

MLOps pour R



We R
Association
Madagascar



DevOps et MLOps



Plan

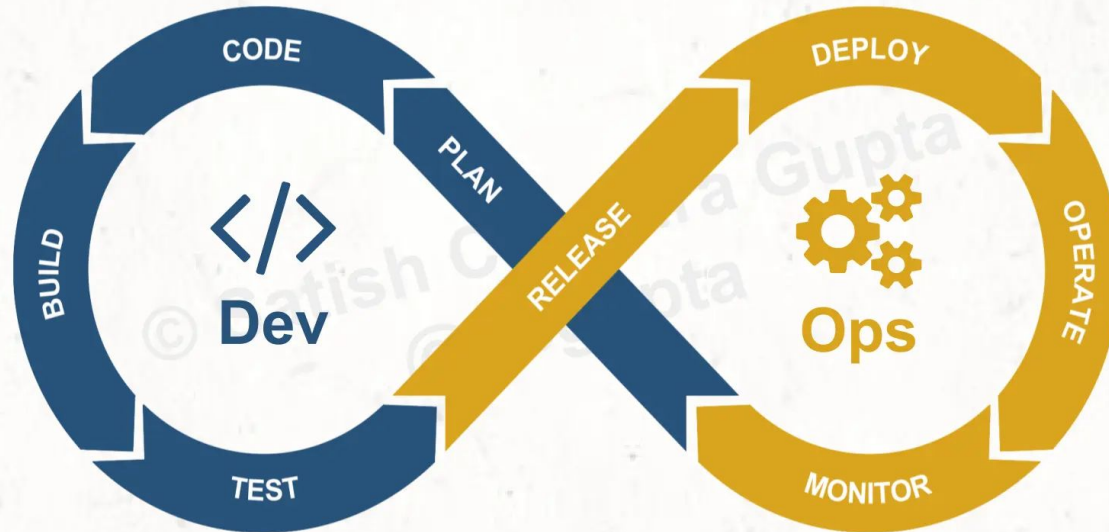
- DevOps et MLOps
- R langage de programmation pour l'analyse de données et le ML
- Processus MLOps sur Azure ML
- Processus MLOps avec Vetiver



DevOps

Software Development

ml4devs.com/mlops-lifecycle 



© Satish Chandra Gupta



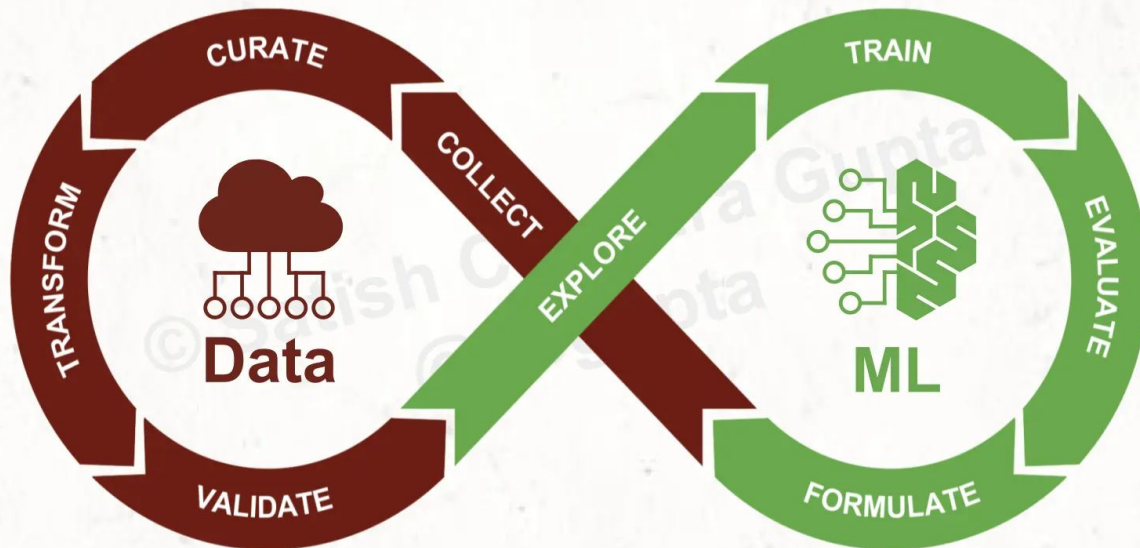
CC BY-NC-ND 4.0 International License
creativecommons.org/licenses/by-nc-nd/4.0/

