HDFS Forcast

Wael Emam

6/25/2018

As a prereq. for this script, I used HDFS Offline Image Viewer with "Delimited processor"

Example: hdfs oiv -i fsimage -p Delimited

```
library(dplyr)
library(anytime)
library(lubridate)
library(prophet)
```

The only three columns we need are Filesize, Replication and ModificationTime. Also by choozing FileSize > 0, removes all directories in the file from our calculation as well as zero size files.

```
files <- fsimage %>%
  filter(FileSize > 0) %>%
  select (FileSize, Replication, ModificationTime)
```

Here I am calculating actual file size on disk (file size * replication factor) and converting size to GB (not really required). And filtering on which day I want to start my calculation from.

```
files_used <- mutate(files, RawSize = (((FileSize/1024)/1024)/1024) * Replication, MTime = anytime(Modi
  select (RawSize, MTime) %>%
  group_by(day=floor_date(MTime, "day")) %>%
  filter (day > '2017-04-01') %>%
  summarize(RawSize = sum(RawSize))
```

Using Prophet library for prediction.

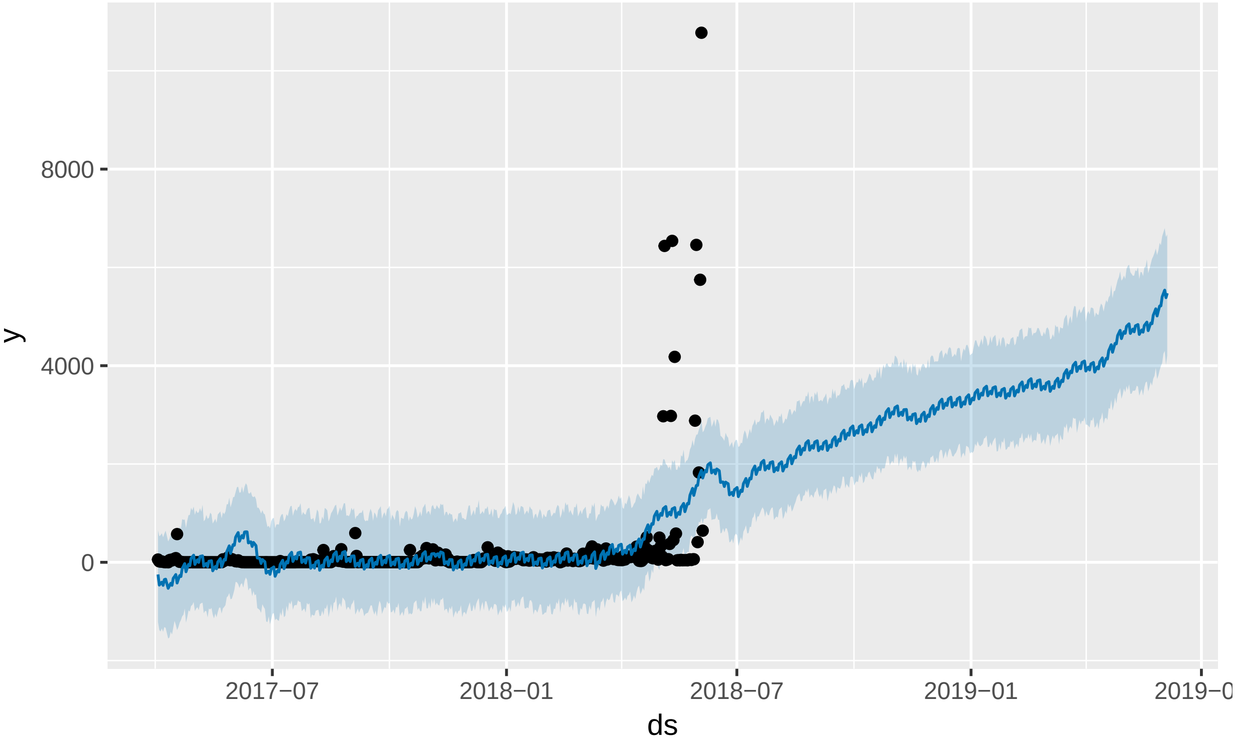
```
# Change column names for prophet to work
names(files_used)[1] <- "ds"
names(files_used)[2] <- "y"
```

```
# Prophet
m <- prophet(files_used, yearly.seasonality=TRUE)
```

