# Rossmann Stores: Daily Customer Prediction

Andrew Sang
2019.03.06
404 Final Project

# Business Question

Rossmann is Germany’s second-largest drugstore chain with over 3,000 drug stores in 7 European countries.

Currently, Rossmann store managers are tasked with predicting daily store sales in order to create effective staffing schedules. This is both time-consuming and is difficult for managers to do, as there are many variables that need to be accounted for. An accurate staffing schedule is key to balancing employee burnout and wasted hourly salaries.

The ask here is to **develop a forecasting method that will predict daily customer volume at the store level**, in order to aid staffing schedules.

# Data & Evaluation Metric

1. **Stores:** 1,115 stores in dataset
    - More: Store Type, Nearest Competitor Distance and Open Date
2. **Dates:** Jan 2013 to July 2015 (942 days)
3. **Day Info:** Open / Promo / State Holiday / School Holiday

- Root Mean Squared Error (**RMSE**) at the Store-Day level
    - This is **similar to the average error for each Store and Day**. The only difference is that there is a bigger penalty placed on larger errors since the errors are squared, averaged, and then square rooted.

# Experiment Setup

Train

Test

Valid

1/1/13

5/9/15

6/20/15

7/31/15

859 days

42 days

42 days

Rossmann Managers are tasked with creating 6 week schedule blocks so we are testing on 6 week chunks.

# Processing Steps

- **Differenced Response Variable:** Most series I observed were stationary, however some were, so differenced in order to make sure that there wasn't some trend that would bias the time series forecast

- Fortunately, **no missing values**, so no need to impute.

- In terms of feature engineering, utilized a lagged version of the response variable (pulled in last 35 days). This was based on looking at the autocorrelation plot as well as testing out different lag lengths.

# Prophet (Baseline)

Prophet is an additive regression model from Facebook.

- Pros
  - Quick / Easy to Implement
  - Automatically picks the “best settings”
  - Recognizes weekly, monthly, yearly seasonality
- Cons
  - Not a lot of fine-tuning that the end user can choose
  - If there is some interaction relationship, model may not pick up on it.

- Results
  - “Error” (RMSE) of 116 per Store/Day

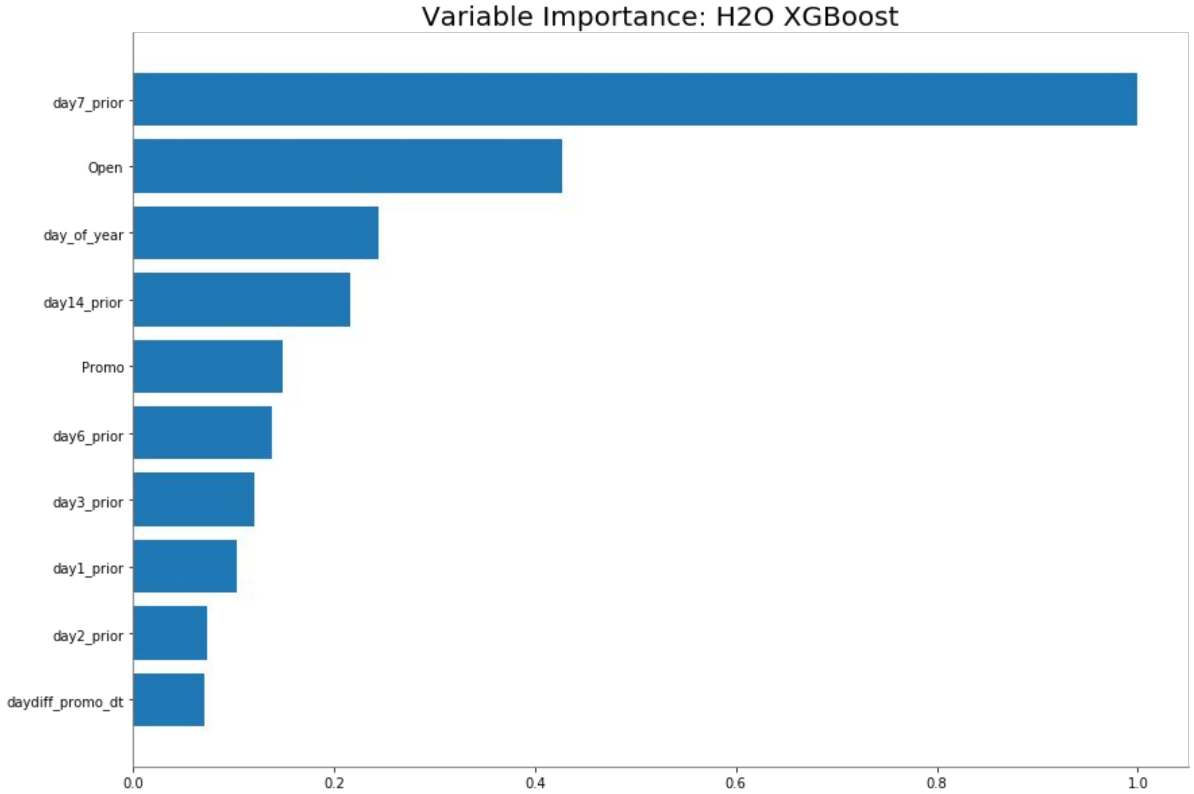
# XGBoost

XGBoost is a regularized tree-based model, where trees are iteratively built on top of each other. The model works by weighting errors that the previous trees got wrong. It controls for overfitting by using a regularization technique.

