Store Item Demand Forecast Challenge

By Mimi Trinh

INTRODUCTION

This report details the submissions for the store item demand forecast challenge on Kaggle from the beginning till the end of the project. The competition provides five years of historical store, item, and sales data and asks to predict three months of sales for 50 different items at 10 different stores. The challenge provides an opportunity to explore different time series techniques on a relatively simple and clean dataset. Once submission is posted on Kaggle, it generates a private score and a public score for each submission to show how well the predicted values match the actual values. Though Kaggle doesn't reveal actual values of the following three months of the sales data, the scores are helpful to determine the best models among all those generated in this project. The score metric used here it SMAPE.

ANALYSIS

Section 1: Problem and Significance

The problem used in this competition is starts with the train dataset of sales of 50 items at 50 stores from 1/1/2013 to 12/31/2017, which makes it a complicated time series analysis since we have four variables in the data: date, sales, store, and item. Essentially, the sales of each item at each store is a five-year time series. Thus, putting it all together, we have 500 time series forecasts to run. In addition, 2016 is a leap year with 366 days, so that will affect the regular frequency of 365 on daily time series. As a result, below are questions that need to be answered to proceed with the project.

- Can we generate a model to run all 500 forecasts at once? Or do we generate a predictive model for each time series?
- How do we handle leap year?
- How do we handle missing value, if any?
- How do we transform the CSV file to a time series in R?

The problem is significant because this competition poses a real-life question that is frequently faced in business such as retail, manufacturing, sale, etc. Academically, the dataset typically only includes one or two variables i.e.: forecast sales given the impact of marketing. This type of analysis is useful on an ad hoc basis such as when the marketing team needs to evaluate the effectiveness of a campaign. However, on a regular basis, constant forecasts need to be generated to stock the items on the store shelves, order parts at the factories, etc. This type of analysis typically involves more variables in the dataset, and the level of forecast is deeper. As a result, the store item demand forecast challenge poses a significant problem that can be applied in real-life business applications.

Section 2: Literature Review

When a time series can be aggregated or disaggregated with multiple categories, we consider it a hierarchical time series. In this case, sale is the target variable that we need to forecast for the next three months or 90 days. Sale can be disaggregated into store and item, so there are two categories in the time series: 10 stores and 50 items. Thus, the dataset in this project represents a hierarchical time series. Using hts() command to address hierarchical time series, R automatically generates the best ARIMA and ETS models that fit the train dataset the most. ARIMA and ETS model classes are chosen for this analysis because they are the most popular methods used in time series and many successful applications. Below are five examples from peer reviewed journals of how ARIMA and ETS are utilized in time series analysis.

- Use ETS model to forecast pig price in China. Specifically, this example uses grey double exponential smoothing, an extension of ETS for trend time series data. The results of the analysis prove to be accurate since they match the intended effects of the government's intervention in the industry (Wu, Liu, and Yang, 2016).
- Use ETS model to forecast monthly inventory of merchant wholesalers in the US. Specifically, this example uses simple exponential smoothing, damped trend exponential smoothing, and ARIMA. The results show that the ARIMA model performs the best among all options (Sbrana and Silvestrini, 2014).
- Use ARIMA model to evaluate the effectiveness of intervention programs. Specifically, the research question asks if the district-wide drug abuse prevention program interrupts the series of drug-abuse behaviors observed in middle school children. If the trend or seasonality is broken, it supports the positive impact of the program (Braden and Gonzalez, 1990).
- Use ARIMA model along with wavelets and artificial neural networks as hybrid models to forecast natural gas prices. The need for this forecast arises as natural gas loses its relationship with crude oil after the unconventional gas revolution (Jin and Kim, 2015).
- Use ARIMA model to measure changes in behavior occurring during nursing period. "A goal of clinical research is to identify, describe, explain, and predict the effects of processes that bring about therapeutic change over an entire course of treatment… Research questions are often: Can patterns of change be reliably identified? Are these patterns of change related to the outcome?" (Jensen, 1990).

