R ♥ Julia

présenté par:
Jérémie Desgagné-Bouchard

EVOVEST

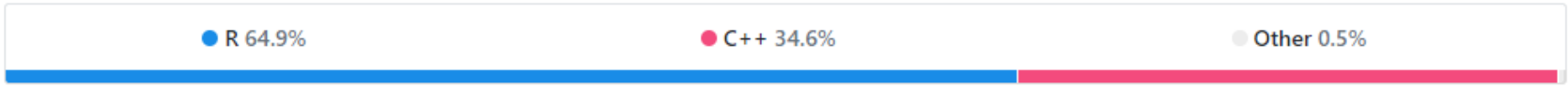
INVESTISSEMENTS ÉVOLUTIFS

L’enjeu des deux langages

R's data.table package extends data.frame: http://r-datatable.com

dplyr: A grammar of data manipulation https://dplyr.tidyverse.org

r data-manipulation grammar



Scalable, Portable and Distributed Gradient Boosting (GBDT, GBRT or GBM) Library, for Python, R, Java, Scala, C++ and more. Runs on single machine, Hadoop, Spark, Flink and DataFlow https://xgboost.ai/

gbdt gbrt gbm distributed-systems xgboost machine-learning

L’enjeu des deux langages

Parallel analytical database in pure Julia http://juliadb.org/

Relax! Flux is the ML library that doesn't make you tensor https://fluxml.ai/

flux machine-learning neural-networks the-human-brian deep-learning data-science

Application Adhoc avec JuliaCall

kaggle instacart

https://www.kaggle.com/c/instacart-market-basket-analysis

Sachant les achats passés, quelle selection d'items lors d'une nouvelle commande maximise le score F1?

Solution:

1. Appliquer un modèle de classification identifiant pour chaque produit précédemment acheté la probabilité d'être ajoutée au panier courant.
2. Soumettre les produits dont la probabilité d'achat est élevée.

Application Adhoc avec JuliaCall

Optimizing F-Measures: A Tale of Two Approaches:
https://arxiv.org/ftp/arxiv/papers/1206/1206.4625.pdf

Algorithm 1 Compute $f_{\beta;1}, \ldots, f_{\beta;n}$, where $\beta^2 = q/r$

1: For $0 \leq i \leq n$, set $C[i]$ as the coefficient of x^i in $[p_1 x + (1 - p_1)] \ldots [p_N x + (1 - p_N)]$;
2: For $1 \leq i \leq (q + r)n$, $S[i] \leftarrow q/i$;
3: **for** $k = n$ to 1 **do**
4: $\quad f_{\beta;k} \leftarrow \sum_{k_1=0}^{n} (1 + r/q) k_1 C[k_1] S[rk + qk_1]$;
5: $\quad$ Divide C by $p_k x + (1 - p_k)$;
6: $\quad$ **for** $i = 1$ to $(q + r)(k - 1)$ **do**
7: $\quad\quad S[i] \leftarrow (1 - p_k) S[i] + p_k S[i + q]$;
8: $\quad$ **end for**
9: **end for**

60% ✓

40% ✓

80% ✓

40% ✗

Application Adhoc avec JuliaCall

Implantation R

```
> system.time(submission <- pred_df_sample %>%
+     group_by(order_id) %>%
+     summarise(products = maximize_expectation(pred, 0, prods = product_id)))
   user  system elapsed
   7.70    0.00    7.71
```

Implantation Rcpp

```
> system.time(submission <- pred_df_sample %>%
+     group_by(order_id) %>%
+     summarise(products = exact_F1_max_none(ps = pred, p0 = 0, prods = product_id)))
   user  system elapsed
   0.26    0.00    0.26
```