scgupta.me 
twitter.com/scgupta 
linkedin.com/in/scgupta 

ML

Model Development

ml4devs.com/mlops-lifecycle



© Satish Chandra Gupta



CC BY-NC-ND 4.0 International License
creativecommons.org/licenses/by-nc-nd/4.0/

scgupta.me

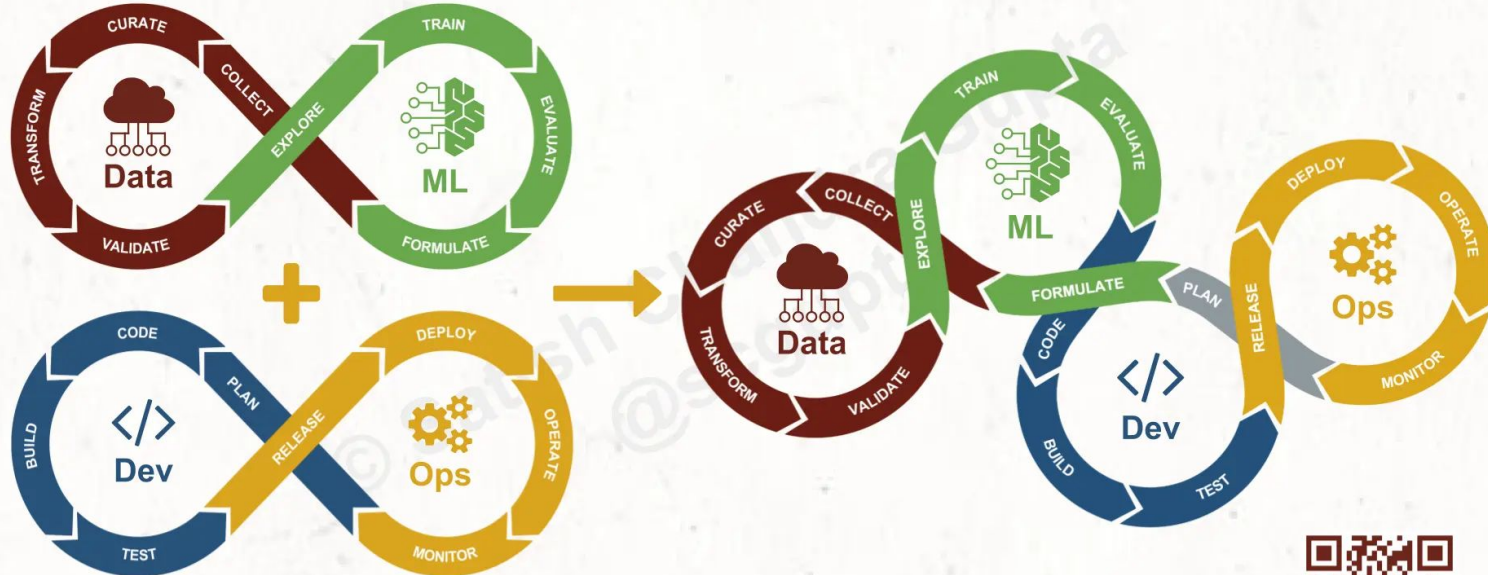
twitter.com/scgupta

linkedin.com/in/scgupta

MLOps

MLOps = DataML + DevOps

ml4devs.com/mlops-lifecycle 



© 2022 Satish Chandra Gupta



CC BY-NC-ND 4.0 International License
creativecommons.org/licenses/by-nc-nd/4.0/

scgupta.me 
twitter.com/scgupta 
linkedin.com/in/scgupta 



DEVOPS	MLOPS
Gestion de code	Gestion de code Gestion des fichiers de données, des notebooks, des documents Rmd
Gestion d'infrastructure	Gestion d'infrastructure Gestion d'environnement
Contrôle de code sources	Contrôle de code sources Suivi des résultats des expérimentations (des modèles) Gestion des jeux de données
Compilation pour obtenir des fichiers exécutables Ressources standards pour la compilation	Entraînement des modèles CPU/GPU pour les entraînements
Gestion des versions des builds	Gestions des versions des modèles
Tests (déterministes) Débogage des erreurs dans le code	Tests (probabilistes) Débogage des erreurs dans le code du modèle et/ou des erreurs dans les jeux de données Perte de performances, ré-entraînements

