Predict Future Sales Development Guide

Oscar Hernandez

Northwestern University

Predict Future Sales Development Guide

This model development guide is focused on covering the steps that were taken to train several models using a time series data. The data are provided by a software firm – 1C Company. The primary goal is to make use of historical time series data along with features that provide various descriptive measures of the observations to achieve accurate forecasts.

Specifically, the data consists of daily sales of various products that were sold at different shops. The dates range from January 2013 to October 2015. All of these shops and products are part of 1C Company. There are over 21,000 unique items, 60 shops and 84 item categories. The trained models will forecast the number of items sold at each shop for November 2015. For example, the forecasts will display how much of Item 1, 2, 3 were sold at Shop A, B, C for all required combinations in November 2015 (Note: the actual data set values are different than what was just stated).

There were many challenges presented during the development. The data used to train the models had be prepared (i.e. feature engineering, outlier detection) from the various files provided by 1C Company. We were not provided a training set in the correct form necessary to train models. Therefore, the data preparation component of this work is one of the first challenges presented. Furthermore, selecting models that can incorporate predictor variables along with the time series data will be another important component of this development guide. Overall, completing multivariate time series analysis and modeling is what made this work challenging.

This exercise is significant because it is applicable to various cross-industry business functions. Arguably, any business that is not making use of forecasting from the data available to them is at a competitive disadvantage. Although there exists plenty of examples that utilize univariate time series data effectively, capturing the complexity of a given business scenario is best served using multivariate time series data. Much of the data that is available today includes many features that can be useful in increasing the predictive accuracy of a model. Therefore, it makes sense to at least consider using the multiple features that are available if it results in better model performance. Overall, the work involved in this development guide serves as an example of utilizing multivariate time series data to aid a business function.

Literature Review

Along with being classified as a multivariate time series data, it can also be described as hierarchical times series data. Hierarchical time series data includes observations that can be labeled into different hierarchies. The data used in this development guide is an example of a 2-level hierarchical structure. We are provided with several store-item sales combinations, which if added up would equal to the total sales of the company. The total sales would be the top level which is called level 0. Level 1 would be the sales from each of the stores grouped by their name and level 2 (bottom-level) would be the sales from all the store-item combinations. Various peer reviewed research involving multivariate and hierarchical time series modelling were studied to learn about the topics.

Zhang, Zhong, Geng and Jiang (2017) developed a vector autoregressive model (VAR) as a multivariate model to forecast the electric vehicle sales in China for both short term and long term time horizons. They compared their results with the forecasts from a univariate time-series model and showed that the use of a multivariate model led to significantly improved predictive accuracy. Demirel, Zaim, Çališkan and Özuyar (2012) modelled natural gas consumption in Istanbul using various model classes such as OLS regression, ARMAX, and Artificial Neural

Network. They incorporated multivariate time series data that includes features such as price and daily temperature.

Pertaining to hierarchical time series forecasting, Hyndman, Ahmed, Athanasopoulos and Shang (2011) utilized a regression model to optimally combine and reconcile the forecasts from a top-down and bottom-up approach. Among other items, their study provided evidence to demonstrate it is possible to effectively combine elements from various approaches when modelling hierarchical time series data. Capistrán, Constandse and Ramos-Francia (2010) developed several models to forecast monthly inflation using individual and hierarchical time series data from the Mexican CPI. This study incorporated both deterministic and stochastic seasonality but did not utilize predictor variables. Terui and Ban (2014) proposed a multivariate time series model for over-dispersed discrete data that included market, manufacturer and product-level information. They incorporated a hierarchical structure as a way to improve predictive accuracy.

These studies provided insight into previous research involving multivariate and/or hierarchical time series modelling. It is reassuring to know the diversity of modelling classes that are applicable to these topics.

Data Preparation

The initial part of our model development is organizing the data to the correct format necessary to train our models. The sales data which we will refer to as the training set for the remainder of this guide included six variables. Table 1 shows the data in the training set. Furthermore, we were provided separate files that included text descriptions of the item categories, items and shops. Also, we are given a file that included the item_category_id variable which showed the item category of every item.

Variable Name	Variable Type	Description
date	date	date of the item sale in format dd/mm/yyyy
date_block_num	categorical	number used to represent a mm/yyyy combination
shop_id	categorical	unique identifier of a shop
item_id	categorical	unique identifier of an item
item_price	continuous	current price of an item
item_cnt_day	continuous	number of items sold on a given date

Table 1: Training Data

The test set includes the store-item combinations that our models will be trained to forecast. We will provide a monthly forecast of these store-item combinations for November 2015. Our training data set includes daily data from January 2013 – October 2015.

