amunategui@gmail.com

amunategui.github.io

Exploring SparkR data frames:

Now, we're ready to access it from SparkR. Hopefully your cluster is ready so, as usual:

- Log into the master cluster
- add a new R user:
    - adduser your_name
    - passwd your_name

Grab the DNS or IP address of the master box, paste it in a browser with the extension :8787 and log into it.

Starting SparkR in RStudio

As usual, let's get the SparkR library setup and instantiate our Spark context:

```
library("SparkR", lib.loc="/root/spark/R/lib")
Sys.setenv(SPARK_HOME="/root/spark")

sc <- sparkR.init()
sqlContext <- sparkRSQL.init(sc)
```

Getting to our S3 data

There are different ways to explore SparkR data frames but a common one is using **dplyr**-based syntax (referred to as verbs). For a good walkthrough of the regular **dplyr** package, check out: https://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html

Let's start by coding our **read.csv** and point it to our S3 link we copied earlier. We also do some basic data cleaning such as changing the '?' to NAs and making the outcome variable a true binary:

```
diabetes <- read.csv('https://s3-us-west-
2.amazonaws.com/amunategui.diabetes/diabetic_data.csv')

# transform all "?" to 0s
diabetes[diabetes == "?"] <- NA

# prep outcome variable to those readmitted under 30 days
diabetes$readmitted <- ifelse(diabetes$readmitted == "<30",1,0)
```

Creating a SparkR data frame

The next step is to convert it to a SparkR data frame so it can be distributed across our cluster.

```
# make a spark data frame
dim(diabetes)
diabetes_sparkdf <- createDataFrame(sqlContext,
diabetes[base::sample(nrow(diabetes), 500),])

# see what we have
printSchema(diabetes_sparkdf)

columns(diabetes_sparkdf)
```

We aren't going to model anything in this section but instead look at different ways of querying this data set.

SparkR Commands

SparkR commands are based on **dplyr** (https://cran.r-project.org/web/packages/dplyr/index.html) tweaked for distributed computing. **dplyr** is a great tool though not all of its functionality is available in SparkR. Here we'll look at the following commands:

- select
- distinct

- filter
- arrange
- agg
- summarize
- groupBy
- mutate

select

```
head(select(diabetes_sparkdf, diabetes_sparkdf$readmitted,
diabetes_sparkdf$gender))
```

distinct

```
head(distinct(select(diabetes_sparkdf, diabetes_sparkdf$age)))
```

filter

```
showDF(filter(diabetes_sparkdf, diabetes_sparkdf$insulin == 'Up'
& diabetes_sparkdf$readmitted==1),3)
```

arrange

```
head(arrange(diabetes_sparkdf,
desc(diabetes_sparkdf$number_diagnoses)))

head(arrange(diabetes_sparkdf,
             desc(diabetes_sparkdf$time_in_hospital),
diabetes_sparkdf$num_medications))
```

mutate

```
number_visits <- select(diabetes_sparkdf,
                        diabetes_sparkdf$medical_specialty,
diabetes_sparkdf$gender,
                        diabetes_sparkdf$number_outpatient,
diabetes_sparkdf$number_emergency,
                        diabetes_sparkdf$number_inpatient)
head(mutate(number_visits,
       total_visits=(diabetes_sparkdf$number_outpatient +
diabetes_sparkdf$number_emergency +
                     diabetes_sparkdf$number_inpatient)))
```

summarize (or agg) / groupBy (or group_by)

```
head(summarize(groupBy(diabetes_sparkdf, diabetes_sparkdf$insulin
== 'Up' & diabetes_sparkdf$readmitted==1),
               count = n(diabetes_sparkdf$insulin == 'Up' &
```

```
diabetes_sparkdf$readmitted==1)))

head(summarize(groupBy(diabetes_sparkdf, diabetes_sparkdf$insulin
== 'Up' & diabetes_sparkdf$readmitted==1),
               count = sum(diabetes_sparkdf$readmitted)))


head(arrange(agg(groupBy(diabetes_sparkdf, diabetes_sparkdf$age),
total_diagnoses = sum(diabetes_sparkdf$number_diagnoses)),
     diabetes_sparkdf$age))
```

magrittr

It's all about pipes! Here is a great introduction from the author: http://www.r-statistics.com/2014/08/simpler-r-coding-with-pipes-the-present-and-future-of-the-magrittr-package/

```
# install.packages('magrittr')
library(magrittr)
groupBy(diabetes_sparkdf, diabetes_sparkdf$age) %>%
  summarize(total_diagnoses =
sum(diabetes_sparkdf$number_diagnoses)) %>%
  arrange(diabetes_sparkdf$age) %>%
       head
```