Predicting Graduate School Admissions at iLink University

SAS Guided Demo

Purpose

This *SAS Guided Demo* gets you started with ethical data analysis in SAS Model Studio. In seeking to select the fall 2025 class for the Masters in Analytics Program at iLink University – a completely made-up institution for higher learning – you'll use the characteristics of successful applicants of the past. Then you are confronted with an important question: *just because you can run a model, should you*? Towards answering that important question, you'll learn how built-in fairness and bias assessment tools in SAS Model Studio help you refine your modeling and get you closer to a solution that is both equitable and empirically rigorous.

Scenario

In this *SAS Guided Demo*, we'll pretend that we're at our first day as Predictive Modelers in the Admissions Department at iLink University. Moreover, our modeling will help decide who gets into the fall 2025 M.S. in Analytics Program. So – congrats on the new job – and thank you for joining us on this analytics adventure!

Learning Objectives

We'll cover a lot of ground in this *SAS Guided Demo*, including:

- Creating SAS Model Studio projects
- Running a series of competing predictive models and evaluating which performs "best"
- Choosing a champion model and selecting the new incoming class of 2025
- Understanding how we can use Fairness + Bias tools to re-evaluate our initial models
- Refining our models to meet global objectives beyond "best fit"

Software

This *SAS Guided Demo* uses SAS Viya for Learners 4, version 2024.09LTS. Other versions can be used, but some screenshots will differ.

Prerequisites

A basic understanding of predictive modeling and machine learning is helpful. Additionally, some experience using the SAS Viya ecosystem is useful, but not required.

Table of Contents

Predicting Graduate School Admissions

Overview

Welcome to iLink University! As a recently hired Predictive Modeler in our Admissions Department, you will help us decide who gets into our fall 2025 M.S. in Analytics Program. Yay!

As this is your first day on the job, I'll guide you through some useful Visual Data Mining and Machine Learning tools in SAS Model Studio. In Part 1 – you'll examine the data and select the incoming 2025 class based upon the tried-and-true process we've used in previous years. Put simply: we use the characteristics from successful applicants in the past to determine the incoming class. In terms of modeling based upon our historical data, you'll use a variable called *Admissions* – which is a 0/1 variable that indicates whether we offered that student admissions to our program. This 0/1 variable can also be interpreted as "Yes" / "No" and the binary nature of this outcome makes it very useful for predictive modeling and machine learning tools.

In Part 2, I'll show you some new + useful features in SAS Model Studio, which fall under the category of **Fairness and Bias assessment** tools. These exciting features allow us to ask the question: "just because we can run a model, should we?" As you learned from your coursework, using historical data to predict new cases can allow existing bias to persist. And since we want the fairest admissions process possible, you will use the Fairness and Bias assessment tools to adjust your model, if needed.

Let's get started!

Part 1: Business as Usual

Overview

Ready, set, go!

I know you're ready to get to work, because you're a Hacker at heart. But before just plunging into the modeling, we should start with a better understanding of the data. For the modeling, we have 1000 admission decisions from the past 5 years. Let's understand the data a bit better, by examining the data dictionary below:

Variable	Label	Definition
Admitted	Admitted (Yes=1)	When Admitted = 1, the student is offered admissions into the iLink University M.S. in Analytics Program
Analytics_Work_Experience	Analytics Work Experience	Number of years working in the field of analytics.
Country_Region	Country Region	Region of the world applying from

Cultural_Identity	Cultural Identity	Cultural identity
Gender	Gender Identity or Gender at Birth	Gender identity or gender at birth
ID	Application ID	Application ID
Legacy_Admission	Legacy Admission	Legacy admission means that either (1) the student's parents attended the university or (2) they previously completed another degree at iLink University.
Mission_Statement	Mission Statement	Optional mission statement (maximum of 100 words)
Standardized_Test_Score	Standardized Test Score	Standardize test score (Z-score)
Strength_of_Recommendation	Strength of Recommendations	Overall strength of recommendations (0 to 5, higher is better)
Undergrad_Degree	Undergraduate Degree Category	Undergraduate degree category
Years_Work_Experience	Years Work Experience	Total years of work experience, all fields

Admissions is the outcome variable that we really care about. As noted, it is a binary variable that can be considered a yes/no variable. The other variables can be used to help us select the top candidates for admissions in the incoming class, based upon historical precedent.