Implantation Julia

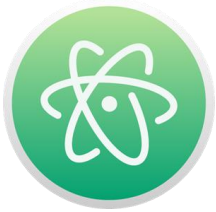
```
> system.time(submission <- pred_df_sample %>%
+     group_by(order_id) %>%
+     summarise(products = julia_F1_max_none(ps = pred, p0 = 0, prods = product_id)))
   user  system elapsed
   0.30    0.03    0.33
```

Environement de développement

Language + console

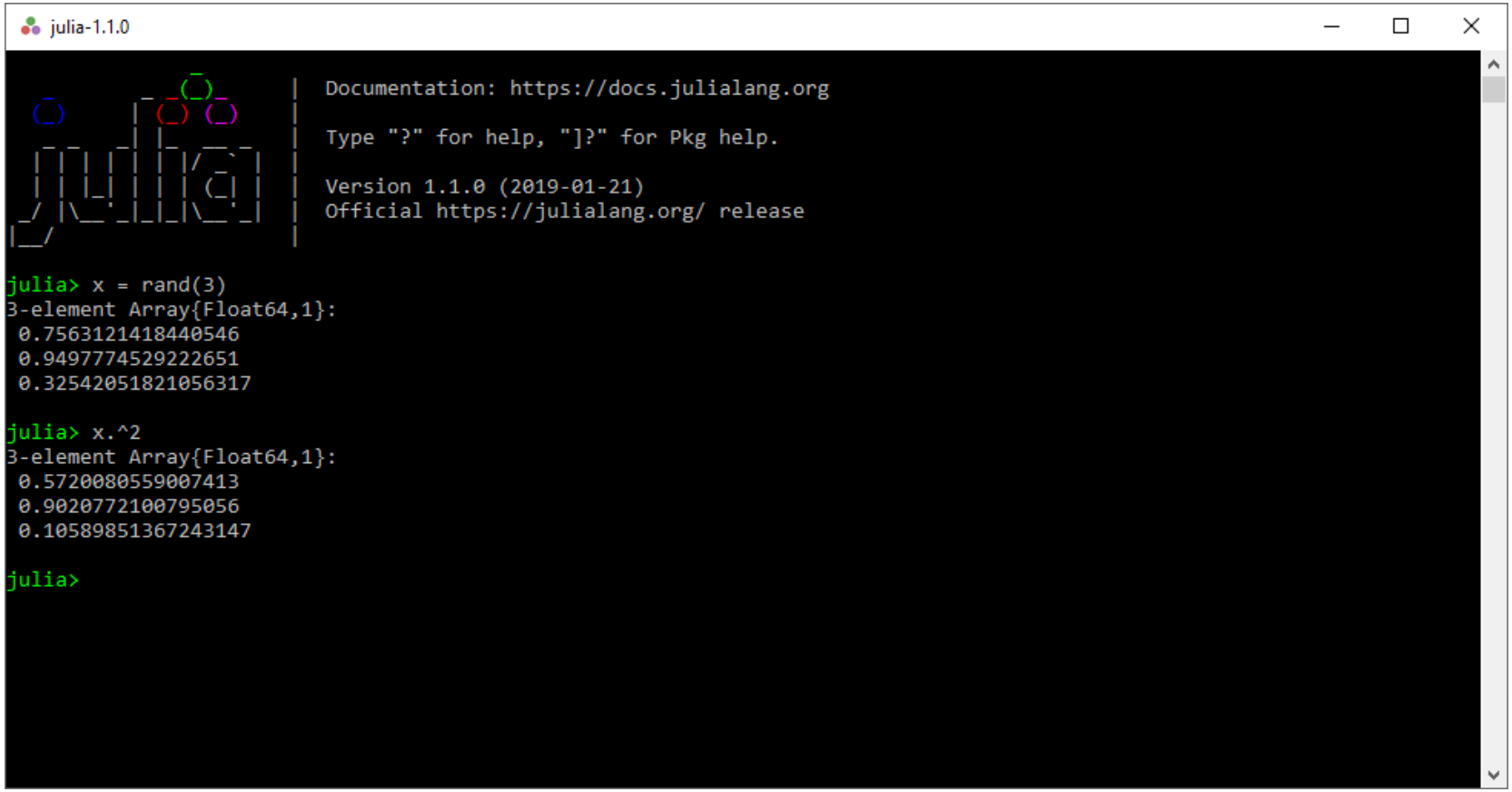
https://julialang.org/downloads/

Environnement de développement

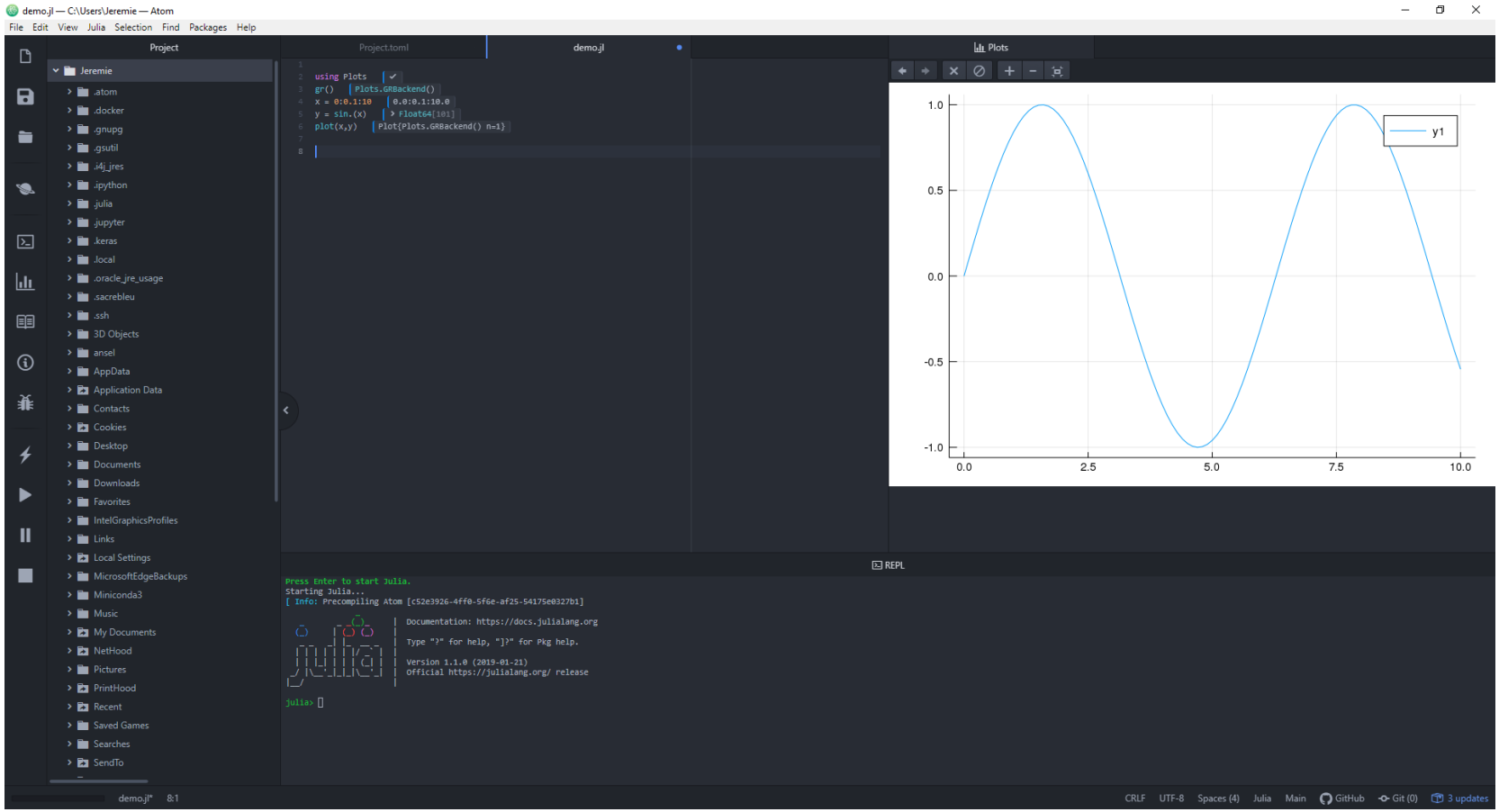


https://junolab.org/

Console


julia-1.1.0
Documentation: https://docs.julialang.org
Type "?" for help, "]?" for Pkg help.
Version 1.1.0 (2019-01-21)
Official https://julialang.org/ release
julia> x = rand(3)
3-element Array{Float64,1}:
0.7563121418440546
0.9497774529222651
0.32542051821056317
julia> x.^2
3-element Array{Float64,1}:
0.5720080559007413
0.9020772100795056
0.10589851367243147
julia>

Juno


demo.jl — C:\Users\Jeremie — Atom
File Edit View Julia Selection Find Packages Help
Project
Jeremie
.atom
.docker
.gnupg
.gsutil
.i4j_jres
.ipython
.julia
.jupyter
.keras
.local