Data Quality Check

The first decision made was to not utilize any of the text data (e.g. store name, item name) to build our models. Also, we decided to not use the item_category_id variable. Our next step involved comparing the store-item combinations with the training set and test set. We noticed that there was a mismatch in the combinations present in both sets. Therefore, since the test set was the store-item combinations that we were going to forecast, we trimmed the training set by first removing stores that were not present in the test set. We further trimmed the training set by removing items that were not present in the test set.

After making the initial changes to our training set, Table 2 highlights descriptive statistics for each of the features except date. We can see that there is no missing data in the training set which is not necessarily an expected result while doing exploratory data analysis. Furthermore, date_block_num, shop_id and item_id represent categorical features so their mean and standard deviation are not of real interest to us.

From a data quality check perspective, we were interested in making sure that there were not any erroneous values. We noticed that we didn't have any negative values for date_block_num, shop_id and item_id which is correct. Negative values exist for item_price and item_cnt_day which could be considered returned items. However, the decision was made to remove these observations from the training set. The last change was to remove observations where item_price was greater than 50,000.

	date_block_num	**shop_id**	**item_id**	**item_price**	**item_cnt_day**
Count	2935849	2935849	2935849	2935849	2935849
Mean	14.57	33.00	10197.23	890.85	1.24
SD	9.42	16.23	6324.30	1729.80	2.62
Min	0.00	0.00	0.00	-1.00	-22.00
25%	7.00	22.00	4476.00	249.00	1.00
50	14.00	31.00	9343.00	399.00	1.00
75%	23.00	47.00	15684.00	999.00	1.00
Max	33.00	59.00	22169.00	307980.00	2169.00
NaN	0	0	0	0	0

Table 2: Descriptive Statistics

Feature Engineering

Our next data preparation step was to create three new features – item_price_month_avg, item_cnt_month_current and item_cnt_month_fcasts. Using Python's agg function, item_price_month_avg was created which represents the average monthly price of a particular store-item combination with respect to a month-year pair. Similarly, agg was used to create item_cnt_month_current which represents the monthly number of items sold for each store-item combination with respect to a month-year pair.

Prior to the creation of the item_cnt_month_fcasts feature, we had to make another adjustment to our training set. We wanted our training set to have a value for each store-item combination for each date_block_num value (0-33). Therefore, we adjusted our training set which resulted in many rows having 0 values since not every store-item combination had existing results in every month-year pair.

The next step was to create the item_cnt_month_fcasts feature (i.e. our Label) which represents the monthly sales of the following month for each store-item at different dates. Given that the item_price_month_current feature represents the sales for the current month, we used the Python shift method to move down the observations for item_price_month_current by one to create the item_cnt_month_fcasts feature. This approach was used to create a training set that included the average item price and total item sales count for each month-year pair along with their respective total item sales count for the subsequent month.

The final part of our data preparation was to create a new test set based from our training set that was just updated. First, we removed the values associated with October 2015 from the

training set. Next, we created a dataframe “temp2” that was the result of keeping only the last observation for each store-item pair in the training set (i.e. the September 2015 observations) and dropping the other ones. We merged temp2 and the original test data provided to create a new test set. This new test set had missing values as a result of the merge. From the Python impyute library we used the mice function which imputed the missing values for the test set using the k-nearest neighbors algorithm. There were no further data preparation steps taken.

Model Development

We chose to develop three models to forecast the monthly sales for each store-item combination in November 2015. The first one that we trained was an XGBoost regression model. In terms of hyperparameter tuning, we chose a max_depth and n_estimators equal to 5 and 20, respectively. Figure 1 shows the feature importance plot for this model. We can see that item_cnt_month_current and item_price_month_avg are the two most important features in this model with respect to forecasting November 2015 sales.

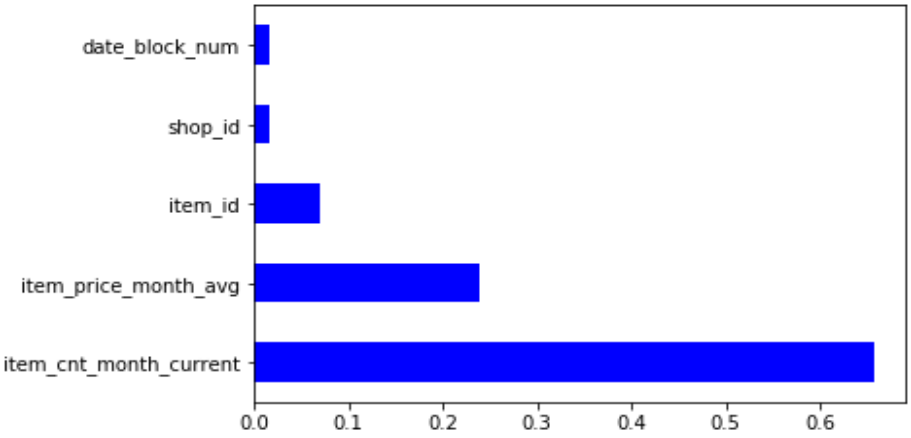

Figure 1: XGBoost Feature Importance

The second model that we trained was as a Random Forest regression model. In terms of hyperparameter tuning, we chose a max_depth and n_estimators equal to 5 and 20, respectively. Figure 2 shows the feature importance plot for this model. We can see that item_cnt_month_current and item_price_month_avg are once again the two most important features with respect to forecasting November 2015 sales. However, the Random Forest model placed a greater weight on the item_cnt_month_current feature compared to the XGBoost model.

