Part 3 Machine learning

```
library(readr)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(xts)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
getwd()
```

```
## [1] "/Users/yingding/Desktop/Forcasting M2/Project and data-20220107/github code/Part 3"
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:xts':
##
##     first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(fpp)
```

```
## Loading required package: fma
```

```
## Loading required package: expsmooth
```

```
## Loading required package: lmtest
```

```
## Loading required package: tseries
```

```
library(fpp3)
```

```
## -- Attaching packages --------------------------------------------- fpp3 0.4.0 --
```

```
## v tibble      3.1.0      v tsibble     1.1.1
## v tidyr       1.1.3      v tsibbledata 0.4.0
## v lubridate   1.7.10     v feasts      0.2.2
## v ggplot2     3.3.5      v fable       0.3.1
```

```
## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x lubridate::date()      masks base::date()
## x dplyr::filter()        masks stats::filter()
## x dplyr::first()         masks xts::first()
## x fabletools::forecast() masks forecast::forecast()
## x tsibble::index()       masks zoo::index()
## x tsibble::intersect()   masks base::intersect()
## x tsibble::interval()    masks lubridate::interval()
## x dplyr::lag()           masks stats::lag()
## x dplyr::last()          masks xts::last()
## x tsibble::setdiff()     masks base::setdiff()
## x tsibble::union()       masks base::union()
```

```
##
## Attaching package: 'fpp3'
```

```
## The following object is masked from 'package:fpp':
##
##     insurance
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(ggplot2)
library(ranger)
library(ftsa)
```

```
## Loading required package: rainbow
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following objects are masked from 'package:fma':
##
##     cement, housing, petrol
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Loading required package: pcaPP
```

```
## Loading required package: sde
```

```
## Loading required package: stats4
```

```
## Loading required package: fda
```

```
## Loading required package: splines
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loading required package: fds
```

```
## Loading required package: RCurl
```

```
##
## Attaching package: 'RCurl'
```

```
## The following object is masked from 'package:tidyr':
##
##     complete
```

```
## The following object is masked from 'package:lmtest':
##
##     reset

## Loading required package: deSolve

##
## Attaching package: 'fda'

## The following object is masked from 'package:forecast':
##
##     fourier

## The following object is masked from 'package:graphics':
##
##     matplot

## sde 2.0.15

## Companion package to the book

## 'Simulation and Inference for Stochastic Differential Equations With R Examples'

## Iacus, Springer NY, (2008)

## To check the errata corrige of the book, type vignette("sde.errata")

##
## Attaching package: 'ftsa'

## The following object is masked from 'package:tidyr':
##
##     extract

## The following objects are masked from 'package:stats':
##
##     sd, var
```

```
library(Metrics)
```

```
##
## Attaching package: 'Metrics'

## The following object is masked from 'package:fabletools':
##
##     accuracy

## The following object is masked from 'package:forecast':
##
##     accuracy
```

```
library(rmarkdown)
library(fable)
```

By comparing with the two models and

1. Prepare train test data, same data is applied to the pyton code

```
data <- read_csv("/Users/yingding/Desktop/Forcasting M2/Project and data-20220107/Projectdata.csv")
```

```
## Rows: 1969 Columns: 19

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (18): Hobbies_CA_1, Household_1_CA_1, Household_2_CA_1, Foods_1_CA_1, F...
## date  (1): date

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
tsdata <- xts(data[,2:19], as.Date(data$date))

split2<- sample(c(rep(0, 0.8 * nrow(data)), rep(1, 0.2 * nrow(data))))
table(split2)
```

```
## split2
##    0    1
## 1575  393
```

```
train2 <- head(tsdata, 1575)
test2 <- tail(tsdata, 394)
```

