STAT 542: Project 2

Ashley Zhang, Shivesh Pathak, Jean Liu, Paulina Koutsaki

March 19, 2017

1 Data manipulation

Not much data manipulation was required for this project. First off, we assign a week, month and year to each the of the measurements in the test and training data using standard conventions from the package lubridate. Second, whenever we get a new chunk of training data we immediately append it to the old training data and then do the training on the new concatenated data. Before we do the training, we also assign the week, month and year to the concatenated training data.

2 Simple method

The simple method uses the same idea as shown in class for the Walmart data. If we want to predict the expected sales for a given week of a given year, we simply take the median of the sales for the same week of the previous year, as well as the week before and after this week. In the situation that no such historical data exists, we give a prediction of 0. For the case where the "historical data" would be in the future, we also give a prediction of 0. Using the metric for error provided on Kaggle, if we set all of the sales predictions to be zero we would get an error of roughly 15,000. However, using this simple method we get an error of about 2-5,000 per month, therefore leading to a decrease by about 3-6 times in the error! The run time for a single month's calculation is around 10 minutes, meaning that for 20 months the total runtime is 200 minutes: a little bit short of two hours. This calculation is pretty accurate and fairly cheap, however we can make improvements on both ends.

3 Random forest

We decided to use Random forest as our second method because the given Walmart data is very complex with many different stores and departments. We assume this ensemble approach will give us a better prediction of our data than predicting with only one regression. The random forest method starts with lots of decision trees, and each tree will branch off making smaller sets of data. Random forest method has relative fast runtime, so we think it will be very efficient for the Walmart data set. We first read-in the train and test data, and merged the type and size for train and test data. We made features for train and test data by subset each of the year, month, and day. We made our

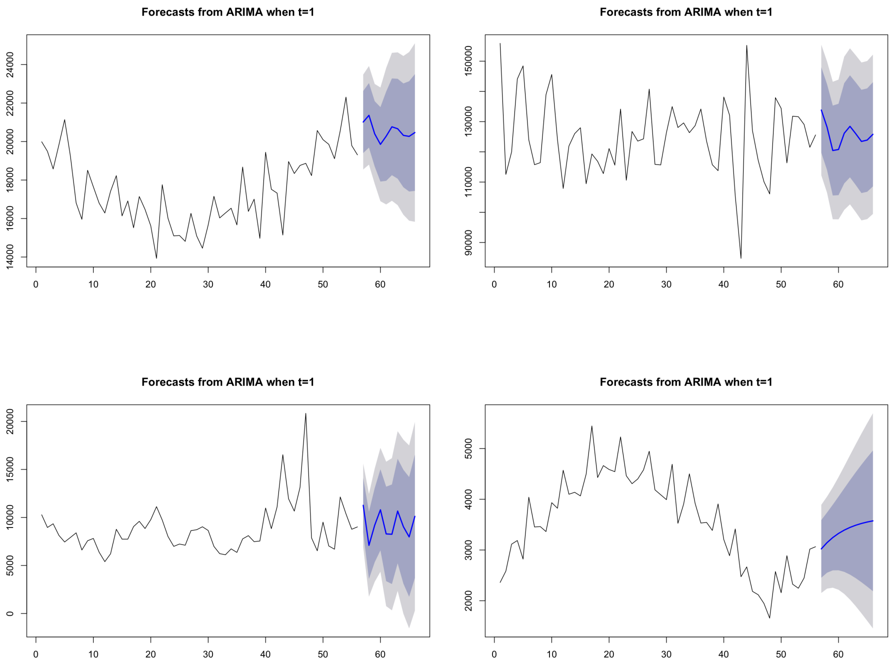

Figure 1: Sales forecast for four store-department combinations

model for submission by selecting only relevant data for store and department. We increased the weights of holiday data by scale of 5 because the MAE is weighted. In the end, we output predictions in a csv file for submission.

4 ARIMA

For our last method, we group the data by Store and Dept and use auto.arima to fit a model to each of the groups. We then loop over every store and department combination and forecast the sales of that month for those pairs that appear in the test set. If we have no history of sales for a store-department combination we need a prediction for (perhaps a specific store added a new department recently), we predict 0. Using this method we get an error of about 1-2,000 per month, a noticeable improvement! However, the price we have to pay is lengthy computations. The run time for a single month's calculation is around 23 minutes, meaning that for 20 months the total runtime is about 7 hours. However, for a store like Walmart, this shouldn't be an issue.