Predict Future Sales
DS 413 – Sec 55
Dalya Adams
5/5/2018

Problem

The problem under consideration is to forecast the total future sales for every product and store in the next month. The data provided to solve this issue is selling location, unique identifiers of the sold item, the number of items sold, the price and the date the item sold. The data analyzed was provided by a Russian Software Firm, 1C.

Significance

This problem, rooted in real world data, represents a real request that data scientists could receive from a company. Forecasting the items and amounts sold at each of a company's locations is essential in properly planning production, logistics, personnel, etc. The quantity of data is also representative of a dataset that we may need to wrangle with in our professional careers. While frustrating, it is a good challenge to overcome, as it prepares us for the future.

Data

The data for this model was provided by the Russian Software Firm, 1C. It consists of 2.9 million observations. These observations are the daily sales by store and item.

```
##       date               date_block_num    shop_id         item_id
##  Min.   :2013-01-01   Min.   : 0.00      Min.   : 0      Min.   :    0
##  1st Qu.:2013-08-01   1st Qu.: 7.00      1st Qu.:22      1st Qu.: 4476
##  Median :2014-03-04   Median :14.00      Median :31      Median : 9343
##  Mean   :2014-04-03   Mean   :14.57      Mean   :33      Mean   :10197
##  3rd Qu.:2014-12-05   3rd Qu.:23.00      3rd Qu.:47      3rd Qu.:15684
##  Max.   :2015-10-31   Max.   :33.00      Max.   :59      Max.   :22169
##    item_price         item_cnt_day
##  Min.   :    -1.0     Min.   : -22.000
##  1st Qu.:   249.0     1st Qu.:   1.000
##  Median :   399.0     Median :   1.000
##  Mean   :   890.9     Mean   :   1.243
##  3rd Qu.:   999.0     3rd Qu.:   1.000
##  Max.   :307980.0     Max.   :2169.000
```

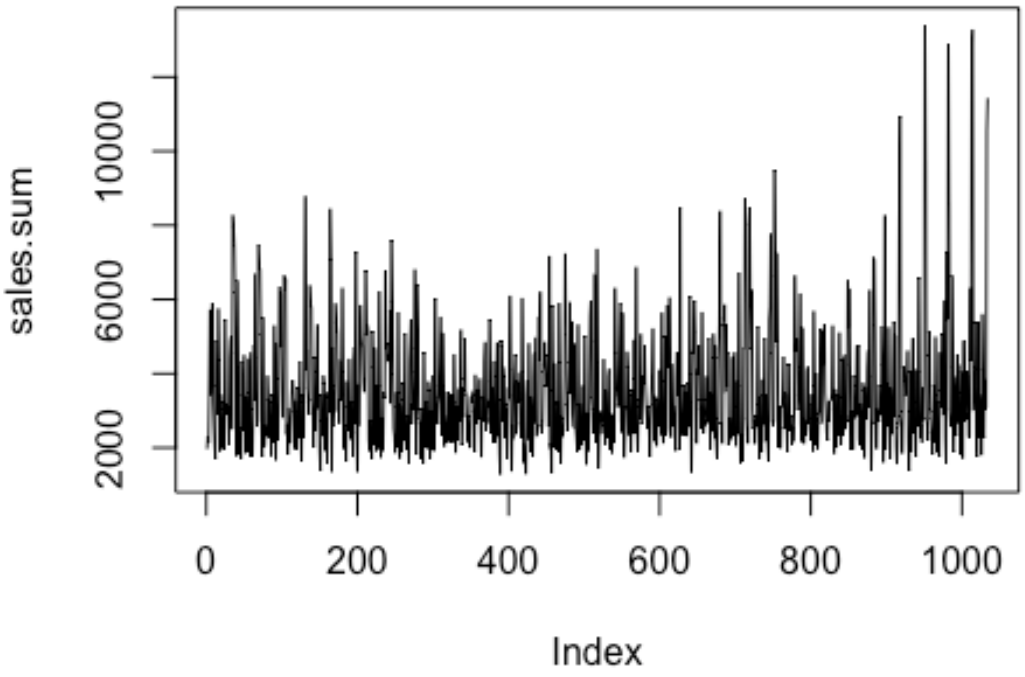
In reviewing the summary stats, we notice 60 shops in the dataset and 22170 items in the dataset. We also notice item prices of $-1 and item daily sales of -22. While negative sales could be possible if items were stolen, a negative price is unlikely. Since supporting information about the dataset it provided in Russian, we will assume that all negative values are errors. The absolute value of all observations are taken to correct the negative values.

```
sales$item_price<-abs(sales$item_price)
sales$item_cnt_day<-abs(sales$item_cnt_day)
```

