Detecting Anomaly with H2O

Deep Learning building an Autoencoder model

Victoria Bolotin

8/10/2019

H2O's Anomaly function

First, we import prostate data set, preprocess it, print a summary.

Second, we runs Deep Learning building an Autoencoder model.

Third, we calculate the reconstruction error

Libraries

```
library(dplyr)
library(recipes)
library(tidyquant)
library(h2o)
```

Data

```
data_tbl <- read.csv("../../h2o_projects/Data/prostate.csv")
data_tbl$X <- NULL
glimpse(data_tbl)
```

```
## Observations: 380
## Variables: 9
## $ ID      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,...
## $ CAPSULE <int> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, ...
## $ AGE     <int> 65, 72, 70, 76, 69, 71, 68, 61, 69, 68, 68, 72, 72, 65...
## $ RACE    <int> 1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 1, 1, 1, 1, 2, 1, ...
## $ DPROS   <int> 2, 3, 1, 2, 1, 3, 4, 4, 1, 1, 4, 2, 4, 4, 1, 2, 1, 2, ...
## $ DCAPS   <int> 1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1, ...
## $ PSA     <dbl> 1.4, 6.7, 4.9, 51.2, 12.3, 3.3, 31.9, 66.7, 3.9, 13.0,...
## $ VOL     <dbl> 0.0, 0.0, 0.0, 20.0, 55.9, 0.0, 0.0, 27.2, 24.0, 0.0, ...
## $ GLEASON <int> 6, 7, 6, 7, 6, 8, 7, 7, 7, 6, 7, 7, 9, 7, 5, 5, 5, 5, ...
```

Preprocessing with recipes

For H2O We need our data only in readable format (numeric and factor), NOT necessary dummy vars, NOT necessary transformations such as YeoJonson and other most preprocessing steps. We just remove zero variance columns, and turning "DPROS", "CAPSULE", "GLEASON" columns to factor.

```
recipe_obj <- recipe(DCAPS ~ ., data = data_tbl) %>%
    step_zv(all_predictors()) %>%
    step_num2factor(CAPSULE, DPROS, GLEASON) %>%
    prep()

recipe_obj
```

```
## Data Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Training data contained 380 data points and no missing data.
##
## Operations:
##
## Zero variance filter removed no terms [trained]
## Factor variables from CAPSULE, DPROS, GLEASON [trained]
```

```
data_tbl <- bake(recipe_obj, new_data = data_tbl)
glimpse(data_tbl)
```

```
## Observations: 380
## Variables: 9
## $ ID      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,...
## $ CAPSULE <fct> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, ...
## $ AGE     <int> 65, 72, 70, 76, 69, 71, 68, 61, 69, 68, 68, 72, 72, 65...
## $ RACE    <int> 1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 1, 1, 1, 1, 2, 1, ...
## $ DPROS   <fct> 2, 3, 1, 2, 1, 3, 4, 4, 1, 1, 4, 2, 4, 4, 1, 2, 1, 2, ...
## $ DCAPS   <int> 1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1, ...
## $ PSA     <dbl> 1.4, 6.7, 4.9, 51.2, 12.3, 3.3, 31.9, 66.7, 3.9, 13.0,...
## $ VOL     <dbl> 0.0, 0.0, 0.0, 20.0, 55.9, 0.0, 0.0, 27.2, 24.0, 0.0, ...
## $ GLEASON <fct> 6, 7, 6, 7, 6, 8, 7, 7, 7, 6, 7, 7, 9, 7, 5, 5, 5, 5, ...
```

```
summary(data_tbl)
```

```
##        ID         CAPSULE      AGE             RACE       DPROS
##  Min.   :  1.00   0:227   Min.   :43.00   Min.   :0.000   1: 99
##  1st Qu.: 95.75   1:153   1st Qu.:62.00   1st Qu.:1.000   2:132
##  Median :190.50           Median :67.00   Median :1.000   3: 96
##  Mean   :190.50           Mean   :66.04   Mean   :1.087   4: 53
##  3rd Qu.:285.25           3rd Qu.:71.00   3rd Qu.:1.000
##  Max.   :380.00           Max.   :79.00   Max.   :2.000
##
##      DCAPS            PSA               VOL         GLEASON
##  Min.   :1.000   Min.   :  0.30   Min.   : 0.00   0:  2
##  1st Qu.:1.000   1st Qu.:  5.00   1st Qu.: 0.00   4:  1
##  Median :1.000   Median :  8.75   Median :14.25   5: 67
##  Mean   :1.108   Mean   : 15.41   Mean   :15.81   6:139
```

```
##  3rd Qu.:1.000   3rd Qu.: 17.12   3rd Qu.:26.45   7:128
##  Max.   :2.000   Max.   :139.70   Max.   :97.60   8: 30
##                                                   9: 13
```

