Featured Prediction Competition

Corporación Favorita Grocery Sales Forecasting

Can you accurately predict sales for a large grocery chain?

$30,000
Prize Money

Corporación Favorita · 1,675 teams · 2 days ago

Shize Su
16th place

my 10 day journey for this competition and solution sharing

22

posted in Corporación Favorita Grocery Sales Forecasting 16 hours ago

As promised in the post https://www.kaggle.com/c/favorita-grocery-sales-forecasting/discussion/47326 , now I am sharing my 10 days journey experience in this competition and my approach^_^

In addition to sharing my approach, I am also most interested in discussing why there is such a big shake-up on the private LB, and I would like to hear your thoughts and/or findings as well^_^

(Warning: This is going to be a long post, hope you have some patience^_^)

As I mentioned in that post, I entered this competition quite late (when there was only 10 days left). At that time I just finished the last master only competition and took a short break. Since I had some spare time, I planed to explore this competition a bit, simply for fun. Finally, I am ranked 8th place on the public LB (score 0.503), but unfortunately shaked down a bit on the private LB (score 0.516). But In general, I am quite happy and satisfied with the score I achieved in this competition, given that I entered this competition so late.

Before introducing the flowchart of my approach, I want to talk a bit about the CV_LB relationship as well as the final private LB shake-up. I would also love to hear your thoughts and findings on this.

As I mentioned in the previous post https://www.kaggle.com/c/favorita-grocery-sales-forecasting/discussion/47326 , my local CV score and public LB score correlates somewhat reasonably well (especially when comparing same type of

models, e.g., comparing different feature engineering ideas for lgb models, or comparing different feature engineering ideas for nn models, for the comparison between nn and lgb models the CV-LB gap was indeed a bit different (nn showed significantly better cv score thank lgb on the similar feature set, while performed slightly worse than lgb on the LB)), which enabled me make progress somewhat smoothly on the public LB and climbed to 8th place on the public LB in the limited 10 days. The final public LB-private LB shake up is much larger than I expected given my somewhat reasonably good CV-LB relationship, and I have't fully figure out why. I did expected some extent of medium shaking up between public and private LB ranking, but definitely not as large as what it turned out to be. I understood that the promotion information is biased between the train and test set, and feature engineering efforts based on such promotion information might be easily over-fitting if not dealt with very carefully, and I guess many competitors relied on feature engineering ideas based on such promotion features to improve the public LB score, which might possibly to some extent explain the big shake-up. But I have no a certain idea on this since I don't have full knowledge about how other competitor did when getting their public LB score. On my side, I did almost zero additional feature engineering efforts based on the promotion variable (except those already used in the shared lgb starter script), and my improvement of public LB score from 0.513 (lgb starter script score) to 0.505 for a single lgb relied on feature engineering efforts that is not related to promotion variable at all, and my local CV-Pub LB relationship is somewhat consistent, though not perfect. I am a bit confused why the CV-LB relationship for my models on the private LB is much less correlated, even if I almost didn't use any additional promotion-based features (compared with the lgb starter script) to achieve my score improvements in CV and public LB. Ironically, it turned out that my 1st entry of this competition, which is a lgb scored 0.511 on public LB and 0.517 on private LB, is my best single model on the private LB. In comparison, my final best single lgb on the public LB (scored 0.505 on public LB) also only gave 0.517 on private LB (and slightly worse than my 1st entry lgb). Namely, all my added feature engineering efforts (which are not related to promotion variable at all) improved both my local CV and public LB score significantly and somewhat consistently, turned out to adding no extra values (and even some small negative effects) to the private LB score, in comparison with my 1st entry lgb in this competition. Bias in promotion variable in train and test set can to some extent explain the shake up on public/private LB for many competitors if they relied on promotion-based features to climb the public LB, but definitely still could not explain many other scenarios, such as my case here that almost no promotion-based engineered features are used to achieve CV/Public LB score improvement, and CV_Public LB patterns are somewhat consistent.