Section 3: Descriptive Statistics and Data Processing

When we first load the train CSV file into R, it's recorded as a data frame with four columns: date as a factor along with store, item, and sales as integer. The descriptive statistics of the dataset is shown in exhibit 1. There's no NA in the dataset, so there's no missing value. The first three columns (date, store, and item) don't have meaningful descriptive statistics whereas the last column (sales) is interesting since that's the target variable. Specifically, the min of 0 shows that there are items at stores with no sale on particular days. Also, there's no data entry error since there's no negative value, which means that we can't have a negative sale. The max of 231 shows the highest record of sales. On average, we can expect to sell 57 units of each item at each store on one day. The median and mean are little different, so the dataset might be slightly skewed but not significantly.

Since store and item variables are in integer format, we convert them to character. For example, store value of “1” is converted to “store01” whereas item value of “1” is converted to “item01.” Then we combine store and item together to create a new variable called “level.” For example, an observation with store=1 and item=1 has a level value of “store01item01.” Since we already have this level variable, there’s no need for the sale and item columns anymore. Thus, we drop them from the dataset.

The next step is to handle the extra date in the 2016 leap year. In this case, 2016 has 366 days, which will make it different from the frequency of 365 days in other years. There’s a possibility that sales significantly go up or down on 2/29/2016, but that’s outside the scope of this project. Therefore, we remove the 2/29/2016 records across all 10 stores and 50 items.

To use the hts() command in R to convert the current data frame into a hierarchical time series, we create 500 columns to represent each level from store01item01 to store10item50 and load the appropriate sales data into each column. We narrow down the dataset to only the sales and 500 level columns along with the first 1825 rows, which represent 365 days of 5 years from 2013 to 2017. Then we convert the data frame into a hierarchical time series using ts() and hts() command in R.

An autoplot is created to visualize the data (exhibit 2), which shows seasonality from 2013 to 2017 for each of the 10 stores. When we set up the hierarchical time series, store is the first category, and item is the second category, which explains why the autoplot is organized by store and not item. The autoplot shows that the movement of each year is very similar to each other, so perhaps we don’t need the full dataset for modeling purpose. However, the autoplot has too much information and can be difficult to view. Thus, we break it apart by first showing only level 1 of store category (exhibit 3). This graph is more obvious that there’s similar seasonality between each year and a slight upward trend from beginning till end. The chart for level 2 of item category (exhibit 4) also confirms that there’s seasonality YOY and a slight upward trend. However, since there are 50 items and only 10 stores, this plot is more difficult to view than the previous one due to too much information. We can’t combine all levels together because figure margins are too large. We unfortunately get an error message from R.

Section 4: Modeling Development and Coding Formulation in R

When we try to include the full dataset of five years of historical data into modeling, R stops working and crashes. The dataset is simply too large for R on a personal computer to handle. Given this constraint and the fact that there’s similar seasonality between each year, we’re confident to use only 2017 data on the first two models and 2016 to 2017 data on the last two models. Specifically, the two model classes we build in this project are ARIMA and ETS since they’re the most popular techniques to handle time series analysis and forecasting with many successfully examples as demonstrated in the literature review section. Both ARIMA and ETS techniques can be developed using forecast() command in R. Specifically, we build the following four models for this project to predict the next 90 periods or three months values of sales. The forecast() command by default only generates aggregate prediction at the top level. Therefore, in order to get prediction at the most detailed level for each of the 10 stores and 50 items, we use the aggts() command after. This is how we formulate the models in R.

- Model #1: ARIMA model using the last year of historical data in 2017
- Model #2: ETS model using the last year of historical data in 2017
- Model #3: ARIMA model using the last two years of historical data in 2016 and 2017
- Model #4: ETS model using the last two years of historical data in 2016 and 2017

After applying the forecast() and aggts() commands to build each model, we convert the predicted time series results into data frames with the correct ID and forecast sales values for each store and each item in the appropriate time order. This step is necessary since we need to write a CSV output file to match the submission template on Kaggle.