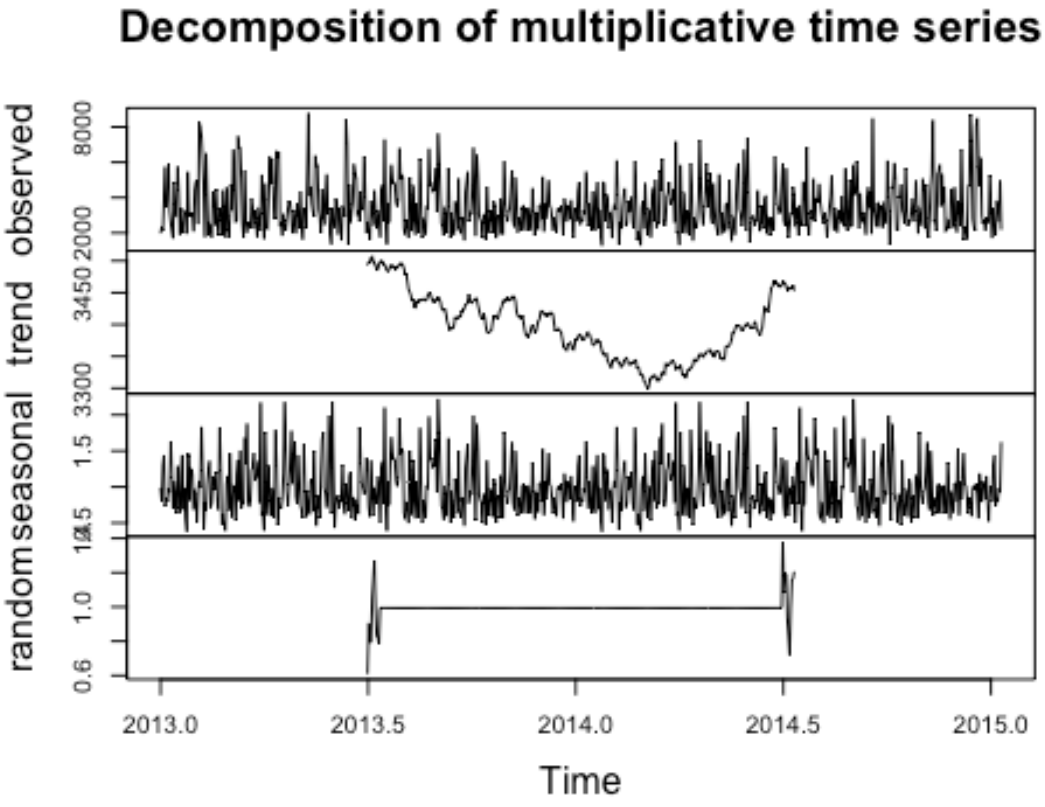
In reviewing the problem, we are asked to predict the monthly sales, so the daily data is summed to the month, allowing easier predictions at the month level.

```
sales_si=aggregate(item_cnt_day~date_block_num+shop_item,FUN=sum,data=salesbr)
sales_shop=aggregate(item_cnt_day~date_block_num+shop_id,FUN=sum,data=salesbr)
sales_item=aggregate(item_cnt_day~date_block_num+item_id,FUN=sum,data=salesbr)
```

