WLOG SOLUTIONS

R+H2O - idealny tandem do analityki predykcyjnej?

Wit Jakuczun, WLOG Solutions

Wstęp

WLOG SOLUTIONS

Plan prezentacji

- Kim jestem?
- Co to jest H2O?
- Dlaczego R+H2O jest dobrym tandemem do analityki predykcyjne?
- Przykłady użycia

Kim jestem?

- Jestem:
  - Właścicielem firmy WLOG Solutions
  - Absolwentem wydziału MIM UW, doktorat z IPI PAN.
- Znam się na:
  - Analizie i syntezie potrzeb klienta.
  - Wdrażaniu analityki w organizacji (np. w oparciu o GNU R).
  - Tworzeniu i implementacja modeli optymalizacyjnych oraz analitycznych.
- Dane kontaktowe:
  - email: w.jakuczun@wlogsolutions.com
  - WWW: www.wlogsolutions.com

Z czego korzystałem przygotowywując się do warsztatu?

- GNU R
  - Używam od 2006 roku praktycznie codziennie.
  - Pierwsze produkcyjne wdrożenie w Mazowieckiej Spółce Gazowniczej w roku 2006.
  - Korzystałem z wersji 3.2.3
- Pakiety analityczne:
  - h2o
  - h2oEnsemble
- Prezentacja powstała w oparciu o:
  - knitr
  - pandoc
  - MikTeX
  - checkpoint

WLOG SOLUTIONS

Co to jest H2O?

H2O to..

Framework do rozproszonego przechowywania danych tabelarycznych "w pamięci" w postaci "klucz-wartość" oraz rozproszonych obliczeń na tych danych.

H2O - spojrzenie ogólne

- Rozproszony **in-memory** storage
  - **key-value** storage
  - Kompresja danych
  - Skalowalny i wydajny
- Wydajna implementacja algorytmów
  - Predykcyjne: Random Forest, GBM, GLM, Deep Learning
  - Analiza skupień: k-means
  - Redukcja wymiaru: PCA, Generalized Low Rank Models
  - Optymalizacja: algorytmy numeryczne (np. BFGS)
- Obliczenia rozproszone
  - Kluster H2O
  - Spark - `http://www.h2o.ai/product/sparkling-water/`

H2O - namiary na informacje

- Strona główna H2O - `http://www.h2o.ai/`
- Prezentacje H2O World 2015 - `http://h2oworld.h2o.ai/`
- Repozytorium GitHub - `https://github.com/h2oai`
- Dokumentacja - `http://www.h2o.ai/docs/`

Dlaczego R+H2O jest dobrym tandemem do analityki predykcyjnej?

Co mnie urzekło?

Test 15 minut: *jeśli w ciągu 15 minut nie będę potrafił zbudować modelu dla danych* iris *to odkładam na półkę.*

Test zaliczony pozytywnie!

Co otrzymujemy z półki?

WLOG SOLUTIONS

- **Bardzo dobra dokumentacja!**
- **Bardzo dobre wsparcie!**
- **Działające** API do R (także Python, Scala, Java)
  - pakiet wspierający h2oEnsemble - `http://learn.h2o.ai/content/tutorials/ensembles-stacking/index.html`
- Interfejs *Flow* - `http://localhost:54321`
- Pełen wykorzystanie mocy obliczeniowej: rdzenie, klaster
- Wsparcie przy doborze parametrów modelu (*grid i random search*)
- Eksport/import modeli w formie binarnej oraz **Plain Old Java**!
- Import danych z plików lokalnych, HDFS, URL
- Integracja ze SPARK - tzw. **Sparkling Water** (wymaga wysiłku)

Model pracy z R+H2O

1. Przygotowanie danych (R)
2. Budowa modelu (H2O)
3. Analiza wyników (R)
4. Raportowanie (R)

Korzyści

- Nie szukam pakietów z różnymi algorytmami
- Nie tracę czasu na dostosowanie się do różnych API pakietów
- Nie tracę czasu na obróbkę techniczną danych (np. braki danych)
- Wyniki modeli mam szybko (na ile pozwalają dostępne zasoby)
- Nie tracę czasu na wyliczanie metryk modeli: AUC, ROC, Lift, F1, MSE, etc.
- **Mam więcej czasu na pracę analityczną i komunikację z klientem!**

Czy H2O nadaje się do środowiska Enterprise?

Pytanie jest źle postawione! Poprawne pytanie brzmi ***Dlaczego korzystać z innych rozwiązań do budowania modeli predykcyjnych?***

Przykłady praktyczne

Opis zadania

Predykcja czy dane zapytanie HTTP jest reklamą.

Przykłady użycia

1. Inicjalizacja h2o
2. Import danych
3. Budowa modelu
4. Dobór parametrów modelu
5. Analiza jakości modelu
6. Eksport modelu
7. Import modelu i scoring

Inicjalizacja h2o

```
library(h2o)

# Można też z linii komend...
# Dostęp do flow z poziomu przeglądarki:
#     http://localhost:54321/flow/index.html
h2o_local <- h2o.init(startH2O = TRUE,
                      nthreads = -1,
                      max_mem_size = "5g")
```

Import danych

```
h2o_train <- h2o.importFile(path = "data/adv_train.csv",
                            destination_frame = "adv_train",
                            header = TRUE,
                            sep = ";",
                            parse = TRUE)

h2o_test <- h2o.importFile(path = "data/adv_test.csv",
                           destination_frame = "adv_test",
                           header = TRUE,
                           sep = ";",
                           parse = TRUE)
```

Budowa modelu

```
rf_model <- h2o.randomForest(training_frame = h2o_train,
                             model_id = "rf_model",
                             x = colnames(h2o_train),
                             y = "advertisement",
                             nfolds = 5)
```

Eksport modelu

```
h2o.saveModel(rf_model, "work/rf_model.h2o")
```

Scoring

```
rf_model <- h2o.loadModel("work/rf_model.h2o")
rf_score <- h2o.predict(rf_model,
                        newdata = h2o_test)
```

Eksport scoringu

```
rf_score <- as.data.table(rf_score)
write.table(x = rf_score,
      file = "work/score.csv",
      col.names = TRUE,
      row.names = FALSE,
      sep = ";")
```

Grid search

```
h2o_rf_grid <- h2o.grid(algorithm = "randomForest",
                        grid_id = "rf_grid",
                        hyper_params = list(
                          ntrees = c(50, 200)
                        ),
                        training_frame = h2o_train,
                        x = colnames(h2o_train),
                        y = "advertisement",
                        nfolds = 5,
                        seed = 1245)
```

Ensemble

```
learner <- c("h2o.glm.wrapper", "h2o.randomForest.wrapper",
             "h2o.gbm.wrapper", "h2o.deeplearning.wrapper")
metalearner <- "h2o.glm.wrapper"

fit <- h2o.ensemble(training_frame = h2o_train,
                    x = colnames(h2o_train),
                    y = "advertisement",
                    family = "binomial",
                    learner = learner,
                    metalearner = metalearner,
                    cvControl = list(V = 5),
                    seed = 1245)
```

WLOG SOLUTIONS

Dziękuję za uwagę