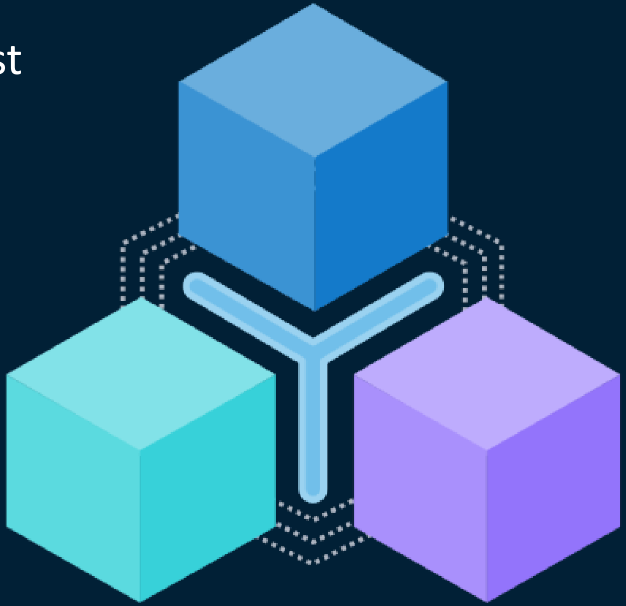
SAS Workbench

A SAS and Python Perspective

Jordan Bakerman – Data Scientist



What is SAS Workbench?

Centralized Coding Application

Web based development environment for SAS, Python, and R practitioners.

IDE agnostic. Choose between VSCode, Jupyter Notebook, etc.

Light Weight and On-Demand

Spin-up and Spin-down CPU/GPU compute resources as needed.

Cloud agnostic. AWS, Azure, GCP, etc.

Start coding in the cloud faster!

SAS Workbench Features

Independent or Symbiotic

Workbench can be a stand-alone application.

Manage, govern, and deploy Workbench models with SAS Viya 4.

Deploy Astores, DATA Step code, or pickle files.

Secure and IT Friendly

Data resides within cloud subscription and firewall.

Access engines available.

Reduces administrative burden for hybrid teams.

Users download OS packages from IT managed list.

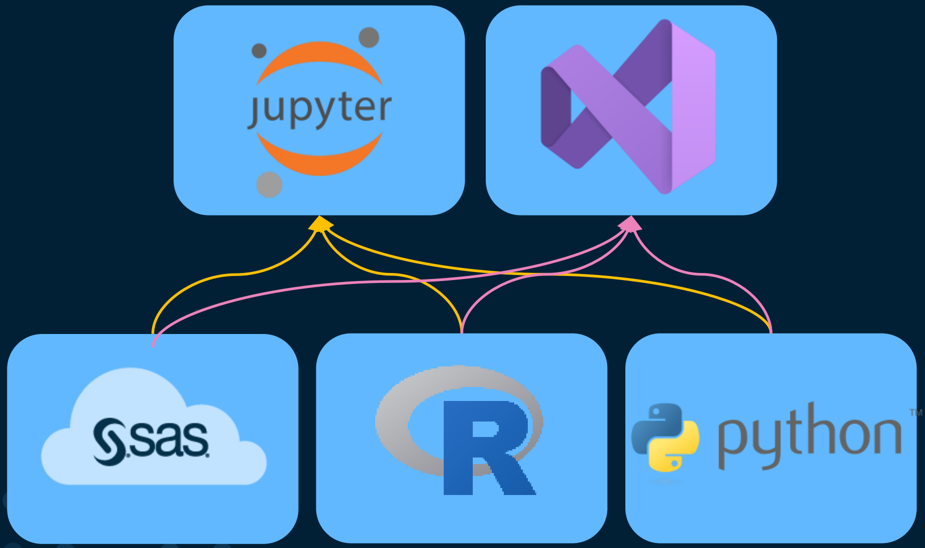
Optimal Cloud Performance

High performance analytics optimized for cloud parallelization.

Workbench APIs allow OS users to leverage SAS algorithms and speed.

