# Favorita

1674 competitors

## Entities:

**Train**

Store (), Product ()

**Test**

Store (), Product ()

## Data:

Date

Store (id, State, city, type, cluster)

Product (id, family, class, perishable)

Onpromotion (Boolean)

Count of sales transactions (train only)

Oil prices

Holidays

Events

**Target:** Unit sales by store/product/date

Benchmarks:

Mean by Product (0.72922)

Last Year (0.91165)

Median of medians by weekday/weekend

## Takeaways:

* Neural networks where the big winners – taking up 6 of top 8 placings
  + Different architectures and features across contestants, but similar results
    - GRU, LSTM, FF, CNN
* Features: median-based methods did surprisingly well
  + Medians of medians for different intervals
  + Weekday vs. weekend
  + Winner did not use this feature engineering!! (same for 4th and 6th place)
  + Using lags/ACF values in neural networks to help learn seasonality and prevent forgetting of long term dependencies works -> Similar findings as from M4
* Segmented approach based on kalman filter and median of medians placed 8th
* Very different training & validation approaches did well
  + Basic time split and window-based sampling for training
  + Reduce whole training set to one row per series and use grouped k-fold to predict other series given current fold series
  + TSCV by number 8
* Ensembling or stacking still used in the top
  + Multiple NN’s
  + Muiltiple checkpoints
  + Weight averaging
  + Stacking of models using XGBoost
* Scaling
  + Median scaling
  + Log1p

## Notable writeups

#1st Place w (0.504/0.509)

* Train only on 3,5 or 1 months from 2017 (different by model)
  + Using more data did not help them!
* Train/test split by time
* Missing values:
  + Filled missing or negative with zeros
* Features:
  + Store, Item, Family, Cluster, Class…
  + Promotion
  + DoW (used in one model)
  + Aggregations:
    - Time windows (1, 3, 5, 7, 14, 30, 60, 140 days)
    - By: Store/item, store/class, item
    - Target: promotion, unit\_sales, zeros
    - Function: Mean, median, max, min, std, days since last, difference between time windows (only for equal tws), Exponentially weighted mean
  + Did not gain use out of holidays!
* Single models:
  + 16 one-step ahead LGBMs (0.506/0.511)
  + 16 one-step ahead NNs (0.507/0.513) (weird NN model – basically dense, but uses LSTM with sequence length 1)
  + 1 LGBM for 16 days (0.512/0.515)
  + 1 CNN Model (sjv) (0.517/0.519)
* Ensembling:
  + Weighted average (0.504/0.509)

2nd Place SoLucky (0.51296)

* Wavenet
* Random slices for training
* Train/test split: Last 16 days as validation
* Uses lags of promotion and sales as in Wikipedia
* Ensembling
  + Average of predictions every X minibatch after warmup
  + Average of 5 models
* Cap to max observed value

3rd Place slonoschildpad (0.51309)

* Ensemble of NN and LGBM
* Models by slonoslon:
  + Trained on 80 days
  + Validated on last 16 days
  + Used mostly NNs:
    - Average of 10 models
    - Average of prediction every epoch
  + Ensemble of: CNN (dilated), LSTM, GRU
  + Embedding for categorical features

4th Place spp (sjv only one part of team) (0.51318)

* Seq2Seq
  + Multi-output
* Dilated causal convolutions
* Decoder modified to use bidirectional LSTM to be able to use future onpromotion values
* Modified architecture to output a Bernoulli parameter to deal with forecasting of zero values
* Features:
  + Log transformed time series
  + Embedding of categoricals
  + Manual features:
    - Lags, diffs, rolling statistics, date features, averages by keys (product/store/..)
* CV:
  + 5% of time series held out -> not time split. To avoid bias towards particular month/week etc of validation set.

