TC Incubator

Forecast the daily traffic from Facebook fan page to content website for TC Incubator

Team 3

Sam Kuo

Astro Yan

Serina Hung

Jay Lee

Summary

TC Incubator is dedicated to strengthening youth innovation, focusing on helping entrepreneurs by providing consultants, TC also operates an official information website called "Jinrih Deliver", which promotes and attracts users to the official website by managing Facebook fan pages. While posting daily, they are constantly trying to improve the content of the post to write more attractive articles. But TC did not use the very serious way to do A/B testing, which caused the social manager is hard to know whether the article improvement affect the traffic. We hope to provide them with a tool that can replace the complexity of traditional A/B testing and help them to achieve the effect of A/B testing - the predicted results as a control group for A/B testing.

We do the forecasting to predict the traffic from the following week, dataset is from the Google Analytic that TC set, but because the original data only have total traffic, we use one series to forecast, in the meanwhile, we take daily post number from 6 kinds of theme and traffic number from past 7 days as external data to predict the following day traffic. After removing the extreme values and some pre-treatments, there are a total of 90 days left, about 3 months of data. External data contains the number of posts for the 6 types of posts, lag-1 of the three popular posts, and lag-1 to lag-7 of the traffic, for a total of 16 columns.

We will use 14 days as the validation period, the rest will be the training period, and try Regression, Neural Network and Ensemble to build the model., and we found out if we only predict 7 days, the accuracy of ensemble will be the highest, if we forecast 14 days then only naive get better performance, other method didn’t work well.

We finally recommend using Ensemble as the predictive model. The final 7-day validation result MAPE can be below 40, indicating that the prediction result is still good. If the training data can be more, we believe it can be more accurate and can be used as an A/B testing control group in the future. , to provide a more rigorous A/B testing method.

We suggest TC to detailedly record the adjustment and promotion they have done in A/B testing, because it can improve the forecasting accuracy, also they can use the tool to do A/B testing, in order to improve the post more efficiently. Moreover, It will be more convenient for marketing team to use Shiny interactive interface, and combine Google API to automate data collection to offer complete solution, in order to help marketing team to make decision and do the A/B testing more convenient and effective.

Buiness Goal

TC do the post planning base on 6 kinds of theme, include Jinrih toolbox, Jinrih planning, Jinrih growth, Jinrih attitude, Jinrih check-in, Jinrih brand. The marketing team expect to let more people browse their official website through their posts in Facebook fan page(that is improving the traffic from Facebook to official website).

Currently, the marketing team from TC decides the quantity and content of daily posts based on their experience, and try to improve the post content by doing rough A/B testing, and the result is usually unstable. The problem is that they have a lack of serious and effective way to improve, that is no control group in their A/B testing, so they are not sure the way they change cause the result.

We offer a suggestion which provide them a tool which they can refer to the control group when they do the A/B testing, we will use forecasting to predict the traffic result as the control group, so that they can compare to the traffic of the revised version(experimental group).

Forecast Goal

Our forecast goal is that we will predict traffic from the Facebook page into the official website in the next 7 days with different themes of post. Different kind of themes and quantity of post will affect traffic, however the post content and the post number will be planned 1 week ago, therefore, we take their post planning as external data to do forecasting, and generate to the control group for A/B testing.

Because the marketing team will complete the post planning 1 week ago, we forecast traffic from the next week, and add the themes and quantity are planned to post next week as external data. The forecast result will be daily traffic for the following week, they can compare with the actual traffic in order to verify the improvement they've done increase or decrease the traffic.[1]

The task details:

- t = Daily
- Yt = Daily traffic from the Facebook into the official website
- K = 1,2,3,4...7(days)
- Ft+k = Predited daily traffic from the Facebook into the official website

Data description

The raw dataset is from Google Analytics which is set up by TC and it contained all facebook traffic from the different devices included the computer FB , laptop FB, mobile FB and undetermined. We merged the four traffic data series into one series as our forecast time series. The raw dataset did not have the classification for the six post themes, so it can not use the six post themes to forecast. We just only can take the six post themes as the external data. The new dataset is the hourly traffic and the time period is from August first to December fifteen twenty eighteen(2018/08/31~2018/12/15).[2]

[1] Figure1. The example of a good revision should behave

[2] Figure2. The dataset how we do data pre-processing. (In Appendix)

Data Pre-Processing

- Step 1 : Aggerate the data from hourly to daily.
- Step 2 : Remove the outlier and the unstable data.
    - Unstable data : The data before the red line(Fig 2), TC just started to operate the facebook fan pages so the traffic was high because the facebook had some mechanics to prompt it.
    - Outlier : The three posts (Fig 2) are the most popular theme in this period.
- Step 3 : Label and map the theme of external data. [3]
    - TC have six posting themes so we add this external information in our dataset.
- Step 4 : Check each lag of value and six theme of post

Methods

- Validation: 14 days
- Benchmak: Navie (last day)
- Model：Regression, Neural Network, Ensemble

We expect that we can do the one week forecast so our we set our validation period more than one week to observe the outcome. In addition, the actual data only we use is only about three months. We evaluated the use of two weeks of validation period so that we had enough data to do the training.