I would really like to hear the thoughts and experiences from other competitors to understand what actually caused such phenomenon. In addition to the promotion variable bias, are there anything else important we missed here that contributed to the shaking up? Or it is just due to the randomness /unstable time series trends, and the shaking is simply due to luck? If many of us are happy to share

our thoughts and findings, maybe we would finally be able to figure out what would be the root cause for such a big shake-up and understand this problem much better, rather than simply jumping to the conclusion that the shaking up is due to lacking of comprehensive cv experiments, or bias for promotion variable.

Now let me start introducing my approach.

First of all, credits are given to those people who shared the great starter scripts in the kernel, especially to @tunguz, @ceshine. Your starter scripts (see links below) gave me a great starting point in this competition, and enable me to quickly work on key feature engineering and modeling experiments for fast progression in this competition. Actually, these shared great starter scripts is also one major motivation for me to enter this competition when there is only 10 days left; usually I will only enter a competition when there is much more time left.

https://www.kaggle.com/tunguz/lgbm-one-step-ahead-lb-0-513
https://www.kaggle.com/ceshine/lgbm-starter

Sample Data Selection:

My final models used only 2017 data to extract features and construct samples. I tried to include more data, such as those 2016-Jul and 2016-Aug data (similar month/week periods as my validation period and LB period) to capture the monthly dynamics, but they only helped marginally or not helpful at all, depending on the models, while take significant longer training time. So I decided to drop those data from my final models.

**train data*：*20170503- 20170719 for cv experiments, and 20170503-20170809 for LB experiments (namely, for LB submissions I retrained the models adding the most recent days data since they are closest to the LB period and might be most valuable)

validation data: 16 days period, 20170726 - 20170810 (and also tried using validation period 20160817-20160901 for some of my nn models at the very end of the competition, but just some simple trial since no much time left for me, and it turned out to perform worse than using 20170726 - 20170810). I computed the CV score for the first 5 days, last 11 days, and the full 16 days (simulating the LB split). I observe somewhat consistent CV-LB score improvement trends in all of my experiments, especially when comparing same type of models (e.g., comparing different lgbs, or comparing different nns). For my feature engineering efforts, I even observed even larger cv score improvement for the last 11 days (similar to the private LB) than the cv score improvement for the first 5 days (similar to the public LB), and thus at some time points neard the end of the competition, I once believed that I probably would climbed up a couple of spots on the private LB, due to such CV-LB score findings. Unfortunately, it didn't work

as nice as I expected and I was shaked down a little bit. I still don't understand why for this.....

Since starting date 2017/08/16 for LB period is Wednesday, and thus it is essential to also set the starting date for the validation period as Wednesday to preserve the DayOfWeek pattern.

One key trick to make my local CV and public LB score more consistent is as follows:

Step I: I extract all the store_item combinations that appeared in the period from 2017-07-01 to 2017-08-15.

Step II: I only keep the training data records (and drop all other records) whose store_item combination appeared in the extracted store_item combinations in Step I. The underlying assumption is that, if a item was not sold (or with identical 0 sales) at all in a store in the period 2017-07-01 to 2017-08-15, it is very likely that it wouldn't be sold (or with identical 0 sales) in the LB period 2017-08-16 to 2017-08-31 either. By looking at the train data, it seems to me that the new store_item combinations are added very slowly in train set, and when considering a short period like 16 days, the amount of newly added store_item combination (with non_zero sales) is almost negiligible.

Step III: I also only keep the test data records whose store_item combination appeared in the extracted store_item combinations in Step I. I always predict zero target values for the other store_item combination that don't belong to the store_item combinations extracted in Step I.

