

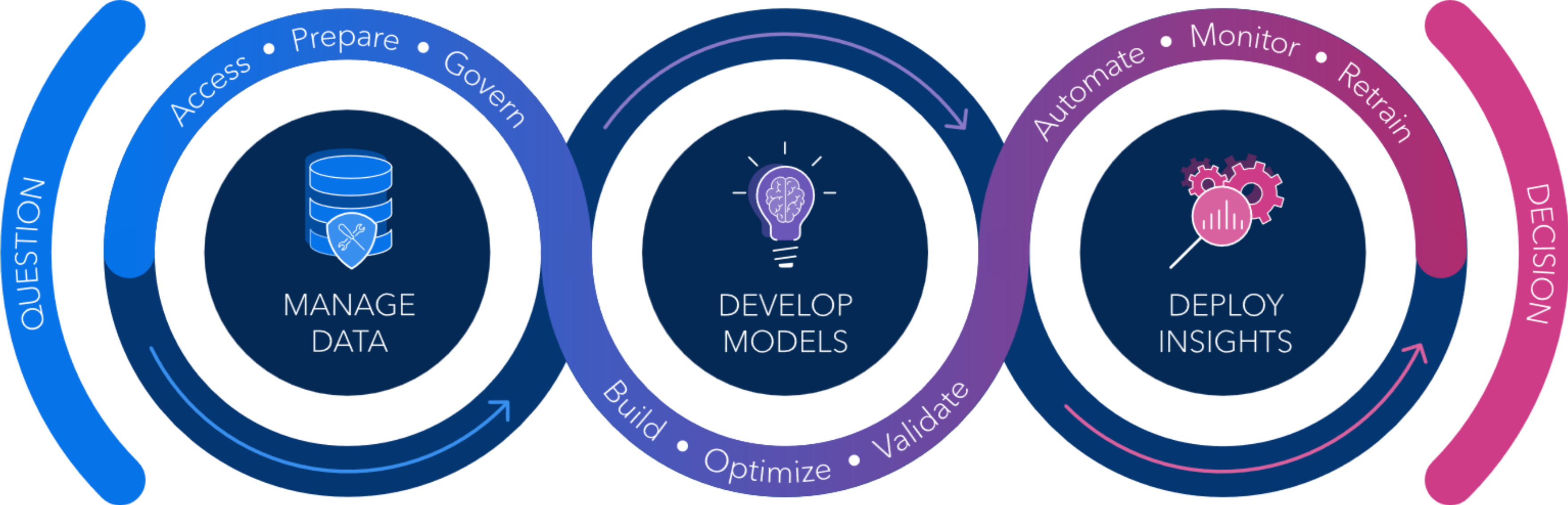
How Do I Manage and Deploy SAS & Open-Source Models with SAS Viya?

Ask The Experts

Jonathon Butow
Yi Jian Ching



SAS Viya – Support entire Analytics Lifecycle

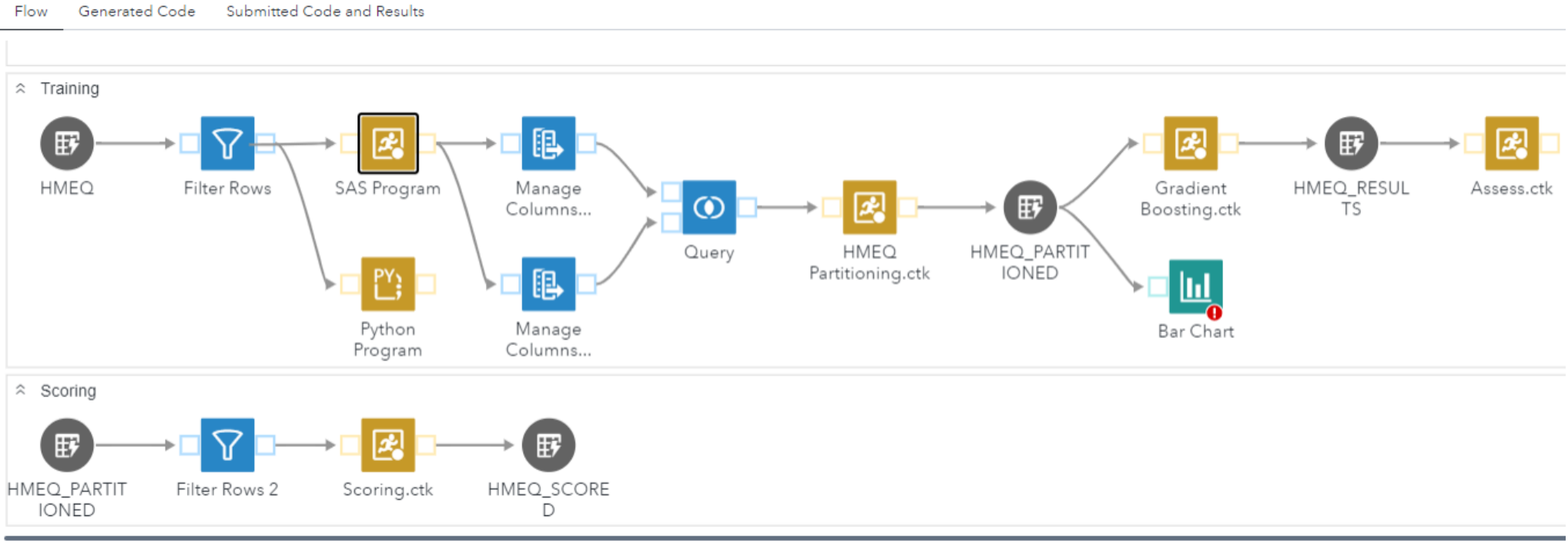
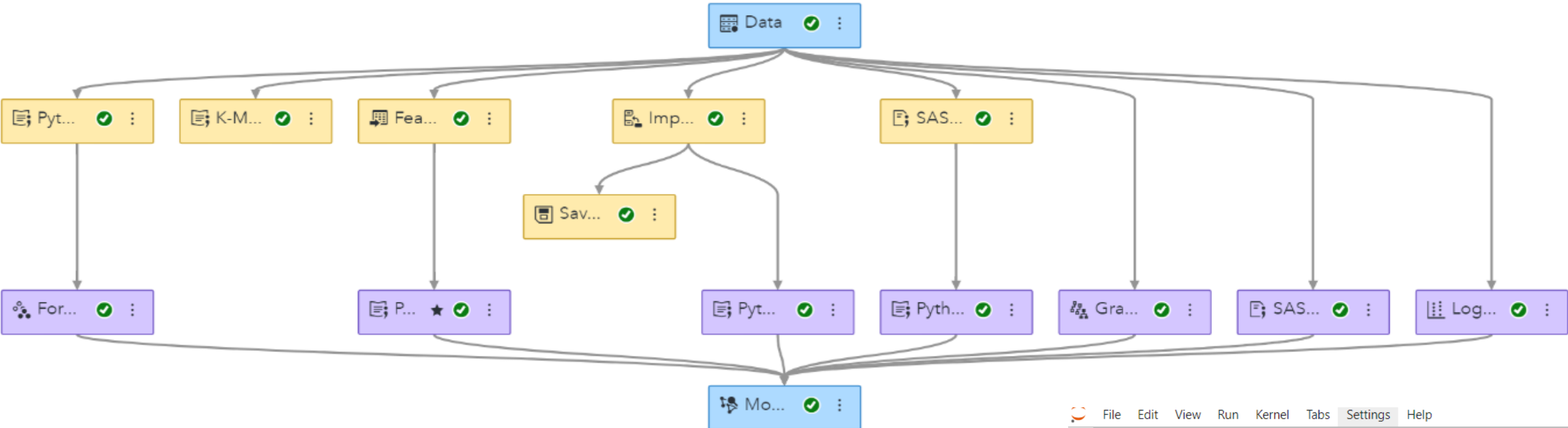


