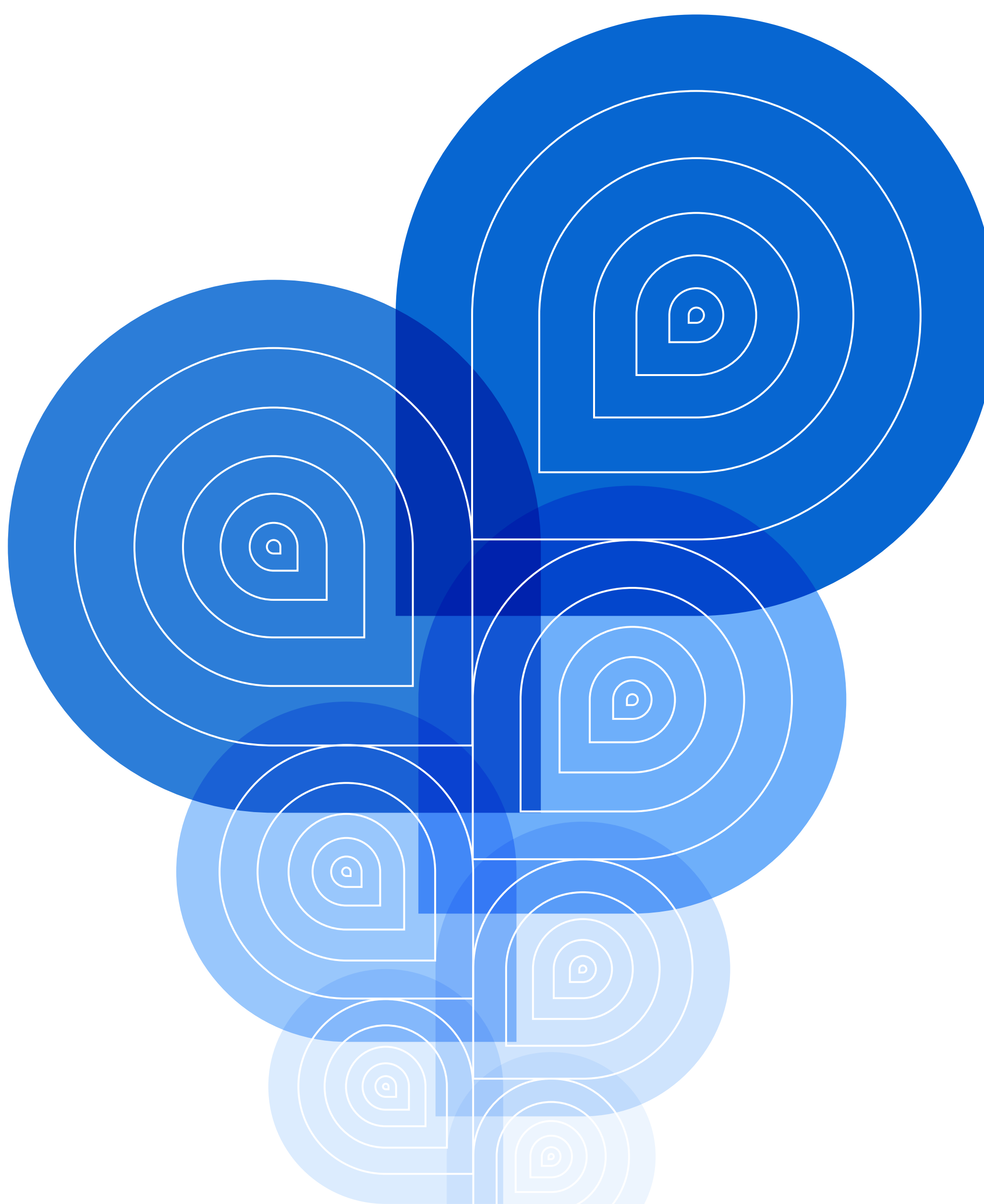


Contents

- 01** Stick with your preferred programming language and coding environment
- 02** Set your own compute power tiers
- 03** Take a point-and-click approach to data prep
- 04** Quickly build and evaluate alternative models
- 05** Many developers, one uniform process
- 06** The race is on
- 07** Resources



More model experiments = more winning models

When most of your experiments fail, how do you avoid failure?

You find a way to run a lot more tests, fast. Identify and set aside failing experiments as soon as their weaknesses become apparent, double down on those that show promise, and keep filling the pipeline with new experiments.

