Rossmann Store Sales

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

Runtime: 2. Apr. – 19 May. ||485 teams entered

**Data:**

* Train.csv (1.02m x 9)
* Test.csv (41.1k x 8)
* Store.csv (1115 x 10)

**Data Description:**

* Historical sales for 1115 Rossmann Stores.
* Train/Test fields (Sales not incl. in test):
  + Store, DayofWeek, Date, Sales, Customers, Open, Promo, StateHoliday, SchoolHoliday
* Store fields:
  + Store, StoreType, Assortment, CompetitionDistance, CompetitionOpenSinceMonth, CompetitionOpenSinceYear, Promo2, Promo2SinceWeek, Promo2SinceYear, PromoInterval

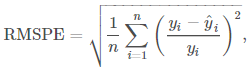
**Goal:**

* Predict Sales in each store on each day.

**Benchmark:**

* Arima (auto.arima): 0.11835 / 0.14323

**Evaluation:**

* Root Mean Square Percentage Error (RMSPE)
  + 
  + N = nrow
  + y = actual sales in single store on single day
  + y\_hat = corresponding prediction

**Summary:**

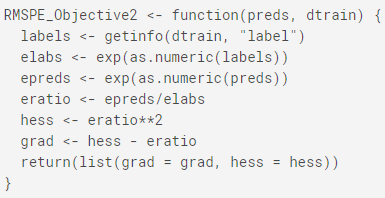
* XGBoost were used by all top-scoring participants
* Making a wide feature pool and then fitting models with differing feature subsets provided good results in most cases.
* Important features:
  + Temporal features: DOW, DOM, etc. as well as average in these horizons.
  + Event counters: Days from/to X events
* Cross-Validation:
  + Those which discuss it, validate models on last 6 weeks of training (correspond to test horizon) – Best used TimeSlice, Most used RandomSample
* Linear Predictions as Feature for XGB gives best of both worlds.

# Contributions

**# 2/1 Place: Gert, Language: Py, Score (Pub/Pri): 0.08936 / 0.10021**

* Features:
  + Recent Data (store specific)
    - Features were extracted with data from:
      * Last quarter, last half year, last year and last 2 years
    - Most contributing factors to variance:
      * Day of week, promotions, and holidays
      * Averages were calculated on combinations of these three
    - Summary features:
      * Centrality: Median, mean, and harmonic mean
      * Spread: standard deviation, skewness, kurtosis, 10%/90% percentiles
    - Additionally, these were summarized on the number of customers instead of the sales amount.
  + Temporal Information:
    - Day counters (express how records relate to certain events or cycles
    - Day counter indicates either number of days before, after or within an event.
    - Events:
      * Promotion Cycle (every 14 days)
      * Secondary promotion cycle (every three months)
      * Summer holidays
      * Store refurbishments
      * Start of competition and start of secondary promotion cycle
    - Day of week, day of month, day / week / month of year
    - Number of holidays during current week, last week and next week.
  + Current Trends
    - Data sets about last quarter and last year similar to data sets for recent features.
    - For each store, fit linear models on: (Ridge Regression is used with default parameters)
      * The day number (extrapolates the trend into the six week period)
      * Day of week
      * Promotions
    - Year over year trend for previous month (not that important)
  + Other information:
    - Store id, assortment, store type
    - Aggregates by store:
      * Average sales per customer, ratio of sales during promotions / holidays / Saturdays
      * Proportion school holidays and proportion of days the store is open.
    - State specific weather:
      * Max temperature, mm precipitation
* Method
  + Validation set: last 6 weeks of training 🡪 TimeSlice
  + All features == Overfitting
  + Method:
    - All Models were made using XGBoost with a focus entirely on feature extraction and selection.
    - Ensembles
      * Handpicked feature combinations
      * Ensemble of 10 models where features are selected at random
      * Final Model: One model with features from 10 best random models + 2 handpicked models
    - Special Traits
      * Trained separate models for May to Sep (test months)
      * Trained “month ahead” models that skipped the most recent month
      * Log transformation of sales
      * Remove zero sales from training period
      * Add constant to all predictions. Adjust for trend
  + Conclusions:
    - Month ahead models performed well, suggesting that recent info was not important
    - Monthly and daily sales average did not improve model
    - Similarly aggregates by day, proportion open and spread of sales were not useful.