Model Development

SAS & Open-Source

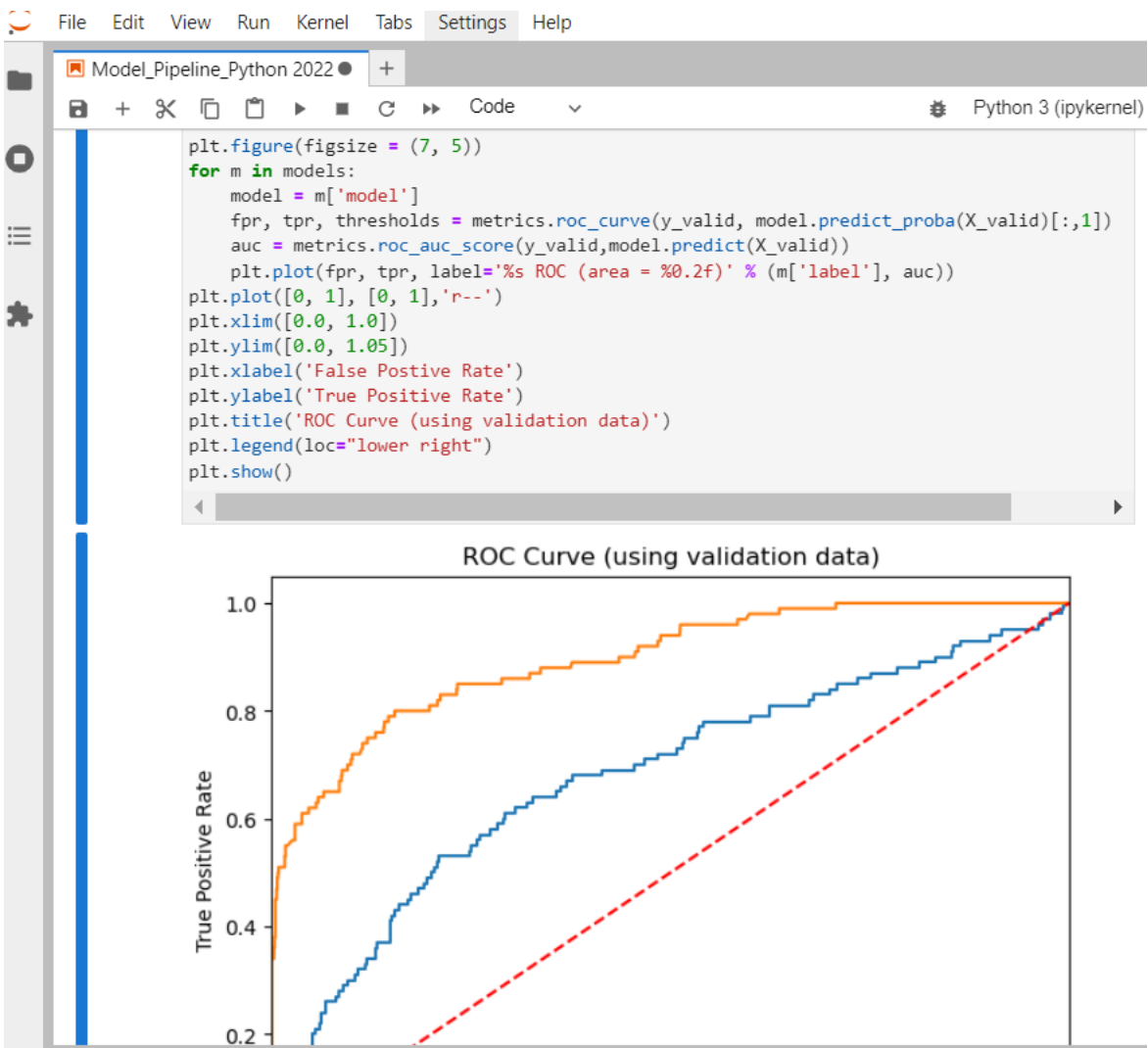


Flexibility in Interfaces



SAS Program

Code	Node	Notes
1	DATA	&_output1;
2	SET	&_input1;
3		ID = _N_;
4	RUN;	



GUI - SAS Model Studio

Run SAS & Open-source model pipelines:

- Create multi-language pipelines between SAS, Python & R
- AI generated diagnostics for SAS & open-source models
- Open APIs to access SAS algorithms in Python & R

The screenshot displays the SAS Model Studio interface. At the top, a pipeline diagram shows a 'Data' node (blue) with a green checkmark, which branches into 'Imputation' (yellow) and 'SAS Code' (yellow), both also with green checkmarks. Below these, three model nodes are shown: 'Python GradientBoo...' (purple), 'R RandomFo...' (purple), and 'Python LogisticRegre...' (purple), all with green checkmarks. A 'Run pipeline' button is visible in the top right. To the right of the pipeline is a configuration panel for the 'Python LogisticRegre...' node, showing a description, an 'Open code editor' button, a language dropdown set to 'Python', and several checkboxes for model options.

Below the pipeline diagram, a 'Cumulative Lift' plot is shown. The plot displays three curves for the VALIDATE, TRAIN, and TEST partitions. The VALIDATE partition has a Cumulative Lift of 5.9, the TRAIN partition has 8.11, and the TEST partition has 5.76. The plot shows that the model's performance is significantly better than random selection, as the cumulative lift values are much greater than 1.

Below the plot, a 'Results' window is open, showing 'Assessment' and 'Fairness and Bias' tabs. The 'ROC Reports' section displays a Receiver Operating Characteristic (ROC) curve. The x-axis is labeled '1 - Specificity' and the y-axis is labeled 'Sensitivity'. The curve is significantly above the diagonal line (x=y), indicating good model performance. The 'Event Classification' section shows a 'Percentage Plot' with a 'Cutoff' of 0.07 and a 'Percent' of 80.

The following text is displayed in the results window:

The VALIDATE partition has a Cumulative Lift of 5.9 in the 10% quantile (depth of 10) meaning there are 5.9 times more events in the first two quantiles than expected by random (10% of the total number of events). Because this value is greater than 1, it is better to use your model to identify responders than no model, based on the selected partition.

The TRAIN partition has a Cumulative Lift of 8.11 in the 10% quantile (depth of 10) meaning there are 8.11 times more events in the first two quantiles than expected by random (10% of the total number of events). Because this value is greater than 1, it is better to use your model to identify responders than no model, based on the selected partition.