Section 5: Performance and Accuracy

Typically, in time series forecasting, we use multiple metrics such as MAE, MAPE, MSE, etc. However, in this competition, since Kaggle has the actual value and their own error metric to score and evaluate submitted models, we follow that approach and use the Kaggle metric, SMAPE, to assess the performance and accuracy for each model. SMAPE stands for systematic mean absolute percentage error and has the following calculation in which A_t represents actual values and F_t represents forecast values.

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$

In Kaggle, we have a private score and a public score. Below are the scores for all four models.

	Model #1	**Model #2**	**Model #3**	**Model #4**
Private Score	**27.71900**	27.78405	27.71900	27.78405
Public Score	**18.28989**	18.42181	18.28989	18.42181

The Kaggle score is based on the error metric of SMAPE, so the lower the score, the better the model. In this case, models #1 and #3 have the same results whereas models #2 and #4 also have the same results. This confirms our theory earlier that we may not need the full dataset to develop models for this project because adding one more year of historical data in 2016 doesn't improve the model fit. The rule of thumb in modeling is the simpler the better as long as we don't lose accuracy. Therefore, models #1 and #2 are better than models #3 and #4. Furthermore, when we examine the scores for models #1 and #2, both private and public scores for model #1 are slightly lower than the ones for model #2. Therefore, we conclude that among all four models, the best option is model #1, which is the ARIMA model applied on the last year of historical data in 2017.

Section 6: Limitations and Recommendations for Future Research

The first limitation of this project is the fact that R crashes and stops working as we load the full dataset from 2013 to 2017 to the model. Therefore, it's recommended that researchers should use a commercial computer with larger memory to run the analysis for future work.

The second limitation is the high SMAPE score of the models. The private leaderboard on Kaggle is calculated with approximately 66% of the test set whereas the public leaderboard is calculated with approximately 34% remaining of the test set. The lowest score of the best winner on private leaderboard is 12.58015 with 56 submission entries in one month. The lowest score of the best winner on the public leaderboard is 13.83614 is 45 submission entries in one month. Those error metrics represent a 1.20% and 32% difference comparing to the best model in this project. Therefore, it's recommended that future researchers should summit more entries and take longer, at least a month, to work on this project.

The third limitation is the lack of diversity in models. In this project, we only use ARIMA and ETS. However, the literature review shows that both ARIMA and ETS models can be applied with other techniques such as wavelets and artificial neural networks generate models with higher performance. As a result, it's recommended to apply other techniques such as wavelets and artificial neural networks to improve this study.

The fourth limitation is the lack of analysis on the impact of leap year. Specifically, we delete all observations on 2/29/2016 in this project to keep the 365-day frequency consistent across all five years. However, future researchers should conduct a separate study to analyze the impact of leap year on 2/29/2016 on sales across 10 stores and 50 items.

CONCLUSION & LEARNINGS

In conclusion, after preparing the data into the hierarchical time series format and develop four models using forecast() command, we determine that model #1 using ARIMA model to apply on only 2017 data is the best model with the lowest SMAPE error score. From building these models, we have the following learnings.

- Sometimes it's not feasible and/or necessary to utilize the full dataset. In our case, the using only one year of data provides the same results as using two years of data.
- Beside hts() and forecast package, there are many other techniques in R to apply on time series analysis such as tbats() and prophet package.
- Sometimes ARIMA and ETS models yield very similar results since the Kaggle scores on these models are very close to each.
- Data preparation is a crucial step of a forecasting project. In this study, if we can't convert the data frame into time series, we can't proceed with the analysis.

Appendix

Braden, Jeffrey P., and Gerardo M. Gonzalez. "Use of time-series, ARIMA designs to assess program efficacy." *School Psychology Review*, vol. 19, no. 2, 1990, pp. 224+, web.a.ebscohost.com.turing.library.northwestern.edu/ehost/detail/detail?vid=0&sid=5c3ebd0d-7d9a-4b23-a3e6-04575ccf1151%40sessionmgr4009&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=9607015442&db=tfh. Accessed 3 Nov. 2018.