From the data, we found that it lack of global trend and the seasonality and it can be seen that there will be different level over time, but the local level still maintained continuously.Thus, we considered that the new data will be affected by the recent data so we choose the naive model with last day as our benchmark.

Although we can not understand the reason for the different level over time, we found some external reasons that the different post themes would affect the traffic so we added the six post theme as our external data to do the analysis. In the case of external data, we choose the model like regression and neural network that can be incorporated into external data.

We used the traffic lag(lag 1~lag 7) because we thought that the traffic for the first seven days was closed related to the traffic of the day, In addition, we took the three most popular themes lag 1 to eliminate some delay factors because effect of post would be last for more than one day. Thus, we used the roll forward everyday to forecast for the next fourteen days.

Result

From Figure 3. left plot which is all the data and validation, both model can capture the level well.

From Figure 3. right plot which only focus on the validation period data,we can see the Linear regression tends to much easier to capture local trend and be over forecast. Compared to the the regression and actual, NNet tend to capture the transition and be under forecast, so we use ensemble to combine the the result from the regression and the NNet model to search for the better result.

[3] Table3. The table combined with our time series and external data (In Appendix)

In addition, since we had used the lag, we must used the predicted data to forecast for future.Therefore, we can found that in the same validation period, the first seven days were better. After more than seven days, it was easy to cause the different level over time and captured the level not well.

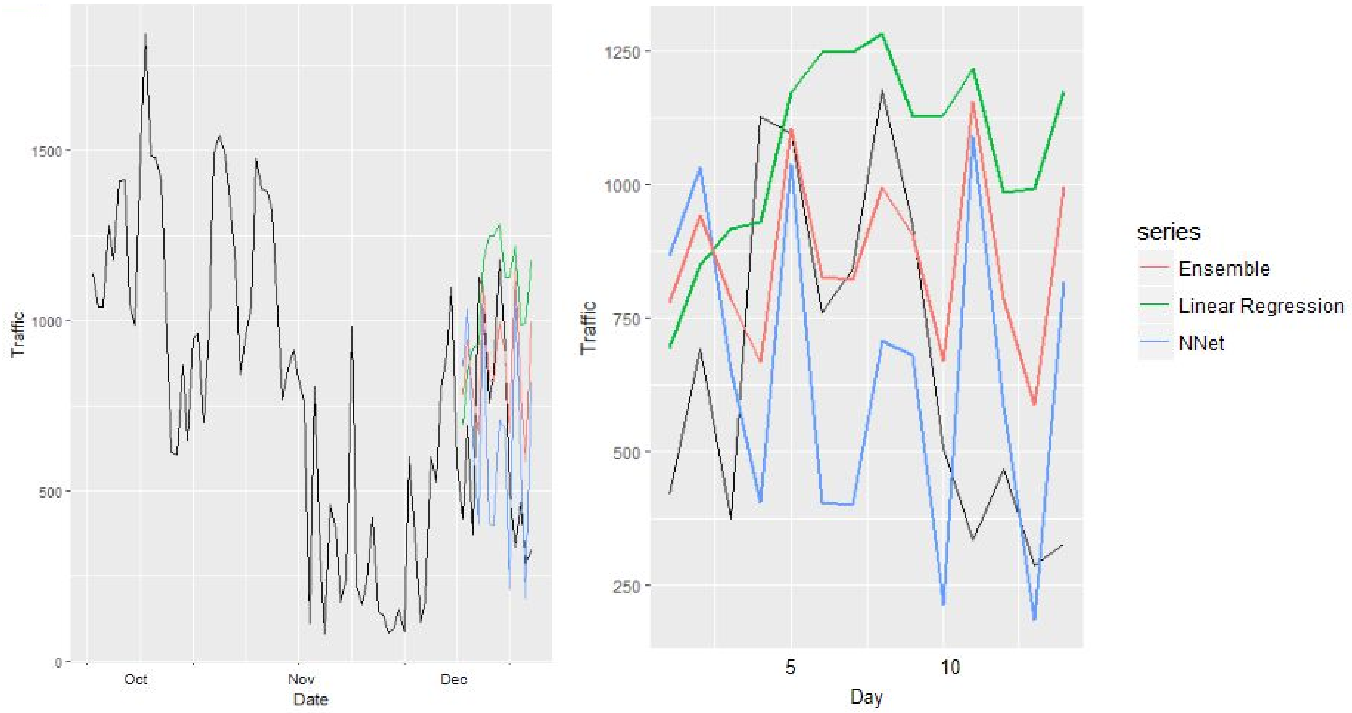
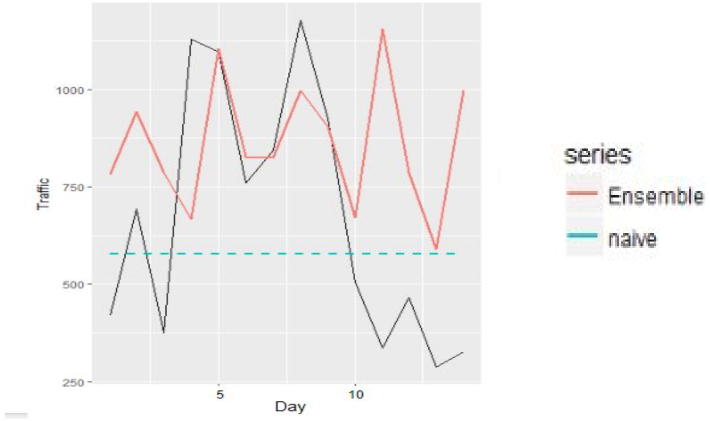
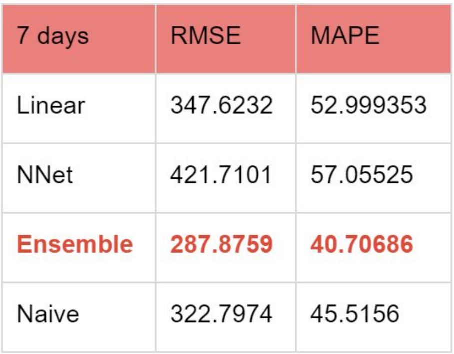