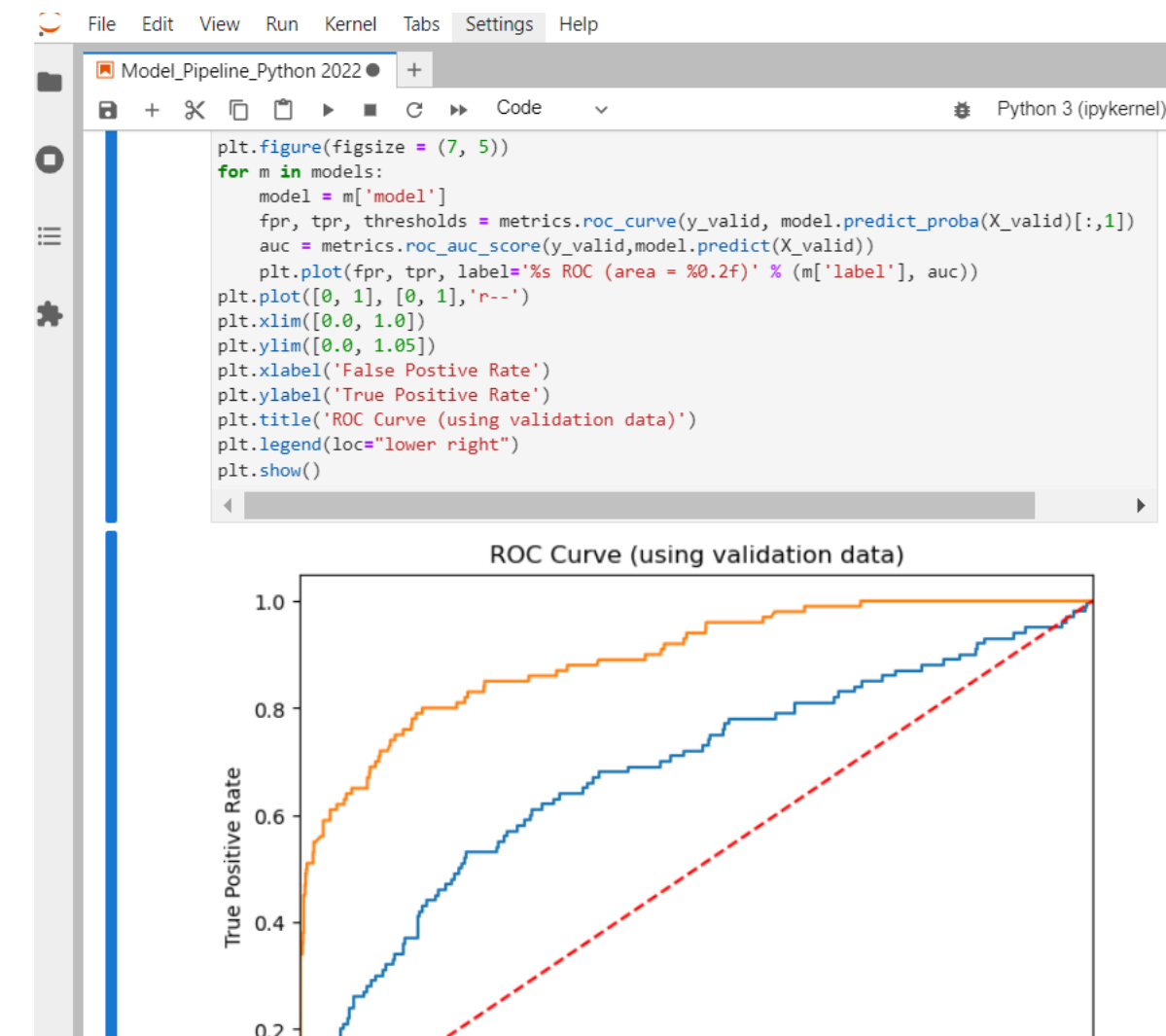
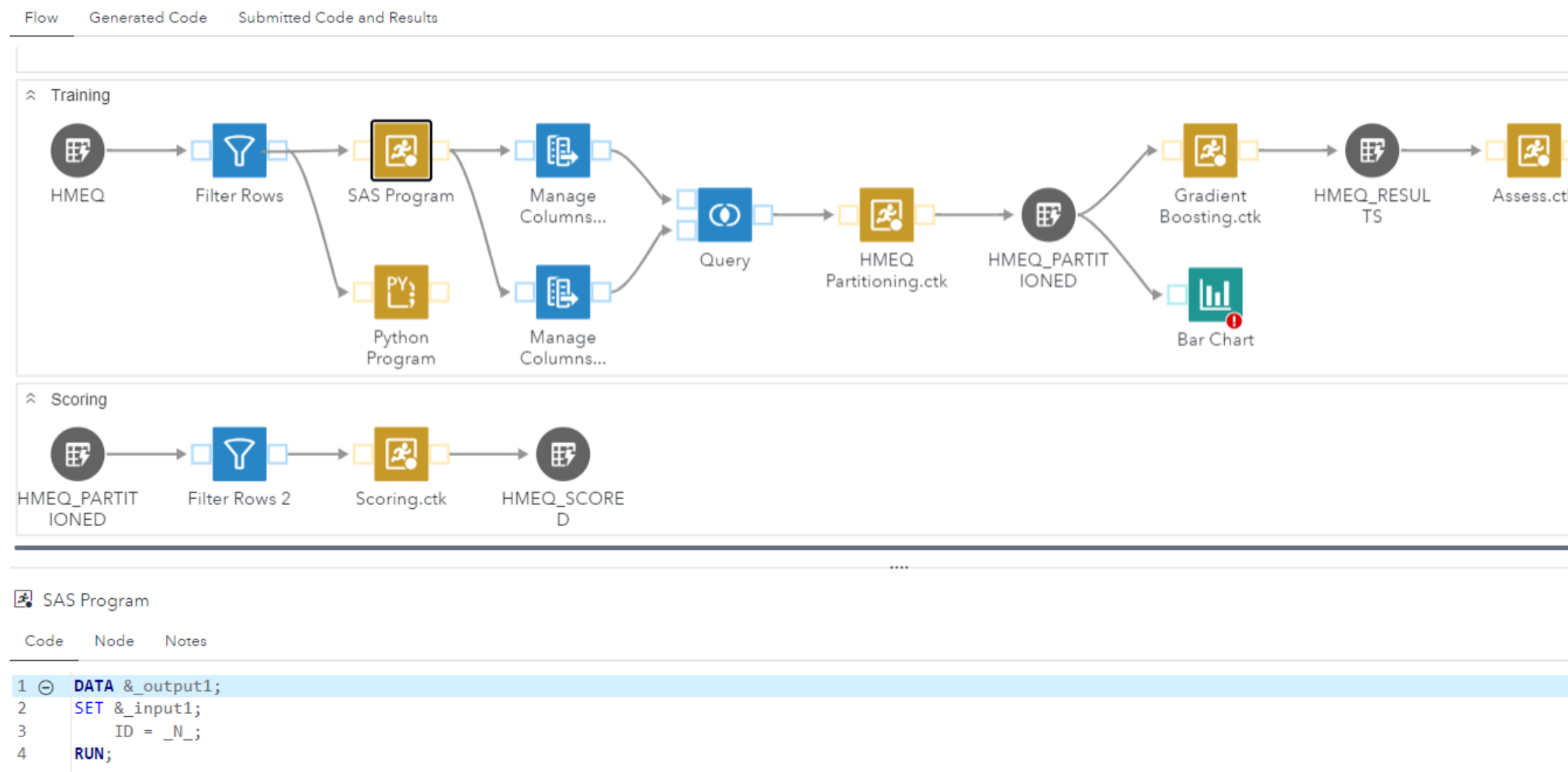
The TEST partition has a Cumulative Lift of 5.76 in the 10% quantile (depth of 10) meaning there are 5.76 times more events in the first two quantiles than expected by random (10% of the total number of events). Because this value is greater than 1, it is better to use your model to identify responders than no model, based on the selected partition.

Cumulative lift is calculated by sorting each partition in descending order by the predicted probability of the target event P_Default1, which represents the predicted probability of the event "1" for the target Default. The data is divided into 20 quantiles (demi-deciles, with 5% of the data in each), and the number of events in each quantile is computed. The cumulative lift for a particular quantile is the ratio of the number of events across all quantiles up to and including the current quantile to the number of events that would be there at random, or equivalently, the ratio of the cumulative response percentage to the baseline response percentage. The cumulative lift at depth 10 includes the top 10% of the data, which is the first 2 quantiles, which would have 10% of the events at random. Thus, cumulative lift measures how much more likely it is to observe an event in the quantiles than by selecting observations at random.

IDEs - SAS & Open-Source

Script in SAS & Open-Source:

- Ability to use IDE of choice for user flexibility
- Model and write in language of choice
- Register models through APIs



