Walmart Store Sales Forecasting

12/19/14

YILIN ZHAO

1) Introduction

a) Aim: Based on the historical sales data in the past three years for 45 Walmart stores, build a strategic model to predict future sales for each department in each store. An additional challenge is to select out the holiday markdown events included in the dataset and to know how the departments are affected.

b) Data background: The historical sales data for 45 Walmart stores located in different regions. Each store contains a number of departments. In addition, Walmart runs several promotional markdown events throughout the year. The four largest of which are the Super Bowl, Labor Day, Thanksgiving, and Christmas. The weeks including these holidays are weighted five times higher in the evaluation than non-holiday weeks.

c) You can get the datasets from:

https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data.

2) Data description

a) Stores.csv: this file has 3 variables which are store, type and size, respectively.

b) Train.csv : historical training data from 2010-02-05 to 2012-11-01. Within this file you can find 5 fields:

- Store- the store number
- Dept- the department number
- Date- the week
- Weekly_Sales- sales for the given department in the given store
- IsHoliday- whether the week is a special holiday we

c) Test.csv- has same fields as train.csv, except have withheld the weekly sales.

d) Feature.csv- This file contains additional data related to the store, department, and regional activity for the given dates. It contains the following fields:

- Store - the store number
- Date - the week
- Temperature - average temperature in the region
- Fuel_Price - cost of fuel in the region

- MarkDown1-5 - anonymized data related to promotional markdowns that Walmart is running. MarkDown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA.
- CPI - the consumer price index
- Unemployment - the unemployment rate
- IsHoliday - whether the week is a special holiday week

The variables that I want to include in my analysis are store, dept, temp, fuel, cpi, unemp, cpi, isholiday, date.

By utilizing date and isholiday, we can generate 6 new fields:

- Holiday: “Christmas”, “Super Bowl”, “Thanksgiving”, “Labor Day”, “NO”
- Year: year of date
- Month: month of date
- Mday: day in month
- Yday: day in year
- WK: week in year

Then merge train.csv, store.csv and feature.csv, test.csv, store.csv and feature.csv to get mytrain and mytest.

Mytrain has 421570 observations and 17 variables.

```
> head(mytrain)
  store       date isholiday dept    sales type   size  temp  fuel      cpi unemp year month mday yday wk holiday
1     1 2010-02-05     FALSE    1 24924.50    A 151315 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
2     1 2010-02-05     FALSE   26 11737.12    A 151315 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
3     1 2010-02-05     FALSE   17 13223.76    A 151315 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
4     1 2010-02-05     FALSE   45    37.44    A 151315 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
5     1 2010-02-05     FALSE   28  1085.29    A 151315 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
6     1 2010-02-05     FALSE   79 46729.77    A 151315 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
```

Table 2.1

We also add a new variable “missing” in mytest, in which 1 represents no data for that particular store and dept combination. There are 36 such observations. Finally mytest has 115028 observations and 17 variables.

```
> head(mytest)
  store       date isholiday dept type   size  temp  fuel      cpi unemp year month mday yday wk holiday missing
1     1 2012-11-02     FALSE    1    A 151315 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No       0
2     1 2012-11-02     FALSE   56    A 151315 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No       0
3     1 2012-11-02     FALSE   24    A 151315 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No       0
4     1 2012-11-02     FALSE   55    A 151315 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No       0
5     1 2012-11-02     FALSE   23    A 151315 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No       0
6     1 2012-11-02     FALSE   22    A 151315 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No       0
```

Table 2.2

We can notice that mytest doesn't have holiday "Labor_Day" and mytrain has no year=2013.