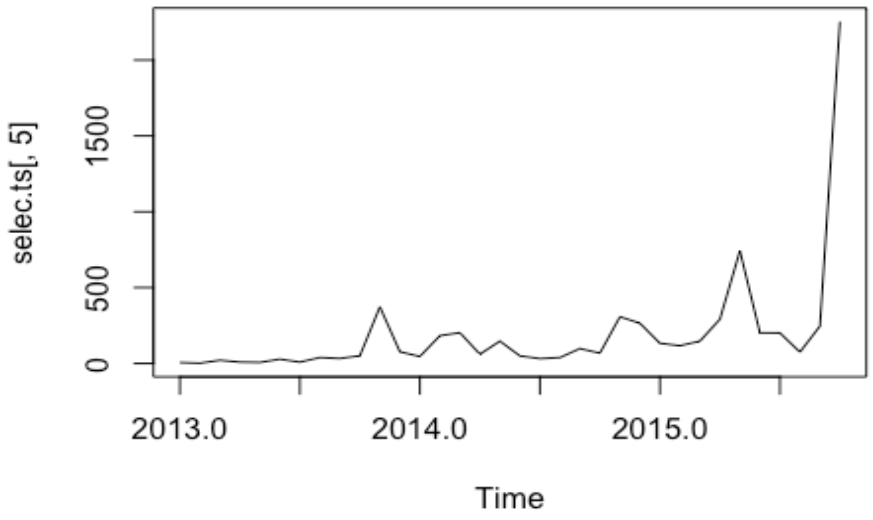
Below we sum all sales by date and plot the results. We can see a seasonal trend, and also notice that the magnitude of sales has increased as time has gone on. This plot is not at an item or store level but the sales of the Russian Software Firm, 1C from January 2013 to October 2015.



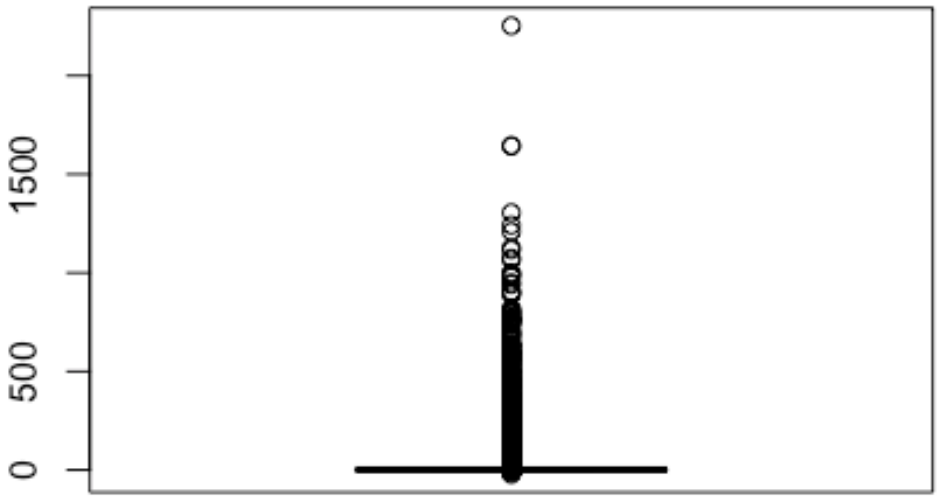
In decomposing the series, the seasonal impact of the dataset remains very visible, but we now notice a downward trend, followed by an upward trend.



Next, we limit the dataset to one store and one item, store 12 and item 11373. We can see an upward trend in this smaller dataset, that was not immediately visible in the plot of the summed sales. The seasonal impact is still visible, although not as predictable.



Next, we look at a boxplot of all the item daily sales. It is immediately visible that the vast majority of the sales are slightly above 0. Values between 100 and 2500 are extreme outliers.



