Walmart Weekly Sales Forecast

Chu-Yun Hsiao, Hao-Chun Niu, Shu-Yun Liu, Yu-Chun Peng

2022.01.14

Executive Summary

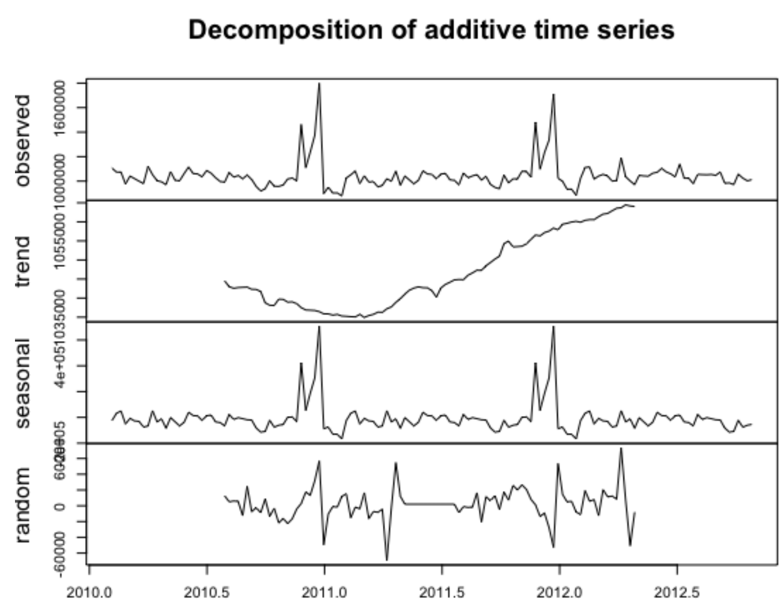
We are data scientists who are passionate about retail industry. We also notice that time series is a relatively important topic in sales prediction. Hence, we decided to take sales data from Walmart (2010~2013) as an example and created a time series model to predict 45 different Walmart stores' sales. We ended up using same parameters and machine learning methods to develop Forecast Models for each store. Thus, we have total 45 Forecasting Models for these 45 stores.

1. Data Description

Before going into the detail of our models, let's look at the data first. Among all the variables, 8 variables were used. The output variable is weekly_sales. The table below shows the name, data type and description of each feature. It's worth noting that since all features are store level, only weekly sales data is department level, we summed up all departments' sales data into single store level sales data.

Features	*Data Type*	*Description*
store	Numeric	The number of each unique store
date	DateTime	The date of weekly sales data
weekly_sales	Numeric	The number of sales of each week
isHoliday	Categorical	The week is a special holiday week of not
temperature	Numeric	The average temperature in the region
fuel_price	Numeric	The cost of fuel in the region
CPI	Numeric	The consumer price index
unemployment	Numeric	The weekly unemployment rate

The graph below shows the decomposition of additive times series of the average weekly sales of all stores. We observed that there is seasonality. The peak of sales often appeared around Thanksgiving and Christmas. Also, the trend of sales number is ascending overtime.



2. Data Visualization

To have a deeper understanding of the data, we visualized the data with Tableau. You can access the dashboard via link1 and link2.

From the visualization, we can find some interesting insights about the data. First, we can tell there's seasonality in the sales data, which will affect how we choose the machine learning method. Secondly, the uptrend of sales is obvious. Surprisingly, unemployment rate does not have strong relation with the number of sales. Last, we observed that CPI is relatively stable and only increased slightly overtime, it does not have significant relation with sales number.

3. Data Split

In order to prevent overfitting, we split the labeled data into three partitions: train, validation and test. We used the train data to build models, used the validation data to select and fine-tune the models and used the test data to evaluate the performance of our selected model. Since we have a time series data, we executed the data split based on the order by time.

Dataset	*Number of Rows*	*Percentage*	*Duration*
Train	113	62.1%	2010-02-05 ~ 2012-03-30
Validation	30	16.5%	2012-04-06 ~ 2012-10-26
Test	39	21.4%	2012-11-02 ~ 2013-07-26

4. External Factor Prediction

Part of our test data has missing values in CPI and unemployment rate, thus we wanted to solve this problem before modeling. There is no data from 2013/5/3 to 2013/7/26. Therefore, we predicted the values ourselves by applying the moving average from the previous 5 weeks' CPI and unemployment rate.

5. Modeling

SARIMAX

First, we tried the SARIMAX model to forecast future weekly sales. SARIMAX models are powerful and widely-applied models to forecast time series data. We can break down the model into five components to get a better understanding of the model structure.

- **S: Seasonality.** Seasonality is that in the time series data, there exists repeatable and predictable patterns.
- **AR: Autoregression.** A model that uses the relationship between an observation and some number of lagged observations
- **I: Integrated**. The use of differencing of raw observations to make the time series stationary.
- **MA: Moving Average.** A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.
- **X: Exogenous Variable.** The exogenous variable is used to model the remaining error, including information that is not recurrent over time.

SARIMAX models are usually denoted as (p,d,q)(P,D,Q) S. Each of the parameters indicates the specific SARIMAX model being used. (p,d,q) are non-seasonal orders, p is the autoregressive order, d is the differencing order and q is the moving average order. On the other hand, (P,D,Q) are seasonal orders, P is the seasonal autoregressive order, D is the seasonal differencing order, Q is the seasonal moving average order and finally S is the number of time steps per cycle.

Before training the model, we first conducted a Dicky-Fuller test to ensure that our data is stationary. The p-value is extremely small (0.0000001), so we were quite certain that we could continue the next steps for modeling.

