BANENE

R 기반의 SAS Viya

강봉주

bonjour.kang@gmail.com

BANENE

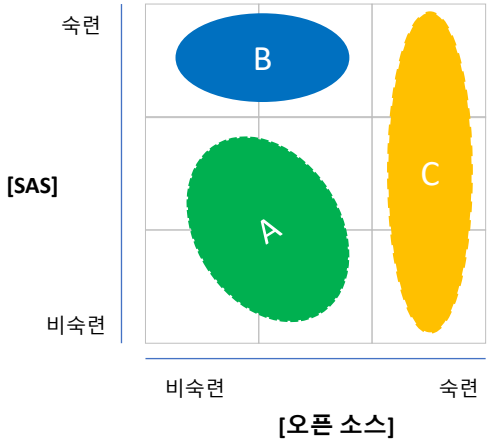
SWAT 패키지

SAS Viya의 오픈소스 인터페이스

강봉주

bonjour.kang@gmail.com

세션 대상자



- A 그룹은 SAS나 오픈소스 둘 다 못하거나 어느 정도 친숙한 그룹
- B 그룹은 SAS의 프로그래밍 방식으로 매우 잘 쓰는 그룹
- C 그룹은 오픈소스 매니아 들이며 다른 툴에 대한 관심이 없는 그룹

사용자 그룹	주 교육 방향	비고
A	시각화 기반의 툴에 대한 교육 'VS, VDMML 기반의 머신러닝'	생산성 향상, 산출물에 대한 이해 증진
B	SAS CASL로 플랫폼 전환 교육 'SAS CASL 기본 교육'	인메모리, 분산환경의 SAS 플랫폼 변환
C	오픈소스 인터페이스에 대한 교육 'SAS 오픈소스 인터페이스: SWAT'	각자의 개발 환경에서 SWAT을 통한 데이터 분석 유사한 패키지보다 우수한 성능 입증

순서

PART 1
- 개요
- 접속
- 카스 서버 데이터 적재
- 액션 집합

PART 2
- 데이터 분할
- 변수 정의
- 분석

개요

SAS Viya, CAS, CAS 액션

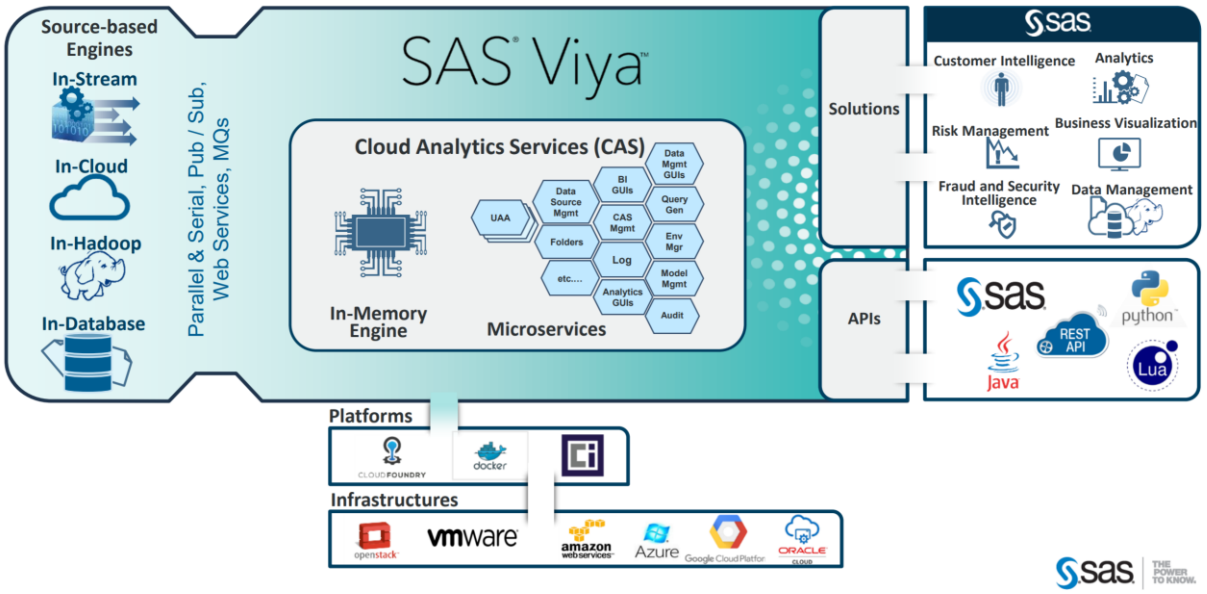
SAS Viya

- Viya는 SAS의 새로운 플랫폼
- BASE SAS 근간의 전통적인 플랫폼에서 벗어나 개방형(open), 클라우드 서비스 가능(cloud-enabled)의 분석 실행 환경

