, compete against yourself setting personal and realistic goals and, above all, enjoy! 4. PS. Don't forget to cross-validate

Comments

No comments yet.

(/)

The best resources and jobs on data science, machine learning, data mining and analytics.

CONTACT

If you want to submit content please contact info@datapreneurs.co (<mailto:info@datapreneurs.co>)

 (<mailto:info@datapreneurs.co>)

 (<https://www.facebook.com/datapreneurs/>)

 (<https://twitter.com/datapreneurs>)

 (<https://t.me/datapreneurs>)

EXPLORE

- ▶ [ABOUT US \(/about/\)](/about/)
- ▶ [TRENDING \(/\)](/trending/)
- ▶ [LATEST \(/latest/\)](/latest/)
- ▶ [JOBS \(/jobs/\)](/jobs/)