That's what developers and data scientists are grappling with today, to identify winning models and deploy them as quickly as possible in real-world environments.

With access to so much data – some of which is tremendously valuable when refined into insights – they're facing pressure to find more and better ways to put it to work. **It's all about more effective model development and management.**

AI is a big part of the answer. But the processes, tools and infrastructure your organization builds to support AI capabilities are just as important. That's where [SAS® Viya® Workbench](#) comes in.

“AWS customers can quickly develop new models using [SAS® Viya® Workbench](#), which integrates with their existing AWS data infrastructure. The platform enables cross-language collaboration between developers using Python, R and SAS.”

-Doug Mbaya, Senior Partner Solutions Architect, AWS

01

Stick with your preferred programming language and coding environment

Are the developers on your team using R? Python? SAS? In Viya Workbench, it doesn't matter. Users simply import their code – in whatever language and environment they want to use – and run it with very minimal modifications. Or they can use existing code and templates (with clear step-by-step instructions and sample code) that come with Viya Workbench to get new ideas for methodologies they may not have considered.

When you're working in Viya Workbench you can run models using the language of your choice without having to learn new coding skills or functions. Instead, you just call up familiar coding environments such as Visual Studio (VS) Code, JupyterLab/ Notebook, or SAS Enterprise Guide (EG). For those who are comfortable working with multiple codes, simply choose the language, code or function that will work best for the task at hand, or test approaches in different languages.

Working in a multi-language environment using Viya Workbench offers development teams **several critical advantages:**

