Labs\_1

Тепсикоев С. А.

2023-03-01

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 1.1.0 ──

## ✔ broom 1.0.4 ✔ recipes 1.0.6  
## ✔ dials 1.2.0 ✔ rsample 1.1.1  
## ✔ dplyr 1.1.2 ✔ tibble 3.2.1  
## ✔ ggplot2 3.4.2 ✔ tidyr 1.3.0  
## ✔ infer 1.0.4 ✔ tune 1.1.1  
## ✔ modeldata 1.1.0 ✔ workflows 1.1.3  
## ✔ parsnip 1.1.0 ✔ workflowsets 1.0.1  
## ✔ purrr 1.0.1 ✔ yardstick 1.2.0

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## ✖ purrr::discard() masks scales::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ recipes::step() masks stats::step()  
## • Learn how to get started at https://www.tidymodels.org/start/

library(arrow)

## Some features are not enabled in this build of Arrow. Run `arrow\_info()` for more information.

##   
## Attaching package: 'arrow'

## The following object is masked from 'package:utils':  
##   
## timestamp

library(dplyr)  
library(ggplot2)

Задание 1: Надите утечку данных из Вашей сети Важнейшие документы с результатми нашей исследовательской деятельности в области создания вакцин скачиваются в виде больших заархивированных дампов. Один из хостов в нашей сети используется для пересылки этой информации – он пересылает гораздо больше информации на внешние ресурсы в Интернете, чем остальные компьютеры нашей сети. Определите его IP-адрес.

1. Импортируем датасет

df\_data <- arrow::read\_csv\_arrow("traffic\_security.csv")

1. Дадим имена признакам

colnames(df\_data) <- c('timestamp','src','dst','port','bytes')  
head(df\_data,3)

## # A tibble: 3 × 5  
## timestamp src dst port bytes  
## <int64> <chr> <chr> <int> <int>  
## 1 1578326400005 16.79.101.100 12.48.65.39 92 11895  
## 2 1578326400007 18.43.118.103 14.51.30.86 27 898  
## 3 1578326400011 15.71.108.118 14.50.119.33 57 7496

1. Очистим датасет, оставив в src ip-адреса, только нашего предприятия

knitr::opts\_chunk$set(  
 df\_data <- df\_data[df\_data$src > 11 & df\_data$src < 15 & df\_data$dst < 11 | df\_data$dst > 15, ]  
)

1. Найдём ip-адрес и максимальное число передаваемых байтов(ответ кто злоумышленник в организации)

knitr::opts\_chunk$set(  
 found\_ip1 <- df\_data %>%  
 group\_by(src) %>%  
 summarise(bytes = mean(bytes)),  
 found\_ip1 <- found\_ip1[which.max(found\_ip1$bytes),],  
 print(found\_ip1)   
)

## # A tibble: 1 × 2  
## src bytes  
## <chr> <dbl>  
## 1 13.37.84.125 192390.

Ответ: 13.37.84.125

Задание 2: Надите утечку данных 2 Другой атакующий установил автоматическую задачу в системном планировщике cron для экспорта содержимого внутренней wiki системы. Эта система генерирует большое количество траффика в нерабочие часы, больше чем остальные хосты. Определите IP этой системы. Известно, что ее IP адрес отличается от нарушителя из предыдущей задачи

## Удаление IP из задания 1

knitr::opts\_chunk$set(  
 df\_data <- df\_data[!(df\_data$src=="13.37.84.125"),]  
)  
head(df\_data,3)

## # A tibble: 3 × 5  
## timestamp src dst port bytes  
## <int64> <chr> <chr> <int> <int>  
## 1 1578326400012 14.33.30.103 15.24.31.23 115 20979  
## 2 1578326400018 12.46.104.126 16.25.76.33 123 1500  
## 3 1578326400021 12.43.98.93 18.85.31.68 79 979

## Извлечение часов

knitr::opts\_chunk$set(  
 df\_data$timestamp <- as.POSIXct(df\_data$timestamp/1000, origin = "1970-01-01", tz = "GMT")  
)  
head(df\_data,3)

## # A tibble: 3 × 5  
## timestamp src dst port bytes  
## <dttm> <chr> <chr> <int> <int>  
## 1 2020-01-06 16:00:00 14.33.30.103 15.24.31.23 115 20979  
## 2 2020-01-06 16:00:00 12.46.104.126 16.25.76.33 123 1500  
## 3 2020-01-06 16:00:00 12.43.98.93 18.85.31.68 79 979

## Удаление NA

knitr::opts\_chunk$set(  
 df\_data <- na.omit(df\_data)  
)

## Извлекаем часы и байты

knitr::opts\_chunk$set(  
 df\_data\_hours\_bytes <- tibble(hours = format(as.POSIXct(df\_data$timestamp), format = "%H%M%S"),  
 bytes = df\_data$bytes)  
)  
head(df\_data\_hours\_bytes,3)