.oracle_jre_usage
.sacrebleu
.ssh
3D Objects
ansel
AppData
Application Data
Contacts
Cookies
Desktop
Documents
Downloads
Favorites
IntelGraphicsProfiles
Links
Local Settings
MicrosoftEdgeBackups
Miniconda3
Music
My Documents
NetHood
Pictures
PrintHood
Recent
Saved Games
Searches
SendTo
Project.toml
demo.jl
using Plots
gr()
x = 0:0.1:10
y = sin.(x)
plot(x,y)
Plots
y1
1.0
0.5
0.0
-0.5
-1.0
0.0
2.5
5.0
7.5
10.0
REPL
Press Enter to start Julia.
Starting Julia...
[Info: Precompiling Atom [c52e3926-4ff0-5f6e-af25-54175e0327b1]
Documentation: https://docs.julialang.org
Type "?" for help, "]?" for Pkg help.
Version 1.1.0 (2019-01-21)
Official https://julialang.org/ release
julia>
demo.jl* &1
CRLF UTF-8 Spaces (4) Julia Main GitHub Git (0) 3 updates

Démarrer un projet/package

```
] generate <NomProjet>
] activate <répertoire du projet>
```

La commande "]" permet d'accéder au à la console du gestionnaire de package Pkg. Alternativement: `Pkg.generate("NomProjet")`

```
; git init
```

La commande ";" active un shell.

Arborescence d'un projet:

```
+-- Project.toml
+-- Manifest.toml
+-- src ->
|              +-- <NomProjet>.jl
```

Propriétés du langage

Langage fonctionnel:

```
function surprise(x)
   x += 1
end
```

Création de structure de données:

```
struct MonObjet
   x
end
```

Multiple dispatch:

```
function surprise(x::T) where {T <: AbstractFloat}
   x += 1
end

function surprise(x::T) where {T <: Int}
   x -= 1
end
```

Intégrer un Package Julia en R

EvoTrees.jl: implantation en Julia d'un algorithme de Gradient Boosted Trees.
https://github.com/Evovest/EvoTrees.jl

EvoTrees: integration du package EvoTrees.jl en un package R.
https://github.com/Evovest/EvoTrees

```
# /R/zzz.R
.onLoad <- function(libname, pkgname) {
  library(JuliaCall)
  JuliaCall::julia_setup()
  JuliaCall::julia_library("EvoTrees")
}
```

```
# /R/EvoTrees.R
#' @export
evo_train <- function(data_train, target_train, params=set_params(), ...) {
  params <- do.call(set_params, params)
  model <- JuliaCall::julia_call("grow_gbtree", data_train, target_train, params, ..., need_return = "Julia")
  return(model)
}
```