Choose the best tool for the job

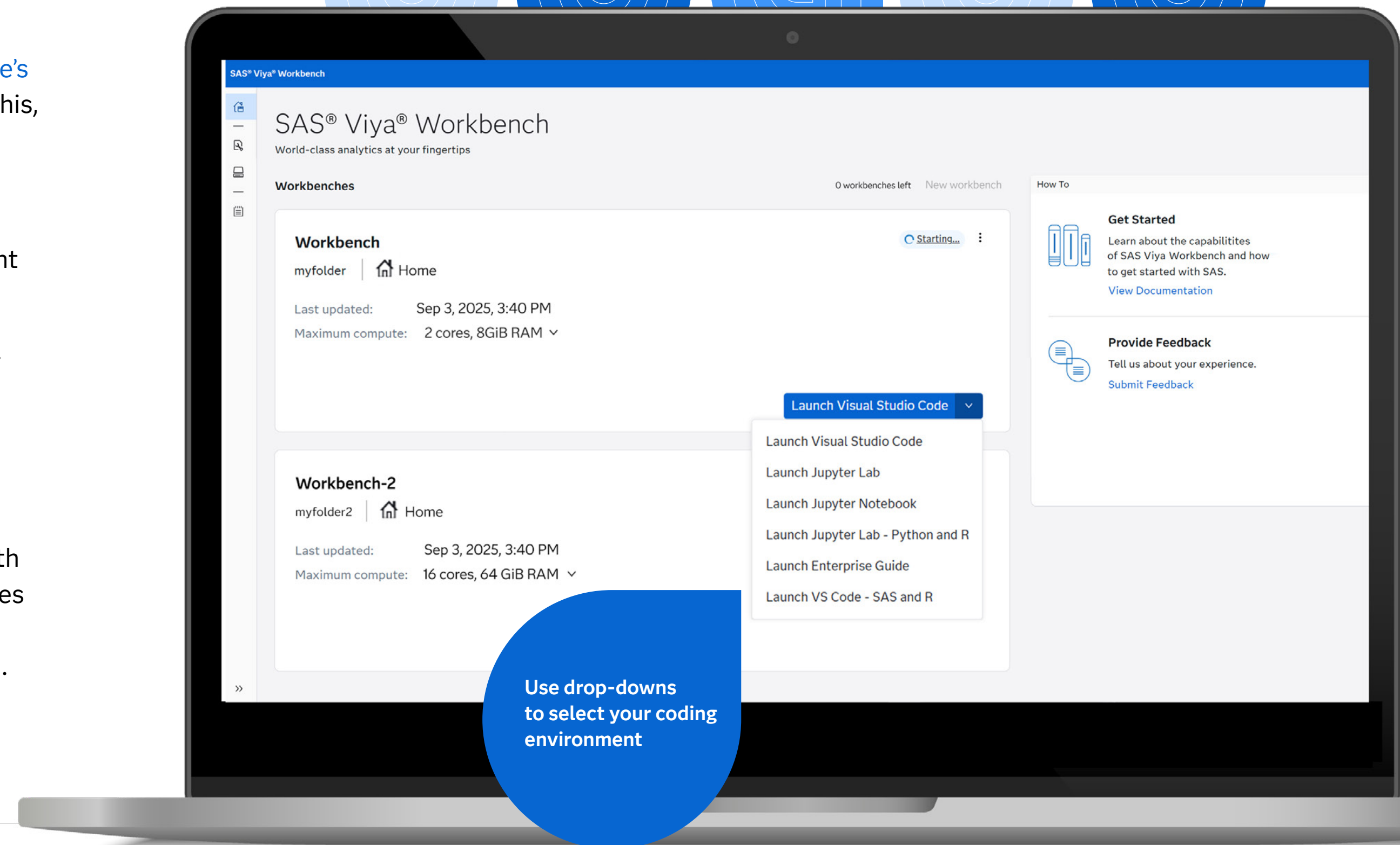
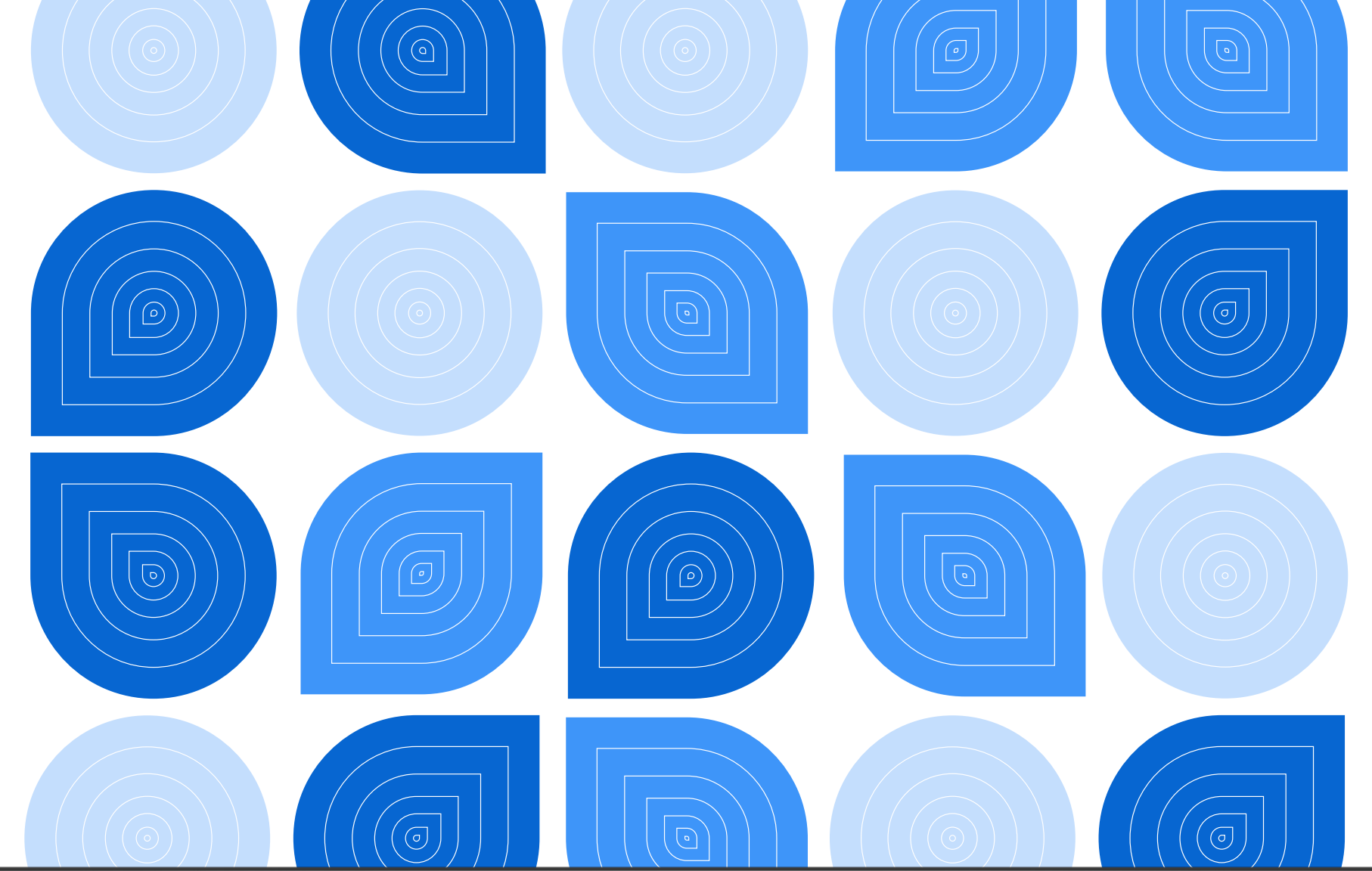
Apply the unique strengths of different coding languages in different parts of the pipeline. [Here's one example](#) of how Viya Workbench handles this, using Python ML pipelines with scikit-learn.