Jense, Louise. "Guidelines for the application of arima models in time series." *Research in Nursing & Health*, vol. 13, no. 6, Dec. 1990, pp. 429-35, search.library.northwestern.edu/primo-explore/fulldisplay?docid=TN_wj10.1002%2Fnur.4770130611&context=PC&vid=NULVNEW&lang=en_US&search_scope=NWU&adaptor=primo_central_multiple_fe&tab=default_t. Accessed 3 Nov. 2018.

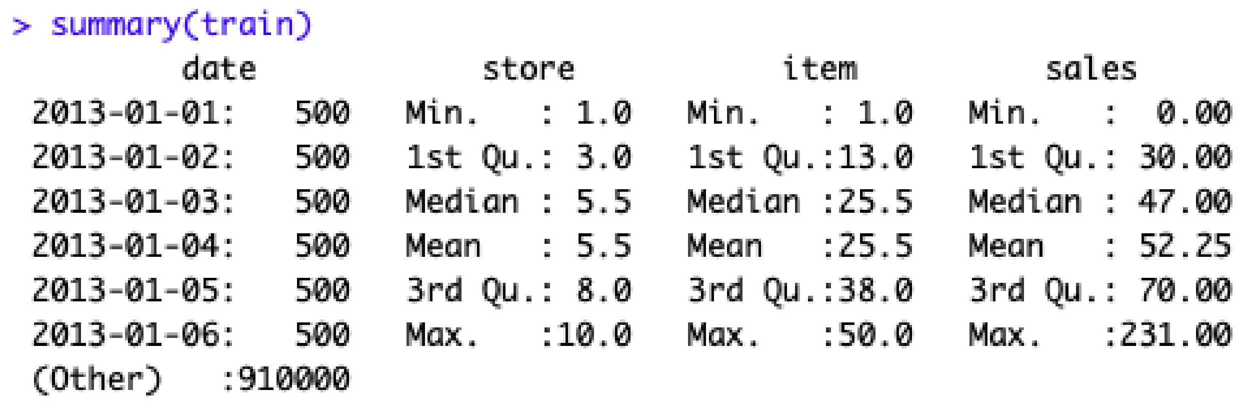
Jin, Junghwan, and Jinsoo Kim. "Forecasting Natural Gas Prices Using Wavelets, Time Series, and Artificial Neural Networks." *PlosvOne*, vol. 10, no. 11, 5 Nov. 2015, search.library.northwestern.edu/primo-explore/fulldisplay?docid=TN_doaj_soai_doaj_org_article_0b68e9245b2847f28258bf8efa756ef0&context=PC&vid=NULVNEW&lang=en_US&search_scope=NWU&adaptor=primo_. Accessed 3 Nov. 2018.

Sbrana, Giacomo, and Andrew Silvestrini. "Random switching exponential smoothing and inventory forecasting." *International Journal of Production Economics*, vol. 156, Oct. 2014, pp. 283-94, www-sciencedirect-com.turing.library.northwestern.edu/science/article/pii/S0925527314001996. Accessed 3 Nov. 2018.

Wu, Lifeng, et al. "Grey double exponential smoothing model and its application on pig price forecasting in China." *Applied Soft Computing*, vol. 39, Feb. 2016, pp. 117-23, www-sciencedirect-com.turing.library.northwestern.edu/science/article/pii/S1568494615006535. Accessed 3 Nov. 2018.