2. Prophet Model

```
#import forecast results from the model
forecast_p <- read.csv('/Users/yingding/Desktop/Forcasting M2/Project and data-20220107/github code/Par

#prepare data for comparision - h = 28
yhat <- tail(forecast_p,28)
actual <- head(test2, 28)
```

```
mat = matrix(ncol = 18, nrow =4)
df_res = data.frame(mat)
names(df_res) = colnames(yhat)[-1]
```

```
#prepare data for comparision
mat = matrix(ncol = 18, nrow =4)
df_res = data.frame(mat)
names(df_res) = colnames(yhat)[-1]
```

```
rownames(df_res) <- c("MAE","MAPE", "MSE", "RMSE")
for (i in 2:length(colnames(yhat))){
  df_res[1,i-1] = mae(actual, yhat[,i])
  df_res[2,i-1] = mape(actual, yhat[,i])
  df_res[3,i-1] = mse(actual, yhat[,i])
  df_res[4,i-1] = rmse(actual, yhat[,i])
}

print(df_res)
```

```
##      Hobbies_CA_1 Household_1_CA_1 Household_2_CA_1 Foods_1_CA_1 Foods_2_CA_1
## MAE  3.483900e+02     3.190381e+02     3.544973e+02 3.435521e+02 3.296063e+02
## MAPE 1.003287e+00     6.043130e-01     5.052791e-01 4.896502e-01 7.886125e-01
## MSE  3.802001e+05     4.211768e+05     4.890513e+05 4.772386e+05 3.962879e+05
## RMSE 6.166037e+02     6.489814e+02     6.993220e+02 6.908246e+02 6.295140e+02
##      Foods_3_CA_1 Hobbies_CA_2 Household_1_CA_2 Household_2_CA_2 Foods_1_CA_2
## MAE     1265.2584 3.268644e+02     3.378379e+02     3.409347e+02 4.543623e+02
## MAPE       6.6093 5.055347e-01     4.991031e-01     4.965983e-01 8.667492e-01
## MSE  1832242.5117 4.521590e+05     4.673187e+05     4.731192e+05 5.762233e+05
## RMSE    1353.6035 6.724277e+02     6.836071e+02     6.878366e+02 7.590937e+02
##      Foods_2_CA_2 Foods_3_CA_2 Hobbies_CA_3 Household_1_CA_3 Household_2_CA_3
## MAE  3.654450e+02 5.434772e+02 3.943166e+02     3.834676e+02     3.289928e+02
## MAPE 5.383347e-01 2.347917e+00 1.396763e+00     1.316753e+00     4.924771e-01
## MSE  4.986366e+05 4.274152e+05 3.678169e+05     3.675996e+05     4.579279e+05
## RMSE 7.061421e+02 6.537700e+02 6.064791e+02     6.062999e+02     6.767037e+02
##      Foods_1_CA_3 Foods_2_CA_3 Foods_3_CA_3
## MAE  3.230886e+02 3.631468e+02 1.744850e+03
## MAPE 5.153701e-01 1.124527e+00 9.055875e+00
## MSE  4.452732e+05 3.727137e+05 3.363844e+06
## RMSE 6.672880e+02 6.105028e+02 1.834078e+03
```

