Assessing Model Bias and Model Health in SAS® Viya

Cat Truxillo, Ph.D.
Director, Analytical Education

About the data

A binary target variable ***Admit*** indicates whether the patient was admitted.

Three variables are identified as potential sources of bias:

Name	Description
gender	female, male, other
race	asian, black, hispanic, other, white
insurance	medicaid, medicare, other, private, self-pay

This workshop uses a data set called dataset named **er_visits_synthetic_biased**_basic. The data set was generated using ChatGPT. While there were additional prompts that ultimately led to the creation of this dataset, this was the original prompt: "I'm looking to showcase my data science platform capabilities. I plan to do this by creating a demo that is specific to the healthcare industry. I want the healthcare customers to be engaged with the demo, so I want the data to have similar qualities to real world health care data. I want to create an AI model whose goal is to determine whether a patient that goes to the emergency room should be admitted to the hospital. Please create a dataset with 10000 observations that has 15 numeric variables and 10 categorical variables." Also, manual edits were added so that black patients and men are more likely to be admitted… this will cause the algorithm to have a harder time predicting for these subpopulations; the bias detection capability to be used to reveal this information. Because this dataset was generated using ChatGPT it should under no circumstances be used for anything other than demonstration purposes.

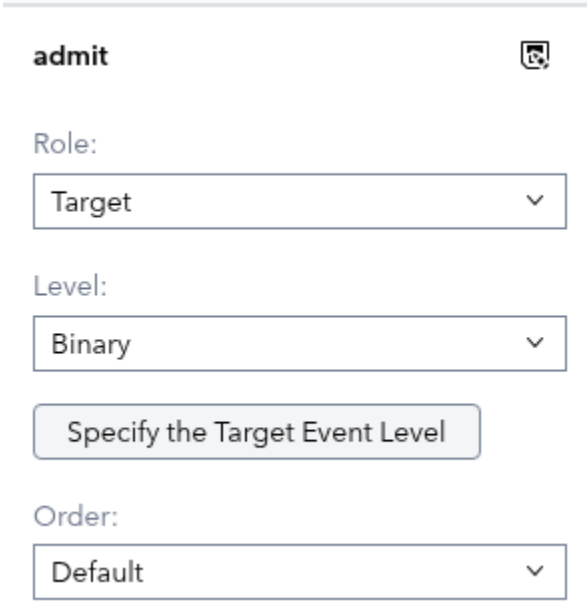
Demonstration Steps

Follow the instructor's instructions to log into the SAS Viya environment.
User: **student**
Password: **Metadata0**
Opt in to assumable groups: **Yes**

1. Click on the applications menu. Select **Build Models**.
2. Select **New projec**t.
3. Name the project **Bias Workshop**.
4. Set the Type to **Data Mining and Machine Learning**.
5. For the template, select **Browse**. Choose **Feature engineering template**. Click OK.
6. For the data, select **Browse → ER_VISITS_SYNTHETIC_BIASED_BASIC**. Click **OK → Save.**

The project is created. In the Data tab, you need to select a few properties for the model pipelines to run.

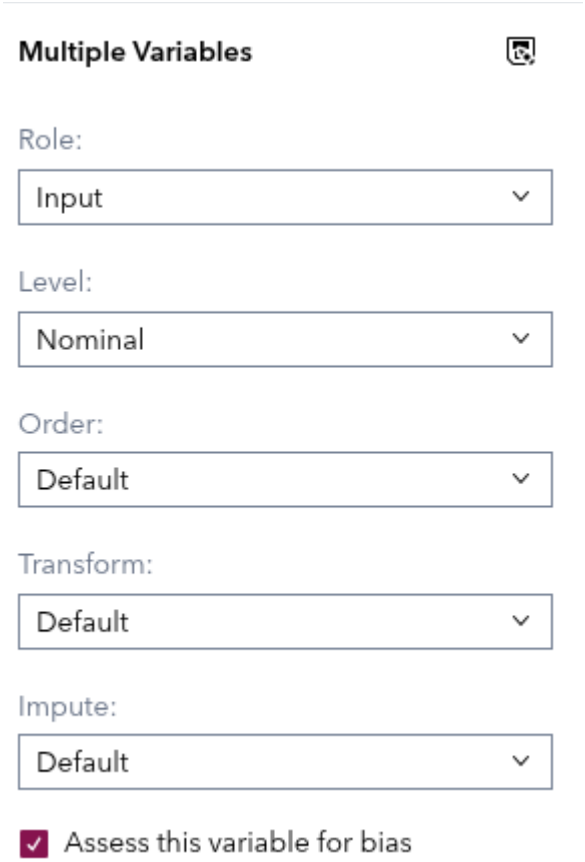
7. First, assign a target variable. Scroll to and select the variable **admit**. On the right, assign the Role as **Target**.
8. Specify the Target Event Level as **admit**.
9. Click **Save.**