Modeling

```
h2o.init()
```

```
##
## H2O is not running yet, starting it now...
##
## Note:  In case of errors look at the following log files:
##     /var/folders/b2/g9_8fkmn6s173cpc9qxj7z840000gn/T//RtmpNP01Pg/h2o_vibolotin_started_from_r.out
##     /var/folders/b2/g9_8fkmn6s173cpc9qxj7z840000gn/T//RtmpNP01Pg/h2o_vibolotin_started_from_r.err
##
##
## Starting H2O JVM and connecting: .. Connection successful!
##
## R is connected to the H2O cluster:
##     H2O cluster uptime:         2 seconds 261 milliseconds
##     H2O cluster timezone:       America/Los_Angeles
##     H2O data parsing timezone:  UTC
##     H2O cluster version:        3.26.0.2
##     H2O cluster version age:    15 days
##     H2O cluster name:           H2O_started_from_R_vibolotin_bam383
##     H2O cluster total nodes:    1
##     H2O cluster total memory:   3.56 GB
##     H2O cluster total cores:    8
##     H2O cluster allowed cores:  8
##     H2O cluster healthy:        TRUE
##     H2O Connection ip:          localhost
##     H2O Connection port:        54321
##     H2O Connection proxy:       NA
##     H2O Internal Security:      FALSE
##     H2O API Extensions:         Amazon S3, XGBoost, Algos, AutoML, Core V3, Core V4
##     R Version:                  R version 3.6.1 (2019-07-05)
```

```
y = "DCAPS"
#x = setdiff(names(train_h2o),y)
x= c("AGE", "RACE", "DPROS","PSA", "VOL", "CAPSULE", "GLEASON")
```

Build autoencoder model & save

```
dl_autoencoder %>%
# save the model
h2o.saveModel(path = "../00_Models/")
```

Load model from a file

```
dl_autoencoder <- h2o.loadModel("../00_Models/autoencoders")
```

Learn more about the model

```
slotNames(dl_autoencoder)
```

```
## [1] "model_id"      "algorithm"     "parameters"    "allparameters"
## [5] "have_pojo"     "have_mojo"     "model"
```

```
dl_autoencoder@model
```

Detect outliers on the prostate dataset

```
anomalies <-  h2o.anomaly(object = dl_autoencoder, train_h2o)

anomalies_tbl <- as.data.frame(anomalies)
head(anomalies_tbl)
```

```
##   Reconstruction.MSE
## 1          0.1519534
## 2          0.6040772
## 3          0.1522961
## 4          2.0240759
## 5          0.1599782
## 6          0.1527333
```