Bring unique skill sets together

Viya Workbench helps developers from different language backgrounds bring their expertise to create more dynamic teams. Each team can communicate easily across different aspects of the pipeline. [Here's how it works](#), drawing from developers skilled in R and SAS.

Embrace evolving tech trends

Each language is being constantly improved with new innovations. By including multiple languages together, teams can make sure they stay competitive and up to date on every innovation.



02

Set your own compute power tiers

Launching and running models at scale is a compute-intensive activity – and costly. Of course, developers who are training and evaluating models can always ask their IT administrators to scale their power resources up or down depending on the model needs, but that can take weeks or even months of waiting.

With Viya Workbench, you configure compute for yourself. Need a small instance for quick testing? Or GPUs for heavy ML training?

Use drop-down menus to adjust cores, memory and GPUs instantly – without IT involvement. Sessions also self-terminate and can be resumed later where you left off. This approach gives you the speed and autonomy you want, while administrators have the ability to set and manage safe boundaries through pre-set tiers.



How it works

Control plane

The control plane runs in an AWS account. It manages tenant integrity, provisioning, authentication and administration.

Data plane

The data plane is deployed in your AWS account/VPC. This is where workbenches run. Data stays here, too – isolated, right-sized and under customer control at all times. So, when you access your workbenches, you're connected directly to your own data plane.

Connection

The control plane and data plane communicate securely through AWS PrivateLink, ensuring that data never enters the public internet.



03

Take a point-and-click approach to data prep

Data preparation is one of the most time-consuming aspects of model development, which leads to bloated model development and iteration cycles. [Smart coding](#) can help – but what if you significantly improve your speed and efficiency? Or enlist more members of your team to assist with data prep?

Using SAS Enterprise Guide integration with Viya Workbench, development teams can use point-and-click data preparation tools that don't require coding – quick, easy, intuitive and reliable. Developers responsible for data cleansing and prep love this approach for the time it shaves from their iteration cycles.



Try the Workbench approach to data prep on your own

We've created a repository of Python notebooks and SAS programs that show how proper data workflows run in Viya Workbench, along with the data required to run them. Just start Viya Workbench and clone the examples repository into the root of your workspace. [Here's how.](#)



Can you use synthetic data in Viya Workbench? Absolutely.

Want to experiment and build models in Viya Workbench without exposing sensitive data? [Here's how to do it](#) using generative adversarial networks (GANs) to create synthetic data that mimics real datasets – with or without Workbench.



04

Quickly build and evaluate alternative models

How long does it take to train and evaluate your models today? Validating, comparing, scoring and preparing models for deployment, which used to take weeks or even months, can now be done in seconds, minutes or hours.

That's a huge difference in the timeline for developing a single model, but at large scale it's transformative – more models and more iterations in much less time, for innovation at light speed.

That's what Viya Workbench is built to do, delivering production-ready, validated results exponentially faster than standard processes.

The secret: combining fast-provisioning, multi-language modeling, easy data preparation, and smart experiment management.

See this video for a more detailed view of how models are built in Viya Workbench.



The synthetic data advantage

Synthetic data capabilities can take things even further by generating realistic datasets on demand, allowing you to jump into modeling faster than before.

Here's how it works in practice

- Use synthetic data to [retain the statistical characteristics of real data](#) without including personally identifiable information.
- Deploy tabular GANs to generate synthetic datasets that preserve business logic, correlations and distributions – enabling models trained on synthetic data to perform close to those built on real data. [Use this checklist](#) to assess your synthetic data.
- Spin up multiple variant pipelines in parallel on synthetic data without generating risk exposure to production or compliance requirements.