(My such validation trick was also partially inspired by the Kaggle Admin Inversion's feedback on the unseen new items in this post , who mentioned that "I'm jumping in here late, regarding the discussion of new products, in order to clarify a bit. As has been mentioned, there are a very small number of new items that were introduced in the test set. That does not explain the apparent high number of new items seen on the first day of the test set. The reason for this is that the training set does not include records for zero sales. The test set, though, includes all store / item combinations, whether or not that item was seen previously in a store." Thus I made an assumption that "apparent high number of new items seen on the first day of the test set" are mostly items that have been seen previously in a store but just with zero sales, while the additional "small number of new items that were introduced in the test set" are those new store_item combinations if we compare later date of the test set and the first date of the test set. Since this amount is small, I assumed that it is safe to simply always predict zero target values for all those store_item combination that don't belong to the store_item combinations extracted in Step I. And such trick indeed made my CV-public LB relationship reasonably consistent, and I thought that I got it. But I don't know why the same consistency didn't preserve well on the private LB.....)

After this 3-step trick and using the 16 days validation period 20170726 - 20170810, (and train period 20170503-20170719), I observed somewhat reasonably consistent CV_LB relationships when testing my feature engineering ideas, especially when comparing same type models (comparing different lgbs, or compering different nns), and it helped me a lot to make smooth progress on the public LB.

Feature Engineering:

1. Just follow the lgb starter scripts and adding more past sales (and DayOfWeek sales) average/ (weekly average) features . I also add (mean, median, std, sum, max, min) for many of such past sales average/ (weekly average) features. This basically gave my 1st entry submission of this competition, which scored 0.511 on the public LB and 0.517 on the private LB. It seems to me that this part didn't break the CV-Public LB-Private LB relationship yet.
2. Adding a little bit more promotion features similar to the starter lgb script (similar to those "promo_14_2017" , "promo_60_2017":, "promo_140_2017" type features, but just used more periods like 7, 21, 35, etc). This improved both the CV and public LB score by about 0.0015. But private LB score dropped from 0.517 to 0.518. I guess bias in promotion variable might to some extent explain such broken in CV-LB score consistency, but can't fully explain it, since CV_public LB consistency is still preserved by adding these promotion based features. These several features are the only additional promotion-based features that I used in my final publicLB 0.505 lgb. From now on, all the feature engineering efforts that improved my public LB score from 0.509 to 0.505 (and also improved my CV score in a similar scale) are all not related to promotion variable at all.
3. Adding the category features from stores.csv and items.csv, and apply label encoding. This improved both the CV and Public LB score by about 0.001+, to get 0.508 public LB score for a single lgb. Private LB score stay the same at 0.518. Not sure why, such category feature should not likely to break CV-LB relationship seems they are no biased. Is that possible this is simply due to that the time series trend pattern in the LB period is not very stable, and luck played a significant role? Or anything important we missed here?
4. Adding statistical features (or "target_encoding" like features for category variables): I use some methods to stat some targets for different keys in different time windows (this is somewhat similar to that one shared the top 1 place solution, with some additional difference being the choice of target and key, as follows:)

key：'store_nbr', 'item_nbr','family', 'class', 'city','state', 'type', 'cluster' (i.e., all the used category variables)

target: I output the lgb feature importance for the above lgb and selecting those most important past sales average/ (weekly average) features & DayOfWeek features, as well as the associated stats features (std, sum, median, etc.). In particular, the complete list of my chosen "target" features are as follows: 'mean_7_2017_01', 'mean_14_2017_01', 'mean_21_2017_01', 'mean_35_2017_01', 'mean_30_2017_01', 'mean_60_2017_01','ahead7_6', 'ahead7_5', 'ahead7_4', 'ahead7_3', 'ahead7_2', 'ahead7_1', 'ahead0_6','ahead0_5', 'ahead0_4', 'ahead0_3', 'ahead0_2', 'ahead0_1', 'mean_4_dow0_2017', 'mean_8_dow0_2017', 'mean_16_dow0_2017', 'mean_4_dow1_2017', 'mean_8_dow1_2017', 'mean_16_dow1_2017', 'mean_4_dow2_2017', 'mean_8_dow2_2017', 'mean_16_dow2_2017', 'mean_4_dow3_2017', 'mean_8_dow4_2017', 'mean_16_dow3_2017','mean_4_dow4_2017', 'mean_8_dow4_2017', 'mean_16_dow4_2017','mean_4_dow5_2017', 'mean_8_dow5_2017', 'mean_16_dow5_2017', 'mean_4_dow6_2017', 'mean_8_dow6_2017', 'mean_16_dow6_2017', 'day_1_2017','day_2_2017','day_3_2017','day_4_2017','day_5_2017','day_6_2017','day_7_2017'

method: mean, median, max, min, std

Note that, in order to avoiding possible future target information leak, such statistical features were generated independent for each week of the training data, validation data, and test data.