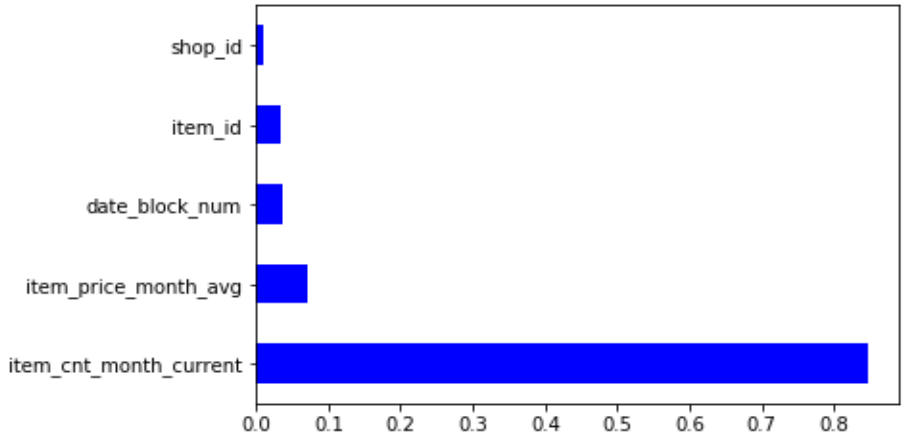

Figure 2: Random Forest Feature Importance

The final model that we trained was an OLS Linear Regression model. Table 3 displays each of the coefficients for the model. One particular item that stands out is the small impact that the item_price_month_avg feature has on the forecasts.

Intercept/Coefficients	Value
intercept	0.065
date_block_num	0.001
shop_id	3.135
item_id	-1.87
item_price_month_avg	-0.0002
item_cnt_month_current	0.77

Table 3: OLS Linear Regression Model Coefficients

Model Comparison

Our next step is to compare each of the model's predictive accuracy. We decided to utilize RMSE as the accuracy metric that would help us determine which model is our best one. The model with the lowest RMSE will be considered the best performing one with respect to predictive accuracy. Table 4 displays the RMSE for each of the models that were trained. The Random Forest Model was chosen as our best performing model since it had the lowest RMSE score.

Model Name	RMSE
XGBoost Model	2.843
Random Forest Model	2.017
OLS Regression Model	2.371

Table 4: RMSE Scores

Conclusion

Limitations

There were several limitations in the development of these models. One of the issues is that we were not able to develop a time series object that could take advantage of the R forecast package, which is considered one of the best packages for time series forecasting. As such, the forecasts that were calculated from our models were not able to take into account any seasonality or trends that exist in the time series data. Also, doing some more feature engineering might have proven useful to improve predictive accuracy although that isn't necessarily the case every time. Nonetheless, being able to compare models with more features would have been an improvement to what was completed in this development guide.

Future Work and Key Learnings

We believe any future model development that involves multivariate time series data should study the effects of probabilistic variables. Although probabilistic variables will add complexity to the model, we believe given the computing power available, incorporating such variables may lead to better forecasts. In other words, being able to account for uncertainty in multivariate time series models is an area that deserves some consideration. Effective data preparation and the importance of efficient coding were the two most significant learnings for us.

References

Capistrán, C., Constandse, C., Ramos-Francia, M. (2010). Multi-horizon inflation forecasts using disaggregated data. *Economic Modelling*, *27*(3), 666-677.

Demirel, O.F., Zaim, S., Çališkan, A., & Özuyar, P. (2012). Forecasting natural gas consumption in Istanbul using neural networks and multivariate time series methods. *Turkish Journal of Electrical Engineering and Computer Sciences, 20*(5), 695-711.

Hyndman, R., Ahmed, R., Athanasopoulos, G., & Shang, H. L. (2011). Optimal combination forecasts for hierarchical time series. *Computational Statistics and Data Analysis, 55*(9), 2579-2589.

Terui, N., & Ban, M. (2014). Multivariate time series model with hierarchical structure for over-dispersed discrete outcomes. *Journal of Forecasting, 33*(5), 376-390.

Zhang, Y., Zhong, M., Geng, N., & Jiang, Y. (2017). Forecasting electric vehicles sales with univariate and multivariate time series models: The case of China. *PLoS ONE, 12(*5*).*