05

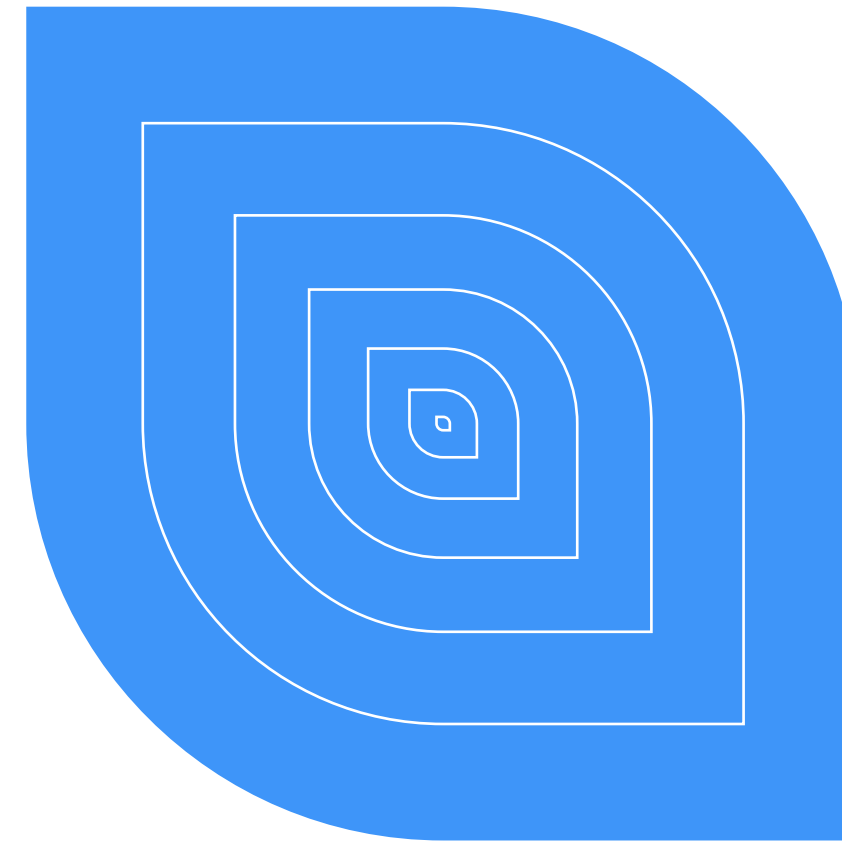
Many developers, one uniform process

What happens when your Python coder is out, and an R developer needs to continue the work? Don't worry – anyone can jump in and begin sharing, versioning and reproducing any developer's model in Viya Workbench regardless of the language used to build it.

And users don't have to fully understand the code a model was written in to get it to run correctly – it works seamlessly. That's part of the beauty of managing model development through a single platform like Viya Workbench. **It just works.**

Plus, through native [Git integration](#) and [CI/CD workflows](#), it's easy to archive, share and continue development on your project with version control. In fact, multi-language teams benefit from reproducibility – when a model developed in Python and another developed in SAS both yield identical results, the resulting consistency provides confidence in your processes.

To put this approach to work and see how it works for yourself, clone the [Viya Workbench examples repository](#) and follow the [Getting Started guide](#). Then you can practice branching, committing and collaborating across codebases.



06

The race is on

Model development is already a competitive differentiator – organizations that are best prepared to quickly develop and deploy successful models have the edge over those that don't. And AI is only going to raise the stakes.

Viya Workbench provides a super-efficient, uncomplicated approach to quickly building smarter, more secure models:

Spin up

a Viya Workbench session in the cloud – in seconds

Configure

server size to match your workload requirements, choosing your cores, memory and GPUs

Minimize

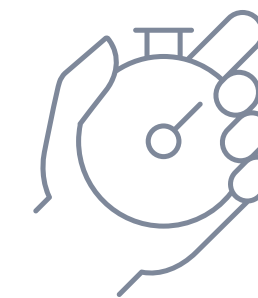
cloud computing costs with self-terminating sessions that resume automatically where you left off

Take advantage

of fast point-and-click self-provisioning that requires minimal IT support

That's how you beat your competitors and shorten the cycle from idea to experiment to production.

See for yourself, using real data in these two use cases from SAS data scientists.



Building a snowboarding dashboard to track performance

In this project, a snowboarder strapped on telemetry sensors, pulled the data into Python, and modeled his performance in Viya Workbench. You can do the same, using the original data or swapping in your own fitness data. [Just clone the data repository](#) and open it in Viya Workbench.