Figure3. Forecast result with validation in 14 days (right one focus on validation only)

Compared to the benchmark, the performance(RMSE,MAPE) of ensemble was better in the validation of the first seven days. However, when the time length extended from the seven days to the fourteen days, the performance of naive was better. In summary, the suitable valid range for the prediction was near seven days.





7 days	RMSE	MAPE
Linear	347.6232	52.999353
NNet	421.7101	57.05525
Ensemble	**287.8759**	**40.70686**
Naive	322.7974	45.5156

Figure4. Compare forecast result to Benchmark(Naive) & The first 7 days of validation performance

Conclusions

We suggest them to record the adjustment and promotion they have done in A/B testing, because variables affect a lot, if the record is more detailed, then the forecast result will be more accurate, and there will be more forecast method can be used, for example, separate series from different theme of posts, then we'll have 6 series to do forecast by using most suitable method.

In the future, they can use the tool to do A/B testing, in order to get a comparable baseline to verify the effect of improvement. But because of the limitation of the tool, we also suggest TC set the forecast period within 7 days to maintain the higher accuracy. It will be more convenient for marketing team to use if the tool be optimized by using Shiny.

In the future, we can combine Google API to automate data collection to offer complete solution, it'll be beneficial for the employee who are not familiar with coding.

Implementation/ Production

1. Update the traffic data(From Google Analytics).
2. Fill the past posting data in the excel file.
3. Get the suitable schedule plan of posting next week in another excel file.[4]
4. Run R code.
5. Use the forecast outcome to try new posting strategy for next week.
6. After one week, social manager can compare the improved post strategy(test group) with prediction(control group) to achieve similar A/B testing effect and prompt the performance of post strategy continuously.

For TC social media manager can use our tool to conduct the A/B testing and see the performance of different post plan. There are four steps that can implement it.

First, update the data, fill the past data in excel, run the code and get the post strategy next week. Manager can use the outcome as the control group of A/B testing. Then,manager can try new post strategy for next week of post theme and as the test group. By doing this, manager can see the effect of the treatment which is the plan of post and find the best plan strategy to create the maximum traffic.

[4] Figure5. The table structure of excel file in step 3.

Appendix

Figure 1. The example of a good revision should behave.