Unsure whether these extreme values are errors or anomalies, we impute all values that lie outside of the 99% confidence interval with the mean of the dataset.

```
fun <- function(x){
  quantiles <- quantile( x, c(.05, .99 ) )
  x[ x < quantiles[1] ] <- quantiles[1]
  x[ x > quantiles[2] ] <- quantiles[2]
  x
}

salesbr$item_cnt_day<-fun(salesbr$item_cnt_day)
salesbr$item_price<-fun(salesbr$item_price)
```

Literature

In reviewing literature, Ramos, Santos and Rebelo (2015) compare the "forecasting performance of state-space and ARIMA models" and find that both methods are similar in one-step and multi-step forecasts, when comparing accuracy based on RMSE, MAE and MAPE. The use of Seasonal ARIMA models for short term sales forecasts is explored in detail by Filler and Digabriele (2008). In their analysis, time series models were effectively used to predict future values, by utilizing the past time dependent behaviors of the data

Silhan (1989) utilized ARIMA models to generate predictions of earnings, sales and margins. Findings from Silhan indicated that ARIMA models used to predict net income was as accurate as ARIMA predictions of net sales and net profit margins. Gharde (2016) provides another academic paper outlining the use of ARIMA models for sales forecasting. Gharde concludes that "forecasting based on one factor" does not yield accurate results and recommends utilizing AI and hybrid models as well as gathering large amount of historical data to improve accuracy of predictions.

In a paper by Kane, Price, Scotch and Rabinowitz (2014), ARIMA models are compared with Random Forests in predictions. Some drawback of utilizing an ARIMA model for forecasting are outlined and compared with the use of Random Forests. The authors explain that ARIMA models suffer from two key drawbacks, the assumed linear relationship between independent and dependent variables and the assumption of a constant standard deviation. When utilizing a Random Forest in place of ARIMA models, accuracy is generally better, but the interpretability of the model is more challenging. This trade off must be considered when choosing between ARIMA models and Random Forest models.

Type of Models

In attempting to forecast the sales by item and store for in the next month, I modeled using three different key models, a naïve forecast, an exponential smoothing model and an ARIMA model. The naïve forecast was split into two different versions, the first version took the last value of sales by store and item and used that as the prediction for the item one month in the future. The second naïve version took the sales by store and item in the final month and used that as the prediction for the store and item one month in the future. If the store and item did not have a sale in the final month of the dataset, a value of 0 was predicted for the one month in the future sale. Since each store and item combination could have at max 33 observations, the naïve model was chosen due to the ease of use. Exponential Smoothing and ARIMA were tested because of their robust approach to modeling time series data.

A zero-inflated model was also attempted in conjunction with the output from the store and item ARIMA model. This model was a logistic regression to predict the probability of selling an item, multiplied by the sales predicted by the ARIMA model, rounded to the nearest whole number. The zero inflated model was only used on the store and item combination.