Model Management

SAS & Open-Source



SAS Model Manager

Govern and deploy all models

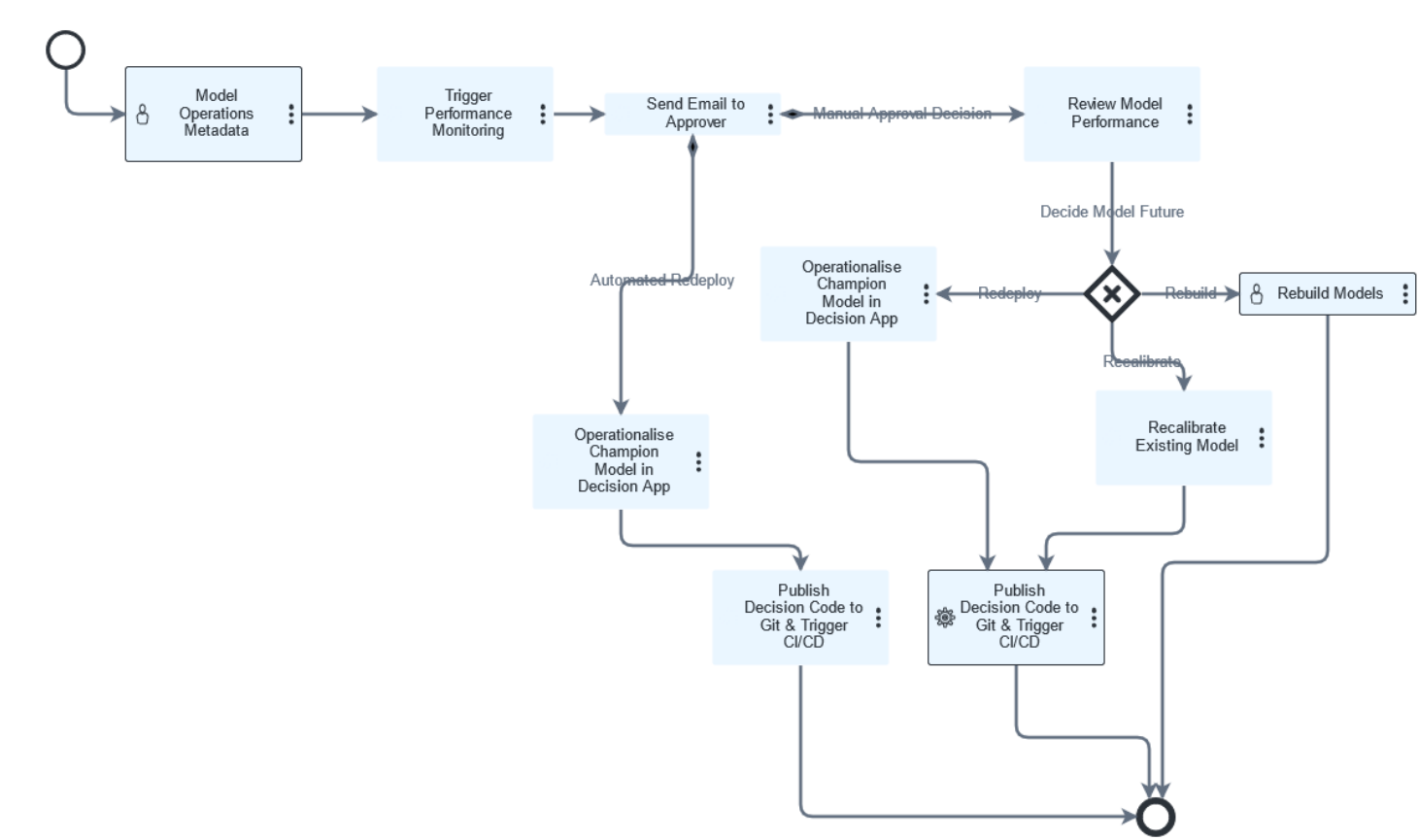
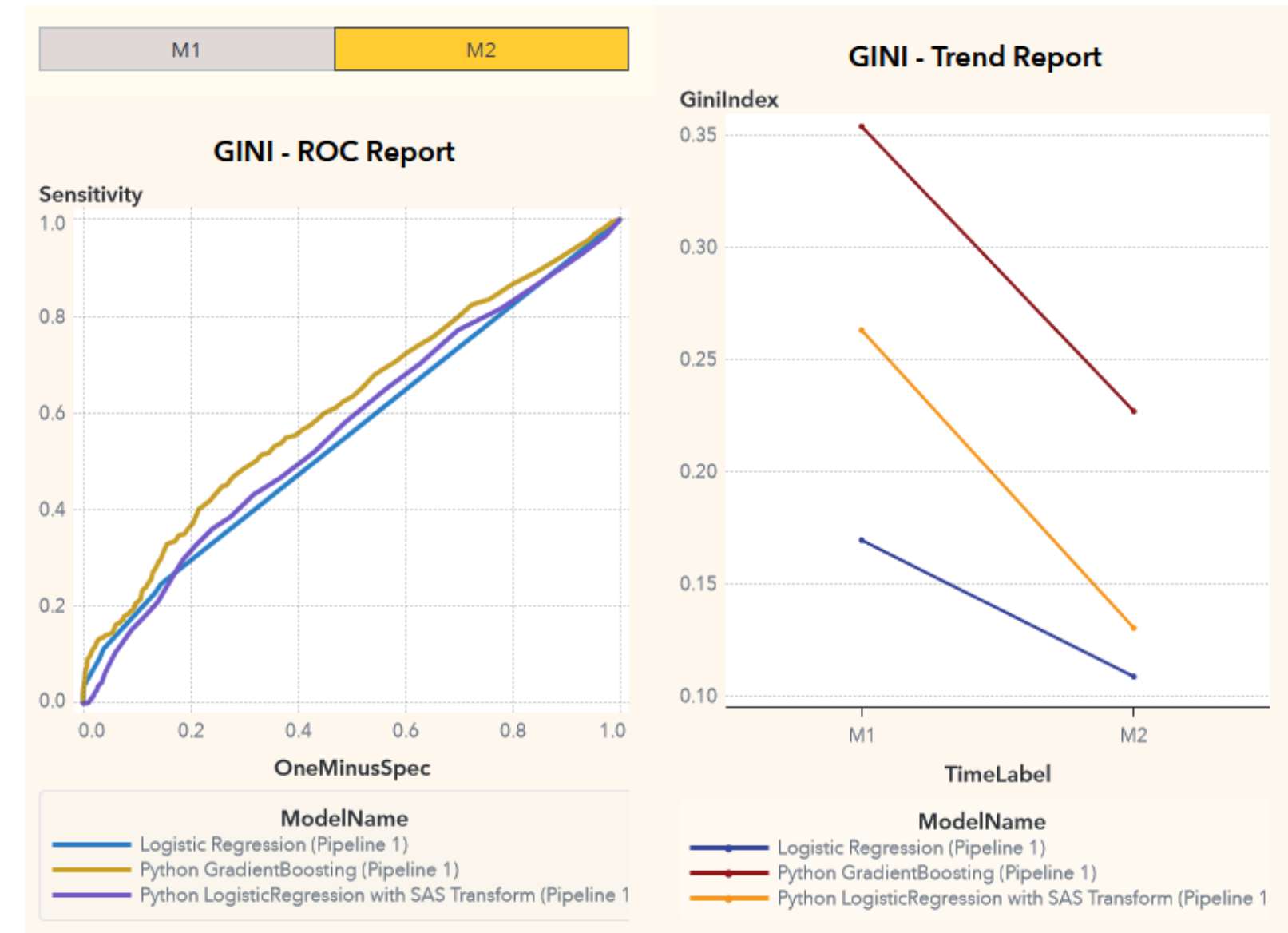
The screenshot displays the SAS Model Manager interface for managing models. The top navigation bar includes a menu icon, the title "SAS Model Manager - Manage Models", and search, notification, and help icons. A left sidebar contains navigation options: Home, Models, Projects (selected), Deployments, and Tasks. The main content area features three dashboards: 1. A card showing "33 Total Number of Projects" with a progress bar at 42% and "14 Projects with Published Models". 2. A donut chart titled "Projects per Operational Status" showing 33 projects, with a legend for Deployed (yellow), In production (purple), and Prototype (blue). 3. A donut chart titled "Projects per Model Function" showing 33 projects, with a legend for Classification (blue), Not specified (orange), Prediction (yellow), and Text categories (purple). Below the dashboards is a table with columns: Name, Published Status, Model Function, Modified By, Date Modified, and Tags. The table lists 14 projects, all of which are "Classification" models. The "Published Status" column shows green circles for published models and a red circle for "Credit Risk - CS_ACCEPTS". The "Tags" column contains labels like "Models published", "Performance results available", "Active workflow", and "Perform".

Name	Published Status	Model Function	Modified By	Date Modified	Tags
<input type="checkbox"/> BBLDEMO	●	Classification	P Piotr.Kaczynski@sas.com	Jul 14, 2023 06:14 PM	Models published Performance results availab
<input type="checkbox"/> Collections_B0	●	Classification	X XinRu.Lee@sas.com	May 24, 2023 07:44 PM	Models published
<input type="checkbox"/> Collections_B1	●	Classification	X XinRu.Lee@sas.com	May 24, 2023 07:44 PM	Models published
<input type="checkbox"/> Collections_B2	●	Classification	X XinRu.Lee@sas.com	May 24, 2023 07:44 PM	Models published
<input type="checkbox"/> Collections_B3	●	Classification	X XinRu.Lee@sas.com	May 24, 2023 07:44 PM	Models published
<input type="checkbox"/> Collections_Optimization1(1)		Classification	X XinRu.Lee@sas.com	May 24, 2023 07:44 PM	
<input type="checkbox"/> Collections_Optimization1(3)		Classification	X XinRu.Lee@sas.com	May 24, 2023 07:44 PM	
<input type="checkbox"/> Collections_Optimization1(4)		Classification	X XinRu.Lee@sas.com	May 24, 2023 07:44 PM	
<input type="checkbox"/> Collections_Recovery_B4	●	Classification	X XinRu.Lee@sas.com	May 24, 2023 07:44 PM	Models published
<input type="checkbox"/> Credit Risk - CS_ACCEPTS	●	Classification	T Thanapon.Treeertkul@sas.com	Sep 1, 2023 01:57 PM	Active workflow Models published Perform
<input type="checkbox"/> Credit Risk Modeling	●	Classification	P peisin.khoo@sas.com	Aug 22, 2023 06:43 PM	Models published Performance results availab
<input type="checkbox"/> Demo test		Classification	Y YiJian.Ching@sas.com	Aug 1, 2023 11:00 AM	
<input type="checkbox"/> FEATIDMM_ResponsiveAI	●	Classification	J Jacky.Long@sas.com	Jun 26, 2023 03:57 PM	Models published
<input type="checkbox"/> HMEQ_TEST		Classification	Y YiJian.Ching@sas.com	Aug 21, 2023 11:50 AM	

Manage SAS & Open-Source Models

Centralised model repository to govern all model types:

- Manage, score and monitor SAS & open-source models
- Deploy into variety of different endpoints and execution types, from batch to API-based containers
- Orchestrate MLOps workflows to evaluate, update and retrain open-source models