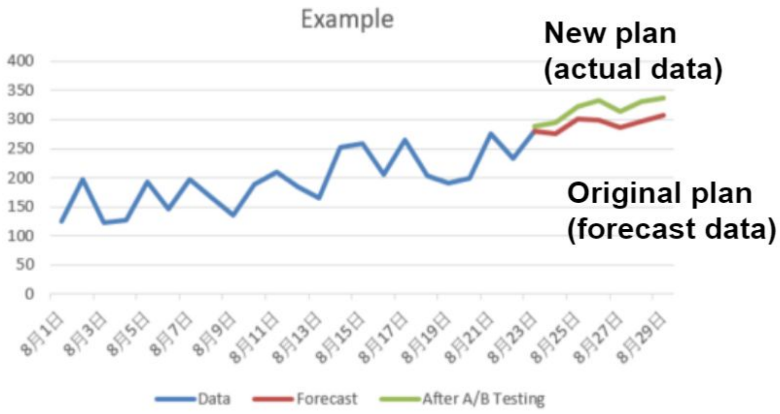


Figure 2. The dataset how we do data pre-processing

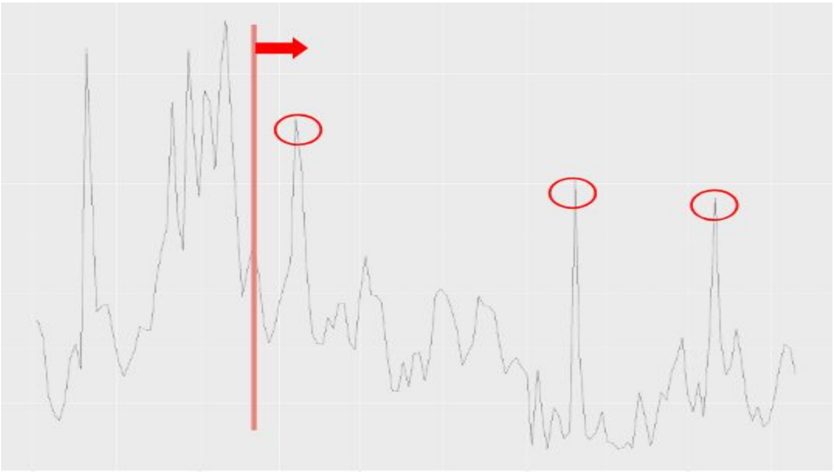


Table 3. The table combined with our time series and external data

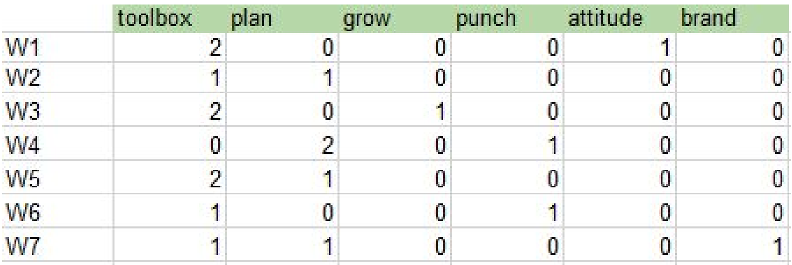
Time	Traffic (daily)	Jinrih toolbox	Jinrih planning	Jinrih growth	Jinrih check-in	Jinrih attitude	Jinrih brand
2018/9/1	(Total in day)	2	1	0	0	0	0
2018/9/2	(Total in day)	1	0	0	0	1	0
...	...						

Table 4. The first 7 days and the all 14 days of validation performance

7 days	RMSE	MAPE
Linear	347.6232	52.999353
NNet	421.7101	57.05525
Ensemble	**287.8759**	**40.70686**
Naive	322.7974	45.5156

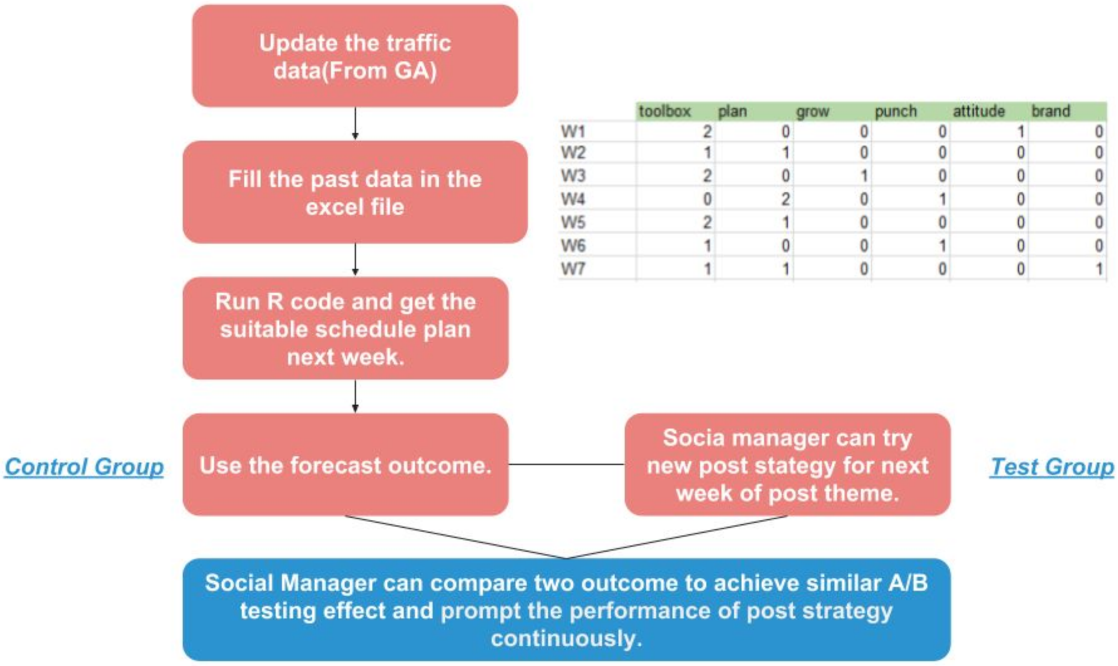
14 days	RMSE	MAPE
Linear	504.17	100.417
NNet	417.9051	68.5472
Ensemble	373.5245	68.40184
Naive	**322.7974**	**45.5156**

Figure 5. The table structure of excel file in Implement step 3.



	toolbox	plan	grow	punch	attitude	brand
W1	2	0	0	0	1	0
W2	1	1	0	0	0	0
W3	2	0	1	0	0	0
W4	0	2	0	1	0	0
W5	2	1	0	0	0	0
W6	1	0	0	1	0	0
W7	1	1	0	0	0	1

Figure 6. Implement Process



Update the traffic data(From GA)

Fill the past data in the excel file

Run R code and get the suitable schedule plan next week.

