kaggle Search kaggle Competitions Datasets Kernels Discussion Jobs

Featured Prediction Competition

Corporación Favorita Grocery Sales Forecasting

Can you accurately predict sales for a large grocery chain?

$30,000
Prize Money

Corporación Favorita · 1,707 teams · 15 hours ago

Overview Data Kernels Discussion Leaderboard Rules Team My Submissions New Topic

Lingzhi
5th place

posted in Corporación Favorita Grocery Sales Forecasting 7 hours ago

First of all, congratulations to the winners. Thanks to @Ceshine Lee, @sjv and others whose codes, kernels and discussions gave me many inspirations. This was my first time to use deep learning and I really learn a lot from previous shared kernels and codes from kagglers.

Model Overview

I build 3 models: a lightGBM, a CNN+DNN and a seq2seq RNN model. Final model was a weighted average of these models (where each model is stabilized by training multiple times with different random seeds then take the average), with special dealing with promotion, which I will discuss later. Each model separately can stay in top 1% in the private leaderboard.

LGBM: It is an upgraded version of the public kernels. More features, data and periods were fed to the model.

CNN+DNN: This is a traditional NN model, where the CNN part is a dilated causal convolution inspired by WaveNet, and the DNN part is 2 FC layers connected to raw sales sequences. Then the inputs are concatenated together with categorical embeddings and future promotions, and directly output to 16 future days of predictions.

RNN: This is a seq2seq model with a similar architecture of @Arthur Suilin's solution for the web traffic prediction. Encoder and decoder are both GRUs. The hidden states of the encoder are passed to the decoder through an FC layer connector. This is useful to improve the accuracy significantly.

Feature Engineering

For LGB, for each time periods the mean sales, count of promotions and count of zeros are included. These features are calculated with different ways of splits of the time periods, e.g. with/without promotion, each weekdays, item/store group stat, etc. Categorical features are included with label encoding.

For NN, item mean and year-ago/quarter-ago sales are fed as sequences input. Categorical features and time features (weekday, day of month) are fed as embeddings

Training and Validation

Will add to this part later.

onPromotion

As you guys all noticed, there are some problems with the promotion feature here. Besides the missing promotion information for zero sales in training data, I found that the promotion info in 8/16 is quite abnormal and unreliable (e.g. there are ~6000 unseen store-item combo on promotion on 8/16). So I don't think it reliable to use PB to infer the distribution of promotion in test set.

The sales ratio of promo/non-promo items in training set is ~2:1 if I fill missing values with 0. I try to infer this ratio in test data. We know that the proportion of 1s in training promo data is underestimated because I fill all missing values with 0. the proportion of 1s in training set is ~6.3%, while in test set is ~6.9% without 8/16. The proportion of missing value is ~40% in training data. If I assume the true distribution is consistent, the proportion of items that are missing in the training data but actually has promotion is (0.069-0.063)/0.4 = ~1.5%. And as the missing values are all 0s, the true ratio would be (0.063*2 + (0.069-0.063)*0)/0.069 = ~1.83. So I guess our training model is around 10% over-estimating the items on promotion.

Then I simple re-train all the models, but without the promotion information. These predictions will of course have a lower sales ratio between promo/non-promo items. Then I average the no-promo predictions with the original predictions with promotion info with weights so that the sales ratio of the final predictions approaches 1.83:1 as I inferred.

This approach is kind of tricky and based on assumptions I cannot validated, but that's the only way I figure out to deal with the promotion bias. However, it seems like it's not useful to the PB. Ironically, the model without any special dealing with promotion bias gives me .510 on PB. I still have no idea why.

I have shared my codes on github https://github.com/LenzDu/Kaggle-Competition-Favorita, but it is still quite messy there. I will organize it when I have time.

Options

Comments (1)

Sort by Hotness ▾

Click here to enter a comment...

🐳鲲(China) • 5 hours ago • Options • Reply

^ 1 ˅

Excellent, It seems both Arthur and SJV's solution are quite generalize to this kind of problems.

Our Team Terms Privacy Contact/Support