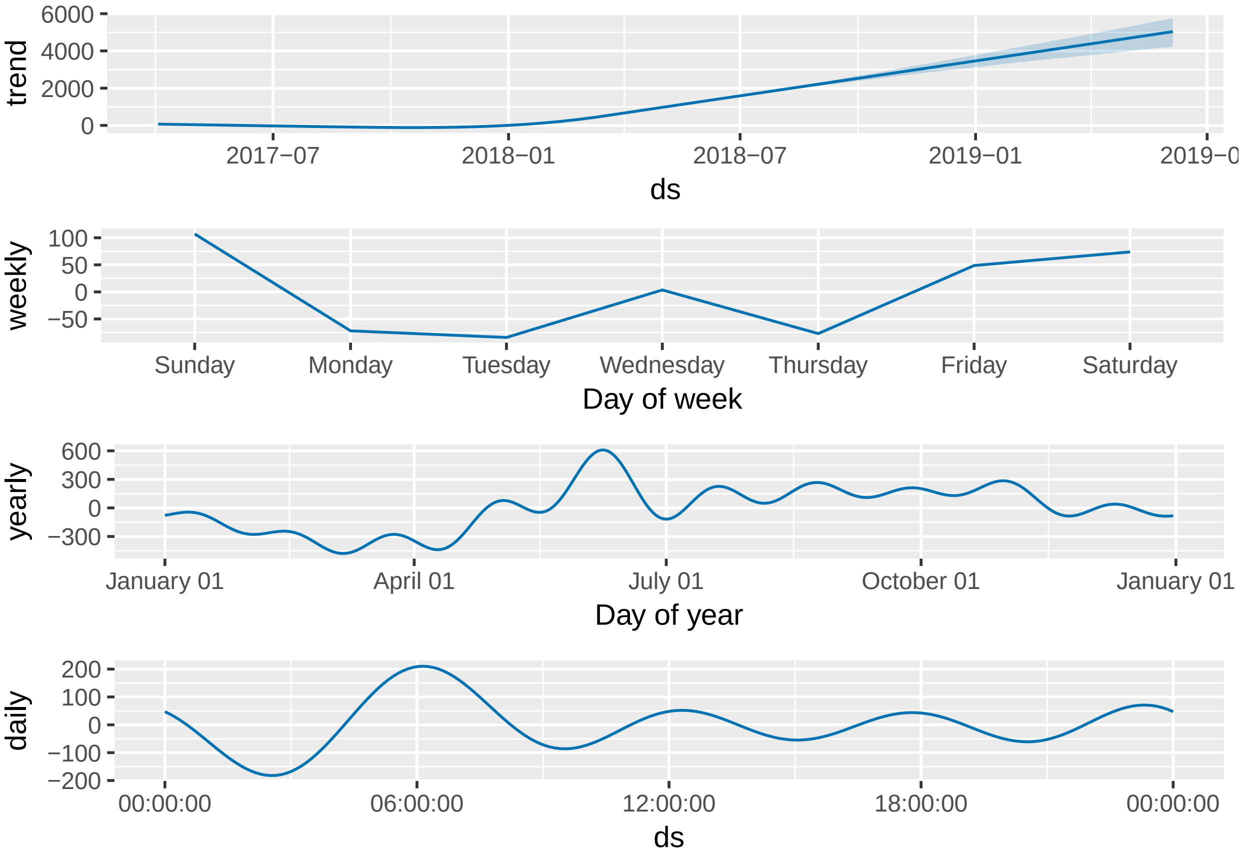
```
## Initial log joint probability = -3.32112
## Optimization terminated normally:
##   Convergence detected: relative gradient magnitude is below tolerance
```

```
# Prediction
future <- make_future_dataframe(m, periods = 365)
forcast <- predict (m, future)
```

```
# Plot
plot(m, forcast)
```



```
prophet_plot_components(m, forcast)
```


6000
4000
2000
0
trend
2017−07
2018−01
2018−07
2019−01
2019−0
ds
100
50
0
−50
weekly
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week
600
300
0
−300
yearly
January 01
April 01
July 01
October 01
January 01
Day of year
200
100
0
−100
−200
daily
00:00:00
06:00:00
12:00:00
18:00:00
00:00:00
ds