R en tant que langage de programmation pour l'analyse de
données et le Machine Learning



R

- Créé par Rosh Ihaka et Robert Gentleman
- Paquets (packages)
- CRAN (<https://cran.r-project.org/>)
- Visualisation de données: ggplot2
- Machine Learning:
 - tidymodels
 - caret
 - xgboost
 - randomForest
 - torch
 - tensorflow for R

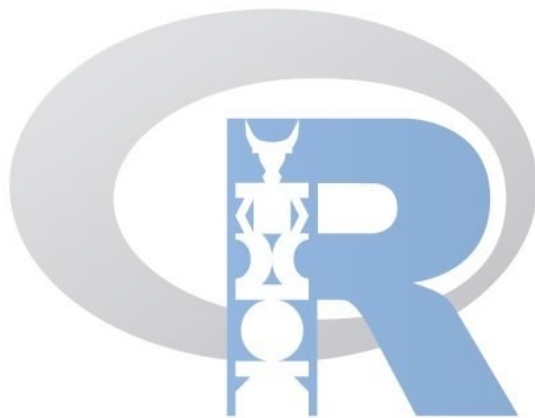


Ressources sur R

- R for Data Science (<https://r4ds.hadley.nz/>)
- An Introduction to Statistical Learning with Applications in R (<https://www.statlearning.com/>)



Processus MLOps avec R sur Azure ML



Azure ML SDK pour R

- Paquet R sur CRAN pour la gestion des services Azure ML: azuremlsdk (<https://cran.r-project.org/web/packages/azuremlsdk/index.html>)
- Création d'espaces de travail, Compute, Modèle et autres outils avec le langage R
- Entraînement, monitoring et déploiement de modèles ML
- Déploiement de modèles en tant que webs services
- Exécution des routines d'entraînements et de déploiements automatiques



Workflow

- Données sur les accidents de voitures aux USA de 1997 à 2002 (<https://rdrr.io/cran/DAAG/man/nassCDS.html>)
- importation des données et traitements
- Création des clusters d'entraînements
- Choix d'un modèle et déploiement en tant que container
- Application Shiny et appel de fonction R par REST endpoint



Préparation et partage des données

```
library(azuremlsdk)  
ws <- load_workspace_from_config()
```

chargement de azuremlsdk
chargement de l'espace de travail avec les clés
d'authentification pré-installées dans l'espace

```
nassCDS <- read.csv("nassCDS.csv")  
# Lots of cleaning code  
saveRDS(accidents,  
        file="accidents.Rd")
```

lecture du fichier csv du jeu de données
code omis
enregistrement des données sous formats Rd
pour données R

```
ds <- get_default_datastore(ws)  
target_path <- "accidentdata"  
upload_files_to_datastore(ds,  
    list("./accidents.Rd"),  
    target_path = target_path,  
    overwrite = TRUE)
```

chargement de azuremlsdk
chargement de l'espace de travail avec les clés
d'authentification pré-installées dans l'espace

Création d'un cluster d'entraînement à deux noeuds

```
ws <- load_workspace_from_config()
compute_target <- create_aml_compute(
  workspace = ws,
  cluster_name = "rcluster",
  vm_size = "STANDARD_D2_V2",
  vm_priority = "lowpriority",
  min_nodes = 1,
  max_nodes = 2)
```

chargement de l'espace de travail

la taille augmente au fur et à mesure que les jobs sont ajoutées à la queue

Entraînement des modèles et choix d'un modèle

```
exp <- experiment(ws, "accident")
est <- estimator(source_directory = ".",
                 entry_script = "accident-glmnet.R",
                 script_params = list(
                   "--data_folder" = ds$path(target_path),
                   "--percent_train" = 0.75),
                 compute_target = compute_target)
run.glmnet <- submit_experiment(exp, est)
```

Enregistrement du modèle

```
model <- register_model(  
  ws,  
  model_path = "outputs/model.rds",  
  model_name = "accidents_model_caret",  
  description = "Predict accident probability")
```

enregistrement du modèle en tant que R model
Object
Option de versionning

```
r_env <- r_environment(name = "basic_env",  
                      cran_packages="caret")
```

Paquets R (CRAN, Github) nécessaires à
l'exécution

```
inference_config <- inference_config(  
  entry_script = "accident_predict_caret.R",  
  source_directory = ".",  
  environment = r_env)
```

Script R qui reçoit un modèle et des données
sous format JSON pour les prédictions