10. Deselect **admit.**

Select variables to assess for bias.

11. Select **gender**, **insurance**, **race**.
12. Set the Role to **Rejected**.
13. Select **Assess this variable for bias**.



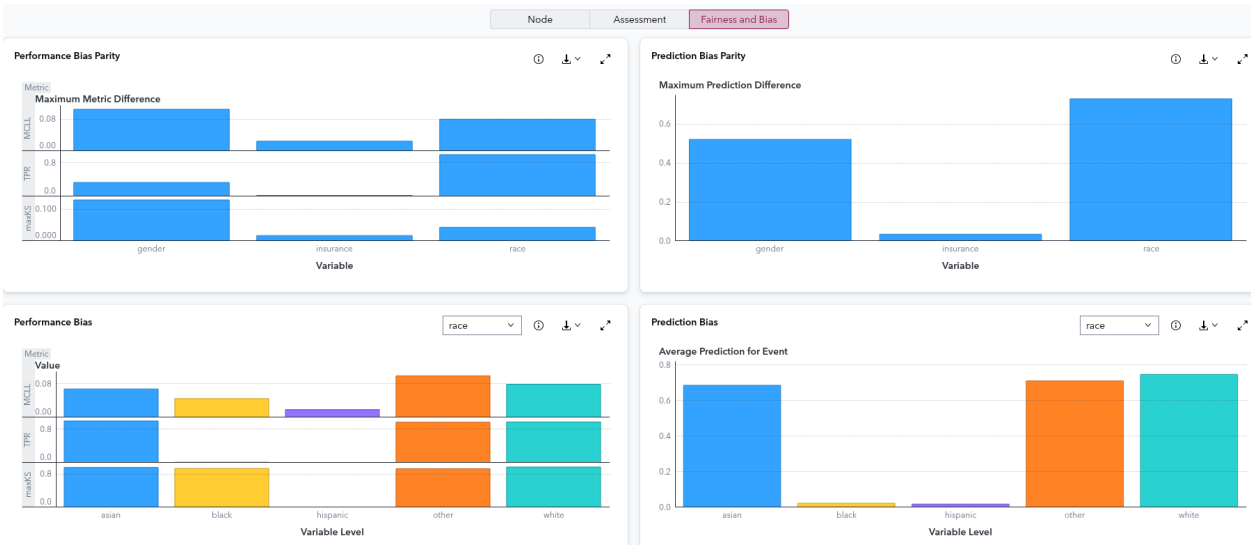
14. Deselect the variables **gender**, **insurance**, and **race**.

Run the pipeline and examine the results.

15. Click the Pipelines tab. Select **Run pipeline**.

The models in this pipeline show very similar performance, and due to differences in random sampling for the validation data, you might have a different champion model. The best model in the pipeline shown here is Autoencoder and Gradient Boosting. Feel free to evaluate any of the models you are most interested in.

