S A S V I Y A D E P L O Y M E N T G U I D E

From Model Studio to Production

A practical path for supervised and unsupervised models

Data Analytics & Data Mining

Instructor & Student Reference — Final Demo Projects

Why This Guide Exists

In this course, students have seen end-to-end demos where Python trains the model and Excel deploys it. That path is clean and intuitive. But for the final team presentations, some students will choose SAS Viya for Learners (VFL) — and deployment there works differently.

This guide covers both the standard supervised path (which works beautifully) and the unsupervised path (clustering, market basket, anomaly detection) which requires a little more thought. The goal is to give you and your students a practical, production-minded answer to the question: "OK, I built a model in SAS — now how do I actually USE it?"

> 🎯 **The big idea:** *Deployment means one thing — taking NEW data the model has never seen and getting predictions (or cluster assignments) out the other end. Everything in this guide serves that simple goal.*

Part 1: Supervised Models — The Standard Path

This is the path students already saw in the AirLincoln no-show homework. It works for:

- Logistic regression (classification)
- Linear regression (numeric prediction)
- Decision trees, random forests, gradient boosting
- Neural networks and SVMs

The 5-Step Workflow

#	Step	What Happens
1	Train in Model Studio	Build your pipeline, run it, pick your champion model based on assessment metrics
2	Register the Champion	Right-click the champion model → Register. This pushes it to Model Manager.
3	Open Model Manager	Navigate to Manage Models → open your project. Your model is now there with metadata.
4	Run a New Test (Score)	In the Scoring tab → New Test. Point it at your scoring dataset (new data without labels).
5	Use the Scored Output	Output lands in a CAS table. Pull it into Visual Analytics, SAS Studio, or export to Excel.

Detailed Walkthrough

Step 1: Train in Model Studio

Students should already be comfortable with this. Build a supervised pipeline, compare models, identify the champion based on the appropriate metric (AUC for classification, RMSE for regression).

> **Teaching moment:** *Remind students that the champion isn't always the most complex model. An interpretable decision tree that stakeholders trust often beats a black-box gradient boosting by 2% accuracy. This echoes the 87% deployment failure conversation.*

Step 2: Register the Champion

1. Right-click the champion model node in the pipeline diagram
2. Select Register Model
3. Confirm the project destination (or create a new one)
4. Wait for confirmation — look for the ✓ checkmark in the Registered column

> **Common student trap:** *If they can't find the Register option, they likely haven't run the full pipeline yet. The model needs to finish training before it can be registered. Also: decision trees register easiest because they produce DS2 score code without needing ASTORE files.*

Step 3: Open Model Manager

Navigate: Applications menu (three bars, top-left) → ANALYTICS LIFE CYCLE → Manage Models. Find your project and open it. You'll see tabs: Models, Variables, Properties, Files, Scoring, Performance, Workflow, History.

Models tab: shows registered models with their function (Classification/Regression), score code type (DS2 multi-type is most common), algorithm, date modified.

Scoring tab: this is where deployment happens. It's the "New Test" button we care about.

Step 4: Run a New Test (the actual scoring)

5. Click the Scoring tab
6. Click New Test
7. Select your champion model from the list
8. Input table: point to your NEW data (the scoring dataset — typically has the same columns as training MINUS the target)
9. Output data library: CASUSER (default is safe)
10. Output table name: give it something memorable like SkyBridge_Scored_Oct2025
11. Click Run

> **The scoring dataset:** *This is the key conceptual point for students. The scoring data should have the same structure as training data MINUS the target variable. The model PRODUCES the target (or probability of the target). If students include the target in scoring data, the model ignores it — but it's a sign they may not fully understand what scoring means.*

Step 5: Use the Scored Output

The output table now contains your original scoring data PLUS new columns:

- **P_TargetName1** (or similar): predicted probability of the event
- **I_TargetName** (or similar): classification decision (Yes/No) based on default 0.5 threshold
- **EM_EVENTPROBABILITY:** the predicted probability column used by Event-Model conventions

From here, three common downstream paths:

- **Option A:** Visualize in SAS Visual Analytics — create a dashboard showing high-risk cases (customers with P > 0.7, for example)
- **Option B:** Export to Excel — in SAS Studio, use PROC EXPORT or download the CAS table as CSV, then students can show it in Excel
- **Option C:** Publish to a destination — in Model Manager, the Publish button can push score code to CAS, Hadoop, or database destinations for automated scoring

Part 2: Clustering — Three Paths for Deployment

Here's the honest answer to your question: clustering deployment in SAS Viya for Learners is different from supervised models, because the "Register → Score" pattern in Model Manager is primarily designed for supervised learning. Clustering has three legitimate paths depending on your use case.

For the Nike customer segmentation demo, I recommend showing Path A (visual, immediate) as the "dashboard deployment" and then Path B (score code) as the "how we'd actually assign a new customer" follow-up. This gives students the complete picture.