SAS Viya, CAS, CAS 액션

CAS(Cloud Analytic Services)

Viya는 인메모리 엔진이며 분산 환경을 지원하는 서비스인 CAS를 포함한 많은 서비스를 갖고 있고 클라우드 서비스가 가능한 분석 환경



SAS Viya, CAS, CAS 액션

CAS Action, Action Set

- 카스 액션은 실행가능한 루틴이며 카스 서버에서 일의 가장 작은 단위
- 분석 알고리즘, 데이터 관리 등에 액션들이 존재
- 액션들은 액션집합으로 범주화 되어 있음
- 예를 들어 테이블 액션집합은 테이블 적재, 테이블 삭제, 테이블 관리 등을 위한 액션으로 구성
- 대부분의 액션은 리스트 데이터 형으로 나옴
- 결과가 나오면 이후에 결과의 특정한 값을 찾거나 이후의 입력으로 연속 사용이 가능

SAS Viya, CAS, CAS 액션

CAS Action, Action Set

디폴트 액션 셋이 아닌 경우에 사전에 반드시 메모리 올리는 작업 필요

Tables Action Set: Syntax

Provides actions for accessing and managing data

Syntax ▾ Examples ▾ Details

Table of Actions

Action Name	Description
addCaslib	Adds a new caslib to enable access to a data source
addCaslibSubdir	Creates a subdirectory in an existing caslib
addTable	Add a table by sending it from the client to the server
alterTable	Rename tables, change labels and formats, drop columns
attribute	Manages extended table attributes
caslibInfo	Shows caslib information
columnInfo	Shows column information

SWAT

Scripting Wrapper for Analytics Transfer:
(분석 전달을 위한 스크립터 포장지)

- 버전: 1.8.1 (Jan 21, 2021)
- 카스 서버 접속 및 실행을 위한 오픈소스 인터페이스
- 결과는 리스트
- `library('swat')`

https://github.com/sassoftware/R-swat

필요한 패키지

```
library('swat')
```

그림 그리기 위한 패키지 외에 단지 swat만 필요!

접속

카스 서버 접속

접속 방법

- CAS 함수 이용

```
# REST API를 통한 CAS 서버 접속 및 세션 생성
conn = swat::CAS(hostname='10.0.11.34', port=8777,
protocol='https')
conn
```

```
CAS(hostname=10.0.11.34, port=8777, username=viyademo05, session=7ef3e646-
7c5f-dd4f-9109-ec120315f240, protocol=https)
```

산출물 유형

산출물

리스트

```
# 서버 정보 확인
out = cas.builtins.serverStatus(conn)
class(out)
```

```
'list'
```

산출물 유형

산출물

리스트

```
# 서버 정보 확인
out = conn.serverstatus()
names(out)
```

```
'About' 'nodestatus' 'server'
```

산출물 유형

산출물

리스트

```
$CAS
'Cloud Analytic Services'
$Copyright
'Copyright © 2014-2018 SAS Institute Inc. All Rights
Reserved.'
$ServerTime
'2021-03-24T04:32:48Z'
$System
$Hostname
'casc.sas.com'
$`Linux Distribution`
'Red Hat Enterprise Linux Server release 7.6 (Maipo)'
$`Model Number`
'x86_64'
$`OS Family`
'LIN X64'
$`OS Name`
'Linux'
$`OS Release`
'3.10.0-957.12.2.el7.x86_64'
```

```
$`OS Version`
'#1 SMP Fri Apr 19 21:09:07 UTC 2019'
$Version
'3.05'
$VersionLong
'V.03.05M0P11062019'
$license
$expires
'17Jan2022:00:00:00'
$gracePeriod
45
$site
'Internal Usage'
$siteNum
70180938
$warningPeriod
47
```

산출물 유형

산출물

리스트

```
# 특정 항목에 대한 값 가져오기
out$About
```

카스 서버 접속

카스 라이브러리

- 카스 서버의 어떤 곳에 데이터가 저장되는지 확인 필요

```
# 라이브러리 정보 확인
cas.table.caslibInfo(conn, active=TRUE)
```

