Automating Data Exploration with R

Basic Feature Engineering

Just like we pulled the first word, word count and character count out of text features, we can also do some basic engineering on numerical and date features.

Dates

A date field can be cast to an integer representation. Day `0` is 1/1/1970 (beginning of Unix time) - as you can imagine, this is very useful for modeling.

```
print(as.numeric(as.Date('1970-01-01')))
```

```
## [1] 0
```

But a date can yield a lot more data than just its integer representation. We'll use the `lubridate` library to asst our extractions:

- Extract day, month, and short and long year
- Day count in year
- Day of the week
- Weekend
- Quarter

Note: you will notice that there is an optional parameter to remove the original date. Some visualization and modeling tools can handle dates automatically but for simplicity here, we want our entire data set to be numerical. If that is an issue or if you want to retain the date for visualization and/or reporting, simply turn the `remove_original_date` off.

```
Feature_Engineer_Dates <- function(data_set, remove_original_date=TRUE) {
     require(lubridate)
     data_set <- data.frame(data_set)
     date_features <- names(data_set[sapply(data_set, is.Date)])
     for (feature_name in date_features) {
          data_set[,paste0(feature_name,'_DateInt')] <- as.numeric(data_set[,featu
re_name])
          data_set[,paste0(feature_name,'_Month')] <- as.integer(format(data_set[,
feature_name], "%m"))
          data_set[,paste0(feature_name,'_ShortYear')] <- as.integer(format(data_s
et[,feature_name], "%y"))
          data_set[,paste0(feature_name,'_LongYear')] <- as.integer(format(data_se
t[,feature_name], "%Y"))
          data_set[,paste0(feature_name,'_Day')] <- as.integer(format(data_set[,fe
ature_name], "%d"))

          # week day number requires first pulling the weekday label, creating the
7 week day levels, and casting to integer
          data_set[,paste0(feature_name,'_WeekDayNumber')] <- as.factor(weekdays(d
ata_set[,feature_name]))
          levels(data_set[,paste0(feature_name,'_WeekDayNumber')]) <- list(Monday=
1, Tuesday=2, Wednesday=3, Thursday=4, Friday=5, Saturday=6, Sunday=7)
          data_set[,paste0(feature_name,'_WeekDayNumber')] <- as.integer(data_set
[,paste0(feature_name,'_WeekDayNumber')])

          data_set[,paste0(feature_name,'_IsWeekend')] <- as.numeric(grepl("Saturd
ay|Sunday", weekdays(data_set[,feature_name])))
          data_set[,paste0(feature_name,'_YearDayCount')] <- yday(data_set[,featur
e_name])
          data_set[,paste0(feature_name,'_Quarter')] <- lubridate::quarter(data_se
t[,feature_name], with_year = FALSE)
          data_set[,paste0(feature_name,'_Quarter')] <- lubridate::quarter(data_se
t[,feature_name], with_year = TRUE)
          if (remove_original_date)
               data_set[, feature_name] <- NULL
     }
     return(data_set)
}
```

Let's test this on our old data set that includes a date field:

```
mix_dataset <- data.frame(
                id=c(10,20,30,40,50),
                gender=c('male','female','female','male','female'),
                some_date=c('2012-01-12','2012-01-12','2012-12-01','2012-05-30','2
013-12-12'),
                value=c(12.34, 32.2, 24.3, 83.1, 8.32),
                outcome=c(1,1,0,0,0))

library(readr)
write_csv(mix_dataset, 'mix_dataset.csv')

mix_dataset <- read_csv('mix_dataset.csv')

mix_dataset <- Feature_Engineer_Dates(mix_dataset)
```

```
## Loading required package: lubridate
```

```
head(mix_dataset)
```

```
##   id gender value outcome some_date_DateInt some_date_Month
## 1 10   male 12.34       1             15351               1
## 2 20 female 32.20       1             15351               1
## 3 30 female 24.30       0             15675              12
## 4 40   male 83.10       0             15490               5
## 5 50 female  8.32       0             16051              12
##   some_date_ShortYear some_date_LongYear some_date_Day
## 1                  12               2012            12
## 2                  12               2012            12
## 3                  12               2012             1
## 4                  12               2012            30
## 5                  13               2013            12
##   some_date_WeekDayNumber some_date_IsWeekend some_date_YearDayCount
## 1                       4                   0                      12
## 2                       4                   0                      12
## 3                       6                   1                     336
## 4                       3                   0                     151
## 5                       4                   0                     346
##   some_date_Quarter
## 1            2012.1
## 2            2012.1
## 3            2012.4
## 4            2012.2
## 5            2013.4
```