Path A: SAS Visual Analytics — Cluster as a Data Item

This is exactly what you instinctively asked about, and yes, it's a valid form of deployment — specifically, deployment for reporting and dashboard purposes.

When to Use Path A

- You want immediate visual segmentation of customers
- The "production" use case is a dashboard that marketers filter by cluster
- Students need a fast, impressive visual for their demo
- The cluster assignments refresh whenever the data refreshes (acceptable for most business users)

How to Do It

12. Open SAS Visual Analytics, create a new report
13. Add your Nike customer dataset (nike_customers_student.csv) as a data source
14. In the data pane, right-click the dataset → New Data Item → Cluster
15. In the Cluster dialog: select the variables you want to cluster on (e.g., purchase_frequency, avg_spend, recency, athleisure_ratio)
16. Specify the number of clusters (K) — start with 4 or 5 for a Nike demo
17. VA adds a new data item called Cluster_ID (or similar) to your dataset
18. Drag Cluster_ID onto any visualization to color/group by segment

Deployment Implications

This IS deployment — just a specific kind. Every time a marketer opens this dashboard, VA recomputes (or reuses) the cluster assignments. New customers that enter the dataset will automatically get a cluster ID when the dashboard refreshes.

> 🎯 **Honest limitation:** *Path A recomputes clusters on every data refresh. This means Customer #1,234 might be in Cluster 2 today and Cluster 3 next month if the underlying distribution shifts. For stable, business-critical segmentation, Path B is better.*

Path B: SAS Studio with PROC KCLUS — Scorable Clustering

This is the "real" deployment path for clustering. You train the clustering model, export the centroids (the cluster centers), and then use those centroids to assign any new customer to a cluster. This matches what Python's sklearn does with kmeans.fit() → kmeans.predict().

When to Use Path B

- Marketing wants stable cluster assignments (a customer stays in the same cluster until you retrain)
- You need to score customers one-at-a-time (e.g., when they sign up)
- The cluster model needs to be part of an automated pipeline
- You want to show students what's happening "under the hood" of clustering

Step 1: Train the Cluster Model with PROC KCLUS

In SAS Studio (under Develop Code and Flows), run this code. It trains k-means with K=4 and saves the centroids.

```
/* Train the clustering model */
proc kclus data=CASUSER.NIKE_CUSTOMERS
           maxclusters=4
           maxiter=50
           seed=42;
   input purchase_frequency avg_spend recency athleisure_ratio / level=interval;
   output out=CASUSER.NIKE_CLUSTERED
          copyvars=(customer_id);
   ods output ClusterCentroids=WORK.centroids;
run;
```

What this produces:

- CASUSER.NIKE_CLUSTERED — your original data WITH a cluster assignment column
- WORK.centroids — a small table with the center of each cluster (this is your deployable model!)

Step 2: Save the Centroids as Your "Deployed Model"

The centroid table IS your deployed model. Here's what it looks like conceptually:

```
Cluster_ID | purchase_frequency | avg_spend | recency | athleisure_ratio
     1     |        12.3        |   245.0   |   14    |      0.72
     2     |         3.1        |    89.0   |   87    |      0.31
     3     |        24.8        |   412.0   |    7    |      0.85
     4     |         7.2        |   156.0   |   45    |      0.48
```

Each row is a cluster centroid — the "typical" customer for that segment.

Step 3: Score a New Customer

To assign a new customer to a cluster, compute the Euclidean distance from the customer to each centroid, and pick the nearest. This is the core of k-means scoring and it's just arithmetic.

```
proc distance data=CASUSER.NEW_CUSTOMERS
              out=CASUSER.DIST_TO_CENTROIDS
              method=euclid;
```

```
   var interval(purchase_frequency avg_spend recency athleisure_ratio);
   id customer_id;
run;
```

Then pick the minimum distance for each customer. Students will love this because it demystifies what clustering "means" — it's literally just finding the nearest center.

Step 4: Deploy via Model Manager (optional advanced step)

You CAN register a KCLUS model in Model Manager by wrapping the score logic in a DS2 or SAS macro. This is overkill for most student demos, but worth mentioning as the "enterprise" version of what they just did.

Path C: Model Studio Data Mining & Machine Learning Pipeline

Model Studio has a second project type (not just "Data Mining Project" but "Data Mining and Machine Learning") that includes clustering nodes as part of a pipeline. This gives you Register/Publish behavior similar to supervised models.