Déploiement

```
aci_config <-  
  aci_webservice_deployment_config(  
    cpu_cores = 1, memory_gb = 0.5)  
  
aci_service <- deploy_model(ws,  
  'accident-pred-caret',  
  list(model),  
  inference_config,  
  aci_config)  
  
accident.endpoint <- get_webservice(  
  ws, "accident-pred-caret")$scoring_uri
```

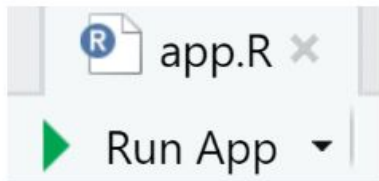
aci = Azure Container Instance

obtention du REST endpoint pour l'utilisation de l'app

Intégration du modèle dans une application Shiny

```
library(httr)
v <- POST(accident.endpoint,
          body=input,
          encode="json")

pred <- content(v)[[1]]
```



Accident Fatality Probability Estimator

Occupant Age:
16 68 95
16 24 32 40 48 56 64 72 80 88 95

Occupant gender:
f

Occupant role:
pass

Vehicle Year:
1,955 2,002
1,955 1,960 1,965 1,970 1,975 1,980 1,985 1,990 1,995 2,000 2,005

Seatbelt:
belted

Airbag:
none

Impact speed:
40-54

Collision type:
notfrontal

Probability (%)