3) Data Preprocessing:

A)Simple Model:

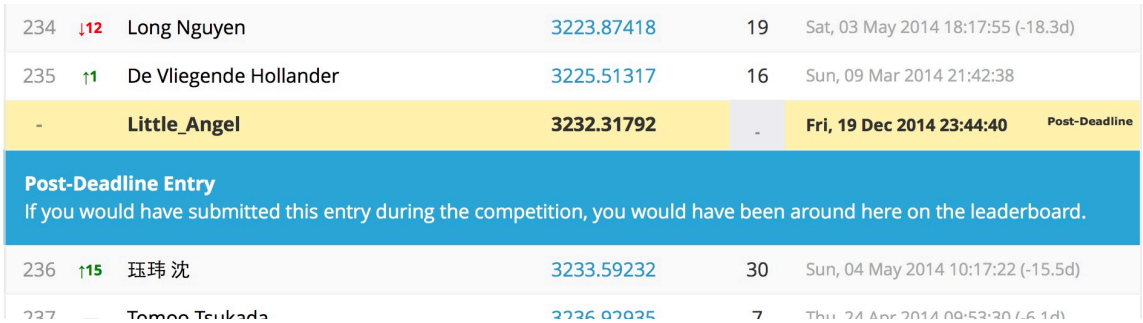
From data preprocessing steps and a quick glimpse of the strong seasonal time series graphs of weekly sales data per store and department combination, I find it that the test dataset ranges from 2012-11-02 to 2013-07-26(39 weeks), and training dataset ranges from 2010-02-05 to 2012-10-26(143 weeks), this means that all test cases could be viewed as future instances of those in the training data. So for any given store and department, we should be able to use historical data from the same week or neighboring weeks, if any, to predict the studied weekly sales.

Heuristics: In many store and department combinations, strong seasonal patterns can be observed, does it make sense to estimate future weekly sales from historical data?

Modeling Framework:

For each department in test:
For each store in test:
If (department, store) also in train:
Predict based on median of 52,53,54 weeks earlier's data in train
Else:
Predict based on median of other stores but same department
For each row in test:
If this row is an holiday:
Predict based on historical weekly sales for the same holiday
For all the rest NAs:
Use median of weekly sales from training data that share the same department

This very simple model ranks at 236 with a score of 3232.31 on the private leader board, better than 66% of all the teams that participated in Kaggle's Walmart Store Sales Forecast project.



234	↓12	Long Nguyen	3223.87418	19	Sat, 03 May 2014 18:17:55 (-18.3d)
235	↑1	De Vliegende Hollander	3225.51317	16	Sun, 09 Mar 2014 21:42:38
-		**Little_Angel**	**3232.31792**	-	**Fri, 19 Dec 2014 23:44:40** **Post-Deadline**

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

236	↑15	珏玮 沈	3233.59232	30	Sun, 04 May 2014 10:17:22 (-15.5d)
237		Tomoo Tsukada	3236.92935	7	Thu, 24 Apr 2014 09:53:30 (-6.1d)

B) Multiple Linear Regression

Mytest dataset has some store and dept combinations that are not included in mytrain. So I find out these rows and put them in a new subset dataset: mytest2. The remain subset is mytest1. And I notice that "isholiday" and "date" can be detailed by "holiday" and "year" "month" "week" while "type" and "size" make no contribution to the analysis. So I remove these variables out. Then I separate the subset mytest1 into 2 two parts. One part has year=2012(lm.mytest1) and the other has year=2013(lm.mytest2). So I build my regression model based on these four subsets.
Here is the summary table of these datasets.

Datasets	lm.mytrain	lm.mytest1	lm.mytest2	lm.mytest3
Obs/vars	421570/17	26774/12	88254/12	36/12
Feature		Year=2012	Year=2013	Store and dept combination missing in lm.mytrain

Table 3.2.1

```
> head(lm.mytest1)
  store dept  temp  fuel      cpi unemp year month mday yday wk holiday
1     1    1 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No
2     1   56 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No
3     1   24 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No
4     1   55 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No
5     1   23 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No
6     1   22 55.32 3.386 223.4628 6.573 2012    11    2  307 44      No
```

Table 3.2.2

```
> head(lm.mytest2)
    store dept  temp  fuel     cpi unemp year month mday yday wk holiday
649     1   21 41.73 3.161 224.081 6.525 2013     1    4    4  1      No
650     1   22 41.73 3.161 224.081 6.525 2013     1    4    4  1      No
651     1   23 41.73 3.161 224.081 6.525 2013     1    4    4  1      No
652     1    8 41.73 3.161 224.081 6.525 2013     1    4    4  1      No
653     1   44 41.73 3.161 224.081 6.525 2013     1    4    4  1      No
654     1   97 41.73 3.161 224.081 6.525 2013     1    4    4  1      No
```