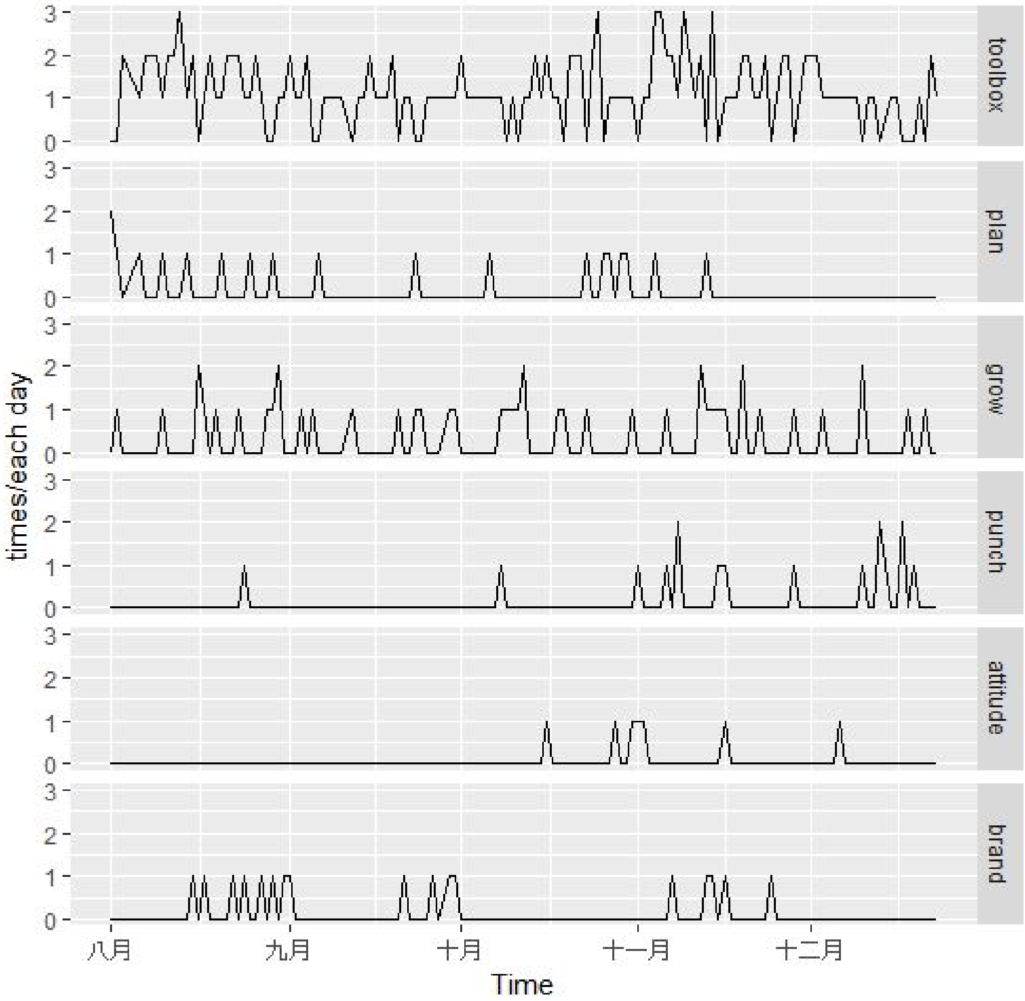
Use the forecast outcome.

Socia manager can try new post stategy for next week of post theme.

Social Manager can compare two outcome to achieve similar A/B testing effect and prompt the performance of post strategy continuously.

	toolbox	plan	grow	punch	attitude	brand
W1	2	0	0	0	1	0
W2	1	1	0	0	0	0
W3	2	0	1	0	0	0
W4	0	2	0	1	0	0
W5	2	1	0	0	0	0
W6	1	0	0	1	0	0
W7	1	1	0	0	0	1

Figure7. External data series (6 type of post category)