100
80
60
40
20
0

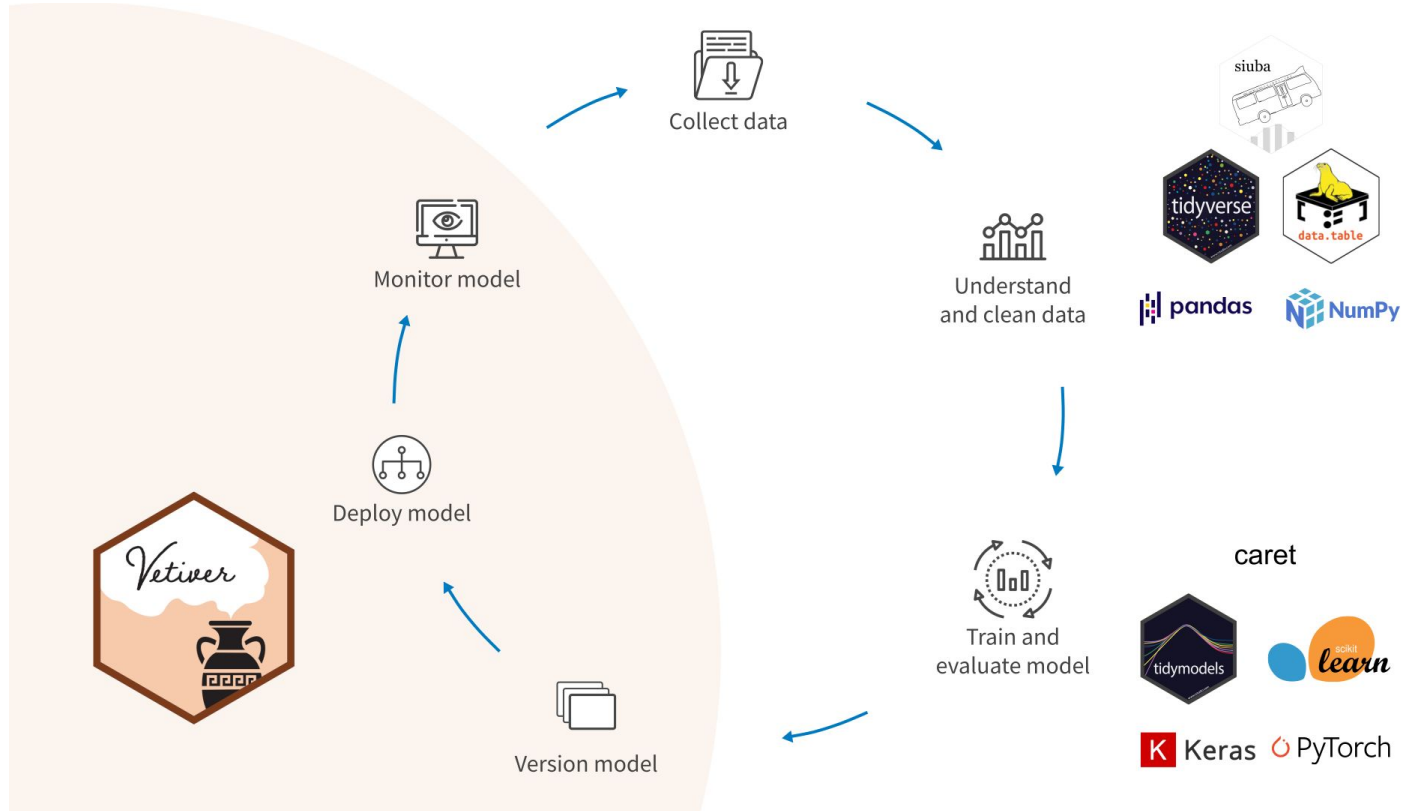


30.03

Processus MLOps pour R avec Vetiver



Vetiver et MLOps



<https://vetiver.rstudio.com/>

- tidymodels ou caret, etc,...
- plumber
- pins



PlumberR

```
# Generated by the vetiver package; edit with care

library(pins)
library(plumber)
library(rapidoc)
library(vetiver)

# Packages needed to generate model predictions
if (FALSE) {
  library(caret)
}
b <- board_folder(path = "pins-r")
v <- vetiver_pin_read(b, "accident_model")

#* @plumber
function(pr) {
  pr %>% vetiver_api(v)
}
```

PlumberR

```
# Generated by the vetiver package; edit with care

library(pins)
library(plumber)
library(rapidoc)
library(vetiver)

# Packages needed to generate model predictions
if (FALSE) {
  library(caret)
}
pin_loc <- pins::github_raw("DimbyTa/r-mlops-vetiver-huggingFace/main/pins-r/_pins.yaml")
b <- board_url(pin_loc)
v <- vetiver_pin_read(b, "accident_model")

#* @plumber
function(pr) {
  pr %>% vetiver_api(v)
}
```

Dockerfile

```
# Generated by the vetiver package; edit with care

FROM rocker/r-ver:4.0.5

ENV RENV_CONFIG_REPOS_OVERRIDE https://packagemanager.rstudio.com/cran/latest

WORKDIR $HOME

RUN apt-get update -qq && apt-get install -y --no-install-recommends \
    libcurl4-openssl-dev \
    libicu-dev \
    libsodium-dev \
    libssl-dev \
    make \
    zlib1g-dev \
    && apt-get clean

COPY vetiver_renv.lock renv.lock

RUN Rscript -e "install.packages('renv')"
RUN Rscript -e "renv::restore()"
COPY plumber.R /opt/ml/plumber.R
EXPOSE 8000
ENTRYPOINT ["R", "-e", "pr <- plumber::plumb('/opt/ml/plumber.R'); pr$run(host = '0.0.0.0', port = 8000)"]
```

Dockerfile

```
# Generated by the vetiver package; edit with care

FROM rocker/r-ver:4.0.5
# Create a non-root user to run the application
RUN useradd -m -u 1000 appuser
USER appuser

ENV RENV_CONFIG_REPOS_OVERRIDE https://packagemanager.rstudio.com/cran/latest
ENV HOME=/home/appuser
WORKDIR $HOME
USER root
RUN apt-get update -qq && apt-get install -y --no-install-recommends \
    libcurl4-openssl-dev \
    libicu-dev \
    libsodium-dev \
    libssl-dev \
    make \
    zlib1g-dev \
    && apt-get clean
USER appuser
COPY vetiver_renv.lock renv.lock
# Create the .cache directory and give appuser permission to write to it
RUN mkdir -p /home/appuser/.cache && chown -R appuser:appuser /home/appuser/.cache
# Create the .cache/pins/url directory and give appuser permission to write to it
RUN mkdir -p /home/appuser/.cache/pins/url && chown -R appuser:appuser /home/appuser/.cache/pins/url
USER root
RUN Rscript -e "install.packages('renv')"
#USER appuser
RUN Rscript -e "renv::restore()"
USER appuser
COPY plumber.R /opt/ml/plumber.R
EXPOSE 7860
ENTRYPOINT ["R", "-e", "pr <- plumber::plumb('/opt/ml/plumber.R'); pr$run(host = '0.0.0.0', port = 7860)"]
```

- Dockerfile
- plumber.R
- vetiver_renv.lock
- <https://github.com/DimbyTa/r-mlops-vetiver-huggingFace>

Merci de votre attention