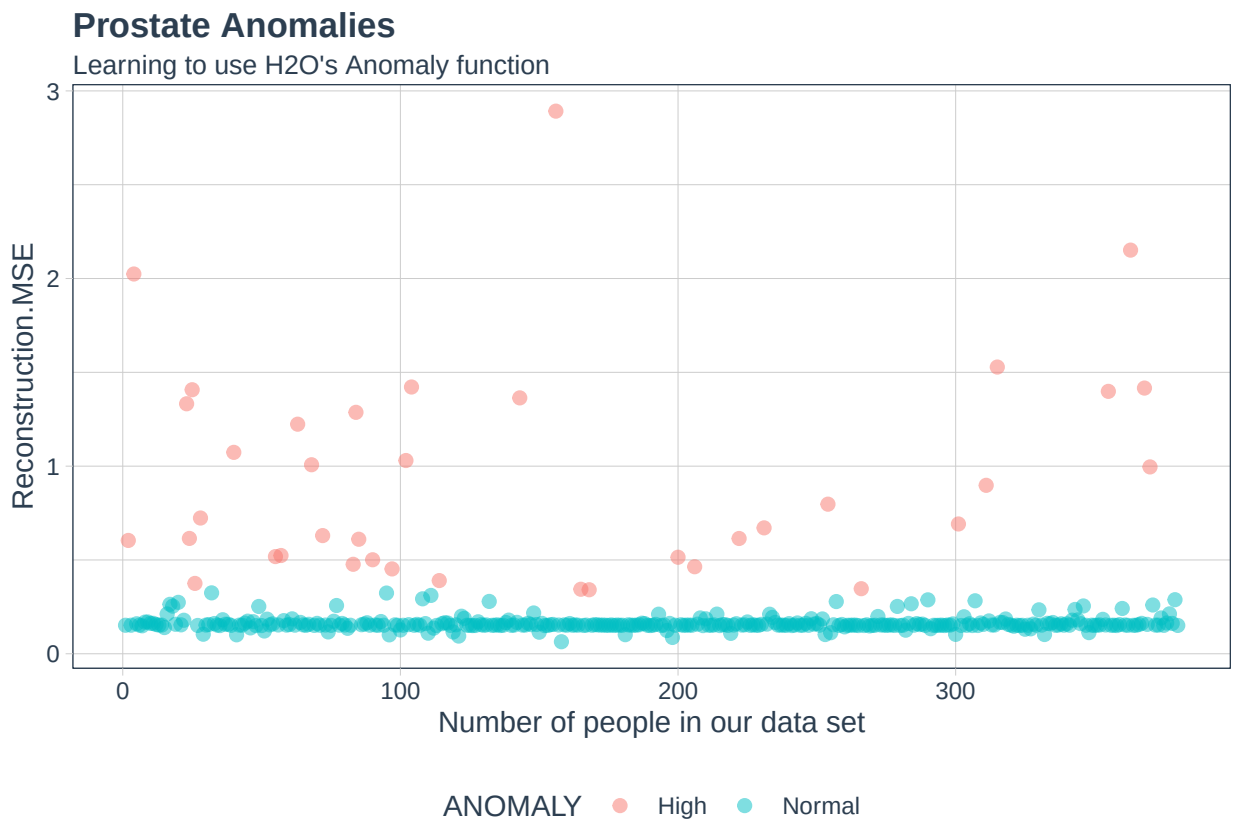
Plot

Plot the reconstruction error and color anomalies for error in the 90th percentile

```
data_transformed_tbl<-
data_tbl %>% cbind(anomalies_tbl) %>%
    mutate(Reconstruction_MSE_binned = ntile(Reconstruction.MSE, 3)) %>%
    mutate(ANOMALY= case_when(
        Reconstruction.MSE > quantile(Reconstruction.MSE, 0.9) ~"High",
        TRUE ~"Normal"
    )) %>%
    select(ID, AGE, Reconstruction.MSE, ANOMALY
)
```

```
#plot
data_transformed_tbl %>%
    ggplot(aes(x= ID, y = Reconstruction.MSE, color = ANOMALY)) +
    geom_point(alpha= 0.5, size = 2)+
```

```
    labs(
        title = "Prostate Anomalies",
        subtitle = "Learning to use H2O's Anomaly function",
        x = "Number of people in our data set",
        y = "Reconstruction.MSE"
    )+
    theme_tq()+
    theme(
        plot.title = element_text(face = "bold") ,
        plot.caption = element_text(face = "bold.italic")
    )
```



Detach h2o

```
#detach("package:h2o", unload = TRUE)
```