**# 20/4 Place: Russ W, Language: R, Score (Pub/Pri): 0.09590 / 0.10621**

* Features:
  + Temporal:
    - WeekofMonth, month, week, day, DayOfWeek, Year
  + Specific Features:
    - Store, Promo, SchoolHoliday, CompDist0, ComOpenSince0, Promo2Since0
  + Averages:
    - MeanLogSalesByStore, MeanLogSalesByState, MeanLogSalesBySTateHoliday, MeanLogSalesByAssortment, MeanLogSalesByPromoInterval, MeanLogSalesByStorePromoDOW, MeanLogCustByStorePromoDOW, MeanLogSalesBySchoolHoliday2Type, MaxTemperature, SunshineHours
* Method:
  + Remove outliers
  + CV over last 3 weeks of training set
  + Objective Function:
    - 
  + Adjusted predictions based on linear model. Adjust for trend
  + Final Model: Geometric Weighted Average of six XGBoost models
* Conclusion
  + Transformations other than Log did not work
  + Google trends did not work
  + Prior detrending did not work

**# 20/4 Place: entron, Language: Py, Score (Pub/Pri): 0.09563 / 0.10583**

* Repo: <https://github.com/entron/entity-embedding-rossmann>
* Features:
  + Number of week/month since completion/promo2
  + Number of and nearest promo2/stateholiday/schoolholiday for previous and next 7 days
  + Storestate, weather, google trends
* Method:
  + Category embedding for categorical variables 🡪 Helps with sparsity and overfitting
  + Neural Network with 3 fully connected layers
  + Final submission: Average of predictions from 10 networks

**# 330/26 Place: Willie Lieao, Language: R, Score (Pub/Pri): 0.10293 / 0.11100**

* Features:
  + Promo lift, Store rank (based on 2013 history), DOW transformations, Christmas days
  + Counters: time since last open / last promo
* Methods:
  + Individual TS and XGB models provide rank > 1000 on leaderboard
    - Combined provide 35 place on private
  + Final Model:
    - Ensemble of Arima, Exponential Smoothing, Exponential Smoothing with seasonal Components, Arima with Covariates On top of XGB Model

**# 22/10 Place: NaiveLearners, Language: R/Py, Score (Pub/Pri): 0.09606 / 0.10839**

* Features:
  + WeekOfYear, QuarterOfYear, WeekStartDate, WeekEndDate, WeekOfMonth, Refurb Flag, DaysBeforeRefurb, DaysSinceRefurb, MedianSales at Store/DOW, Receipt – ratio of total sales, Storetype/Assortment flag, Quadratic Trend by Store, SpecialDay Flag (Holidays), IsPreviousHoliday, IsNextHoliday, TotalSchoolHoliday in week/month, IsSchoolHolidayAdjacentState, Sales/Customer/Store, Sales/Customer/DOW, SaturedayRatio/Promo, SundayRatio/Promo, Promointerval, CompetitionEntered, ChristmasFlag, HolidayFactor, CompetitionFactor, kMeansCluster/StoreSale, DaysToNextOpen, DailySalesVariation/Promo/DOW/Store, DailySalesVariation/Promo/Month/Store
* Method:
  + Tested Models:
    - XGB (Different feature subsets) – R XGBOOST
    - XGB per store (Different feature subsets) – R XGBOOST
    - Auto-Arima with and without Fourier – R Forecast
    - XGB with Auto-arima stacked predictions – R XGBOOST & Forecast
    - XGB on residuals of auto-arima at store – R XGB & Forecast
    - Neural Network – Python Keras
    - RF – R & SKLEARN
    - ET SKLearn
  + Best Model: Harmonic mean of 7 XGB models with different features combined with individual XGB at store level and some post processing for special stores

**# 1/7 Place: SDNT, Language: ??, Score (Pub/Pri): 0.08932 / 0.10784**

* Included XGB Poisson Regression on Customers
* PayDay Loans – I.e. month day 28,29,30 is Monday or 289 is either Thursday or Friday 🡪 Evident increase in sales

**# 41/23 Place: Rhinodaveb, Language: Py, Score (Pub/Pri): 0.09810 / 0.11060**

* Features:
  + Day/Week in Month
  + Month, Quarter, Year
  + Week in Year
  + Promo, Open, Holiday 🡪 All offset by 1 in each direction
  + Number of days Open in week
* Method:
  + Ensemble of 5 models:
    - Top public script XGB
    - Personal XGB with data from all stores
    - XGB for individual stores
    - SKlearn GradientBoostingRegressor – individual store models
    - Support Vector Regression – Individual store models