5th Place Lingzhi (0.51340)

* Ensemble of:
  + LGBM
    - Similar to public kernels with additional features, data and periods
  + CNN + DNN
    - DNN to raw sales sequence. Embedding of cats + and future promotions
    - Directly output 16 values
  + Seq2Seq RNN
    - GRU -> similar architecture to Suilin in WTF
* Ensembling
  + Weighted average
  + For each NN, average of multiple training runs
* Features:
  + For LGBM
    - For each period:
      * Mean sales, count of promos and count of zeros
      * By key: store/item/(promo/weekday/)
    - Label encoded cat features also
  + For NN
    - Item mean
    - Quarter/Year ago
    - Embeddings categoricals and time: weekday, dom…
* Validation:
  + 16 days validation set
  + Only data from 2017 used for train

6th Place Nicolas (one model on team) (0.51467)

* LGBM on 2017 data
* Over 500 features
  + Aggregations by store/item
    - Median of Moving average
    - DoW average (over time periods 7, 14, 28, 56, 112)
    - Days since first sale in train
    - Quantiles for diff. time periods
    - Promotion
    - Averages for diff. time periods
    - Future onpromotion sum
    - Mean non zero salaes
  + Frequency encoding by item,store,family,class
  + Item-cluster mean for past 5 days
  + OHE of categoricals
  + Weekend and weekday means
  + Means of last 16 days
  + Quantiles of last 16 days
  + Past count of zero sales for diff. time periods
  + Past count of promo days for diff. time periods
  + Past promo sales avg
  + Past dow quantiles
* Last 16 days as validation
* 16 models – one for each horizon
* Trained 5 models using same features and averaged
* Then ensemble with partner

8th Place: CPMP/Giba (0.51504)

* Ensemble of NN and LGBM
* NN:
  + FF (3 layers, ReLu)
  + Embed class and store in 4 dimensional vectors (appended at sec level)
  + Outputs of length 16
  + Last level is 1d conv with output, on promotion input and product of x and on promo input.
* LGB:
  + Similar to starter scripts, same features as NNs
  + One model per day
* LGB Classifier:
  + Predict whether sales are zero, used as part of ensemble
* Validation:
  + Train on December 1 2015 and forward
    - Condense all history to one observation based on features
    - Done for 17 periods -> 17 obs per store/item pair present
  + 2 validation periods:
    - Last 16 days
    - From 17-32 days ago -> for Early Stopping
  + Grouped 10-fold CV on training periods
    - No item/store pairs in diff. folds
  + “We then train one model per fold, as usual in kfold cv, and keep the epoch with the best average cv value on the folds to compute the prediction on the validation data using the average of the fold models predictions. We then use that prediction score as our measure of the trained model”
  + Kfold for parameters, Validation for perf measurement.
* Used sales from a year ago to capture weekly seasonality
  + But training on this (as in WTF) did not work -> seasonality not strong enough?
* Features:
  + Aggregations over time windows
    - Mean
    - Max
    - Proportion of zero
    - Proportion of promo
  + Sales for each of last 7 days
  + Avg sales per weekday over last 8 weeks
  + Sales 364 days ago, promo status 364 days ago
  + Class of item
  + Store
  + Didn’t use oil price or item id

12th Place Antklen (0.51566)

* LGBM and NN
  + Label encoding for LGBM, Embedding for NN
* Two approaches (for both NN and LGBM)
  + One overall model for all horizons
  + One model per horizon
    - Not all features used -> horizon specific, e.g. dow features
* Validation
  + Last 16 days
* Training
  + Tested various levels of data included, from 6 16-day periods to 104
* Features:
  + Time windows:
    - Mean sales for item/store
    - Mean sales for item/store/dow
    - Mean promotion for item/store
    - Mean sales for item/store/dom
    - Mean # zero sales for item/store
    - Mean sales store/item (family, class, city, type, cluster)
    - Mean sales by item
    - Mean # zero sales by store
  + Differences between time window stats
  + LR for each store/item to incorporate trend
  + Mean promo for all 16 days

13th Place Louis T (0.51572)

* DNN
  + 6 layers, Batch norm
* Uses last 2 years of data
* Sliding window
* Separate models for perishable and non-perishable
* Ensembled with public kernel
* Last 16 (32) days as validation set
* Many class based features

15th Place Team Lottery (0.51636)

* Ensemble of LGBM and NN
  + Starter script
* Added features
  + Zero sales counts in windows
  + Latest promo in last 30 days
* NN: feedforward
  + Same features
  + Preprocessed using “RankGauss”
  + Batchnorm, big dropout.
* Last 16 days as validation set