# Grupo Bimbo Demand Forecasting

1968 competitors

## Entities:

**Train**

Stores (880.604) Product (1799), Depot (552), Routes (3603), Channel (9)

**Test**

Stores (745.164), Product (1522), Depot (552), Routes (2608), Channel (9)

## Data:

Week

Depot (Id, Town, State)

Sales Channel (Id)

Route (Id)

Store (Id, Name)

Product (Id, Name)

Sales (pesos, units)

Returns next week (pesos, units)

**Target:** Adjusted demand (Sales – Returns) by Week/Store/Depot/Store/Channel/Route/Product

Benchmarks:

Log Means by Product/Client/Agencia with intelligent fallbacks for non-observed pairs + Adjustments based on LB: 0.4700/0.48818

Medians by Product/Client with fallbacks: LB: 0.50758/0.52838

## Takeaways:

* New ways of ensembling used/stacking:
  + Linear models as features for XGBoost
  + Stacking used by winner
  + Ensembles of multiple models with diff. features
* Xgboost once again big winner
* Key topic due to sparsity of data: How do you encode/extract signal from categorical variables?
  + Sparse linear models (FF and FFM) for handling high cardinality categorical variables
  + Target encoding
* Top performers only trained on 1 or 2 weeks of data, with the remaining data used for feature generation
* Cross-validation setup to mimic leaderboard once again dominated
  + Iterated predictions used by some, direct by others
* Lags and aggregates by categorical variables and combinations were key once again
* Text mining was used to extract more info on the characteristics of the time series
* Seasonality not important due to short dataset available / data very sparse
  + Likely why time series models were not really used
* Simple benchmark gets around 0.48 on private leaderboard -> winner improved by around 10%

## Notable writeups

#1st Place Team Slippery Appraisals (0.42701/0.44260)

* 2nd level ensembling -> stacking
  + Weighted average of ExtraTrees and Linear model
* Validation sets:
  + Week 8 + 9 as validation set on first level
  + Week 9 as validation set on second level
* Reduced training sets
  + Week 6 + 7 or only week 7 for validation set prediction
  + Week 8 + 9 or only week 9 for test set prediction
* Recursive/iterated forecast to calculate lag 1 for test sets second week
* Mostly Xgboost on 1st level
* Features
  + 1-3 week lags of target grouped by factors & combinations (**Key**)
  + Aggregations (min, max, mean, sum) grouped by factors & combinations
    - Previous weeks only??
    - For target
    - For returns and sales
  + Frequency of factors & combinations
  + Product Information using Text Mining
    - Short names
    - Weight
    - Brand
    - # Pieces
  + Grouped products by cluster and used cluster as feature
  + More than 300 features generated
  + Xgboost variable importance used to select features (top 175)
* Ensembling
  + Multiple models trained by diff. people w. diff settings & diff. features

3rd Place Team Mystic (0.42999/0.44408)

* Training set: week 7
* Validation set: W 8+ 9
* Features:
  + Data used:
    - W 3-6 for training set
    - W 3-7 for W8
    - W 3-8 for W9
  + Aggregates by categorical vars and interactions
    - Max
    - Min
    - Median
    - Mean
    - Std
  + Lag 1-3 of Sales & Returns
  + Linear model predictions for capturing interactions
    - Using Vowpal Wabbit, FTRL with Pypy, libFM, libFFM
* One single xgboost model

4th Place: Gilberto & Regis (0.42999/0.44469)

* Week 9 for validation
* Ensembling
  + Weighted avg of 7 models
  + Models without lag 1 features
  + Models with lag 1 based on iterated prediction (by product, client)
  + XGBoost (100+ features)
  + Linear Regression
  + libFM (only sparse features)
* Features:
  + Lag based features
* Single models:
  + XGBoost: 0.433 on public

16th Place ML-Bure (0.43561/0.45133)

* Only trained on week 9
* Features
  + Avg. by client, product, product/city, product/client,
  + Avg by client/product at lag 1, 2, 3, 4, 5
  + Lin reg. FTRL on historical average demand by client/product id & hashed all ids
  + Lin reg. FTRL on all hashed ids and poly4 interaction

18th Place: CPMP (0.43612/0.45287)

* One model per productid
* Week 9 as validation set
  + Local CV well correlated with actual LB
* Train models on 6-8
* Previous four weeks to calculate features:
* Aggregates by categoricals
  + Lags
  + Averages,
  + Std
  + Non-zero values
  + For sales & returns
* For horizon two predictions, only use lag 2 or more features
* Products with very poor results were pooled into one model to allow for pooling of data
* XGBoost used -> Single model

24th Place: aldente (0.43953/0.454562)

* 2 Levels model
  + Level 1: FTRL using W 3-9 predictions used as feature
  + Level 2: Xgboost ensembles
* Strongest features are ftrl preds + lag1 + lag 2 + lag3
* Features similar to 12th place