Formulation

After correcting extreme outliers and errors, each store, item and store and item combination were run through a naïve forecast, an exponential smoothing model and an ARIMA model. The code below outlines the loops which provided one-month forecasts for each store, item and store and item combination.

```
for (i in 0: 60){
  tempdata<-subset(sales_shop,sales_shop$shop_id==i)
  if (length(tempdata$date_block_num)>0) {
    ##Naive Forecast for shop
    myf[i]<-meanf(tempdata$item_cnt_day, h=1)$mean
    ##ETS for shop
    myf1[i]<-forecast(ets(tempdata$item_cnt_day), h=1)$mean
    ##ARIMA for shop
    myaa2<-auto.arima(tempdata$item_cnt_day)
    myaa_f2<-forecast(myaa2, h=1)
    myf2[i]=myaa_f2$mean
  } else {
    myf[i]=0
    myf1[i]=0
    myf2[i]=0
  }
}

for (i in 0: 21807){
  tempdata<-subset(sales_item,sales_item$item_id==i)
  if (length(tempdata$date_block_num)>0) {
    ##Naive Forecast for item
    myf3[i]<-meanf(tempdata$item_cnt_day, h=1)$mean
    ##ETS for item
    myf4[i]<-forecast(ets(tempdata$item_cnt_day), h=1)$mean
    ##ARIMA for item
    myaa5<-auto.arima(tempdata$item_cnt_day)
    myaa_f5<-forecast(myaa5, h=1)
    myf5[i]=myaa_f5$mean
  } else {
    myf3[i]=0
    myf4[i]=0
    myf5[i]=0
  }
}

for (i in 0: 424124){
  tempdata<-subset(sales_si,sales_si$shop_item==i)
  if (length(tempdata$date_block_num)>0) {
    ##Naive Forecast by shop and item
    myf6[i]<-meanf(tempdata$item_cnt_day, h=1)$mean
    ##ETS by shop and item
    myf7[i]<-forecast(ets(tempdata$item_cnt_day), h=1)$mean
    ##ARIMA by shop and item
    myaa8<-auto.arima(tempdata$item_cnt_day)
    myaa_f8<-forecast(myaa8, h=1)
```

```
   myf8[i]=myaa_f8$mean
  } else {
   myf6[i]=0
   myf7[i]=0
   myf8[i]=0
  }
}

###Logistic regression
sales_agg=aggregate(item_cnt_day~date_block_num+shop_item,FUN=sum,data=salesbr)
cnt<-sales_agg[sales_agg$date_block_num == 33, "item_cnt_day"]
itm<-sales_agg[sales_agg$date_block_num == 33, "shop_item"]
uni_item<-data.frame(shop_item=unique(sales_agg$shop_item))

##Join itm and cnt, to have dataset of item and last use
cnt_itm<-data.frame(
  shop_item=itm,
  count=cnt)
#join cnt_itm to uni_item and all NA values become 0.
#Outer Join
mjoin<-merge(uni_item,cnt_itm, by="shop_item",all = TRUE)
mjoin[is.na(mjoin)]=0
## for each shop_item average item price and count of dateblocknum
sales_price=aggregate(item_price~shop_item,FUN=mean,data=salesbr)
sales_count=aggregate(date_block_num~shop_item,FUN=length,data=salesbr)

##Join with mjoin
new_mjoin<-merge(mjoin, sales_price, by="shop_item",all = TRUE)
new_mjoin_2<-merge(new_mjoin, sales_count, by="shop_item",all = TRUE)

##If 1 or greater, convert to 1, if 0 make it 0
new_mjoin_2$log_count<-ifelse(new_mjoin_2$count>='1', 1, 0)

###mjoin is target variable
model <- glm (log_count ~ item_price + date_block_num, data = new_mjoin_2, family = binomi
al)
summary(model)
predict <- predict(model, type = 'response')
new_predict<- melt(predict)

##new_predict joins with item_shop
new_predict<-data.frame(prediction=new_predict, shop_item=unique(sales_agg$shop_item))
new_mjoin_2<-merge(new_mjoin_2, new_predict, by="shop_item",all = TRUE)
mynew_8=merge(test, new_mjoin_2, by="shop_item", all.x = TRUE)
mynew_8[is.na(mynew_8)]=0
mynew_8$shop_id=NULL
```

```
mynew_8$item_id=NULL
mynew_8$shop_item=NULL
mynew_8$shop_item_num=NULL
mynew_8$count=NULL
mynew_8$item_price=NULL
mynew_8$date_block_num=NULL
mynew_8$log_count=NULL
mynew_8$value[is.na(mynew_8$value)]=0

##join sales_pred and value on ID
mynew_8=merge(sales_pred_item_shop_mynew_7, mynew_8, by= "ID" , all = TRUE)
pred_8<-(mynew_8$item_cnt_month*mynew_8$value)
pred_8[is.na(pred_8)]=0
pred_8[pred_8<0]=0
pred_8=round(pred_8,0)
mynew_8<-data.frame( ID= mynew_8$ID, item_cnt_month=pred_8)
write.csv(mynew_8, "~/sales_pred_item_shop_mynew_8.csv", row.names=FALSE)
```

Performance / Accuracy

The three main classes of models implemented for this analysis, naïve, ETS and ARIMA, were implemented at a store level, an item level and an item by store level. The models at the store level performed the worst, based off of the Root Mean Square Error (RMSE) metric. The naïve model, has a RMSE of 30.57, the ETS prediction by store has an RMSE of 19.46 and the ARIMA model scores 14.68. This level of prediction is not practical and the poorly performing RMSE are expected. The assumption of predicting at the store level is that all items in each respective store will all sell the same. Experience as consumers lead us to believe this is extremely unlikely.