Workbench Inheritance from SAS Viya

Language and IDE Agnostic



Multithreaded Algorithms

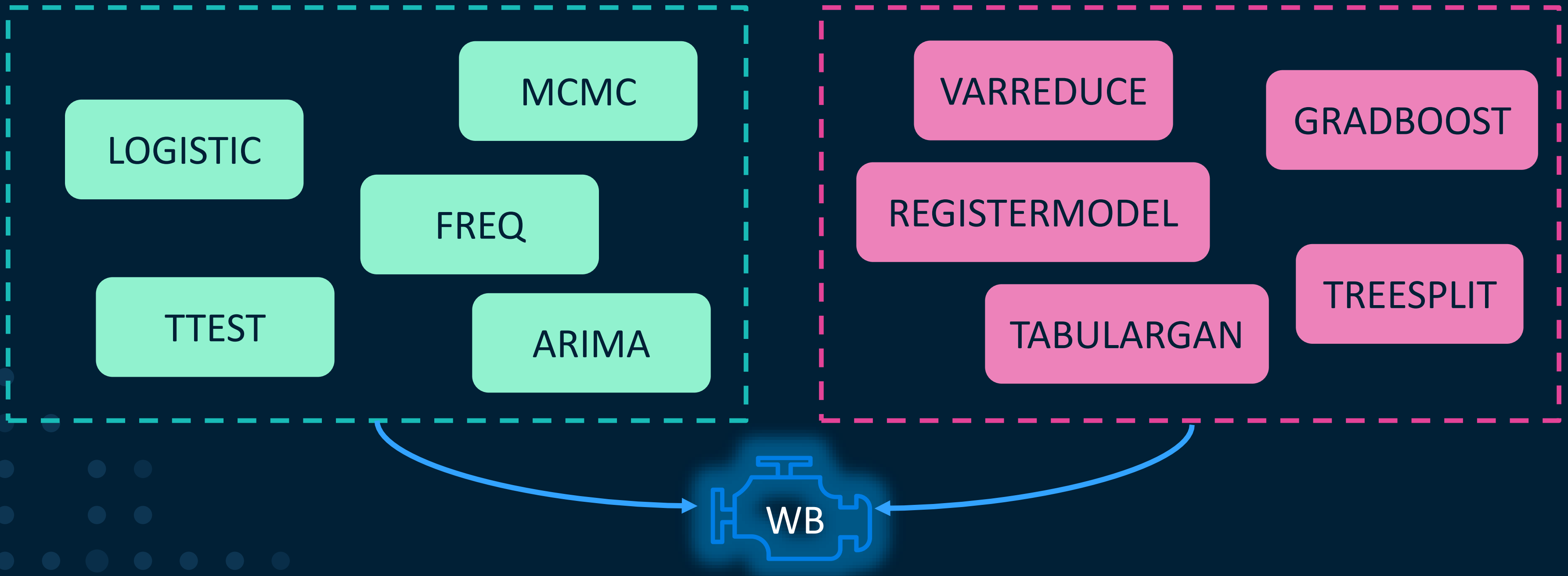
- High performance parallel processing algorithms.
- SAS open-source packages wrap up SAS algorithms have a look and feel of OS syntax.

SAS Workbench Improvements

	SAS Viya	SAS Workbench
Engines	Compute CAS	Single Unified Engine
SAS Code	SAS 9 code runs on Compute and SAS Viya code runs on CAS	Both SAS 9 and SAS Viya code runs on a single unified engine.
API Code	The SWAT package is meant to have a look and feel of open-source code.	The sasviya.ml package code is meant to appear identical to scikit-learn code.

SAS Code on Workbench

Workbench enables SAS coders to leverage SAS 9 procedures and the next generation of SAS Viya procedures on a single application and engine.



Scikit-Learn VS Workbench

Scikit-Learn

```
from sklearn.linear_model
import LogisticRegression

mymod = LogisticRegression(
    fit_intercept = True,
    max_iter      = None
    tol           = .0001
)

mymod.fit(x_trian, y_train)
```

OS

Workbench

```
from sasviya.ml.linear_model
import LogisticRegression

mymod = LogisticRegression(
    fit_intercept = True,
    max_iter      = None,
    tol           = .0001
)

mymod.fit(x_trian, y_train)
```

WB

Model Governance

The SASCTL package enables users to move deployment artifacts to SAS Model Manager for deployment and model governance. Deployment artifacts include metadata, score code, and model comparison statistics.

Workbench OS APIs

Submit Astore files or DATA Step score code to SAS Model Manager

Pure OS

Submit Python pickle files and R RDA files to SAS Model Manager

SAS Pickle

Convert a SAS model built with SAS to a pickle file

SAS Workbench

This demonstration illustrates how to leverage SAS Viya cloud algorithms from the Python API in SAS Workbench.