$CASLibInfo =

	Name	Type	Description	Path	Definition	Subdirs	Local	Active	Personal	Hidden	Transient
CASUSERHDFS(viyademo05)	HDFS	Personal HDFS Caslib	/user/viyademo05/		1	0	1	1	0	1

카스 서버 데이터 적재

카스 서버 데이터 적재

적재 방법

카스 테이블은 서버에 있는 데이터이며 로컬에서는 단지 뷰임

```
# 특정 URL에 있는 데이터 올리기
tbl = cas.read.csv(conn, 'https://github.com/bong-ju-
kang/data/raw/master/bank.csv',
                   casOut=list(name='bank', replace=TRUE))
class(tbl)
```

```
'CASTable'
```

카스 서버 데이터 적재

데이터 보기 및 요약

액션 셋을 이용한 테이블 보기 및 요약

table, simple, ...

```
# 메타데이터 정보
cas.table.columnInfo(conn, table='bank')
```

$ColumnInfo =

Column	Label	ID	Type	RawLength	FormattedLength	Format	NFL	NFD
age		1	double	8	12		0	0
job		2	varchar	13	13		0	0
marital		3	varchar	8	8		0	0
education		4	varchar	9	9		0	0
default		5	varchar	3	3		0	0
balance		6	double	8	12		0	0
housing		7	varchar	3	3		0	0

카스 서버 데이터 적재

데이터 보기 및 요약

액션 셋을 이용한 테이블 보기 및 요약

table, simple, ...

```
# 데이터 일부 보기
cas.table.fetch(conn, table='bank', to=5)
```

$Fetch =

Index	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pday
1	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	79	1	-
2	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	220	1	33
3	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	185	1	33
4	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	199	4	-
5	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	226	1	-

카스 서버 데이터 적재

데이터 보기 및 요약

액션 셋을 이용한 테이블 보기 및 요약

table, simple, …

```
# 데이터 요약
cas.simple.summary(conn, table='bank')
```

$Summary =

Column	Min	Max	N	NMiss	Mean	Sum	Std	StdErr	Var	USS	CSS	CV
age	19	87	4521	0	41.1700951	186130	10.576211	0.15729425	1.118562e+02	8168580	5.055902e+05	25.68906
balance	-3313	71188	4521	0	1422.6578191	6431836	3009.638142	44.76071639	9.057922e+06	50092108080	4.094181e+10	211.55039
day	1	31	4521	0	15.9152842	71953	8.247667	0.12266308	6.802402e+01	1452621	3.074686e+05	51.82231
duration	4	3025	4521	0	263.9612917	1193369	259.856633	3.86470681	6.752547e+04	620218345	3.052151e+08	98.44498
campaign	1	50	4521	0	2.7936297	12630	3.109807	0.04625047	9.670897e+00	78996	4.371246e+04	111.31778
pdays	-1	871	4521	0	39.7666445	179785	100.121124	1.48904720	1.002424e+04	52459009	4.530956e+07	251.77162

액션 집합

액션 집합(action set) 호출하기

builtins 액션 집합

actionSetInfo, loadActionSet 이용

```
# 현재 설정된 액션 집합 보기
cas.builtins.actionSetInfo(conn)
```

$setinfo =

actionset	label	loaded	extension	version	product_name	user_defined
accessControl	Access Controls	1	tkacon	3.05.000	tkcas	false
accessControl	Access Controls	1	casmeta	3.05.000	tkcas	false
builtins	Builtins	1	tkcasablt	3.05.000	tkcas	false
configuration	Server Properties	1	tkcascfg	3.05.000	tkcas	false
dataPreprocess	Data Preprocess	1	tktrans	3.05.000	crsstat	false
dataStep	DATA Step	1	datastep	3.05.000	tkcas	false
percentile	Percentile	1	tkcasptl	3.05.000	crsstat	false
search	Search	1	casidx	3.05.000	crssearch	false

액션 집합(action set) 호출하기

builtins 액션 집합

actionSetInfo, loadActionSet 이용

```
# 필요한 액션 집합 적재 하기
cas.builtins.loadActionSet(conn, actionSet="sampling")
cas.builtins.loadActionSet(conn, actionSet="regression")
cas.builtins.loadActionSet(conn, actionSet="astore")
```

데이터 분할

데이터 분할

sampling 액션 집합

stratified 이용

```
# 데이터 분할
cas.sampling.stratified(conn,
    # 데이터와 층화변수 지정
    table=list(name='bank', groupby='y'),
    # 산출 데이터 지정
    output=list(casOut=list(name='bank_part', replace=TRUE),
copyVars='ALL'),
    # 표본 추출 비용 지정
    samppct=70,
    # 표본 추출 변수 지정: 표본이면 1 아니면 0
    partind=TRUE
)
```