So in total there are about 1400+ such statistical features, and expand my full feature set size to about 2300+. I then run lgb and output the feature importance, and only select the most important 400 features and use them for my later modeling (for lgb and nn). Such feature selection reduce the training time significantly, and also improved CV and public LB score by about 0.0006 simply by itself.

Such statistical features are very helpful for improving my CV and public LB score. In particular, it improved my single lgb score from 0.508 on public LB to 0.505 on the public LB, and similar scale of improvement is observed in local CV. However, the private LB score only improved slightly from 0.518 to 0.517.

My nn models were basically trained on the same feature set as my best single lgb, except that I replacing the label-encoding category features by one-hot encoding (i.e., dummie features) category features, since label encoding of category variables doesn't make much sense for models like nn. My best single nn scored 0.506 on public LB & 0.520 on private LB, and an average of two nn with some small difference gave me 0.505 score on the public LB and 0.518 on the private LB. Note that I have only explored nn model when there is only 2 days left (after I am almost done for lgb model and feature engineering) and there is no much time for me to tune the nn though. I just made it somehow work.

Ensemble of my best lgb and nn gave my final score, 0.503 on public LB and 0.516 on private LB. The ensembling step of nn and lgb does seem to preserve the

CV_Public LB_Private LB relationship, namely, they improved the score all by about 0.002.

To sum up, based on my experience, what seems to preserve the CV_Public_Private score relationships are:

a) the feature engineering effort 1 described above (adding more past sales weekly average, DayOfWeek sales, as well as the associated mean, std, min, max, sum stats features, this preserve CV-Public-Private relationship nicely)

b) feature engineering effort 3 described above ("target_encoding" like statistical features for category variables, this preserves the CV_Pub_Private relationship somehow ok, but not perfect, CV_Pub score improved by about 0.003, private LB score improved by about 0.001),

c) ensemble of nn and lgb.

What seems to not preserving the CV_Public_Private score relationships are:

a) the feature engineering effort 2 described above (i.e., adding some more promotion-based features similar as the lgb starter scripts, i.e., similar to "promo_14_2017" , "promo_60_2017":, "promo_140_2017" type features, but just used more periods like 7, 21, 35, etc). This might be somehow understood due to the bias in promotion variable.

b) Adding the category features from stores.csv and items.csv, and apply label encoding. This one didn't preserve the CV_Pub_Private relationship makes me completely confused, since they are not biased between CV_Pub_Private. And note the the CV_Pub score relationship is still nicely preserved when adding such category variables, not sure why it didn't work on private LB.

c) for the feature engineering effort 4 described above ("target_encoding" like statistical features for category variables), they improved both CV and public LB score by about 0.003+, but only improved private LB score by about 0.001 also make me a bit confused. Such features are not related to the promotion variable at all, and why it broke the CV_Private LB relationship (given that the CV_Public LB was still nicely preserved)?

Single Model:

public LB 0.505/ private LB 0.517, single lgb

public LB 0.506/ private LB 0.520, single nn

average of 2 nn with small differences gave public LB 0.505/ private LB 0.518.

Final Ensemble:

0.54-0.46 weighted average of lgb and nn, public LB 0.503/ private LB 0.516.