Exhibit

Exhibit 1



```
> summary(train)
         date              store            item           sales
 2013-01-01:   500   Min.   : 1.0   Min.   : 1.0   Min.   :  0.00
 2013-01-02:   500   1st Qu.: 3.0   1st Qu.:13.0   1st Qu.: 30.00
 2013-01-03:   500   Median : 5.5   Median :25.5   Median : 47.00
 2013-01-04:   500   Mean   : 5.5   Mean   :25.5   Mean   : 52.25
 2013-01-05:   500   3rd Qu.: 8.0   3rd Qu.:38.0   3rd Qu.: 70.00
 2013-01-06:   500   Max.   :10.0   Max.   :50.0   Max.   :231.00
 (Other)   :910000
```

Exhibit 2

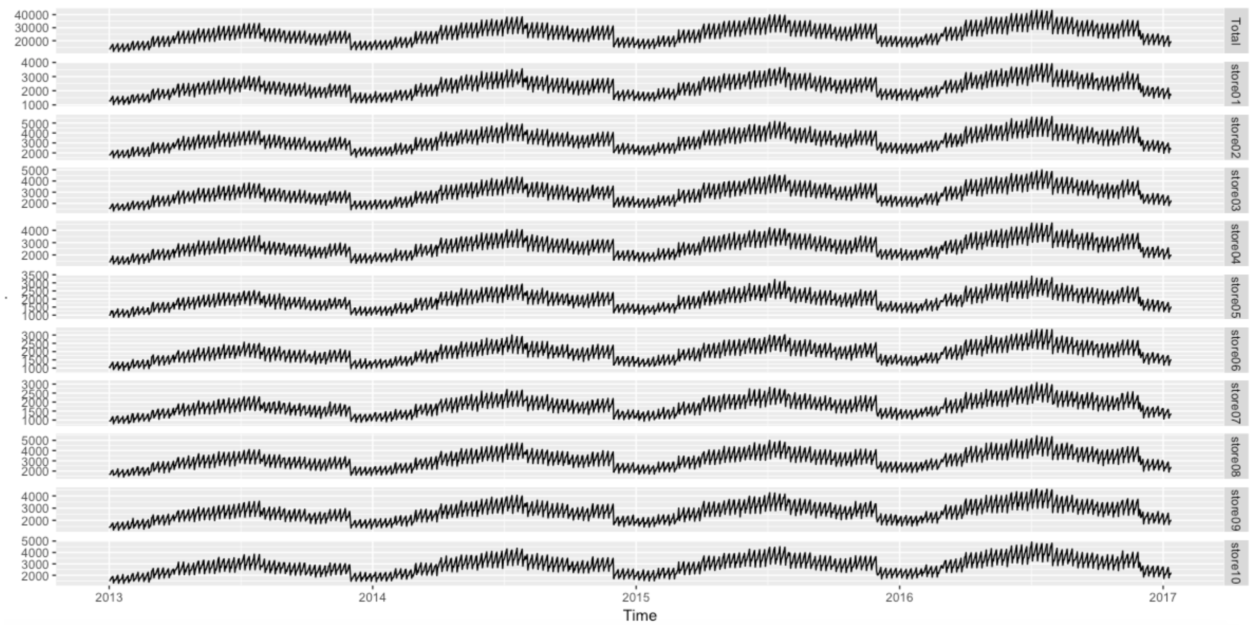


Exhibit 3

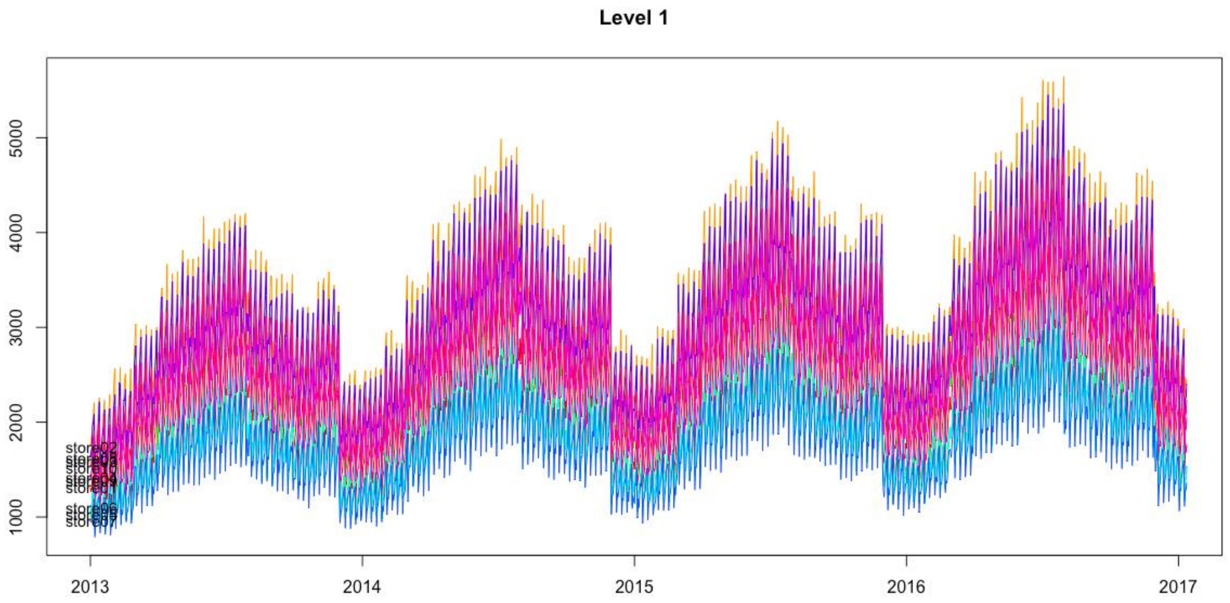
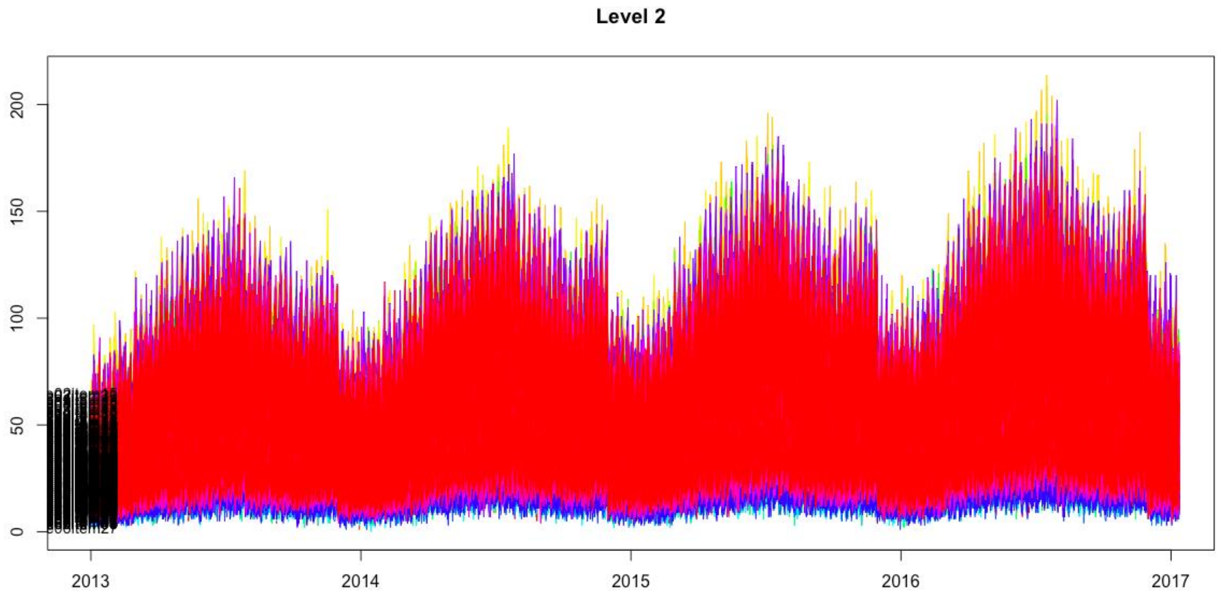


Exhibit 4


Level 2