## # A tibble: 3 × 2  
## hours bytes  
## <chr> <int>  
## 1 160000 20979  
## 2 160000 1500  
## 3 160000 979

## Разбиваем время на часы минуты и секунды

library("stringr")

##   
## Attaching package: 'stringr'

## The following object is masked from 'package:recipes':  
##   
## fixed

knitr::opts\_chunk$set(  
 df\_data$hour <- with (df\_data,format(as.POSIXct(df\_data$timestamp), format = "%H")),  
 df\_data$minutes <- with (df\_data,format(as.POSIXct(df\_data$timestamp), format = "%M")),  
 df\_data$sec <- with (df\_data,format(as.POSIXct(df\_data$timestamp), format = "%S"))  
)  
head(df\_data,3)

## # A tibble: 3 × 8  
## timestamp src dst port bytes hour minutes sec   
## <dttm> <chr> <chr> <int> <int> <chr> <chr> <chr>  
## 1 2020-01-06 16:00:00 14.33.30.103 15.24.31.23 115 20979 16 00 00   
## 2 2020-01-06 16:00:00 12.46.104.126 16.25.76.33 123 1500 16 00 00   
## 3 2020-01-06 16:00:00 12.43.98.93 18.85.31.68 79 979 16 00 00

## Группируем по времени

knitr::opts\_chunk$set(  
df\_data\_hours\_bytes <- df\_data\_hours\_bytes %>%  
 group\_by(hours) %>%  
 summarise(bytes = mean(bytes))  
)  
head(df\_data\_hours\_bytes,3)

## # A tibble: 3 × 2  
## hours bytes  
## <chr> <dbl>  
## 1 000000 24886.  
## 2 000001 14100.  
## 3 000002 20128.

## Кластерезуем

knitr::opts\_chunk$set(  
 kclust <- kmeans(na.omit(df\_data\_hours\_bytes), centers = 2)  
)

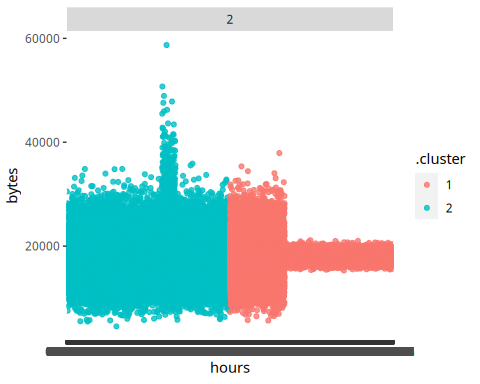
## Эксперимент с кол-вом кластеров

knitr::opts\_chunk$set(  
 kclusts <-  
 tibble(k = 2) %>%  
 mutate(  
 kclust = map(k, ~kmeans(df\_data\_hours\_bytes, .x)),  
 tidied = map(kclust, tidy),  
 glanced = map(kclust, glance),  
 augmented = map(kclust, augment, df\_data\_hours\_bytes)  
 )  
)

knitr::opts\_chunk$set(  
 clusters <-  
 kclusts %>%  
 unnest(cols = c(tidied)),  
   
 assigments <-  
 kclusts %>%  
 unnest(cols = c(augmented)),  
   
 clusterings <-  
 kclusts %>%  
 unnest(cols = c(glanced))  
)

### Графический вид

knitr::opts\_chunk$set(  
plot1 <-  
 ggplot(assigments, aes(x = hours, y = bytes)) +  
 geom\_point(aes(color = .cluster), alpha = 0.8) +  
 facet\_wrap(~ k)  
)  
plot1

 ### Поиск IP в 07:20:30

df\_data\_hours\_bytes <- df\_data[df\_data$hour=="07" & df\_data$minutes=="20" & df\_data$sec=="30",]  
  
found\_ip2 <- df\_data\_hours\_bytes %>%  
 group\_by(src) %>%  
 summarise(bytes = mean(bytes))  
  
found\_ip2 <- found\_ip2[which.max(found\_ip2$bytes),]  
  
print(found\_ip2)

## # A tibble: 1 × 2  
## src bytes  
## <chr> <dbl>  
## 1 12.55.77.96 656178.

Ответ: 12.55.77.96

Задание 3: Надите утечку данных 3 Еще один нарушитель собирает содержимое электронной почты и отправляет в Интернет используя порт, который обычно используется для другого типа трафика. Атакующий пересылает большое количество информации используя этот порт, которое нехарактерно для других хостов, использующих этот номер порта. Определите IP этой системы. Известно, что ее IP адрес отличается от нарушителей из предыдущих задач.