That's all for my approach in this 10 days journey of this competition. Overall speaking, I am satisfied with what I achieved in this competition given the limited time, and I have learned a couple of useful things from competitors and my experiments when addressing this time series challenge. To the end, I would really love to hear about your thoughts and findings about what preserves the CV_LB relationships and what didn't preserve the CV_LB relationships, such as sth similar to what I shared above. For now I still couldn't fully understand why the relationship is not preserved on the private LB, given that CV_Pub LB relationship is somewhat consistent for me. By combining our findings, maybe we could have a deep understanding about why there would be such a big shake up on the private LB, rather than simply jumping to the conclusion that it is due to lacking of comprehensive CV experiments or just luck/randomness. Maybe we missed sth important but subtle, which contributed to the big shake-up on the final private LB, and if we figured out that, maybe there is indeed a nice way to preserve the CV_Pub_Private score relationship very well if done appropriately.

Finally, thanks to Kaggle and the sponsor to host such a great competition, and thanks to all other competitors who made my journey in this competition fun! And, congratulation to the winners!

Best regards,

Shize

Options

Comments (24) Sort by Hotness

Click here to enter a comment...

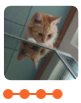
Lingzhi • (5th in this Competition) • 11 hours ago • Options • Reply 1

For me, the CV and LB performances were not consistent, and not consistent even between different models, this made me confused for a long time. My Seq2Seq model gave me the best CV result but was stuck at .512 at LB, my CNN model was slightly worse and not stable in CV but got .505 in LB. Finally these two models both achieved .514 in PB. Generally, for me the CV and PB are quite consistent but the LB is problematic.

Actually after noticing the strange pattern of promotion infos in 8/16, which is stated in my solution, I told myself the LB is problematic and cannot be trusted. The CV cannot be trusted either because of the biased promotion info. I have to say there is some luck here to bet on the distribution of the private test data.

Shize Su • (16th in this Competition) • 10 hours ago • Options • Reply 0

@Lingzhi: You mentioned that your CV and Private LB is somewhat consistent, but not the public LB. Mine is the other way, namely, CV and Public LB is consistent, but not the private LB. May I know which validation periods/method you were using?



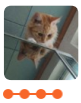
Lingzhi • (5th in this Competition) • 10 hours ago • Options • Reply 0

I only used 7/26-8/10 for validation for the whole time, and used similar ideas as yours: keeping the store-item combos that are in test data and have values during 2017 only.

Shize Su • (16th in this Competition) • 10 hours ago • Options • Reply 0

@Linzhi Hmm, then it is quite strange why my models preserve CV_Public LB score consistency, while your models preserve CV_Private LB score consistency, given that we were using similar CV mechanism...

Hmm, after a second thought, a possible explanation is that this might be due to the different type of feature engineering efforts (and/or modeling approaches, I didn't try any sequence models in this competition) between us (at least this is the only thing I can imagine for now). Namely, some type of feature engineering efforts might work for validation period and public LB period, while some other type of feature engineering efforts might work for validation period and private LB, due to some subtle difference between public LB and private LB that we haven't identified yet.



Lingzhi • (5th in this Competition) • 10 hours ago • Options • Reply 1

This is what I am confused too. Btw I also found item_nbr overfitted my models and I dropped it.

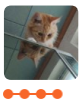
feng • (184th in this Competition) • 7 hours ago • Options • Reply 0

@Lingzhi I use you shared LGBM starter code, but what confused me is that when i add some new features, the local cv decrease, but the public lb score increase, why? It's overfitting the validation data?

feng • (184th in this Competition) • 6 hours ago • Options • Reply 0

@Shize su. From you shared methods, you are using 20170503- 20170719 for training, 20170726 - 20170810 for validation. When you add new features every time, using the validation score to evaluate it, if the new features decrease the validation score for the first 5 days, last 11 days, and the full 16 days then add this features. Is my understand right?

What confused me is that how to debug the overfitting? Can you give some insights of it? Thanks.



Lingzhi • (5th in this Competition) • 6 hours ago • Options • Reply 0

I am not sure, maybe it was overfitted to the first 5 days but failed to generalize to 6th ~ 16th day, or maybe the increase in LB is just random noise that cannot be relied on. As I said, I don't think the test data in LB is reliable in this competition.

CPMP • 6 hours ago • Options • Reply 0

> Btw I also found item_nbr overfitted my models and I dropped it.