Predicting *Survivor* winners with 47 seasons of data

This project is a great tool for experimenting with pipelines in Viya Workbench. It uses scikit-learn to forecast the outcomes of this reality show pioneer – you can rerun the exact analysis yourself. [Run the notebook](#) and tweak the feature set, add synthetic data, or test new algorithms in Viya Workbench.

07

Resources

[SAS Viya Workbench Documentation](#)

Overview, onboarding, user guides and APIs

[Administrator's Guide](#)

Compute allocation, GPU scaling and workspace management

[Run Example SAS Programs](#)

Sample SAS code packaged into Viya Workbench

[Viya Workbench Demo](#)

Showcases the experience for Python coders

[Viya Workbench Examples on GitHub](#)

Canonical repository and reproducible workflows

[Viya Workbench Demos on GitHub](#)

Industry projects for banking, healthcare and more

[Viya Workbench "Getting Started" Community Hub](#)

Step-by-step tutorials, workplace setup, Git integration

[SAS Developer Site](#)

Developer-focused content and use cases

[SAS Support Page](#)

Troubleshooting and articles

Technical blogs

[Using Synthetic Data to Bridge Production and Development](#)

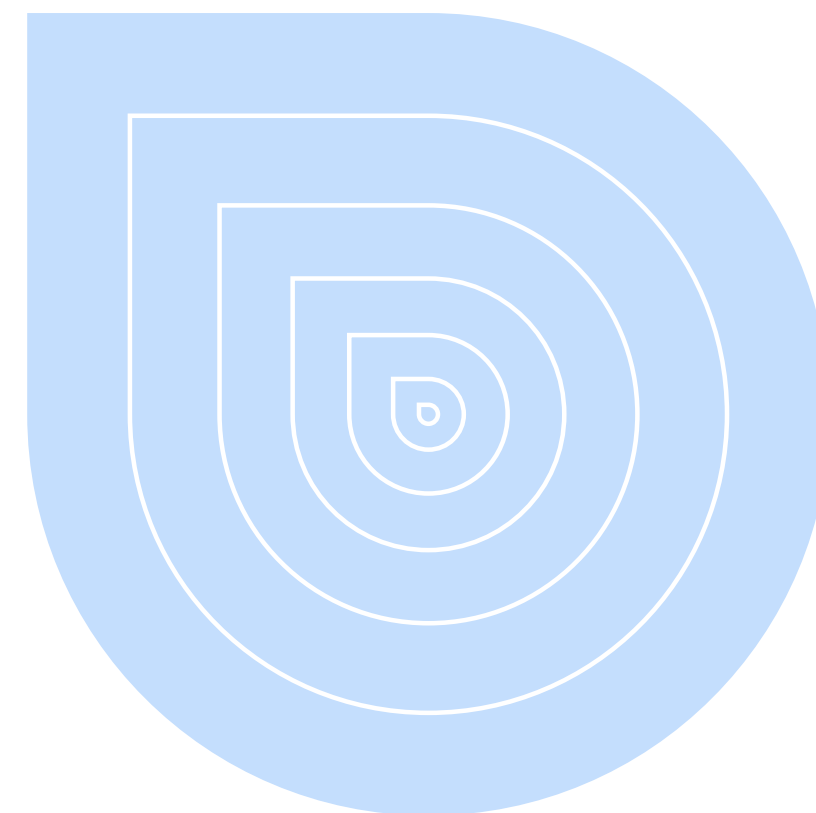
[From Slopes to Stats Telemetry and Python Analysis](#)

[Predicting Survivor Winners with ML Pipelines](#)

[Python ML Pipelines with Scikit-Learn](#)

[Build Modern Workflows with GitHub Actions](#)

[Building Data Pipelines with R and SAS](#)





SAS and AWS together deliver powerful analytics capabilities designed for enterprise security and scale. Whether you're on-prem, cloud-curious or navigating compliance across hybrid environments, SAS Viya on AWS gives you the flexibility to modernize at your pace. Our joint solutions empower organizations – from public sector to life sciences – to unlock real-time insights, accelerate AI adoption, and meet regulatory demands with confidence. SAS brings deep analytics expertise; AWS brings proven cloud agility. Together, we're built for what's next.

To learn more, visit sas.com/en_us/software/viya/workbench.html



To contact your local SAS office, please visit: sas.com/offices