Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



**Лабораторна робота №5**

*З дисципліни:*

**“** **Видуботок великих даних”**

**Виконала:**

Студентка групи КН-318

Яворська М.О.

**Перевірила:**

професор Шаховська Н.Б.

Львів-2019

**Хід роботи**

**1) Querying MySQL**

• The following analysis requires the RMySQL, Hmisc, and plyr libraries, along with dependencies.

install.packages("RMySQL")

install.packages("Hmisc")

install.packages("plyr")

install.packages("RSQLite")

library(RMySQL)

library(Hmisc)

library(plyr)

library(RSQLite)

• The following demonstrates a simple select query using RMySQL along with some basic R-side refinement of the retrieved dataset.

con <- dbConnect(RSQLite::SQLite(), ":memory:")

dbWriteTable(con, "mtcars", mtcars)

rs <- dbSendQuery(con, "SELECT \* FROM mtcars WHERE cyl = 4;")

dbFetch(rs)

dbClearResult(rs)

dbDisconnect(con)

• First we calculate the total drat

summa <- dbSendQuery(con, "SELECT SUM(drat) FROM mtcars ;")

dbFetch(summa)



• Find mean for drat

average <- dbSendQuery(con, "SELECT AVG(drat) FROM mtcars ;")

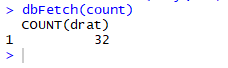
dbFetch(average)



• Find count of rows using SQL-commands

count <- dbSendQuery(con, "SELECT COUNT(drat) FROM mtcars ;")

dbFetch(count)



install.packages("RMySQL")

install.packages("Hmisc")

install.packages("plyr")

install.packages("RSQLite")

install.packages("hflights")

library(hflights)

library(RMySQL)

library(Hmisc)

library(plyr)

library(RSQLite)

library("DBI")

library(ggplot2)

con <- dbConnect(RSQLite::SQLite(), ":memory:")

dbWriteTable(con, "mtcars", mtcars)

rs <- dbSendQuery(con, "SELECT \* FROM mtcars WHERE cyl = 4;")

dbFetch(rs)

dbClearResult(rs)

summa <- dbSendQuery(con, "SELECT SUM(drat) FROM mtcars ;")

dbFetch(summa)

average <- dbSendQuery(con, "SELECT AVG(drat) FROM mtcars ;")

dbFetch(average)

count <- dbSendQuery(con, "SELECT COUNT(drat) FROM mtcars ;")

dbFetch(count)

str(hflights)

library(devtools)

install\_github(repo = "mongosoup/rmongodb")

library(rJava)

library(rmongodb)

library(dplyr)

library(magrittr)

library(dplyr)

library(RMongo)

#install.packages("pipeR")

library(pipeR)

conn\_mongo <- mongoDbConnect("flightsData")

airlines\_mongo <- dbGetQuery(conn\_mongo, "airlines", "",0,16)

airports\_mongo <- dbGetQuery(conn\_mongo, "airports", "",0,1397)

flights\_mongo <- dbGetQuery(conn\_mongo, "flights", "",0,336776)

xid<-dbGetQuery(conn\_mongo, 'airports', '{"AIRPORT": "Adak Airport"}')

xid$X\_id

distance=flights\_mongo[15]

carrier=flights\_mongo[20]

airtime=flights\_mongo[11]

flights\_mongo %>%

filter(CANCELLED== 0) %>%

mutate(speed = distance/AIR\_TIME)

group\_by(carrier) %>%

summarize(mean\_speed = mean((DISTANCE/AIR\_TIME)\*60,na.rm = TRUE)) %>%

arrange(desc(mean\_speed)) %>%

with(barplot(mean\_speed,names.arg = carrier,

main = "Average flight speed", xlab="Airline", ylab="Miles per hour"))

flights\_mongo %>%

filter(CANCELLED== 0) %>%

mutate(speed = DISTANCE / AIR\_TIME) %>%

group\_by(speed) %>%

summarize(mean\_speed = mean((DISTANCE/AIR\_TIME)\*60,na.rm = TRUE)) %>%

arrange(desc(mean\_speed)) %>%

with(barplot(mean\_speed,names.arg = carrier,

main = "Average flight speed", xlab="Airline", ylab="Miles per hour"))