The next level of prediction was by item. The naïve model, ETS and ARIMA model were all calculated at the item level. The assumption here is that each item will follow the same sales pattern, regardless of store. The Naïve item model achieved a RMSE of 28.25, the ETS by item received a score of 16.83 and ARIMA by item received a RMSE of 16.75.

The final models by store and item combination scored the best by far. Intuitively, this makes sense as each item and store combination are unique and the way that an item sells will differ based on the store the item is at and the item itself. Basing selection of the final model off of RMSE, the naïve model scored 1.52, while the ETS score 1.76 and ARIMA model scored 1.45. The zero-inflated model scored the highest with a score of 1.25.

Limitations

The limitations of these models are based mainly off of the sheer size of the data. As the dataset was approximately 4 million observations, the computational power required to process individual models was great. This slowed the iterative process to fine tune models and parameters. Another possible limitation relates to the not creating additional observations for the model. In the current dataset, dates are missing when an item did not sell, creating observations of 0 for these would likely impact the accuracy of the ETS and ARIMA models, as each ETS and ARIMA model are building based off of only the observations present in the dataset, whereas the missing observations of 0 sales likely has a strong bearing on the future sales of the item.

Another limitation of the models I designed is related to the preparation of the dataset. All extreme outliers were imputed. These extreme outliers, if correct sales, would likely have had a profound impact on the future prediction. This is mainly due to the max observations per store and item being 33. Since a max of 33 observations were used to model each store and item combination, removing the extreme outliers made the model more stable, but may have hampered the models' ability to identify new trends by store and item.

Future Work

In the future, more computational power would be arranged for to ensure that I was able to iteratively fine tune the models and parameters. Other options that may have made the models more accurate would be to: include exogenous variables, such as price of the item, split item sales into fast movers and slow movers, utilizing a zero inflated model to predict probability of sales and sales forecast, as well as machine learning techniques such as gradient boosting and random forests. The models were considered but were postponed as the time series nature of the data dictated the initial use of time series models. Also, the computational difficulties when utilizing time series models would have likely been much more extreme for gradient boosting or tree-based models.

Learning

One of the main takeaways from this problem is experience on how to tackle a problem like this. A forecast for a dataset with a multitude of categorical variables, in this case store and item combination, is not unlikely. Having an idea of how to address this issue is the first step in building the data intuition necessary to become a successful data scientist. Another takeaway, on a personal level, is related to the realization that utilizing GPUs for computation power is a necessity in the current world of "big data" and will only become more important. The realization is that as a data scientist, I must ensure that I know how to use the tools, like TensorFlow and Keras to solve big data problems in a timely manner.

References

Filler, M., & Digabriele, J. (2008). SHORT-TERM SALES FORECASTING USING A SEASONAL ADJUSTMENT MODEL. Valuation Strategies, 11(5), 6-17.

Gharde, A. (2016). Influence of factors on clothing sales and its future trend: Regression analysis and time series forecast of clothing sales. Journal of Textile and Apparel, Technology and Management, 10(2), 1-11.

Kane, M. J., Price, N., Scotch, M., & Rabinowitz, P. (2014). Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks. BMC Bioinformatics, 15(1), 276

Ramos, P., Santos, N., & Rebelo, R. (2015). Performance of state space and ARIMA models for consumer retail sales forecasting. Robotics And Computer-Integrated Manufacturing, 34(2), 151-163.

Silhan, P. (1989). Using Quarterly Sales and Margins to Predict Corporate Earnings: A Time Series Perspective. Journal of Business Finance & Accounting, 16(1), 131-141.