## Удаление IP из задания 2

knitr::opts\_chunk$set(  
 df\_data <- df\_data[!(df\_data$src=="12.55.77.96"),]  
)  
head(df\_data,3)

## # A tibble: 3 × 8  
## timestamp src dst port bytes hour minutes sec   
## <dttm> <chr> <chr> <int> <int> <chr> <chr> <chr>  
## 1 2020-01-06 16:00:00 14.33.30.103 15.24.31.23 115 20979 16 00 00   
## 2 2020-01-06 16:00:00 12.46.104.126 16.25.76.33 123 1500 16 00 00   
## 3 2020-01-06 16:00:00 12.43.98.93 18.85.31.68 79 979 16 00 00

## Извлекаем часы и байты

knitr::opts\_chunk$set(  
 df\_data\_ports\_bytes <- tibble(src = df\_data$src,  
 port = df\_data$port,  
 bytes = df\_data$bytes)  
   
)  
head(df\_data\_ports\_bytes,3)

## # A tibble: 3 × 3  
## src port bytes  
## <chr> <int> <int>  
## 1 14.33.30.103 115 20979  
## 2 12.46.104.126 123 1500  
## 3 12.43.98.93 79 979

## Группируем по портам

knitr::opts\_chunk$set(  
df\_data\_ports\_bytes <- df\_data\_ports\_bytes %>%  
 group\_by(src, port) %>%  
 summarise(bytes = mean(bytes)),  
  
df\_data\_ports\_bytes <- tibble(port = df\_data\_ports\_bytes$port,  
 bytes = df\_data\_ports\_bytes$bytes)  
)

## `summarise()` has grouped output by 'src'. You can override using the `.groups`  
## argument.

head(df\_data\_ports\_bytes,3)

## # A tibble: 3 × 2  
## port bytes  
## <int> <dbl>  
## 1 22 995.  
## 2 23 1000.  
## 3 25 42

## Кластерезуем

knitr::opts\_chunk$set(  
 kclust <- kmeans(na.omit(df\_data\_ports\_bytes), centers = 2)  
)

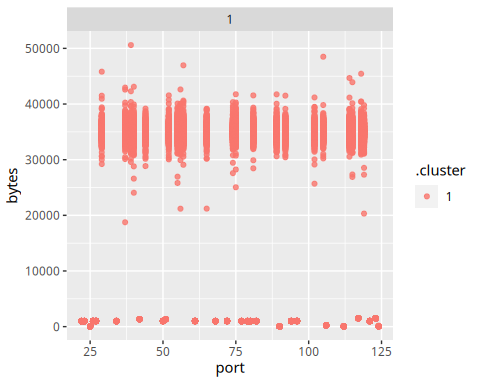
## Эксперимент с кол-вом кластеров

knitr::opts\_chunk$set(  
 kclusts <-  
 tibble(k = 1) %>%  
 mutate(  
 kclust = map(k, ~kmeans(df\_data\_ports\_bytes, .x)),  
 tidied = map(kclust, tidy),  
 glanced = map(kclust, glance),  
 augmented = map(kclust, augment, df\_data\_ports\_bytes)  
 )  
)

knitr::opts\_chunk$set(  
 clusters <-  
 kclusts %>%  
 unnest(cols = c(tidied)),  
   
 assigments <-  
 kclusts %>%  
 unnest(cols = c(augmented)),  
   
 clusterings <-  
 kclusts %>%  
 unnest(cols = c(glanced))  
)

### Графический вид

knitr::opts\_chunk$set(  
plot1 <-  
 ggplot(assigments, aes(x = port, y = bytes)) +  
 geom\_point(aes(color = .cluster), alpha = 0.8) +  
 facet\_wrap(~ k)  
)  
plot1

 ### Поиск IP в 07:20:30

df\_data\_ports\_bytes <- tibble(src = df\_data$src,  
 port = df\_data$port,  
 bytes = df\_data$bytes)  
  
df\_data\_ports\_bytes <- df\_data\_ports\_bytes %>%  
 group\_by(src, port) %>%  
 summarise(bytes = mean(bytes))

## `summarise()` has grouped output by 'src'. You can override using the `.groups`  
## argument.

df\_data\_ports\_bytes <- df\_data\_ports\_bytes[df\_data\_ports\_bytes$port=="39",]  
  
found\_ip3 <- df\_data\_ports\_bytes %>%  
 group\_by(src, port) %>%  
 summarise(bytes = mean(bytes))

## `summarise()` has grouped output by 'src'. You can override using the `.groups`  
## argument.

found\_ip3 <- found\_ip3[which.max(found\_ip3$bytes),]  
  
print(found\_ip3)

## # A tibble: 1 × 3  
## # Groups: src [1]  
## src port bytes  
## <chr> <int> <dbl>  
## 1 13.36.102.77 39 50597.

Ответ: 13.36.102.77