

Chapter. 05

R시계열 라이브러리 활용

| Prophet by 페이스북북

FASTCAMPUS
ONLINE

금융공학/퀀트 I

강사. 김경륜

I Prophet by 페이스북

Prophet 공식사이트 : <https://facebook.github.io/prophet/>

Prophet은 페이스북에서 만든 시계열예측 오픈소스 패키지
R과 Python에서 설치하여 사용가능

Prophet 특징 (일별데이터에 적합)

1. 비선형추세 산출(Non-linear trend)
: 시계열 Automatic Chang-Point Detection + 베이지안 방법(prior for trend)
2. 일별 및 연도별 계절성 계산(daily and yearly seasonality)
: Fourier terms 이용하여 multi-seasonality 적용 (daily : freq=7, yearly : freq=365.25)
3. 휴일효과적용가능
4. Saturating Forecast(예측치의 범위 및 limit을 설정)

참고논문: <https://peerj.com/preprints/3190/>

I Prophet을 사용한 예측

1. prophet 및 dplyr 패키지를 로딩

```
library(prophet)  
library(dplyr)
```

2. 데이터를 불러온 후 log변환 (Prophet에서는 log변환을 권고)

```
df <- read.table('C:/example_wp_peyton_manning.csv', sep=";", header=T) %>%  
  mutate(y = log(y))
```

3. 모델링

```
m <- prophet(df)
```

4. 예측 (향후 365일 예측)

```
future <- make_future_dataframe(m, periods = 365) # 예측치를 저장할 공간  
forecast <- predict(m, future) # 예측치 산출
```

5. 그래프

```
plot(m, forecast) #전체 데이터 및 예측치 plot  
prophet_plot_components(m, forecast) # trend, weekly/yearly seasonality plot
```

I Saturating Forecast in prophet

1. 데이터를 불러온 후 log변환

```
df <- read.table('C:/example_wp_R.csv', sep=",", header=T) %>%  
  mutate(y = log(y))
```

2. Cap설정(maximum value)

```
df$cap <- 8.5  
# df$y <- 10 - df$y; df$cap <- 6; df$floor <- 1.5; future$cap <- 6; future$floor <- 1.5
```

3. 모델링

```
m <- prophet(df, growth = 'logistic')
```

4. 예측

```
future <- make_future_dataframe(m, periods = 1826)  
future$cap <- 8.5  
fcst <- predict(m, future)
```

5. plot

```
plot(m, fcst)  
prophet_plot_components(m, forecast)
```

Adding Monthly Seasonality

1. 데이터를 불러온 후 log변환

```
df <- read.table('C:/example_wp_R.csv', sep=",", header=T) %>%  
  mutate(y = log(y))
```

2. Weekly를 Monthly로 변경

```
m <- prophet(weekly.seasonality=FALSE)  
m <- add_seasonality(m, name='monthly', period=30.5, fourier.order=5)
```

3. 모델링

```
m <- fit.prophet(m, df)
```

4. 예측

```
future <- make_future_dataframe(m, periods = 365)  
forecast <- predict(m, future)
```

5. plot

```
plot(m, fcst)  
prophet_plot_components(m, forecast)
```

Adding holiday effect

playoffs 날짜 데이터 만들기

```
playoffs <- data_frame(  
  holiday = 'playoff',  
  ds = as.Date(c('2008-01-13', '2009-01-03', '2010-01-16',  
                 '2010-01-24', '2010-02-07', '2011-01-08',  
                 '2013-01-12', '2014-01-12', '2014-01-19',  
                 '2014-02-02', '2015-01-11', '2016-01-17',  
                 '2016-01-24', '2016-02-07'))),  
  lower_window = 0,  
  upper_window = 1  
)
```

슈퍼볼 날짜데이터 만들기

```
superbowls <- data_frame(  
  holiday = 'superbowl',  
  ds = as.Date(c('2010-02-07', '2014-02-02', '2016-02-07'))),  
  lower_window = 0,  
  upper_window = 1  
)
```

holiday 데이터 만들기

```
holidays <- bind_rows(playoffs, superbowls)
```

I Adding holiday effect

holiday변수를 이용하여 모델링

```
m <- prophet(df, holidays = holidays)
forecast <- predict(m, future)
```

holiday효과산출

```
forecast %>%
  select(ds, playoff, superbowl) %>%
  filter(abs(playoff + superbowl) > 0) %>%
  tail(10)
```

holiday효과 plot

```
prophet_plot_components(m, forecast)
```

Cross Validation in prophet

1. 데이터를 불러온 후 log변환

```
df <- read.table('C:/example_wp_peyton_manning.csv', sep="," , header=T) %>%  
  mutate(y = log(y))
```

2. 모델링

```
m <- prophet(m, df)
```

3. Cross-Validation

```
df.cv <- cross_validation(m, horizon = 365, units = 'days')  
head(df.cv)
```

4. 결과추출

```
df.cv$cutoff <- as.Date(df.cv$cutoff) #POSIXct포맷을 date포맷으로 변환  
unique(df.cv$cutoff) # cutoff날짜확인  
filter(df.cv, cutoff=="2011-01-19") # cutoff별 예측결과 추출
```