Demo

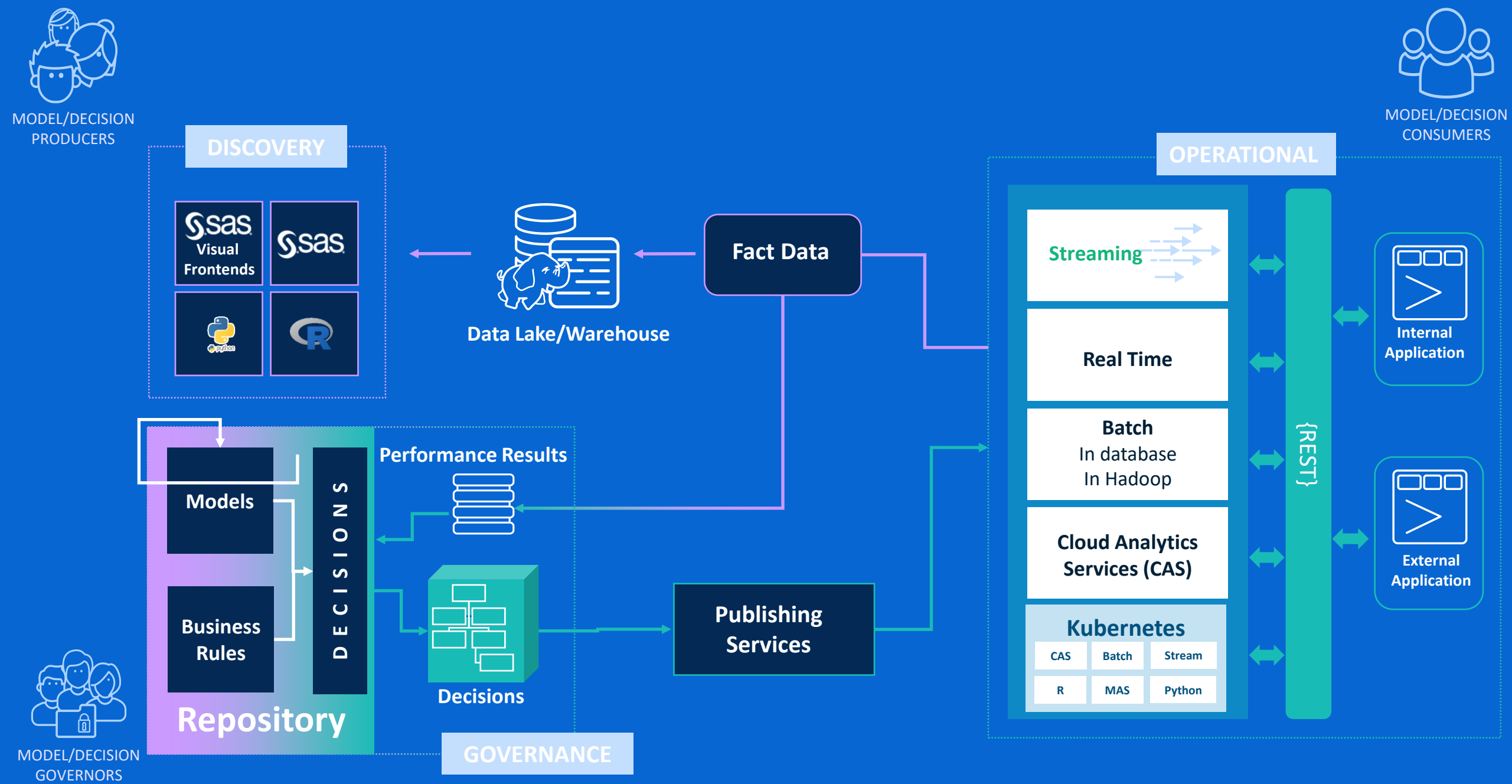
SAS ModelOps Deployment

The last mile?

Jonathan Butow

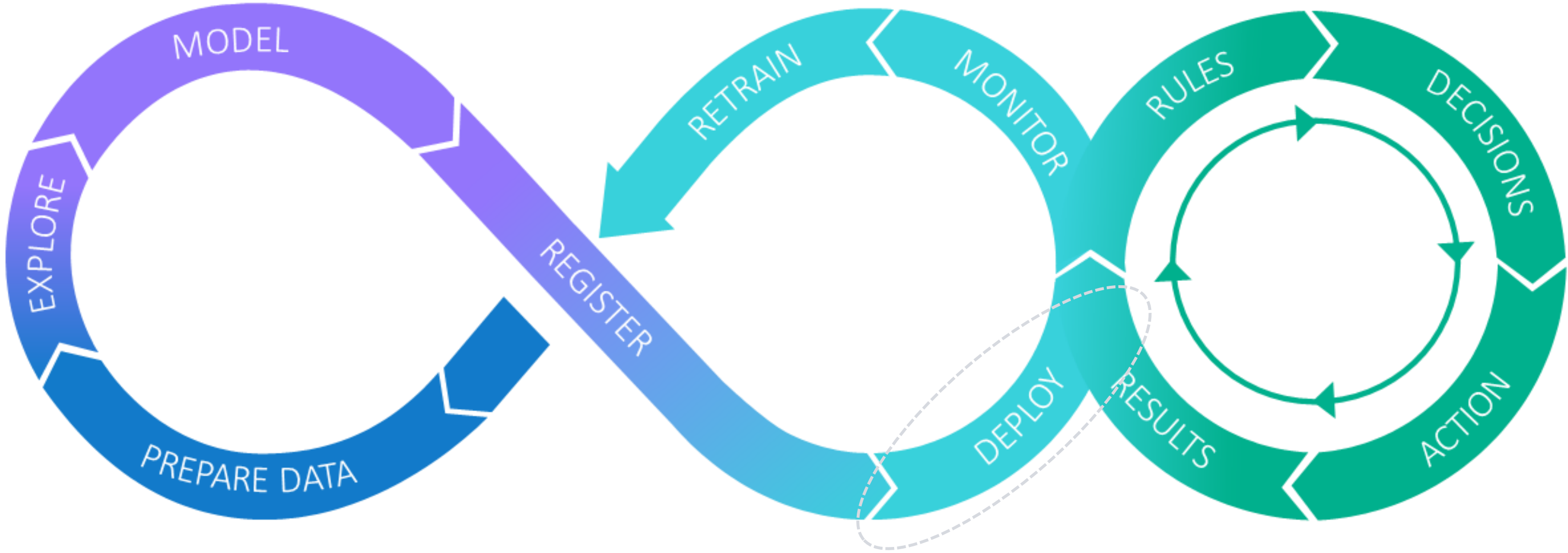


Operational Analytics (Process Flow)



Deployment and Publishing

Term	Definition	Comments
Deploy	Get software up and running in an environment. Can apply to models, jobs, applications or the software of the SAS platform itself.	Deployment is a process involving integration of components and testing. Code and other objects are typically deployed to a staging environment for testing before being deployed to production. Deployment may be automated using pipelines.



ModelOps Deployment

Model Formats

Common modeling languages and the outputs they generate

Execution Runtimes

Engines which are able to run these outputs

Conceptual Architecture

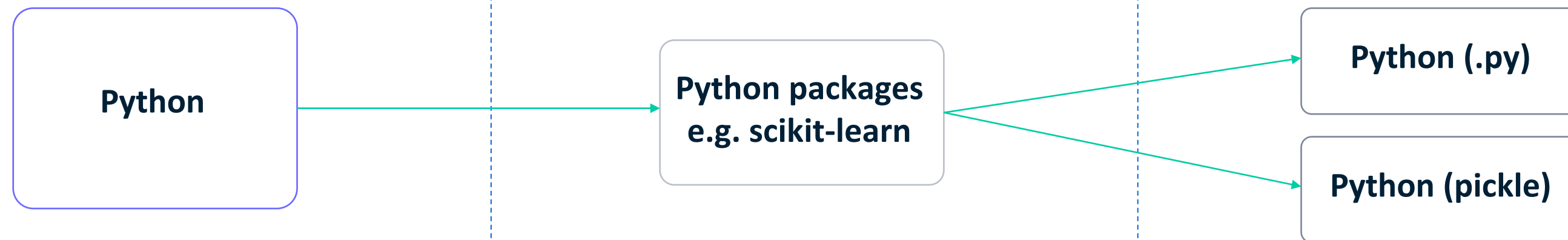
The processes which move models to execution runtimes