데이터 분할

sampling 액션 집합

stratified 이용

$OutputCasTables

	casLib	Name	Label	Rows	Columns
CASUSERHDFS(viyademo05)		bank_part		4521	18

$STRAFreq

ByGrpID	y	NObs	NSamp
0	no	4000	2800
1	yes	521	365

데이터 분할

훈련데이터, 검증데이터 지정

defCasTable 함수 이용

```
# 훈련 데이터
train_casdf = defCasTable(conn, 'bank_part', where='_PartInd_=1')
head(train_casdf)

# 검증 데이터
valid_casdf = defCasTable(conn, 'bank_part', where='_PartInd_=0')
head(valid_casdf)

# 전체 데이터
all_casdf = defCasTable(conn, 'bank_part')
head(all_casdf)
```

변수 정의

변수 정의

table 액션집합

columnInfo 이용

```
# 데이터 정의
casdf = defCasTable(conn, 'bank')
```

변수 정의

table 액션집합

columnInfo 이용

```
# 변수들을 리스트로 만들기
meta = cas.table.columnInfo(conn, table='bank')
allvars <- as.list(meta$ColumnInfo$Column)
allvars
```

1. 'age'
2. 'job'
3. 'marital'
4. 'education'
5. 'default'
6. 'balance'
7. 'housing'
8. 'loan'
9. 'contact'
10. 'day'
11. 'month'

변수 정의

table 액션집합

columnInfo 이용

```
# 목표 변수 정의
target <- 'y'

# 입력 변수 정의
xvars = setdiff(allvars, as.list(target))

# 범주 입력 변수 정의
class_vars = allvars[meta$ColumnInfo$Type == 'varchar']

# 값 정의
event <- 'yes'
non_event <-  'no'
```

변수 정의

table 액션집합

columnInfo 이용

```
# 목표변수의 표본 분포
freq = cas.simple.freq(conn, table='bank', inputs='y')$Frequency
freq
```

Column	CharVar	FmtVar	Level	Frequency
y	no	no	1	4000
y	yes	yes	2	521

분석: 모델적합 (로지스틱 회귀)

모델 적합

regression 액션집합

logistic 이용

모델 적합

```
# 로지스틱 회귀 모델 적합
lr = cas.regression.logistic(conn,

    # 데이터 지정
     table=all_casdf,

    # 범주 변수 지정
    classVars=class_vars,

    # 주의 사항: 종속변수와 설명변수는 리스트 형식으로 입력
    # 종속변수, 설명변수 지정
    model=list(depvar=list(list(name=target,
options=list(event=event))),
           effects=list(list(vars=xvars))
           ),
```

모델 적합

```
 # 훈련 데이터와 검증 데이터 지정
    partByVar=list(name="_partind_", train="1", valid="0"),

    # 변수 선택 방법 지정
    selection=list(method="FORWARD"),

    # 모델 저장: ASTORE
    savestate=list(name='logistic_model_astore', replace=TRUE)
)
```

모델 적합

regression 액션집합

logistic 이용

```
# 모델 결과가 갖고 있는 정보
names(lr)
```

'ClassInfo' 'ModelInfo' 'NObs' 'OutputCasTables' 'ResponseProfile' 'SelectedModel.Dimensions' 'SelectedModel.FitStatistics' 'SelectedModel.GlobalTest' 'SelectedModel.ParameterEstimates' 'SelectionInfo' 'Summary.ConvergenceStatus' 'Summary.SelectedEffects' 'Summary.SelectionReason' 'Summary.SelectionSummary' 'Summary.StopReason' 'Timing'

모델 적합

regression 액션집합

logistic 이용

```
# 모델 선택 요약 정보
lr$Summary.SelectionSummary
```

Control	Step	EffectEntered	nEffectsIn	SBC	OptSBC
	0	Intercept	1	2271.056	0
-	1	duration	2	1783.222	0
	2	poutcome	3	1662.095	1
	3	month	4	1703.575	0
	4	contact	5	1706.240	0
	5	loan	6	1700.154	0

모델 적합

regression 액션집합

logistic 이용

```
# 저장된 모델 내용: 이진 파일
model <- defCasTable(conn, 'logistic_model_astore')
cas.table.fetch(conn, table=model, to=100)
```