The strength of using SARIMAX is that it is easy and can automatically generates the best values for (p,d,q)(P,D,Q) S. The best parameter found after automatic selection is SARIMAX(2, 0, 2)x(1, 0, 0)52.

After training the model with the training data and evaluating the performance based on validation data, ***MAPE = 5.7%*** using SARIMAX. Despite the result seemed satisfying, it took too much time to run the models, therefore we kept on investigating other models to forecast the weekly sales.

Prophet

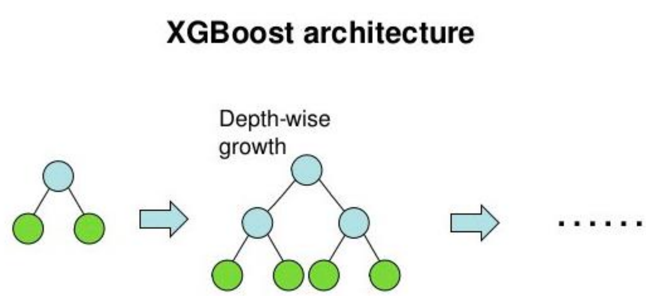
Secondly, we tried the Prophet Model to forecast Walmart's weekly sales. Prophet is based on a generalized additive model, which means that it consists of nonlinear terms that are added together. Three different nonlinear terms included in Prophet are: Trend, Seasonality (weekly, monthly, yearly), and Holiday effects. Since Prophet is good at working with time series that have strong seasonal effects, we thought it would be a good choice for us to use when dealing with Walmart's historical dataset containing an obvious seasonality.

After executing hyperparameter tunning using grid-search with cross validation, the best model we found is the model with s***easonality_prior_scale = 20***, ***changepoint_prior_scale = 0.1***, and ***holidays_prior_scale = 10.***

Fitting the parameter back to our Prophet Model, we then trained the model with the train data and evaluated the performance based on validation data. The overall performance is ***MAPE = 11.83%.***

Ensemble Model – Prophet + XGBoost

Thirdly, although Prophet is good at dealing with time series that have strong seasonal effects, one of its drawbacks is that the regressors are modelled linearly, which against the fact that most data are not linear in real life. On the other hand, XGBoost Model is an optimized distributed gradient boosting library designed to be highly efficient and outstanding in learning non-linear decision boundaries, but is poor at coping with trend and seasonality data. Hence, we came up with this ensemble model combining Prophet and XGBoost as an optimization solution that can complement the drawbacks of each model with their respective advantage. Below is the plot of a simple XGBoost's structure.

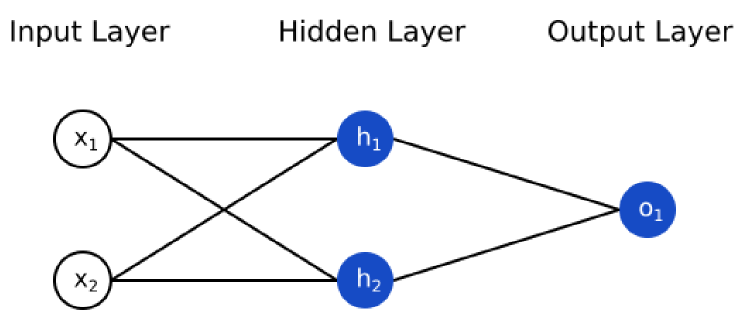


The ensemble process can be divided into three steps. Firstly, we trained the Prophet Model with the best parameter we found in the previous hyperparameter tuning section and made the prediction on the train data. Secondly, we extracted the "trend", "holiday", "weekly", and "yearly" columns from the prediction and combined these columns with the original train data to create a new train dataset. Lastly, we trained the XGBoost model with the new train dataset and evaluated the performance based on validation data. The overall performance is ***MAPE = 5.9%***.

Neural Network

Lastly, we tried the Neural Network Auto Regression Model to forecast the weekly sales. This method is a really widely used and powerful method. It has three important strengths. First, it is simple to use and does not require too much manual optimization. Next, it is extremely accurate and powerful. Hence, we also have to be aware of overfitting. Yet, by splitting the train, validation, and test datasets, we mostly avoided the overfitting problem. Last but not least, it is extremely fast. Hence, it is the perfect tool for large datasets.

Generally speaking, Neural Network works like human brain. By accepting the inputs and aggregating them with the hidden layers, it can accurately predict the outcome variable. Below is the plot of a simple Neural Network's structure.



After trying different arguments and optimization, we found the best model is the model with ***size =10*** and ***decay = 0.1***, meaning that there will be 10 nodes in the hidden layer and weight decay with 0.1.

We trained the model with the train data and evaluated the performance based on validation data. Eventually, the performance is outstanding with ***MAPE = 5.5%***.

6. Conclusion

In the end, we chose the Neural Network Models as our final model. There are 2 main reasons. First, it has the best performance. Second, its time cost is much less than the other methods. To be more specific, its computing time is ***100 times less*** than SARIMAX's computing time. Therefore, we chose the Neural Network Models as our final model.

Method Name	**MAPE**	**Time Cost (mins)**
Neural Network Regression	5.5%	2
SARIMAX	5.7%	150
Prophet + XGBoost	5.9%	37
Prophet	11.83%	145

The plot shows the past data and final forecast of average weekly sales of all 45 stores. The red line represents the past data, and the blue line represents the forecast data. As you may tell, we greatly captured the trend and seasonality within the sales.