Table 3.2.3

```
> head(lm.mytest3)
      store dept  temp  fuel      cpi unemp year month mday yday wk holiday
78749    37   29 59.43 3.207 222.2776 6.228 2012    11   30  335 48      No
78777    37   29 68.04 3.198 222.3255 6.228 2012    12    7  342 49      No
78829    37   29 56.66 3.168 222.3838 6.228 2012    12   14  349 50      No
78900    37   29 63.08 3.098 222.5040 6.228 2012    12   21  356 51      No
79012    37   29 49.08 3.161 222.7443 6.266 2013     1    4    4  1      No
79080    37   29 54.15 3.243 222.8644 6.266 2013     1   11   11  2      No
```

Table 3.2.4

```
> head(lm.mytrain)
  store dept    sales  temp  fuel      cpi unemp year month mday yday wk holiday
1     1    1 24924.50 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
2     1   26 11737.12 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
3     1   17 13223.76 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
4     1   45    37.44 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
5     1   28  1085.29 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
6     1   79 46729.77 42.31 2.572 211.0964 8.106 2010     2    5   36  6      No
```

Table 3.2.5

Build Linear Model

First, I transform store, year, month, mday, dept into factors and then build different model for different test dataset. And I also transform the response from sales to log(sales+4990) since the minimum sales is less than 0.

For lm.mytest1 in each department, I use all of the remain variables as predictors. The regression model is.

log(sales+4990)~store+temp+fuel+cpi+unemp+year+yday+mday+month+wk+holiday

For lm.mytest2.

There are 76310 “NA” in lm.mytest2. And I check out that all the missing values are in cpi and unemp so I think they are not good predictors in this model. And year 2013 is not in the training data. Since I decide to treat “year” as a factor, I remove out “year” in this model. So for lm.mytest2 in each department, the linear regression model is

log(sales+4990)~store+temp+fuel+month+as.numeric(yday)+mday+wk+holiday

```
> sum(is.na(lm.mytest))
[1] 76310
> sum(is.na(lm.mytrain))
[1] 0
> sum(is.na(lm.mytest1))
[1] 0
> sum(is.na(lm.mytest2))
[1] 76310
```

Output 3.2.1

We have already known the rows of missing department and store combinations. So I choose store as the index of the loop and run the model for all the departments in this store to get an estimate value for these combinations. The linear model is:

log(sales+revise)~temp+fuel+cpi+unemp+as.numeric(yday)+mday+month+as.numeric(wk)+holiday

I also generate a text file include all the R square for each model. We can notice that except those departments that have errors during the iteration, most of the R square is greater than 0.7, which implies the model might fit the data well. (see output table 3.2.2, highlight parts occur errors)

```
> r2
 [1] 0.8362211 0.9828933 0.8783515 0.9748868 0.9455607 0.7323954 0.9389563 0.9894786
0.9327853 0.9811337 0.9299346
[12] 0.9539794 0.9823371 0.9600439 0.8377911 0.9716358 0.7586817 0.8091510 0.9515286
0.9518656 0.9324370 0.9630364
[23] 0.8539810 0.9605174 0.9440510 0.8701445 0.8507546 0.9257619 0.9320697 0.8093263
0.8788653 0.8746014 0.9281385
[34] 0.8584302 0.6162132 0.7690113 0.9208847 1.0000000 0.9822150 0.6864821 0.9715289
1.0000000 0.9278305 0.1419410
[45] 0.9683453 0.0980285 0.4931118 0.9611350 0.8832248 0.2493640 0.9018986 0.5057279
0.8833647 0.6738832 0.7401763
[56] 0.5792589 0.8240656 0.0000000 0.9103277 0.9014234 0.9645189 0.9753967 0.8099508
0.4121850 0.9775411 0.9906567
[67] 0.9802299 0.9252609 0.9250373 0.8662274 0.9742035 0.9843936 0.9736480 0.9799429
0.9864151 0.9822749 0.9731281
[78] 0.9712082 0.9805393 0.9760610 0.7380803
```

Table 3.2.6

C) Improve model

After first running the three regression models, we can notice there are several warnings during the iteration. So we have to revise the errors and refit the model.

