Grupo Bimbo Inventory Demand

<https://www.kaggle.com/c/walmart-recruiting-sales-in-stormy-weather>

Runtime: 8. Jun. – 31 Aug. ||1968 Teams / 2268 Competitors entered

**Data:**

* Train.csv (?? X ??)
* Test.csv (7m X 7)
* Clente\_tabla.csv – Client names (935k X 2)
* producto.tabla.csv – Product names (2592 x 2)
* town\_state.csv – town and state (790 x 3)

**Data Description:**

* Dataset consist of 9 week of sales transactions.
* Every week, products are delivered to vendor.
* Each transaction consist of sales and returns. Returns are products that are unsold and expired. The demand for a product is defined as the sales this week subtracted by the return next week.

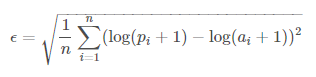
**Goal:**

* Predict demand of a product for a given week, at a particular store

**Benchmark:**

* Medians, Score = 0.52718 (<https://www.kaggle.com/dukebody/benchmarkmedians-0-52718-on-leaderboard>)

**Evaluation:**

* Root Mean Squared Logarithmic Error (RMSLE)
  + 
  + N = observations, p\_i = prediction, a\_i = actual, log(x) = natural log

**Summary:**

* A

# Contributions

**# 2/1 Place: The Slippery Appraisals, Language: Py/R, Score (Pub/Pri) 0.42701 / 0.44260**

* Repo:
  + <https://www.kaggle.com/lyytinen/grupo-bimbo-inventory-demand/basic-preprocessing-for-products>
  + <https://www.kaggle.com/ybabakhin/grupo-bimbo-inventory-demand/products-clustering>
  + <https://www.kaggle.com/mlandry/grupo-bimbo-inventory-demand/h2o-gbm/run/263502>
* Features:
  + Product Feats:
    - Short names, Weight, Brand, etc.
    - Clusters (kMeans)
  + Numerical Feats:
    - Mean, Min, Max, Sum grouped by factors and combinations (Product ID, Client ID, Agency ID, etc.)
    - Similar aggregates calculated by variable factors (Ventahoy, VentaUniHoy, DevUniProxima, DevProxima)
  + Most Important:
    - 1-3 weeks lags of target grouped by factors and combinations, aggregated (min, max, mean, sum) of target by factors and combinations, frequency features of factor variables
  + Only included the 175 top features, as approximated by XGBoost feature importance
* Method:
  + 2-level ensembling.
    - 1st level (Using week 9 as validation set): XGBoost
    - 2nd level: ExtraTrees classifier and linear model
    - Final model: Weighted average of 1st and 2nd level.
  + Training:
    - 6-7 week training data to predict 8-9 weeks sales (validation)
    - 8-9 weeks for predction of 10-11 week sales (test)
    - Since 1 week lag is used as feature this had to be calculated for week 11. Thus, week 10 was predicted and then used in calculating features for week 11.
      * A similar approach was used in validation to predict week 9.
    - Using only data from week 7 to predict weeks 8-9 and data from week 9 to predict weeks 10-11 provided better results than using two weeks.
  + Prediction:
    - 1st level used XGBoost classifiers. 2nd level used validation results of the first level as training sets for predicting test set. Here, a linear model and model based on ExtraTreesClassifier was used. Lastly, the results from the 2nd level models were averaged using weighted averages.