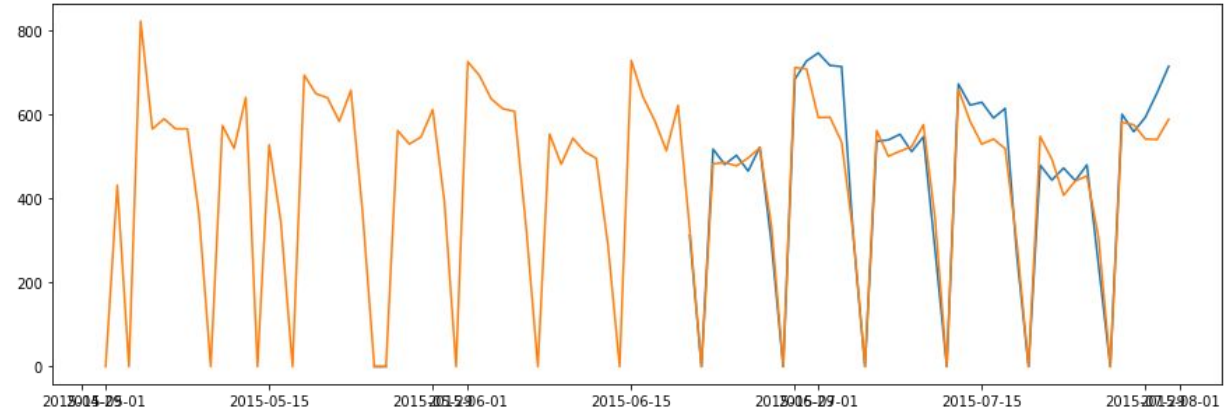
Results: RMSE of **94** (from 116)

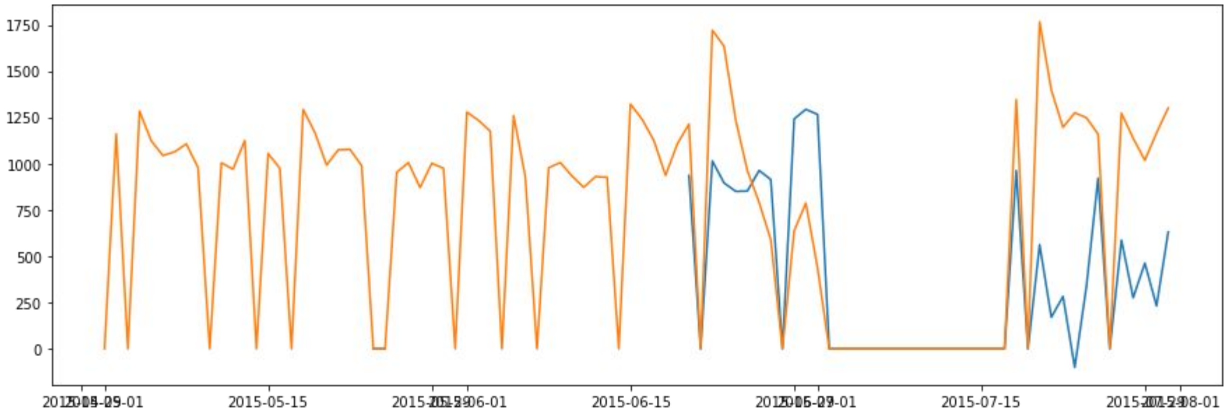
- Can think of this as on average, the model was off by 94 customers
- The middle 50% of the errors fell between -63 and 26 customers
- For comparison, the middle 50% of stores had between 506 and 848 customers on any given day

# XGBoost Variable Importance Plot



# XGBoost (Good / Bad) Areas





- Does well when there is very well defined seasonality (weekly)
- Does poorly when there are large gaps in time when the stores are closed for a prolonged period (rare)

# Business Application

- **Output:** For a first iteration, it would make sense to deploy this as some sort of web app that managers could enter customer sales in. Ideally, this data would already be available and we could run the model and provide a scheduled automated output to managers.

- **Analysis Next Steps:** However, the next task should be to find out what is the optimal level of staffing that we need, given we know the number of customers in a store on any day?
  - Consider: wages/cost, productivity, satisfaction, employee turnover

# Recap and Next Steps

- **Improvements:**
  - Model did not perform very well for stores that were closed for a prolonged period of time. Consider generating a separate model trained on Stores who had prolonged closures.

  - Alternatively, train each store on its own model to see if that improves accuracy. Also, see if it’s possible to grab more information, if we could get the actual city/country of the store as well as location, we could layer in demographic and weather data to see if that is predictive.