ITMAGINATION
Imagine it. Done.

Spark+R – wydajne obliczenia w chmurze

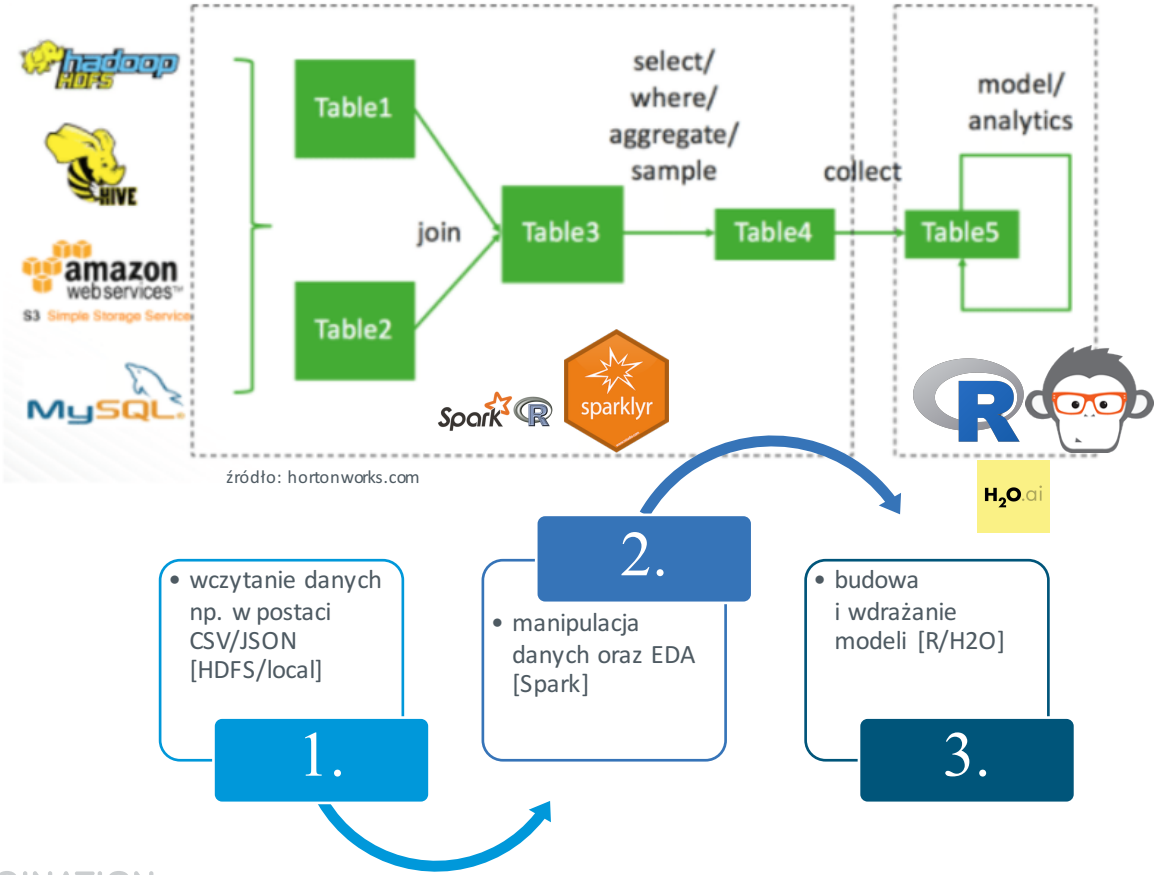
Wiktor Zdzienicki
Data Science

www.itmagination.com

Facebook: **R Memes For Statistical Fiends**

Why Spark?

Data Science Workflow



Zastosowania:

- Big Data, Small Learning – przetworzenie dużej ilości danych → agregacja lub próbkowanie → budowa modelu
- Partition - Aggregate – zrównoleglenie funkcji
- Large Scale Machine Learning – trenowanie algorytmów na wielkich zbiorach danych

Why Spark?

Porównanie narzędzi

VS.

	SparkR	sparklyr
twórcy	UC Berkeley	RStudio
aktualizacja	powiązana z releasami Sparka	niezależny pakiet
składnia	predefiniowane ekwiwalenty, tworzenie UDF	dplyr / filozofia tidyverse
machine learning	MLlib	dostosowany MLlib, integracja z H2O (rsparkling)

Use-case – model scoringowy

Nawiązanie połączenia i wczytanie danych


```r
library(SparkR)

sparkR.session()

# local
credit_loc <- read_csv("credit.csv")
credit <- as.DataFrame(read_csv("credit.csv"))

# file
credit <- read.df("credit.csv", "csv")

# Hive
credit <- sql("FROM credit SELECT *")
```


```r
library(sparklyr)
library(dplyr)

sc <- spark_connect()

# local
credit_loc <- read_csv("credit.csv")
credit <- copy_to(sc, credit_loc, overwrite = TRUE)

# file
spark_read_csv(sc, "credit", "credit.csv")

# Hive
credit <- tbl(sc, "credit")
```

Use-case – model scoringowy

Badanie i transformacja danych


```r
# sample rows
head(credit)

# specific
filter(credit, credit$Purpose == 3)

# distribution
agg(groupBy(credit, credit$Type_of_apartment), count = n(credit$Type_of_ap
artment))

# subset
credit_subset <- select(credit, c("Creditability", "Account_Balance"))
credit_subset <- filter(credit_subset, credit_subset$Account_Balance != 4)
credit_subset$duration_yr <- Duration_of_credit_month_ / 12
```


```r
# sample rows
credit

# specific
filter(credit, Purpose == 3)

# distribution
credit %>%
    count(Type_of_apartment)

# subset
credit_subset <- credit %>%
    select(Creditability, Account_Balance) %>%
    filter(Account_Balance != 4) %>%
    mutate(duration_yr = Duration_of_credit_month_ /12)
```

Use-case – model scoringowy

Podział zbioru i modelowanie


```
# train & test
credit_sets <- randomSplit(credit,
c(7, 3), 2)
credit_train <- credit_sets[[1]]
credit_test <- credit_sets[[2]]

# logit
model <- spark.logit(credit_train,
Creditability ~ .)
```


```
# train & test
credit_sets <- credit %>%
    sdf_partition(training = 0.7, test = 0.3)

# logit
model <- credit_sets$training %>%
    ml_logistic_regression(Creditability ~ .)

# WŁĄCZENIE H2O #
library(rsparkling)
library(h2o)

train <- as_h2o_frame(sc, credit_sets$training)
test <- as_h2o_frame(sc, credit_sets$test)

model <- h2o.glm()

# # # # # # # # #
```

Use-case – model scoringowy

Predykcje, performance i zapis modelu


```r
# zapis modelu
write.ml(model, sciezka)


# predykcje
predictions <- predict(model, credit_test)
collect(predictions)

# performance
table(predictions$Creditability, predictions)
```


```r
# zapis modelu
ml_save(model, sciezka)


# predykcje
predictions <- sdf_predict(model, credit_sets$test) %>%
    collect

# performance
table(predictions$Creditability, predictions)
```

ITMAGINATION
Imagine it. Done.

Dziękuję za uwagę!

Wiktor Zdzienicki
Data Science
wiktor.zdzienicki@itmagination.com

www.itmagination.com