2. LSTM Model

```
#import forecast results from the model
forecast_l <- read.csv('/Users/yingding/Desktop/Forcasting M2/Project and data-20220107/github code/Par
#prepare data for comparision - h = 28
yhat_l <- tail(forecast_l,28)
yhat_l <- xts(yhat_l[,2:19], as.Date(yhat_l$date))
actual <- head(test2, 28)
```

```
mat = matrix(ncol = 18, nrow =4)
df_res_2 = data.frame(mat)
names(df_res_2) = colnames(yhat_l)[-1]
```

```
#prepare data for comparision
mat = matrix(ncol = 18, nrow =4)
df_res_2 = data.frame(mat)
names(df_res_2) = colnames(yhat)[-1]
rownames(df_res) <- c("MAE","MAPE", "MSE", "RMSE")
for (i in 2:length(colnames(yhat_l))){
  df_res_2[1,i-1] = mae(actual, as.numeric(yhat_l[,i]))
  df_res_2[2,i-1] = mape(actual, as.numeric(yhat_l[,i]))
```

```
  df_res_2[3,i-1] = mse(actual, as.numeric(yhat_l[,i]))
  df_res_2[4,i-1] = rmse(actual, as.numeric(yhat_l[,i]))
}

print(df_res_2)
```

```
##   Hobbies_CA_1 Household_1_CA_1 Household_2_CA_1 Foods_1_CA_1 Foods_2_CA_1
## 1 6.248793e+02     6.270582e+02     5.090765e+02 4.502174e+02 5.113087e+02
## 2 2.937061e+00     2.953863e+00     2.179148e+00 1.784051e+00 2.199315e+00
## 3 4.859466e+05     4.908688e+05     4.096162e+05 3.782206e+05 4.082164e+05
## 4 6.970987e+02     7.006203e+02     6.400127e+02 6.149964e+02 6.389181e+02
##   Foods_3_CA_1 Hobbies_CA_2 Household_1_CA_2 Household_2_CA_2 Foods_1_CA_2
## 1 5.111815e+02    539.77287        540.66672     3.466126e+02 4.669280e+02
## 2 2.218202e+00      2.37855          2.39367     9.978391e-01 1.885032e+00
## 3 4.192466e+05 420440.79418     420485.98765     3.844328e+05 3.888420e+05
## 4 6.474926e+02    648.41406        648.44891     6.200264e+02 6.235720e+02
##   Foods_2_CA_2 Foods_3_CA_2 Hobbies_CA_3 Household_1_CA_3 Household_2_CA_3
## 1 4.395724e+02 6.601648e+02 5.030401e+02     4.912168e+02     5.487551e+02
## 2 1.738826e+00 3.171762e+00 2.130125e+00     2.081829e+00     2.462267e+00
## 3 3.930222e+05 5.276030e+05 3.986027e+05     4.092415e+05     4.372327e+05
## 4 6.269148e+02 7.263629e+02 6.313499e+02     6.397198e+02     6.612357e+02
##   Foods_1_CA_3 Foods_2_CA_3 Foods_3_CA_3
## 1 5.358037e+02 5.273126e+02           NA
## 2 2.343146e+00 2.303866e+00           NA
## 3 4.174561e+05 4.157895e+05           NA
## 4 6.461084e+02 6.448174e+02           NA
```

```
print(df_res) #summary of prophet mode;
```

```
##      Hobbies_CA_1 Household_1_CA_1 Household_2_CA_1 Foods_1_CA_1 Foods_2_CA_1
## MAE  3.483900e+02     3.190381e+02     3.544973e+02 3.435521e+02 3.296063e+02
## MAPE 1.003287e+00     6.043130e-01     5.052791e-01 4.896502e-01 7.886125e-01
## MSE  3.802001e+05     4.211768e+05     4.890513e+05 4.772386e+05 3.962879e+05
## RMSE 6.166037e+02     6.489814e+02     6.993220e+02 6.908246e+02 6.295140e+02
##      Foods_3_CA_1 Hobbies_CA_2 Household_1_CA_2 Household_2_CA_2 Foods_1_CA_2
## MAE     1265.2584 3.268644e+02     3.378379e+02     3.409347e+02 4.543623e+02
## MAPE       6.6093 5.055347e-01     4.991031e-01     4.965983e-01 8.667492e-01
## MSE  1832242.5117 4.521590e+05     4.673187e+05     4.731192e+05 5.762233e+05
## RMSE    1353.6035 6.724277e+02     6.836071e+02     6.878366e+02 7.590937e+02
##      Foods_2_CA_2 Foods_3_CA_2 Hobbies_CA_3 Household_1_CA_3 Household_2_CA_3
## MAE  3.654450e+02 5.434772e+02 3.943166e+02     3.834676e+02     3.289928e+02
## MAPE 5.383347e-01 2.347917e+00 1.396763e+00     1.316753e+00     4.924771e-01
## MSE  4.986366e+05 4.274152e+05 3.678169e+05     3.675996e+05     4.579279e+05
## RMSE 7.061421e+02 6.537700e+02 6.064791e+02     6.062999e+02     6.767037e+02
##      Foods_1_CA_3 Foods_2_CA_3 Foods_3_CA_3
## MAE  3.230886e+02 3.631468e+02 1.744850e+03
## MAPE 5.153701e-01 1.124527e+00 9.055875e+00
## MSE  4.452732e+05 3.727137e+05 3.363844e+06
## RMSE 6.672880e+02 6.105028e+02 1.834078e+03
```

Summary of analysis: By comparing the results of two models, the model prophet have relatively lower erros than the LTSM model for 28 days forecast. For prophet model, the variables Foods_3_CA_1 has

very large erros. For LTSM models, the erros are mainly coming from the variables Hobbies_CA_2 and Household_1_CA_2 In general, the performance of the models vary across the variables.