When to Use Path C

- You want the full Model Manager experience for clustering
- Enterprise deployment with monitoring is a requirement
- You're combining clustering with downstream supervised models (cluster ID as a feature)

How to Do It (High Level)

19. Create a new Model Studio project, type = Data Mining and Machine Learning
20. Add your data source (Nike customers)
21. In the pipeline, right-click Data node → Add → Unsupervised Learning → Clustering
22. Configure the Clustering node (variables, K, algorithm)
23. Optional but powerful: add a Segment Profile node after Clustering to describe each segment
24. Run the pipeline, right-click the Clustering node → Register Model
25. Now it's in Model Manager and can be scored via New Test like supervised models

> ⚠ **Honest caveat:** *VFL sometimes limits which nodes are available depending on license level. If the Clustering node doesn't appear in your students' environment, fall back to Path A or Path B — both are pedagogically valid and arguably easier to explain.*

Part 3: Which Path for Which Demo?

Here's a decision table mapped to your specific final demo projects:

Demo	Technique	Recommended SAS Path
#1 SkyBridge Preventive Maintenance	Logistic Regression	Standard supervised: Model Studio → Register → Model Manager → New Test. Same workflow as AirLincoln no-show homework.
#2 UrbanBites Recommenders	Association Rules / Collaborative Filtering	SAS Studio code path: use PROC ASSOC for market basket. Export rules table as CSV → show in Excel as "if customer bought X, recommend Y" lookup.
#3 Nike Clustering	K-Means Clustering	Combination: Path A (VA cluster object for the dashboard visual) + Path B (PROC KCLUS with centroid export) for the scoring story.
#4 ShieldScore Anomaly Detection	Multiple models (classification + isolation forest)	Standard supervised path for the classification piece; Path B-style centroid/threshold scoring for the unsupervised piece.
#5 FleetPulse Time Series + Clustering	Time Series + Clustering	Time series: PROC TSMODEL or Forecasting Server pipeline. Clustering: Path B for stable segment assignment.

Part 4: What Students Should Show in Their Presentation

For team demos, regardless of which path they choose, students should cover these five points. This creates a consistent rubric across different tools (Python+Excel OR SAS Viya).

1. The Business Question

- What decision does this model help someone make?
- Who is the stakeholder? (Operations manager, marketing lead, fraud analyst, etc.)
- What's the cost of being wrong?

2. The Data Story

- Where did the data come from?
- What features matter most?
- What did they do about missing values, outliers, class imbalance?

3. The Model (briefly)

- Which technique and why?
- How did they pick it over alternatives?
- What's the champion model performance?

4. The Deployment Path

- Where does the model live?
- How does a new customer/case get scored?
- Who uses the output and how?

> **This is the MVP of the whole presentation.** *87% of ML projects die because nobody can answer these three sub-questions. If your students can, they're already in the 13% that succeeds.*

5. The Risk Mitigation

- What could go wrong with this model?
- How will they monitor for drift?
- What's the retraining cadence?

Part 5: Common Student Issues & Quick Fixes

Symptom	Likely Cause & Fix
"I can't find the Register option"	Pipeline hasn't completed running, or they're right-clicking the wrong node. Make sure the model node shows a green checkmark first.
"New Test fails with a variable mismatch error"	Scoring dataset has different column names or types than training data. Open both tables in SAS Studio and compare structure.
"Autotuning gives an error"	Expected — autotuning is disabled in VFL. Use default hyperparameters or tune manually via properties panel.
"Clustering node doesn't appear in Model Studio"	Wrong project type. Create a new project and select "Data Mining and Machine Learning," not the default option. If still missing, fall back to Path A or B.
"The scored output has weird column names like EM_PROB_1"	That's normal. EM_ = Event Model. EM_EVENTPROBABILITY is your predicted probability. Rename it on the way into Excel if needed.
"VA cluster item looks nothing like what PROC KCLUS gave me"	Different algorithms. VA uses its own clustering engine (often k-means variant) with different defaults. That's why Path A and Path B can disagree — document which one is the "source of truth."
"My CSV won't import into VFL"	File size limits or encoding issues. Try: open in Excel → save as CSV UTF-8. Also check that there are no special characters in column names.

Part 6: The Methodology-Over-Tools Message

The most important thing students should take from this is your core teaching philosophy: methodology over tools.

Whether a team uses Python+Excel or SAS Viya, the steps are the same:

- Define the business question (the Prediction in PAIR)
- Prepare the data (clean, transform, engineer features)
- Train the model and pick a champion
- Evaluate honestly (AUC, RMSE, precision/recall based on cost asymmetry)
- Deploy (register, score new data, pipe results to a consumer)
- Monitor (watch for drift, retrain when needed)

The tool is just the instrument. A violin and a piano play different sounds, but they both play music. Python and SAS both deploy models — they just do it with different buttons. If students understand the methodology, they can pick up any tool their employer uses.

> **Close the loop:** *When students present, ask them the same question regardless of tool: "Where does this model live, and who calls it to make decisions?" If they can answer clearly, they've got deployment. That's the whole point.*

★ **Methodology over tools. Always.** ★