16. Right-click the Autoencoder and Gradient Boosting model and select **Results**. Click the Fairness and Bias tab.



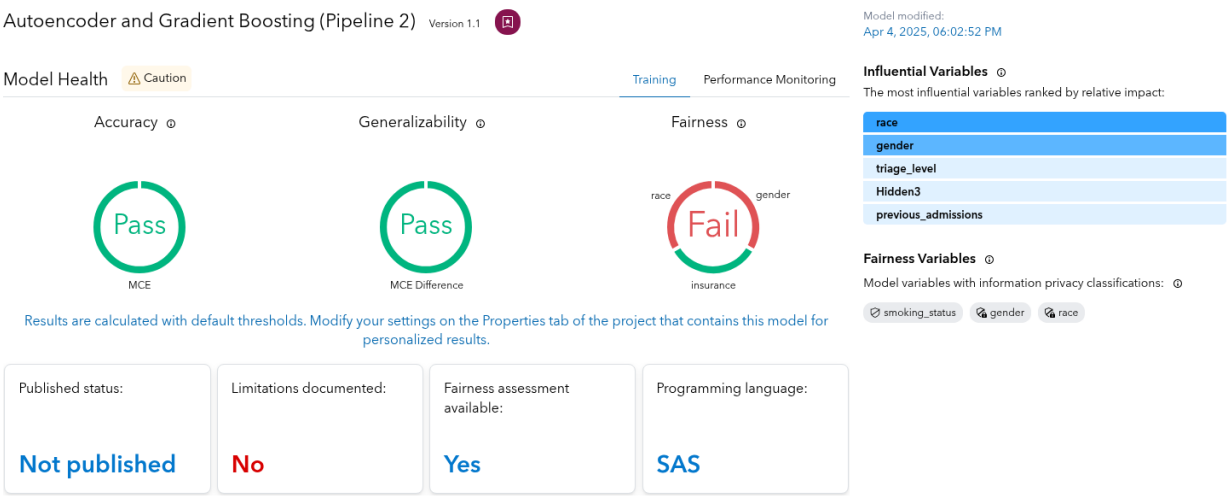
17. Close the Autoencoder and Gradient Boosting Results.

Now, register the winning model from your pipeline.

18. Select the **Pipeline Comparison** tab.
19. Right-click the model and select **Register Model**. Click **OK**. Click **Close**.
20. Once the models are registered, select **Manage Models** from the app menu.
21. Select Models on the left.

22. Click on the **Autoencoder and Gradient Boosting** model.
23. Look over the Model Card.

The Model Card is like a nutrition label for your model. At this point, it tells you about the health of the model on the training data, using a default set of measures.



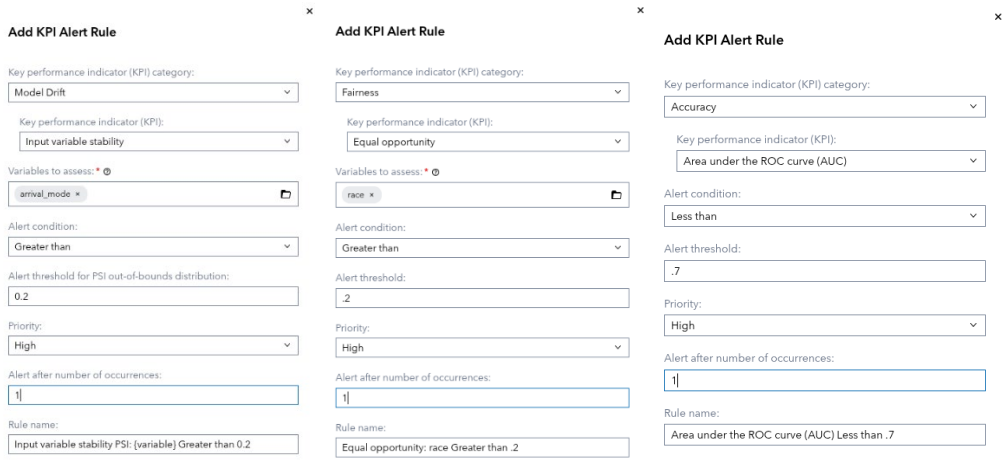
You can evaluate the health of a model over time by assessing it against data acquired after the model is deployed.

Start by loading quarterly follow-up data into memory

24. From the Applications menu, select **Manage Data**.
25. Select **Import**--> **Local Files**.
26. Navigate to the location your instructor tells you (this might be */workshop/HOWCAT/*). Select the data sets recommended by your instructor (*ervisitssynthetic_**). Click **Open**.
27. For each table, under **Step 2 – Output File Options**, set the option for *If table name exists:* to **Replace File**.
28. Click **Import all**.