Code:

```
## Load libraries
library(forecast)
library(ggplot2)
library(lubridate)
library(tidyverse)
library(xts)

# Read data
library(readxl)
df <- read_excel("BAFT/Project/8_10_data.xlsx")

# 轉成一個df
col_name <- c('date', 'from', 'value')
for(n in 1:(length(names(df))/3)){
  num <- 3*n
  df_temp <- df %>%
    select(names(df)[(num-2):num])
  names(df_temp) <- col_name
  if(n==1)
    df_all <- df_temp
  else
    df_all <- rbind(df_all, df_temp)
}

#分割各檔
df.Facebook_PicSee <- df_all %>%
  filter(from == 'Facebook_PicSee') %>%
  select('date', 'value')
df.facebook.no <- df_all %>%
  filter(from == 'facebook') %>%
  select('date', 'value')
df.facebook <- df_all %>%
  filter(from == 'facebook.com') %>%
  select('date', 'value')
df.l.facebook <- df_all %>%
  filter(from == 'l.facebook.com') %>%
  select('date', 'value')
df.m.facebook <- df_all %>%
  filter(from == 'm.facebook.com') %>%
  select('date', 'value')

# 執行各檔
change_data <- function(data){
  data.xts <- xts(data$value,
                  order.by = as.POSIXlt(as.character(data$date),
format="%Y%m%d%H"))
  df.zero <- xts(rep(0, 2208), order.by = as.POSIXlt("2018-08-01 00:00")
+ (0:2207)*3600)
  data.xts <- merge(data.xts, df.zero)
  data.xts$data.xts <- na.fill(data.xts$data.xts, 0)
  return(data.xts)
}

df.Facebook_PicSee.xts <- change_data(df.Facebook_PicSee)
df.facebook.no.xts <- change_data(df.facebook.no)
df.facebook.xts <- change_data(df.facebook)
df.l.facebook.xts <- change_data(df.l.facebook)
df.m.facebook.xts <- change_data(df.m.facebook)
df.all.xts <- df.facebook.no.xts + df.facebook.xts + df.l.facebook.xts +
df.m.facebook.xts

autoplot(df.all.xts$data.xts)

#write.csv(as.data.frame(df.Facebook_PicSee.xts$data.xts),
'Facebook_PicSee.csv', row.names=TRUE)
#write.csv(as.data.frame(df.facebook.no.xts$data.xts),
'facebook.no.csv', row.names=TRUE)
#write.csv(as.data.frame(df.facebook.xts$data.xts), 'facebook.csv',
row.names=TRUE)
#write.csv(as.data.frame(df.l.facebook.xts$data.xts), 'l.facebook.csv',
row.names=TRUE)
#write.csv(as.data.frame(df.m.facebook.xts$data.xts), 'm.facebook.csv',
row.names=TRUE)
#write.csv(as.data.frame(df.all.xts$data.xts), 'all.facebook.csv',
row.names=TRUE)

# 轉成天-總數
df.all.xts <- apply.daily(df.all.xts, sum)
colnames(df.all.xts) <- "value"
df_temp <- data.frame(date = index(df.all.xts), value=df.all.xts$value)
rownames(df_temp) <- NULL
df_temp$date <- as.character(df_temp$date)
df_temp$date <- substr(df_temp$date, start = 1, stop = 10)

df.all.xts.2 <- xts(df_temp$value,
                    order.by = as.POSIXct(df_temp$date),
format="%Y%m%d")
colnames(df.all.xts.2) <- "value"
```

```
#### 額外資訊(External information)
df_ext <- read_excel("BAFT/Project/8-11_post.xlsx",
                     sheet = "Overview", col_types = c("date", "blank",
"numeric",
                                                       "numeric",
"numeric",
                                                       "numeric",
"numeric", "numeric",
                                                       "numeric",
"numeric", "numeric",
                                                       "date"))
df_ext[is.na(df_ext)] <- 0
head(df_ext)

#colnames(df_ext) <- c("date", "like_num", "message_num", "share_num",
"toolbox", "plan",
#                      "grow", "punch", "attitude", "brand",
"hyperlink", "date_time")

df_ext_2 <- df_ext %>%
  select(c("date_time", "toolbox", "plan", "grow", "punch", "attitude",
"brand", "date"))

df_ext_2 <- aggregate (cbind(toolbox, plan, grow, punch, attitude,
brand) ~ date,
                       data = df_ext_2, FUN = sum)
df_ext_2$date <- as.character(df_ext_2$date)

#### Merge them (流量資料&額外資料)
df.sum <- merge(df_temp, df_ext_2, by = "date", all.x = T)
df.sum[is.na(df.sum)] <- 0

### Time Series
# 暫時取8-9月
#df.sum <- df.sum[1:61,]