Model Formats

Coding Language

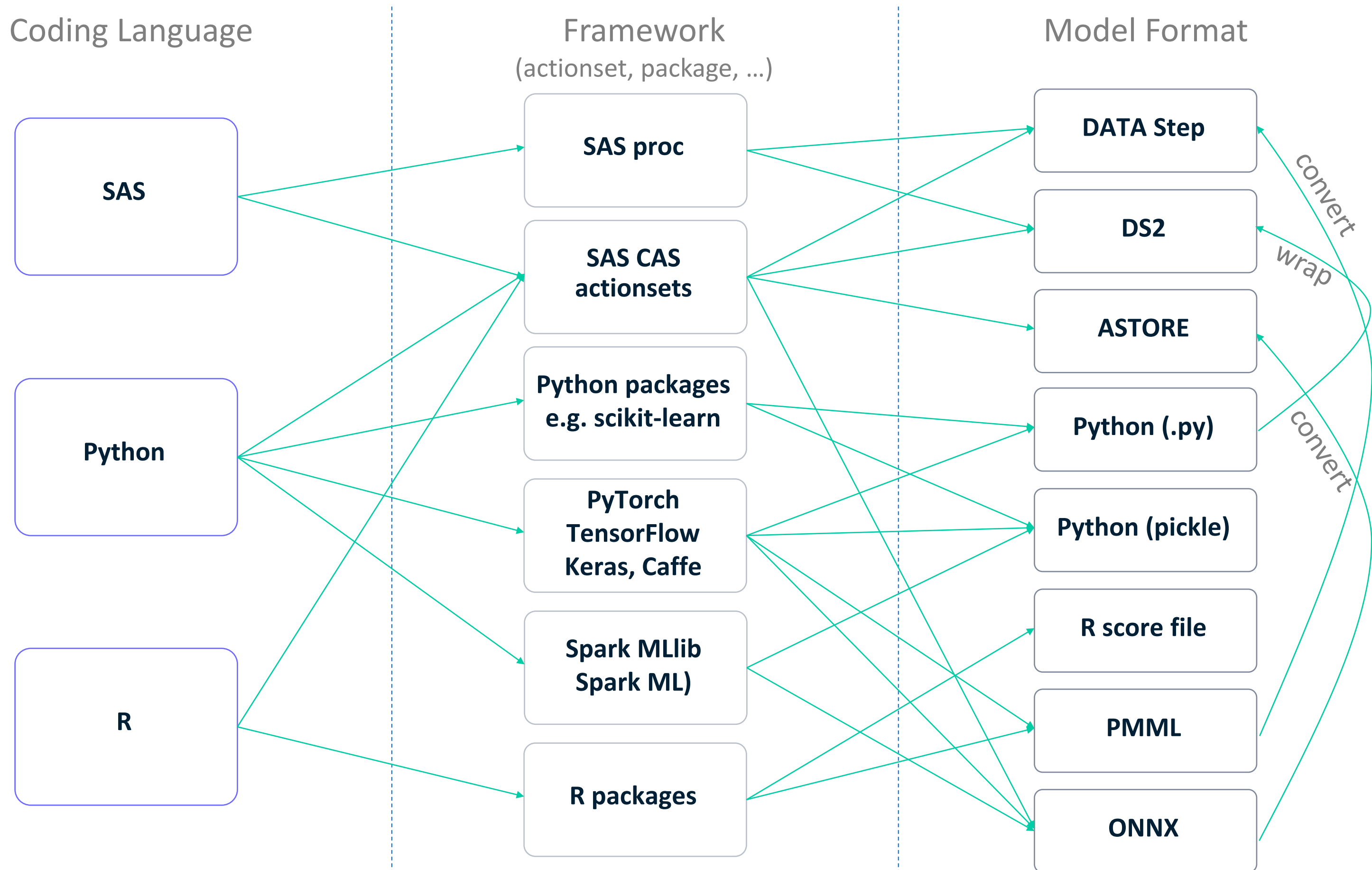
Framework (actionset, package, ...)

Model Format



For scoring:
Model gets deployed to a run-time engine which needs to have the same framework

Which model output format can be generated by which coding language (using which framework)?



Coding Language

SAS

Python

R

Framework (actionset, package, ...)

SAS proc

SAS CAS
actionsets

Python packages
e.g. scikit-learn

PyTorch
TensorFlow
Keras, Caffe

Spark MLlib
Spark ML)

R packages

Model Format

DATA Step

DS2

ASTORE

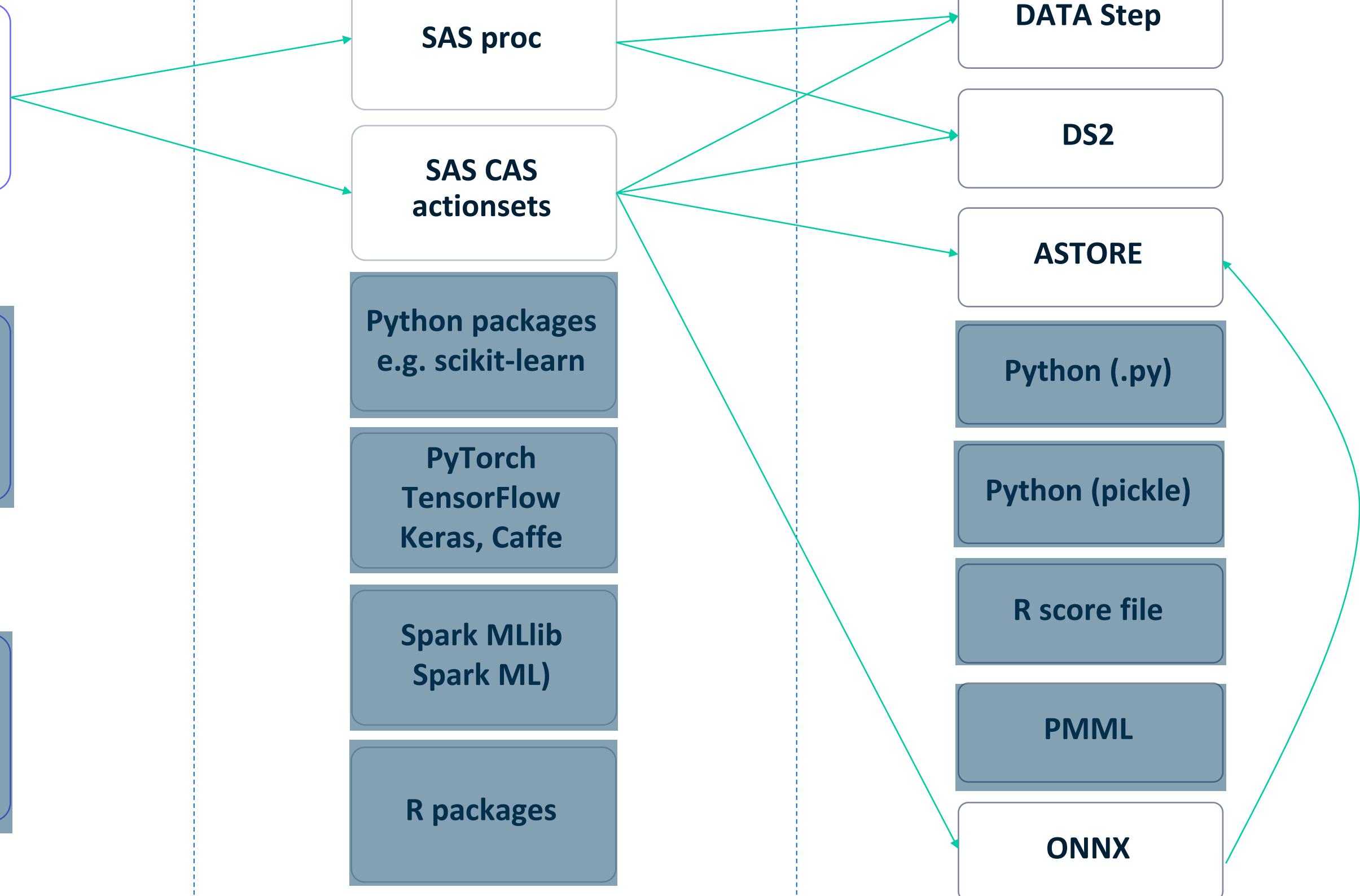
Python (.py)

Python (pickle)

R score file

PMML

ONNX



Coding Language



```
proc forest data=&dset. ntrees=10 minleafsize=5
seed=1859355710;
  target &target. / level=nominal;
  input &nom_input. / level=nominal;
  input &int_input. / level=interval;
  partition rolevar=_partind_(train='0'
  validate='1');
  output out=mycas.ap_scored_forest
  copyvars=( _partind_ &target);
  title "Random Forest";
  savestate rstore=mycas.model_forest
run;
```

Framework (actionset, package, ...)



Decision Tree Action Set

- dtreeCode
- dtreeExportModel
- dtreeMerge
- dtreePrune
- dtreeScore
- dtreeSplit
- dtreeTrain
- forestCode
- forestScore
- forestTrain
- gbtreeCode
- gbtreeScore
- gbtreeTrain

Model Format

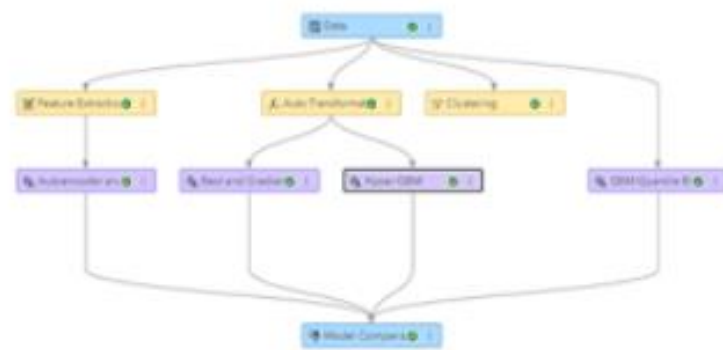


```
SAVESTATE RSTORE=CAS-libref.data-table;
```

Coding Language



SAS visual interfaces interact with the same action sets



Framework (actionset, package, ...)



