Rossmann

3303 Competitors (almost 6x Walmart!)

**Entities:**

Stores (1115 Train, 856 Test)

**Data**

Stores (Id, Type, Assortment)

Date

Holiday

Promotion Information

Competition (Distance, Time since Opening)

Open

Customers

External: Weather, Google Trends

**Target: Sales by store by day**

**Benchmarks**

ARIMA Models around 0.12/0.118 on public LB

**Summarizing:**

* Features/Variables
  + Feature engineering/selection more important than model tuning
  + Counters and Time until were used heavily
    - Important for holidays and promotions
  + External variables used by top competitors:
    - Promotions
    - Holidays
    - Weather
    - Competitor information
    - Some used google trends, but not necessary
  + Averages/aggregates by categorical features and combinations
    - Over time windows or over whole history
* Correct trend/bias using linear model on validation set
* Dealing with Overfitting
  + Choosing the correct validation set -> Time series split!
  + Many overfitted public LB
  + Use local CV and check on LB
* Models
  + Xgboost performed well
  + Global models dominated
  + Time series could work as part of stacking, but not on its own due to the effect of promotions, holidays etc.
  + Ensembling works
    - Multiple models
    - Multiple seeds/training runs
    - 5% reduction in error measure by Gert
* Winner: 3% till next competitor on private, 10% to number 26.

**First Place: Gert 0.08936/0.10021**

* Best individual model XGBoost 0.1058 on private
* Lots of feature engineering
  + Centrality measures by store/weekday/promo during last 3,6,12,24 months
  + Spread measures tried – only few kept due to overfitting
  + Ridge regression for each store on weekday/promo/dayssince2000 on last 3 to 12 months to capture linear trend
  + Number of holidays this week, last week, next week
  + Store Aggregates (encodings)
    - Avg sales per customer
    - Promo sales ratio
    - Saturday sales ratio
    - Holiday sales ratio
  + Day Counters (Before, During, After)
    - Promo
    - Promo2
    - Holidays
    - Refurbishments
  + Weather
    - Max temp
    - Precipitation
  + Summer holiday info
  + Time features
    - Month, Year, Day of Month, Week of Year, Day of Year
    - Overfitted before features were reduced using feature selection
* CV using Last 6 weeks of training
* Feature selection:
  + using CV and models trained on random subsets of features (500 models)
  + Ensembles pairs and checked score, kept the best using backwards elimination
  + Combined features selected in the best models and trained one model
* Ensembling
  + Models trained only on May to Sept (from all years)
  + Models not using data from most recent month to create preds for September
  + Models fit on both customers and sales
* Very few submissions -> relied on internal CV and used LB only to check progress against overfitting

**Second Place: Nima Shahbazi 0.09072/0.10386**

* Discovered relationship with zero sales and boosted sales before/after (closures)
  + -1 on closures and counter before/after to allow for learning this
  + Time into promotion /holiday etc, -> time till end
  + Counters before/after holidays?
* Used time series features: MA’s
  + Fast moving average (low k)
  + Slow moving average (low k)
  + For both sales and customers over different windows
  + Split by store, dow, promotion
* Used XGBoost and ensembling of multiple model

**Third Place: entron 0.09563/0.10583**

Neural Network with Entity embeddings

Embeddings + Input -> Concatted

Dropout (0.02) -> Dense (1000) -> Relu -> Dense (500) -> Relu -> Dense(1) -> Sigmoid

5 NN’s trained and average to reduce training variability.

* For all categoricals: date information (year, month, day, dow, week) and state, store
* Promo variable used
* Promo Interval2
* Promo interval
* HasCompetitionMonths
* HasPromo2Weeks
* Months since Latest Promo2
* Time till/since promo/holiday (7 day window size)
* Holidays/Promo in upcoming 7 days
* Holidays/Promo in last 7 days days
* Open
* Holidays
* CompetitionDistance
* Weather
  + Temp
  + Humidity
  + Wind
  + Cloud
  + Weather Event
* Google Trend
* StoreType
* Assortment
* State

**Fourth Place: Russ W 0.09590/0.10621**

* Ensemble of XGboost models -> 6 models
* 25 features at most per model
* Final 3 weeks as validation, since final 6 weeks had a very hot week he was afraid would not represent test set
* Removed outliers
* Best single model: 0.10713 on Private
  + WoM
  + Month
  + Week
  + Day
  + Store
  + Promo
  + DoW
  + Year
  + School Holiday
  + CompDist
  + CompOpenSince
  + Promo2Since
  + MeanLogSalesBy
    - Store
    - State
    - StateHoliday
    - Assortment
    - PromoInterval
    - StorePromoDow
    - SchoolHoliday2Type
  + MeanLogCustomerBy
    - StorePromoDow
  + Max Temp
  + Sunshine hours
* Did not use hold out when computing these features
* Custom Hessian in Xgboost -> not sure why this worked
* Avg results from three diff. seeds
* Linear adjustment by use of linear model on holdout set to correct bias?

**10th Place: Bishwarup B 0.9606/0.10839**

* Models tested
  + XGB
  + XGB per store
  + Auto arima (+ with fourier)
  + XGB with auto.arima stacked
  + XGB on the residuals of auto.arima
  + Neural network
  + RF
  + Extreme Trees
* Best model:
  + Harmonic mean of 7 XGB Models + 1 store level XGB model
  + Postprocessing for some problematic stores
* Features:
  + WoY, Quarter, WeekStartDay, WeekEndDay, WoM, Refurbishment
  + Counters:
    - Day before refurbishment
    - Days since refurbishment
    - Holidays
    - Time in promocycle
  + Encodings:
    - Median by store/dow
    - Mean Sales per Customer by store
    - Mean sales per customer by store/dow
    - Ratio of promo/non promo sales by store/dow
    - Median customer by store
    - Median customer by store/dow
    - Saturday & Sunday ratio by store/promo
    - Var by store/promo/dow
    - Var by store/promo/month
  + Pct of storevolume out of store/state/type/assortment cluster
  + Store/Assortment type variable
  + KMeans cluster based on sales
  + Stores with competitor entrances in train period
  + Weather
  + Trends

**26th Place: Willie Liao 0.10293/0.11100**

* Combined Xgboost and time series models: Stacking
* Removed promo lift and holidays based on previous year for time series models
* Time series models:
  + ARIMA + X
  + ES + ES Seasonal
* Use Xgboost on residuals:
  + Store rank, i.e. a type of categorical target encoding
  + Weekday rank
  + Added Christmas shopping days
  + Counter: days since last open, days since last promo
  + Google trends
  + Weather