#df.ts <- ts(df.sum$value, start = c(1,1), freq = 24)
df.ts <- ts(df.sum$value)
autoplot(df.ts)

nvalid <- 7
ntrain <- dim(df.sum)[1]-nvalid
train.ts <- window(df.ts, start = c(1, 1), end = c(1, ntrain))
valid.ts <- window(df.ts, start = c(1, ntrain + 1), end = c(1, ntrain +
nvalid))
```

```
# 新增dummy
matrix <- df.sum[, -1]

# train和valid長度，可改
len <- dim(df.sum)[1]
train <- matrix[1:(len-nvalid),]
valid <- matrix[((len-nvalid)+1):len,]

## tslm
ytrain.ts <- ts(train$value)
yvalid.ts <- ts(valid$value)
xtrain <- train[,-1]
xvalid <- valid[,-1]
formula <- as.formula(paste("ytrain.ts", paste(c("trend",
colnames(xtrain)), collapse="+"),
                            sep="~"))
tslm_m <- tslm(formula, data=xtrain, lambda=1)
tslm_m.pred <- forecast(tslm_m, newdata = xvalid)
autoplot(tslm_m.pred)
lm_m<- lm(value ~ 0 + ., data = train)
summary(lm_m)
accuracy(tslm_m.pred$fitted,ytrain.ts)

autoplot(train.ts) +
  autolayer(tslm_m.pred$fitted)

autoplot(valid.ts) +
  autolayer(tslm_m.pred$mean)

autoplot(df.ts) +
  autolayer(tslm_m.pred$mean)


# NNet
df.nnetar <- nnetar(ytrain.ts, xreg=xtrain, repeats = 5, size=10)
df.nnetar.pred <- forecast(df.nnetar, xreg=xvalid)
accuracy(df.nnetar.pred$fitted,ytrain.ts)

autoplot(train.ts) +
  autolayer(df.nnetar.pred$fitted, series = "train")

autoplot(valid.ts) +
  autolayer(df.nnetar.pred$mean)
```

```
# Compare
autoplot(df.ts) +
  autolayer(tslm_m.pred$mean)+
  autolayer(df.nnetar.pred$mean)



#### 找lag
library(forecastxgb)
xg <- xgbar(train.ts)
xg$fitted
summary(xg)

#### 找lag(+external data)
xg <- xgbar(train.ts, xreg=xtrain)
xg$fitted
summary(xg)

#### Merge them (流量資料&額外資料)
df.sum <- merge(df_temp, df_ext_2, by = "date", all.x = T)
df.sum[is.na(df.sum)] <- 0

## 加入lag-1
df.sum.lag.1 <- lag(df.sum$value, 1)
df.sum$lag.1 <- df.sum.lag.1/mean(df.sum$value)
df.sum.lag.2 <- lag(df.sum$value, 2)
df.sum$lag.2 <- df.sum.lag.2/mean(df.sum$value)
df.sum.lag.3 <- lag(df.sum$value, 3)
df.sum$lag.3 <- df.sum.lag.3/mean(df.sum$value)
df.sum.lag.4 <- lag(df.sum$value, 4)
df.sum$lag.4 <- df.sum.lag.4/mean(df.sum$value)
df.sum.lag.5 <- lag(df.sum$value, 5)
df.sum$lag.5 <- df.sum.lag.5/mean(df.sum$value)
df.sum.lag.6 <- lag(df.sum$value, 6)
df.sum$lag.6 <- df.sum.lag.6/mean(df.sum$value)
df.sum.lag.7 <- lag(df.sum$value, 7)
df.sum$lag.7 <- df.sum.lag.7/mean(df.sum$value)


df.sum.toolbox.lag.1 <- lag(df.sum$toolbox, 1)
df.sum$toolbox.lag.1 <- df.sum.toolbox.lag.1
df.sum.plan.lag.1 <- lag(df.sum$plan, 1)
df.sum$plan.lag.1 <- df.sum.plan.lag.1
df.sum.grow.lag.1 <- lag(df.sum$grow, 1)
df.sum$grow.lag.1 <- df.sum.grow.lag.1


#### 重跑一次 ####
# 刪除NA
df.sum <- na.omit(df.sum)

### Time Series
# 暫時取8-9月
#df.sum <- df.sum[1:61,]
#df.ts <- ts(df.sum$value, start = c(1,1), freq = 24)
df.ts <- ts(df.sum$value, start = c(1,1))

nvalid <- 14
ntrain <- dim(df.sum)[1]-nvalid
train.ts <- window(df.ts, start = c(1, 1), end = c(1, ntrain))
valid.ts <- window(df.ts, start = c(1, ntrain + 1), end = c(1, ntrain +
nvalid))
# 新增dummy
matrix <- df.sum[, -1]


# train和valid長度，可改
len <- dim(df.sum)[1]
train <- matrix[1:(len-nvalid),]
valid <- matrix[((len-nvalid)+1):len,]

## tslm
ytrain.ts <- ts(train$value)
yvalid.ts <- ts(valid$value)
xtrain <- train[,-1]
xvalid <- valid[,-1]
formula <- as.formula(paste("ytrain.ts", paste(c("trend",
colnames(xtrain)), collapse="+"),
                            sep="~"))
tslm_m <- tslm(formula, data=xtrain)
summary(tslm_m)
#tslm_m.pred <- forecast(tslm_m, newdata = xvalid)
#autoplot(tslm_m.pred)

# 開始準備回圈
xvalid_result <- train

# 放新結果之list
new_result <- c()
```

```
## 寫迴圈做forward
#
tail(xvalid_result, 5)

num <- 1
for(num in c(1:nvalid)){
  xvalid_result <- rbind(xvalid_result, valid[num,])
  xvalid_result[dim(xvalid_result)[1],]$value <- result

  xvalid_result[dim(xvalid_result)[1], c(2:7)] <- xvalid[num, c(1:6)]