Decision Tree Action Set

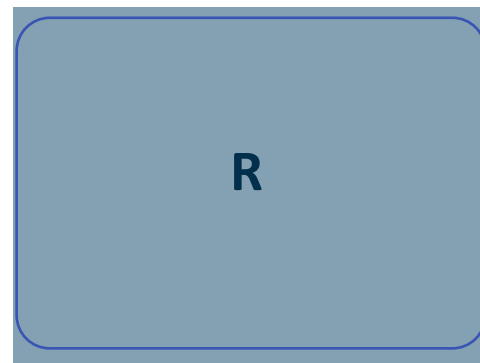
- dtreeCode
- dtreeExportModel
- dtreeMerge
- dtreePrune
- dtreeScore
- dtreeSplit
- dtreeTrain
- forestCode
- forestScore
- forestTrain
- gbtreeCode
- gbtreeScore
- gbtreeTrain

Model Format

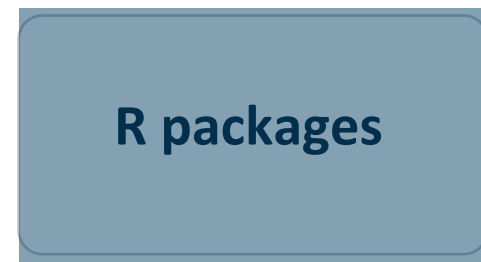
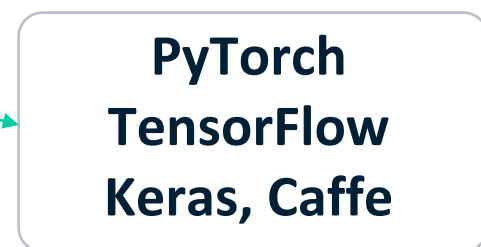
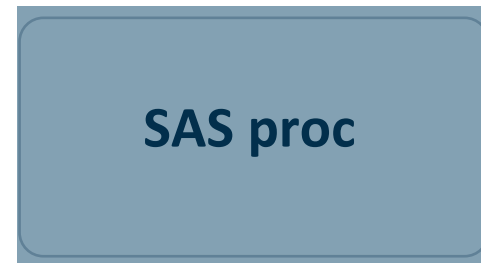


```
SAVESTATE RSTORE=CAS-libref.data-table;
```

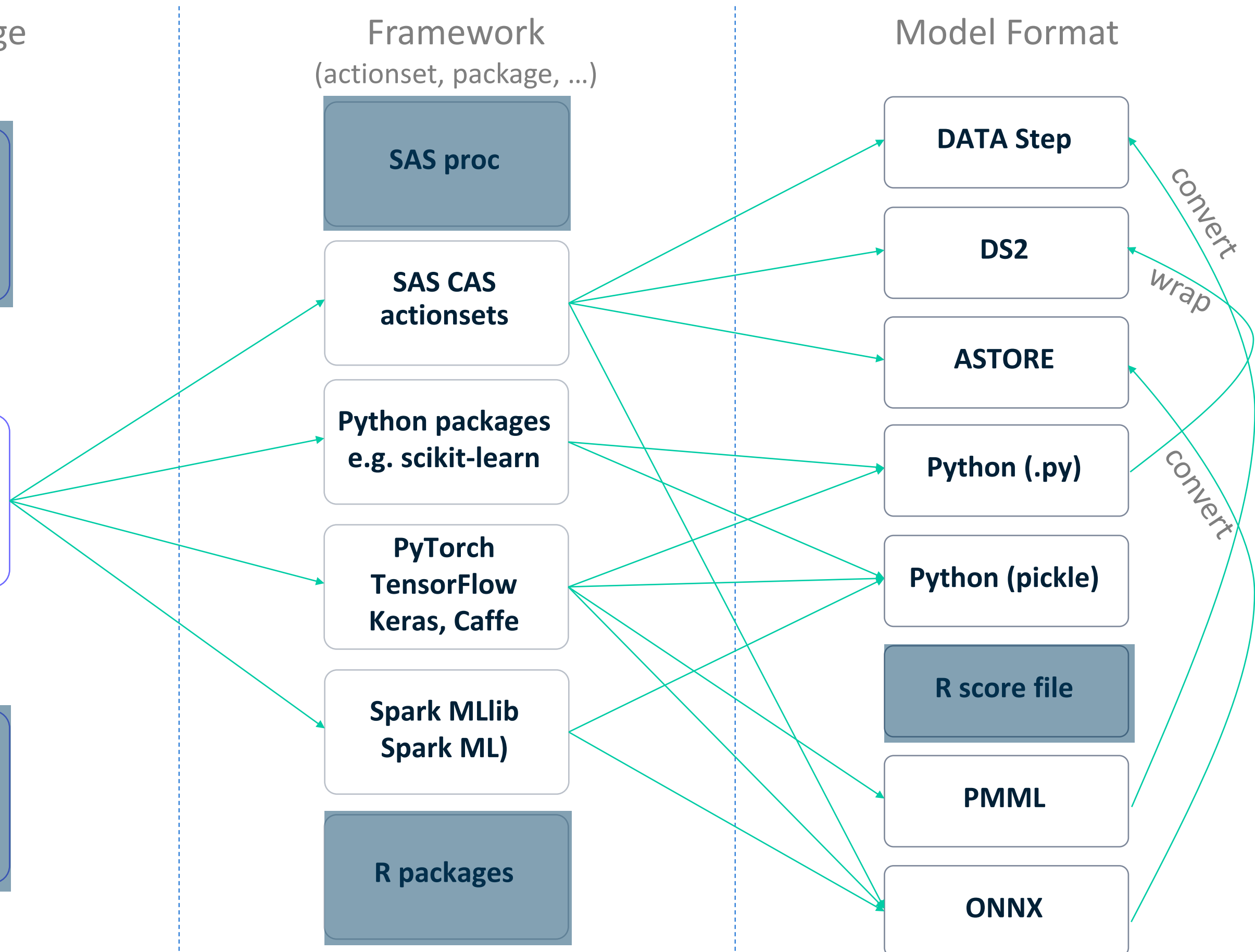
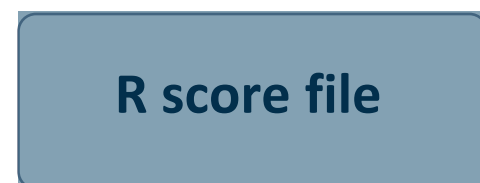
Coding Language



Framework (actionset, package, ...)



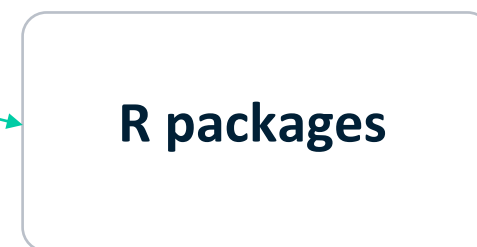
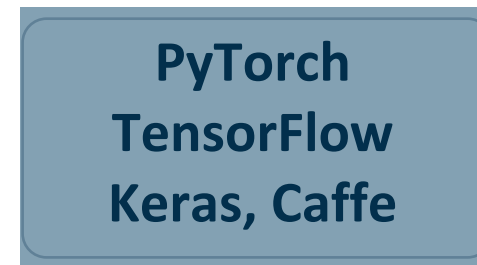
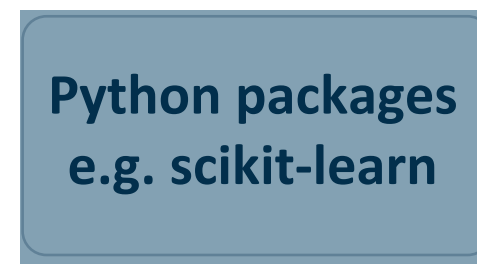
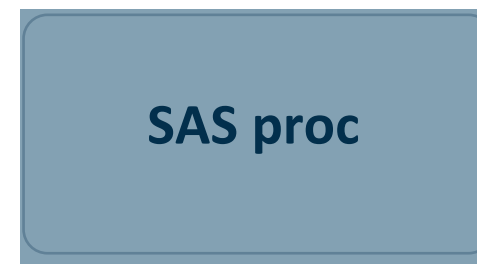
Model Format



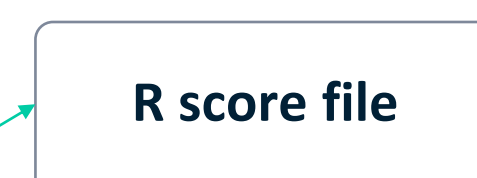
Coding Language



Framework (actionset, package, ...)