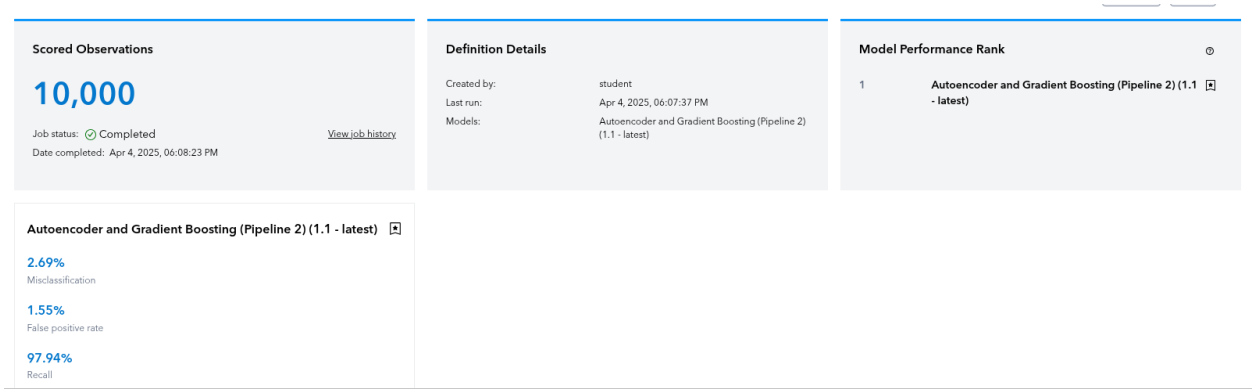
Return to Model Manager

29. Select **Applications → Manage Models**. You might have to click the project link at the top of the model card.
30. Click the project's Properties tab.
31. Select **Model Evaluation**. Scroll to the KPI Alert Rules. Click **+**.
32. Select the desired KPIs. For example, you might set Accuracy: AUC < .7; Fairness: Equal opportunity(race) > .2; Model drift: input stability (by predictor) >.2.

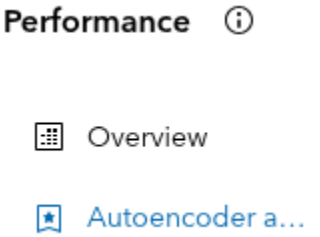


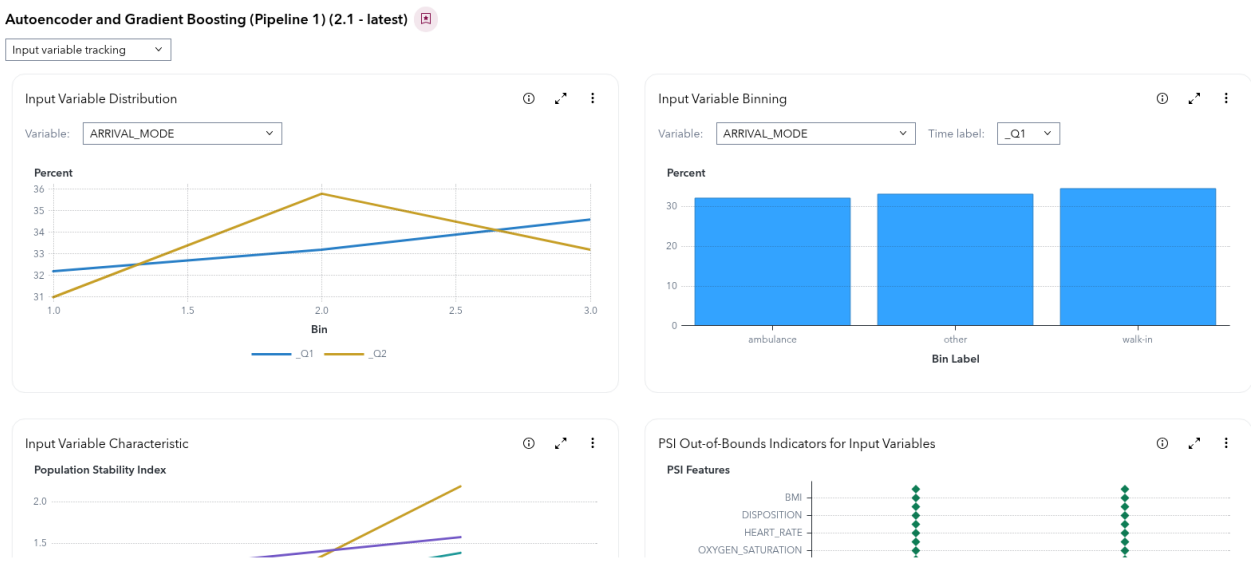
33. Click **OK** for each KPI you select. Click **Save**.
34. Click on the Performance tab and select **New definition**.
35. Click **Tables**.
36. Select **Use a library that contains a specified prefix.**
37. Select next to CAS library. Select **Public → OK**.
38. Next to Prefix type **ERVISITSSYNTHETIC**.
39. Click **Save**.
40. Click **Run → Run now**.