Thanks for confirming this.

feng • (184th in this Competition) • 5 hours ago • Options • Reply 0

@Lingzhi The 16 days validation score is decreased after add some new features, maybe as you said "overfitted to the first 5 days but failed to generalize to 6th ~ 16th day". How to debug it when it's happening? Or what's you next steps? Thanks for response.

CPMP • 13 hours ago • Options • Reply 1

Thanks for sharing. I found that using item_nbr was leading to overfit in my local CV. This may explain part of the shakeup.

Shize Su • (16th in this Competition) • 12 hours ago • Options • Reply 0

Thanks for the info. But why? There should not be any bias for item_nbr between public LB and private LB, and thus this factor cannot explain why my CV_Pub LB relationship is

consistent, but the consistency broken in private LB..... As I mentioned above, with my CV mechanism, adding item_nbr as well as other category variable improved both my local CV score and Public LB score consistently by about 0.001+, while private LB score stayed the same. This seems indeed very strange to me, and I have no clues why.... Could anyone else confirmed whether they encountered similar situations (i.e., improve both CV and public LB score consistently, but not private LB) when adding the category variables?

CPMP • 12 hours ago • Options • Reply 2

I'm not sure how to express the gut feeling I have here. Let me reverse the question: why would item_nbr help at all? Why would be the pattern for sales for a given item be significantly different from another one? In the WTF competition I did not use the page either as a feature. It forces the model to generalize on how sales evolve over time across items, instead of learning a pattern for each item, which increases variance and the risk of overfit. But I may be totally misled here, i welcome other feedback too.

The other reason may be your choice of validation period: the test period ends with month end, i.e. a payday. We selected our validation periods to end with paydays, with the hunch that this was a significant pattern. I must say this was Giba's idea, I was tempted to do exactly as you did ;)

Nicolas • 12 hours ago • Options • Reply 1

In my case, adding:
- Frequency Encoding (store, item, family, class)
- One-Hot_Encoding (clusters, states, types, cities, families)

Did the following:

Days 1 --> 5

- Improvement on public leaderboard of less than .001.
- Improved CV of first 5 days by .0013

Days 6 --> 16

- Improved private LB score by .002 (or less)
- Unfortunately, I do not have the CV total error for days 6-16, but the average 'day_error' improvement was .0018.

Shize Su • (16th in this Competition) • 11 hours ago • Options • Reply 2

Thanks for the feedback, CPMP.

> Let me reverse the question: why would item_nbr help at all? Why would be the pattern for sales for a given item be significantly different from another one?

For me the underlying assumption is that, the sales pattern for different items (there are about 10k different items) are possibly somewhat different from each other, though the extent of dissimilarity is hard to tell. At least, it is possibly fair to say that the past sales data for the same item is possibly a bit more helpful than the past sales data for a different item. According to my intuition, adding the item_nbr feature (as well as other category features like store_nbr, family, cluster, etc.) would help the lgb model to more easily identify those most similar records and gave them a bit more weights, while without the item_nbr, since there are about 10k items in total, such info might be just buried in the large number of records and harder for lgb to identify. The same is for other category variables like store_nbr, etc. Anyway, the key for feature engineering efforts is just to make those information more easily captured by the models like lgb. Theoretically speaking, all the information are contained in the given raw data, our feature engineering efforts just convert those information into a form which is easier for the machine learning models to capture, rather than creating new information.

Also note that, adding all those category variables (store_nbr, item_nbr, family, class, cluster, etc.) in total gave about 0.001+ improvement in both my CV and Public score, which is not very large but should be nontrivial. It might not necessarily to require "the pattern for sales for a given item being 'significantly' different from another one" in order for item_nbr to be helpful in such a extent of 0.001+ score improvements. As long as there are some nontrivial dissimilarity among the sales patterns for different items (and since there are about 10k items in total, I do bet this is true), it is reasonable to accept that adding item_nbr to be helpful (though it is also acceptable if adding item_nbr not helpful, when considering the generalization mentioned by you), namely, adding item_nbr being helpful seems not to be a weird result for me.