According to the print output, I notice that when dept.names=c(38,42,58,63,81) new levels of some variables may appear in the prediction procedure. So I build new model for each of these departments according to the errors. The main approach is transforming the factor variables which cause the error into numeric.
After refit the model, I produce a new r square for each of these departments(see output 3.2.8). And now there are only 15 NAs in mypred.

Obs	Deptnames=38	Deptnames=42	Deptnames=58
R-square	0.01915273	0.9514291	0.7924736

Table 3.2.7

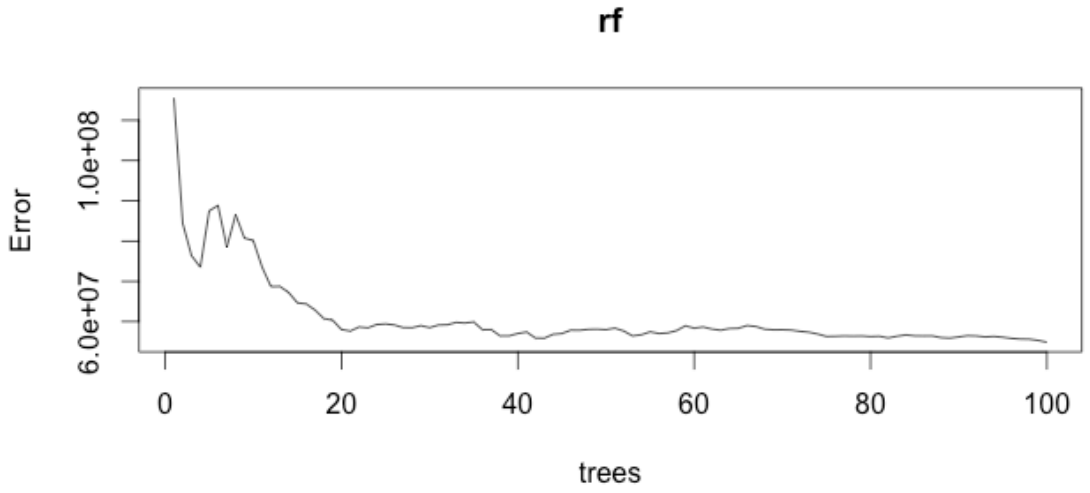
This model ranks 334 with a score 3710.

C) Random Forest:

Random forest is a very representative and easy to use ensemble learning method. As can be seen from lots of machine learning tasks, ensemble learning or bagging methods have greater power than simple linear regression. It constructs a forest of decision trees, and for each tree, it selects a subset of features and it outputs the class that is the mode of the classes from all the trees in the forest. The modeling framework is very similar to multiple linear regression.

```
For each department in test:
    For each store in test:
        If #(dept, store in train) =0 | #(dept,store in test) = 0:
            skip
         rf.tmp <- rpart(sales~ temp+fuel+year+month+mday+yday
                +wk,data = mytrain[tmp.train.id,], method="anova")
        Else:
            Use simple model or linear regression
For all the rest NAs:
    Use median of weekly sales from training data that share the same department
```

Take store 1 and department 1 as an example, the error rate stabilizes fast.



Plot 3.3.1

Importance of variables are listed as follows

> importance(rf,type=1)

	%IncMSE
temp	7.50218932
fuel	-1.98086861
year	0.02987743
month	10.97110482
mday	2.45044107
yday	4.27236387
wk	4.40504457

This table represents how much removing each variable reduces the accuracy of the model.

This model scores 3510 with a rank of 309.