**# 1/3 Place: Team Mystic, Language: Py, Score (Pub/Pri) 0.42690 / 0.44408**

* Features:
  + Shallow Features:
    - Max, Min, Median, Mean, Standard Deviation all on interactions of features.
    - VentaUniHoy, VentaHoy, DevUniProxima, DevProxima, DemandaUniEquil (of last 3 weeks)
  + Deep Features:
    - Trained linear models to get “deep” interaction signal: Vowpal Wabbit, FTRL, libFM, libFFM were all used.
  + Shallow features are simple descriptive statistics, while deep features are linear interaction between all features. Shallow describe the rough shape of these interactions, while deep model all interactions and give predictions based on all features.
  + Think of shallow and “big picture”! and deep as “detailed”
* Method:
  + Used data from weeks 3-6 to generate features for week 7, week 3-7 to generate features for week 8-9. Then train on week 7 to predict week 8 and 9.
  + Similarly, used data from week 3-8 to generate features for week 9 to predict week 10&11.
  + Constructed a single XGBoost model including Shallow and Deep features.
* Conclusions:
  + Predictions from linear models were the most important features

**# 4/4 Place: Gilberto & Regis, Language: Py, Score (Pub/Pri) 0.42999 / 0.44469**

* Method:
  + Uses week 9 for validation
  + Final submission is a weighted average of 7 models.
  + Models used:
    - XGB (+100 features: many combinations of lags)
    - LR
    - libFM (Sparse features only))

**# 11/16 Place: ML-Bure, Language: Py, Score (Pub/Pri) 0.43561 / 0.45133**

* Features:
  + Average Demand for: Client, Product, Product/City, Product/Client
  + Weighted average demand for product by total demand for each client
  + Lag 1, 2, 3, 4, 5 average demand for product/client
  + Predicted demand using FTRL with following features:
    - Historic average demand (before week 9) for each client
    - ProductID, PreviouslyOldProductID: Feature value = Average demand of previously sold product
    - Hash IDs and give them value 1.0
    - Output = demand for each row
  + Predicted demand using FTRL with following features:
    - Hash IDs and give them value 1.0
    - Poly 4 interaction of IDs and give them value 1.0
    - Output = demand for each row
* Method:
  + Treated entire test set as if only one week (week 10), meaning that lag 1 values for week 10 and 11 are the same.
  + XGBoost with 12 features for each row.

**# 15/18 Place: CPMP Language: Py, Score (Pub/Pri) 0.43612 / 0.45287**

* FOCUS ON MEMORY EFFICIENCY
* TimeSlice CV 🡪 Use week 9 as validation and train on week’s 6-8
* Features:
  + Lags & averages on various groupings
  + # of non-zero values
  + Standard deviation
* Method:
  + 1 week predictions:
    - Used past 4 weeks to build features:
      * Weeks 3-5 for 6, 3-6 for 7, 4-7 for 8, 5-8 for 9 and 6-9 for test (weeks 10-11)
      * Missing features are treated as NaN.
    - Models were trained on week’s 6-8 with lag features and validated on week 9.
    - For test predictions week 7-9 with lag features were used.
  + 2 week predictions:
    - CV is similar to 1 week, with one week omitted
      * E.g. weeks 4-6 used to build features for week 8, 7 is omitted.
  + Decomposition lost cross product interactions, to cope with this:
    - Features are based on groupings across products
    - All products with poor local CV are grouped in smaller data set on which XGBoost is applied. (Works well for 1 week predictions but not 2 week.)
  + DID NOT USE ENSEMBLING
* Conclusions:
  + Since individual models are fitted to each, time is a hindrance. Almost 1500 models to build.
  + 1 week ahead models did as well as 2 week ahead models.

**# 41/24 Place: Bimbokémon GO Language: Py, Score (Pub/Pri) 0.43953 / 0.45462**

* Method:
  + Level 1: FTRL with Vowpal Wabbit with week 3-9 to be used as feature in Level 2 (vw\_ftrl)
  + Level 2:
    - XGB with 16 features (Similar to ML-Bure + \_IDs)
      * 2 weeks in each model: (4,9), (5,9), (6,9), (7,9), (8,9)
    - XGB with less features
      * 3 weeks in each model: (6,7,9), (6,8,9), (7,8,9)
    - Strongest features: vw\_ftrl, lag1, lag2, lag3
  + Whole test set is considered as week 10. Both week 10 & 11 use week 9 as lag1, week 8 as lag2, etc.