What confused me was why adding such category variables (item_nbr, store_nbr, family, cluster, etc.) helps both my CV and public LB, but not private LB. But your raised point about pay day in validation period is indeed a factor that I never considered in my experiments, and it might possibly explain the broken CV_LB relationship on private LB, though I am not sure since I didn't do experiments on this. Also note that the top 1 team also used exactly the same validation period (2017/07/26-2017/08/10) as me, but did not suffer the same issue on private LB, so I am not sure whether the issue is indeed associated with my validation period choice...

Shize Su • (16th in this Competition) • 11 hours ago • Options • Reply

1

> Improved private LB score by .002 (or less)

@Nicolas: So it seems that adding those category variables seems indeed contribute to score improvements in all of your CV, Public LB and Private LB scores, which is different from mine. May I know which validation periods you were using? Also 16 days period starting at 2017/07/26? Or sth else?

CPMP • 11 hours ago • Options • Reply 0

@Shize, you did notice yourself that "3b) Adding the category features from stores.csv and items.csv, and apply label encoding " broke the CV LB relationship. I tend to believe that it leads to overfit. Good point re team #1 using the same validation period as you. I haven't studied their code yet, so I cannot comment further.

Shize Su • (16th in this Competition) • 11 hours ago • Options • Reply 0

@CPMP: Yeah, the thing made me confusing is that adding category features from stores.csv and items.csv only broke my CV_Private LB relationship, and the CV_Public LB relationship was still preserved nicely. That's what made me confused. If it also broke my CV_LB relationship, then I would say maybe it is over-fitting to CV and has worse generalization. But since it improved both CV and Public LB score in a similar scale but not improved the private LB score, this made me confused.

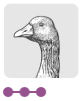
CPMP • 11 hours ago • Options • Reply 0

This dataset was confusing in many ways for me. For instance I tried to use sales from same class as features, it improved CV nicely, and resulted into a public LB of 0.77 !?! I may have done something wrong as I was quite feverish that day, but it made me very cautious about not overfitting by adding too many features. I'm impressed that you and others (eg @Nicolas) managed to add so many features without overfitting.

Shize Su • (16th in this Competition) • 11 hours ago • Options • Reply 1

@CPMP: Yeah, I just have a feeling that it is probable that we might still miss sth important but sbtle, and if we identify that one, maybe it is possible to achieve consistent CV_Public LB_Private LB score relationship, rather than making the final private LB somehow a lottery. My feeling was due to that : 1) My CV_Public LB relationship is reasonably consistent for my feature engineering ideas; 2) It seems to me that there is no too much difference/bias between public LB and private LB, in terms of those feature engineering efforts I tried. Thus, unless I missed anything, I have such a feeling that achieve consistent CV_Public LB_Private LB score relationship might be possible, there might be just some subtle issue that is hard to figure out that we didn't

find it. That's the main motivation why I initiated this discussion about the CV_LB relationship question here.



steelrose • (36th in this Competition) • 5 hours ago • Options • Reply

0

had a similar experience, simply adding the item_nbr + store_nbr basically "broke" the public LB score; when I check my submissions, I have one with private 0.517 but public 0.908 (and another one 0.517 and 0.699), the CV scores of these two compared to my other submissions were almost the same; there really is something "weird" in the public LB

btw. I was also validating on 2017-07-26 and I believe that was the key for staying in the "medal area"

Weijie Gao • (164th in this Competition) • 15 hours ago • Options • Reply

1

Thanks for the write up! Very informative :)

Kevin S. • (1382nd in this Competition) • 10 hours ago • Options • Reply

0

awesome! Thanks for sharing your approach.

feng • (184th in this Competition) • an hour ago • Options • Reply

0

hi, @Su. The "statistical features were generated independent for each week of the training data" is still confused to me.
For the training data (20170503- 20170719), we should construct the features and labels. If using the sales 2017/7/4-2017/7/19 as the labels, how to construct the features for it? or what's the labels in you training data? Thanks for you time.