  # 新一筆external
  xvalid_result[dim(xvalid_result)[1],]$lag.1 <-
tail(lag(xvalid_result$value, 1)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$lag.2 <-
tail(lag(xvalid_result$value, 2)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$lag.3 <-
tail(lag(xvalid_result$value, 3)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$lag.4 <-
tail(lag(xvalid_result$value, 4)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$lag.5 <-
tail(lag(xvalid_result$value, 5)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$lag.6 <-
tail(lag(xvalid_result$value, 6)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$lag.7 <-
tail(lag(xvalid_result$value, 7)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$toolbox.lag.1 <-
tail(lag(xvalid_result$toolbox, 1), 1)
  xvalid_result[dim(xvalid_result)[1],]$plan.lag.1 <-
tail(lag(xvalid_result$plan, 1), 1)
  xvalid_result[dim(xvalid_result)[1],]$grow.lag.1 <-
tail(lag(xvalid_result$grow, 1), 1)
  #  df.nnetar.pred <- forecast(df.nnetar, xreg=tail(xvalid_result[,
-1], 1))
  tslm_m.pred <- forecast(tslm_m, newdata = tail(xvalid_result[, -1],
1))

  result <- as.integer(tslm_m.pred$mean)
  xvalid_result[dim(xvalid_result)[1],'value'] <- result
  # 放入新結果
  new_result <- c(new_result, result)
}

accuracy(new_result, yvalid.ts)
```

```
autoplot(train.ts) +
  autolayer(tslm_m.pred$fitted)

new_result.ts <- ts(new_result, start=c(ntrain+1, 1))
autoplot(valid.ts) +
  autolayer(new_result.ts)

autoplot(df.ts) +
  autolayer(new_result.ts)

tslm.ts <- new_result.ts

### NNet
df.nnetar <- nnetar(ytrain.ts, xreg=xtrain, repeats = 5, size=10)
#df.nnetar.pred.test <- forecast(df.nnetar, xreg=xvalid)

# 創一新df，暫存新external data

# xvalid_result為新的參考，用以forecast
xvalid_result <- train

# 放新結果之list
new_result <- c(tail(train$value, 1))

## 寫迴圈做forward
#
tail(xvalid_result, 5)

num <- 1
for(num in c(1:nvalid)){
  xvalid_result <- rbind(xvalid_result, valid[num,])
  xvalid_result[dim(xvalid_result)[1],]$value <- result

  xvalid_result[dim(xvalid_result)[1], c(2:7)] <- xvalid[num, c(1:6)]
  # 新一筆external
  xvalid_result[dim(xvalid_result)[1],]$lag.1 <-
tail(lag(xvalid_result$value, 1)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$lag.2 <-
tail(lag(xvalid_result$value, 2)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$lag.3 <-
tail(lag(xvalid_result$value, 3)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$lag.4 <-
tail(lag(xvalid_result$value, 4)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$lag.5 <-
tail(lag(xvalid_result$value, 5)/mean(df.sum$value), 1)
```

```
  xvalid_result[dim(xvalid_result)[1],]$lag.6 <-
tail(lag(xvalid_result$value, 6)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$lag.7 <-
tail(lag(xvalid_result$value, 7)/mean(df.sum$value), 1)
  xvalid_result[dim(xvalid_result)[1],]$toolbox.lag.1 <-
tail(lag(xvalid_result$toolbox, 1), 1)
  xvalid_result[dim(xvalid_result)[1],]$plan.lag.1 <-
tail(lag(xvalid_result$plan, 1), 1)
  xvalid_result[dim(xvalid_result)[1],]$grow.lag.1 <-
tail(lag(xvalid_result$grow, 1), 1)
  df.nnetar.pred <- forecast(df.nnetar, xreg=tail(xvalid_result[, -1],
1))

  result <- as.integer(df.nnetar.pred$mean)
  result_temp <- (tail(new_result, 1) + result)/2
  xvalid_result[dim(xvalid_result)[1],'value'] <- result_temp
  # 放入新結果
  new_result <- c(new_result, result_temp)
}

num <- 2

new_result <- new_result[-1]
tail(xvalid_result, 10)
tail(valid, 12)
accuracy(df.nnetar.pred$fitted,ytrain.ts)
accuracy(new_result,yvalid.ts)

autoplot(train.ts) +
  autolayer(df.nnetar.pred$fitted, series = "train")

new_result.ts <- ts(new_result, start=c(ntrain+1, 1))
autoplot(valid.ts) +
  autolayer(new_result.ts)


# Compare
autoplot(df.ts, xlab = "Date", ylab = "Traffic") +
  autolayer(tslm.ts, series = "Linear Regression") +
  autolayer(new_result.ts, series = "NNet")
```