It might take a few minutes to complete the Performance report.



41. On the left-hand side of the Performance tab, select the **Autoencoder and Gradient Boosting Model**.





You can see the performance of the variables over time.

Return to the model card.

42. Click the Models tab, and select the name of the model.
43. Click Model Audit. It takes several minutes for the performance report results to populate, and you might need to refresh the browser.

Model Audit

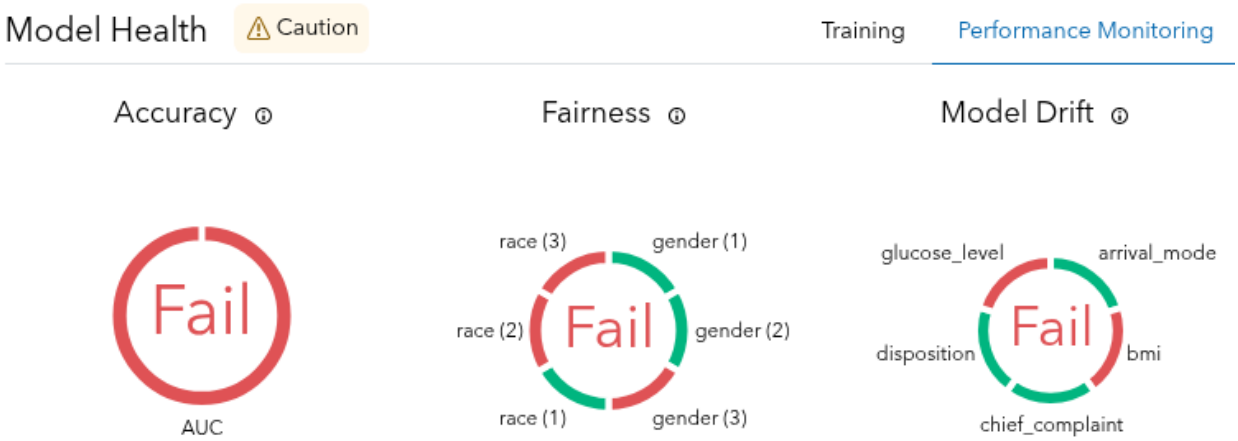
Performance Monitoring Last Run

Model last run: Apr 8, 2025, 05:59:15 PM
Data used: ERVISITSSYNTHETIC_02_Q2

Project last run: Apr 8, 2025, 05:59:15 PM
Data used: ERVISITSSYNTHETIC_02_Q2

KPI	Alert Condition	Alert Threshold	Value	Status
Accuracy				
Area under the ROC curve (AUC)	Less than	0.7	0.551	Fail
Fairness				
Demographic parity: gender (1)	Greater than	0.2	0.021	Pass
Demographic parity: gender (2)	Greater than	0.2	0.021	Pass
Equalized odds: gender (3)	Greater than	0.2	0.361	Fail
Demographic parity: race (1)	Greater than	0.2	0.071	Pass
Equal opportunity: race (2)	Greater than	0.2	0.529	Fail
Equalized odds: race (3)	Greater than	0.2	0.529	Fail

Scroll up to the Overview and select the Performance Monitoring tab.



As you add more tables to the library used by the performance monitoring, the model card information will update as well. If you register multiple models, each will have its own model card that you can evaluate.

The results in this model card are useful for demonstrating and documenting the steps you are taking to reduce risk due to bias, drift, model inaccuracy, and more.