$Fetch =

Index	_index_	
1	0	GB8QETMiADMzAQIxATMBIzMAFBQAIAMBBAkAAAAggwoAAAAAAAz3S0SsSRxauWv6++Vr3iiSgcZ8A
2	1	GB8QETMiADMzAQIxATMBIzMAFBQAIAMBBAkAAAAggwoAAAAAAAz3S0SsSRxauWv6++Vr3iiSgcZ
3	2	GB8QETMiADMzAQIxATMBIzMAFBQAIAMBBAkAAAAggwoAAAAAAAz3S0SsSRxauWv6++Vr3iiSgcZ8A

분석: 점수 산출

점수 산출

astore 액션집합

score 이용

```
# 점수 산출
cas.astore.score(conn,
                 # 점수 산출 대상 파일 지정
                  table=valid_casdf,
                 # ATORE 모델 지정
                  rstore='logistic_model_astore',
                 # 복제할 변수 지정
                  copyvars=allvars,
                 # 점수 산출 저장 테이블 지정
                  casout=list(name='logistic_scored_2',
replace=TRUE)
                 )
```

점수 산출

astore 액션집합

score 이용

```
# 산출된 점수 확인
score <- defCasTable(conn, 'logistic_scored_2')
cas.table.fetch(conn, table=score, to=5)
```

$Fetch =

Index	P_yyes	P_yno	I_y	age	job	marital	education	default	balance	...	loan	contact	day	month
1	0.03438616	0.9656138	no	30	unemployed	married	primary	no	1787	...	no	cellular	19	oct
2	0.09608288	0.9039171	no	33	services	married	secondary	no	4789	...	yes	cellular	11	may
3	0.03188268	0.9681173	no	41	entrepreneur	married	tertiary	no	221	...	no	unknown	14	may
4	0.06625451	0.9337455	no	39	services	married	secondary	no	9374	...	no	unknown	20	may
5	0.03863166	0.9613683	no	43	admin.	married	secondary	no	264	...	no	cellular	17	apr

분석: 모델 평가

모델 평가

percentile 액션집합

assess 이용

```
# 모델 평가
logitAssess = cas.percentile.assess(conn,
    # 평가를 위한 데이터 지정
    table='logistic_scored_2',
    # 실제 목표변수 지정
    response=target,
    # 예측 변수 지정
    inputs=list(list(name=paste0('P_', target, event))),
    # 이벤트 값 지정
    event=event,
    # 이벤트 범주를 제외한 계산된 범주들을 목록화 하여 표시
    pVar=list(paste0('P_', target, non_event)),
    pEvent=list(non_event)
)
```

모델 평가

percentile 액션집합

assess 이용

```
# 평가 결과 보기
names(logitAssess)
```

'FitStat' 'LIFTInfo' 'ROCInfo'

모델 평가

percentile 액션집합

assess 이용

```
# 적합 통계량
logic_fitstat = logitAssess$FitStat
print(logic_fitstat)
```

```
  NOBS       ASE  DIV      RASE       MCE      MCLL
1 1356 0.08009682 1356 0.2830138 0.1039823 0.2736808
```

모델 평가

percentile 액션집합

assess 이용

```
# ROC 정보
logit_rocinfo = logitAssess$ROCInfo
logit_rocinfo
```

Variable	Event	CutOff	TP	FP	FN	TN	Sensitivity	Specificity	KS	...	F_HALF	FPR	
P_yyes	yes	0.00	156	1200	0	0	1.0000000	0.00000000	0	...	0.1397849	1.0000000	0
P_yyes	yes	0.01	156	1200	0	0	1.0000000	0.00000000	0	...	0.1397849	1.0000000	0
P_yyes	yes	0.02	156	1200	0	0	1.0000000	0.00000000	0	...	0.1397849	1.0000000	0
P_yyes	yes	0.03	156	1136	0	64	1.0000000	0.05333333	0	...	0.1465064	0.9466667	0

모델 평가

percentile 액션집합

assess 이용

```
# ROC 곡선
plot(x=logit_rocinfo$FPR, y=logit_rocinfo$Sensitivity, type='l',
col='blue',
    xlab='FPR', ylab='TPR(sensitivity)', main='ROC Curve for
Logistic Model')
abline(c(0,0), c(1, 1), lty=2, col='gray')
legend(0.6, 0.2,c("Logistic","Random"), lwd=c(1,2),
col=c("blue","gray"), bty='n')
```

모델 평가

percentile 액션집합

assess 이용