Model Format



convert

convert

Execution Engines

Mapping output formats to execution engines

Which engine can ingest which model format?

	Batch					On demand	
	SPRE	CAS	Python / R	In-DB	Spark	MAS	Python / R
DATA step							
DS2							
ASTORE							
Python .py							
Pickle file							
R score file							
PMML							
ONNX							

Mapping output formats to execution engines

Which engine can ingest which model format?

	Batch					On demand	
	SPRE	CAS	Python / R	In-DB	Spark	MAS	Python / R
DATA step				(using EP)	(using EP)		
DS2				(using EP)	(using EP)		
ASTORE				(using EP)	(using EP)		
Python .py		PyMAS				PyMAS	
Pickle file		PyMAS				PyMAS	
R score file							
PMML				Snowflake			
ONNX				Azure Synapse			

Mapping output formats to execution engines

Which engine can ingest which model format?

Common arrangement for batch processing

	Batch					On demand	
	SPRE	CAS	Python / R	In-DB	Spark	MAS	Python / R
DATA step				(using EP)	(using EP)		
DS2				(using EP)	(using EP)		
ASTORE				(using EP)	(using EP)		
Python .py		PyMAS				PyMAS	
Pickle file		PyMAS				PyMAS	
R score file							
PMML				Snowflake			
ONNX				Azure Synapse			

Mapping output formats to execution engines

Which engine can ingest which model format?

	Batch				On demand		
	SPRE	CAS	Python / R	In-DB	Spark	MAS	Python / R
DATA step				(using EP)	(using EP)		
DS2				(using EP)	(using EP)		
ASTORE				(using EP)	(using EP)		
Python .py		PyMAS					
Pickle file		PyMAS					
R score file							
PMML				Snowflake			
ONNX				Azure Synapse			

Common arrangement for on-demand processing

Mapping output formats to execution engines

Which engine can ingest which model format?

	Batch					On demand	
	SPRE	CAS	Python / R	In-DB	Spark	MAS	Python / R
DATA step				(using EP)	(using EP)		
DS2				(using EP)	(using EP)	wrapper	
ASTORE				(using EP)	(using EP)		
Python .py		PyMAS				PyMAS	
Pickle file		PyMAS				PyMAS	
R score file							
PMML				Snowflake			
ONNX				Azure Synapse			

For running a Python model from MAS

Mapping output formats to execution engines

Which engine can ingest which model format?

	Batch					On demand	
	SPRE	CAS	Python / R	In-DB	Spark	MAS	Python / R
DATA step				(using EP)	(using EP)		
DS2				(using EP)	(using EP)		
ASTORE				(using EP)	(using EP)		
Python .py		PyMAS				PyMAS	
Pickle file		PyMAS				PyMAS	
R score file							
PMML				Snowflake			
ONNX				Azure Synapse			

You can run PyMAS from CAS too

Mapping output formats to execution engines

Which engine can ingest which model format?

	Batch					On demand	
	SPRE	CAS	Python / R	In-DB	Spark	MAS	Python / R
DATA step				(using EP)	(using EP)		
DS2				(using EP)	(using EP)		
ASTORE				(using EP)	(using EP)		
Python .py		PyMAS				PyMAS	
Pickle file		PyMAS				PyMAS	
R score file							
PMML							
ONNX					Azure Synapse		

Proc SCOREACCEL

Run in Hadoop or Teradata using Model Manager or Proc SCOREACCEL

Mapping output formats to execution engines

Which engine can ingest which model format?

	Batch					On demand	
	SPRE	CAS	Python / R	In-DB	Spark	MAS	Python / R
DATA step				(using EP)	(using EP)		
DS2				(using EP)	(using EP)		
ASTORE				(using EP)	(using EP)		
Python .py		PyMAS				PyMAS	
Pickle file		PyMAS				PyMAS	
R score file							
PMML				Snowflake			
ONNX				Azure Synapse			

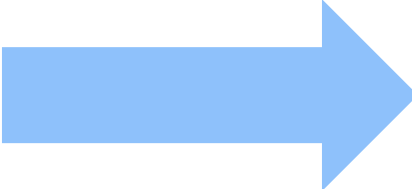
Conceptual Architecture

Deploying decision logic

Sandbox

The Sandbox environment contains several visual components: a decision flow diagram with nodes and arrows, a bar chart with blue bars, and a network diagram with nodes and connections.

- Create Decision Flow
- Diagram Test



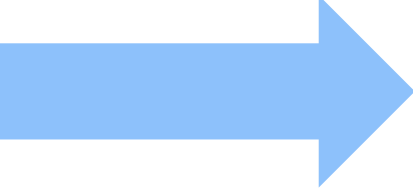
Validation

```

[sas@sasserver dqdashboard]$ ll
total 17280
-rw-rw-r-- 1 sas sas 3132 Apr 7 07:56 addCasLib.log
-rw-rw-r-- 1 sas sas 443 Apr 7 07:56 addCasLib.sas
-rwxr-xr-x 1 sas sas 7290656 Sep 27 2018 ADDRESS.sashdat
-rwxr-xr-x 1 sas sas 8672 Sep 10 2018 BDN_TERMIDS.sashdat
-rw-r----- 1 sas sas 152 Apr 2 09:00 config.json
-rw-rw-r-- 1 sas sas 819697 Apr 3 11:06 DQ_Rules_and_Dashboard.json
-rwxr-xr-x 1 sas sas 975 Apr 3 11:39 lo.sh
-rwxr-xr-x 1 sas sas 185472 Nov 13 2018 MonitorRules.sashdat
-rwxr-xr-x 1 sas sas 9165808 Aug 20 2018 PERSON.sashdat
-rw-rw-r-- 1 sas sas 187944 Apr 3 11:06 SAS_Code_for_Dashboard.json
-rw-rw-r-- 1 sas sas 9112 Mar 26 2019 TEST_HIST.sashdat
-rwxr-xr-x 1 sas sas 131 Apr 2 11:45 up.sh
[sas@sasserver dqdashboard]$
    
```

ID	NAME	TYPE	STATUS	DATE	TIME	USER	GROUP	SIZE	PERMISSIONS
1	10000	FILE	OK	2019	03	11:06	sas	819697	rw-rw-r--
2	10001	FILE	OK	2019	03	11:39	sas	975	rwxr-xr-x
3	10002	FILE	OK	2018	11	13	sas	185472	rwxr-xr-x
4	10003	FILE	OK	2018	08	20	sas	9165808	rwxr-xr-x
5	10004	FILE	OK	2019	03	11:06	sas	187944	rw-rw-r--
6	10005	FILE	OK	2019	03	11:06	sas	9112	rw-rw-r--
7	10006	FILE	OK	2019	03	11:06	sas	819697	rw-rw-r--
8	10007	FILE	OK	2019	03	11:39	sas	975	rwxr-xr-x
9	10008	FILE	OK	2018	11	13	sas	185472	rwxr-xr-x
10	10009	FILE	OK	2018	08	20	sas	9165808	rwxr-xr-x
11	10010	FILE	OK	2019	03	11:06	sas	187944	rw-rw-r--
12	10011	FILE	OK	2019	03	11:06	sas	9112	rw-rw-r--
13	10012	FILE	OK	2019	03	11:06	sas	819697	rw-rw-r--
14	10013	FILE	OK	2019	03	11:39	sas	975	rwxr-xr-x
15	10014	FILE	OK	2018	11	13	sas	185472	rwxr-xr-x
16	10015	FILE	OK	2018	08	20	sas	9165808	rwxr-xr-x
17	10016	FILE	OK	2019	03	11:06	sas	187944	rw-rw-r--
18	10017	FILE	OK	2019	03	11:06	sas	9112	rw-rw-r--
19	10018	FILE	OK	2019	03	11:06	sas	819697	rw-rw-r--
20	10019	FILE	OK	2019	03	11:39	sas	975	rwxr-xr-x

- Decision Logic Test
- Decision Performance Test



Live

The Live environment displays a 'Customs Inspection' web form with fields for Category, Entry Point, Origin, Fragile, Payment Method, Price, Weight, and Volume. Below the form, a message states: 'Probability that our package is suspicious: 73.520%' and 'Automated Decision: High Priority! Please inspect the package'. To the right, a data table is visible.

- Production Execution

Example