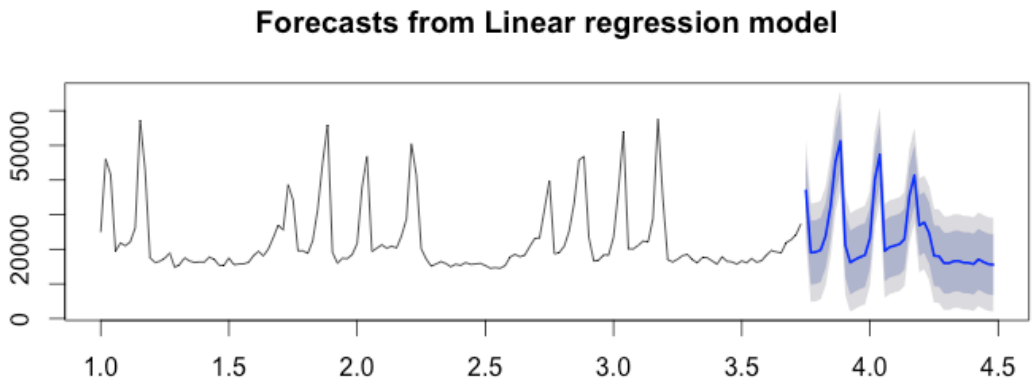
D) Models that Integrate Time Series:

Intuition: As can be seem from previous approaches to this problem, none of them gets to the very 10% of all the contestant, this leads me to think of a significantly different approach than linear regression modeling. Bearing in mind the fact that for a given store and department, historical data shows a strong seasonal patterns(of period around 52), it prompts to think will integrating time series models into my model yield better results?

Initial experiment:

To test for my hypothesis, I took a subset of training data that constrains store = 1 and dept = 1, there are 143 weekly sales data in training and 39 data points in the test set. I

used the implementation of "tslm"- time series linear model to fit a time series regression model and use it to predict for the test data. The result is as shown below. As can be seen, intuitively, time series based modeling may yield good results even without much help from other features. After some research, I decided to turn to time series modeling and forecasting thanks to the easy to use R package "forecast"



Plot 3.4.1

Modeling Framework:

For each department in test:
For each store in test:
If #(dept, store in train) = 143 and #(dept,store in test) =39:
Time series regression and predict these 39 points in test
Else:
Use simple model or linear regression
For all the rest NAs:
Use median of weekly sales from training data that share the same department

Models Used:

forecast.tslm:

```
horizon = nrow(tmp.test) # number of future predictions needed
s = ts(tmp.sales,frequency = 52)

model = tslm(s~trend+season) # no other features are needed

fc = forecast(model,h= horizon) # tslm prediction
pred = as.numeric(fc$mean)
```

"trend" and "season" are automatically created on the fly from the time series characteristics of weekly sales data.

forecast.stlf: ARIMA Model(Autoregressive Integrated Moving Average)

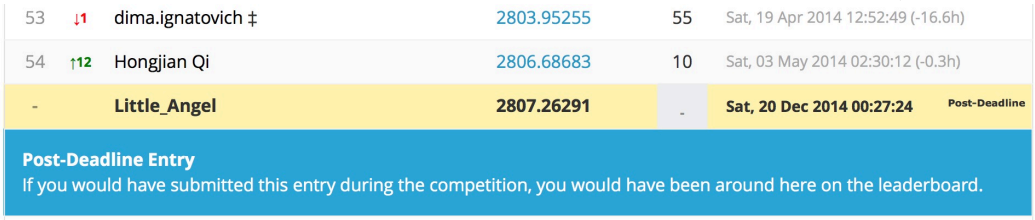
Change

model = try(tslm(s~trend+season))
fc = forecast(model,h= horizon)

to

fc <- stlf(s, h=horizon, s.window=3, method = "arima", ic = "bic")

ARIMA model is one of the most common time series models, so I also give it a try.



It turned out surprisingly well. It scored 2807 ranking 55 in the private leader board. It proves my hypothesis that time series modeling might be one of the ideal solutions.

4)Discussion

Comparing the 4 models we can apparently notice that when we introduce time series into our model, the prediction will be much more precise. Actually, we can get good result based on time series even without much help from other features. So we can see that simple model also works better than linear regression since it also consider the development of the process of time. The multiple linear regression generate most errors since we consider some of the variables as factors and sometimes the test dataset and train dataset may not matched. Random Forest is better than multiple regression and when take a look at the importance of variables we can notice that most of them are related to time.