With a better understanding of the data, the rest of the project flows with steps the data analyst in you (generally) knows by heart:

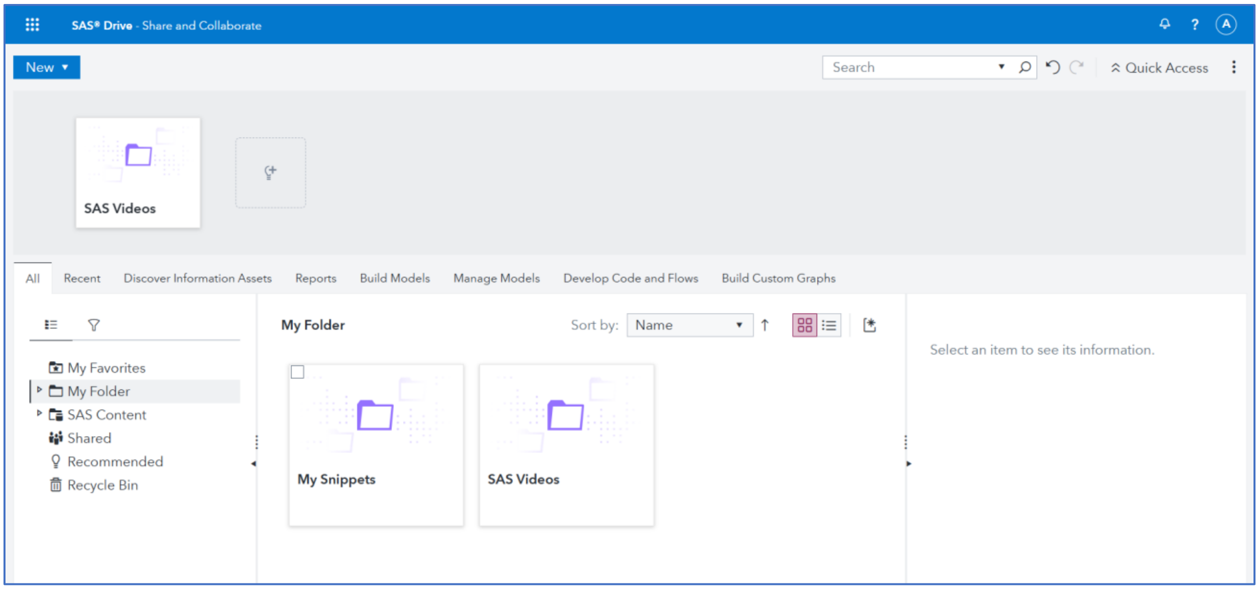
- Start a SAS Model Studio project.
- Explore the data and get to know it A LOT better.
- Run a bunch of predictive models using the Visual Data Mining and Machine Learning tools in SAS Viya.
- Find the best model predicting the historical sample. Crown it the champion.
- Finally, you'll then apply that champion model to a new data set and then choose the 40 students that will be admitted as part of the incoming class of 2025. They will all accept – because our program is awesome.

Let's do it!

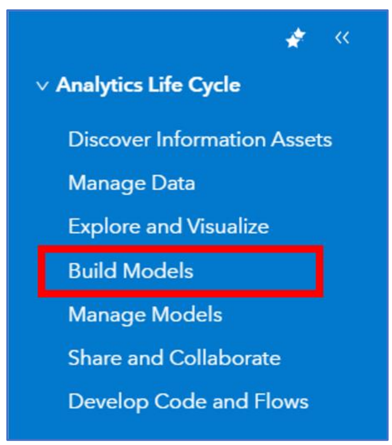
Create a SAS Model Studio Project

A good data adventure often starts with software.

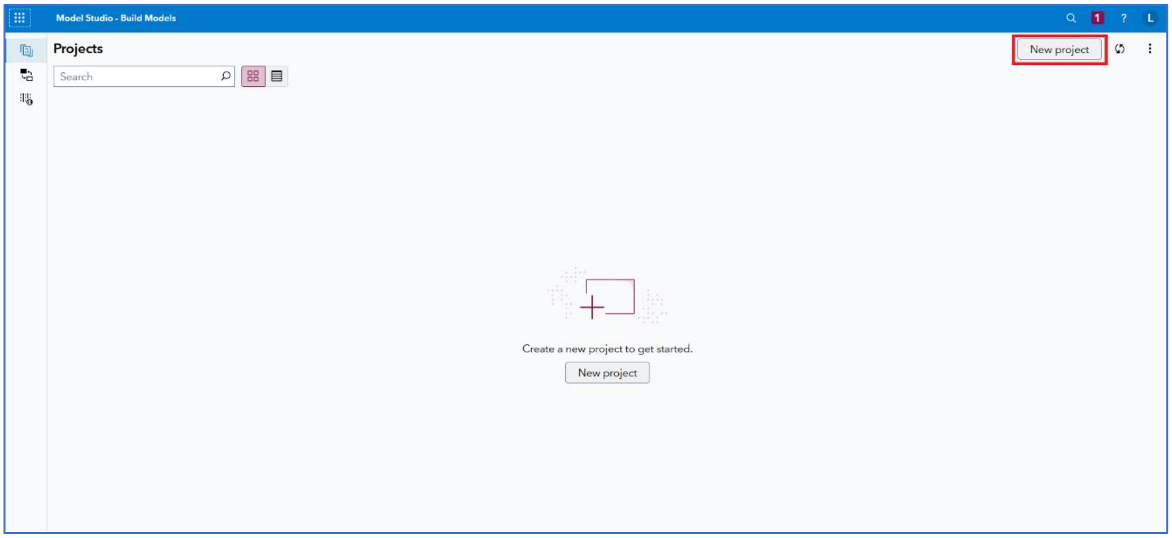
- As such, log in to **SAS Viya for Learners**.
- Once in, you'll land in **SAS Drive**, which will always be your starting point in SAS Viya for Learners. It will look something like the following:



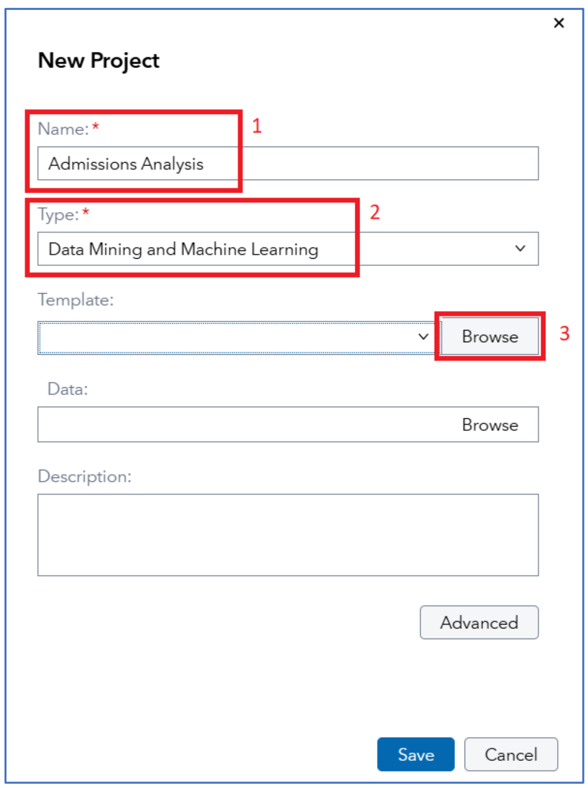
- But, we won't linger here long – as we want to get right into **SAS Model Studio**. From the top left corner, select the **Applications menu** and click **Build Models** from the **Analytics Life Cycle** options.



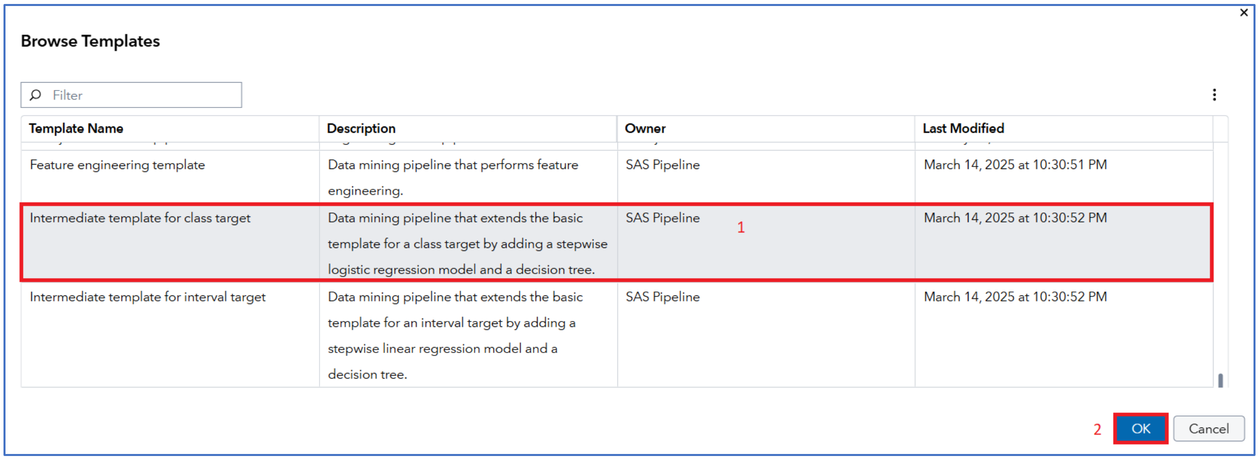
- Welcome to **SAS Model Studio**! Your environment may already have shared projects out there. For now, ignore them. Instead find and click that **New Project** button:



- The **New Project** window will step you through setting up a new project. To start, give it a useful Name, such as *Admissions Analysis*. Keep the **Type** of project at the default, which is *Data Mining and Machine Learning*. Next, find the **Browse** button, under the **Template** section. Those first three steps, succinctly summarized:

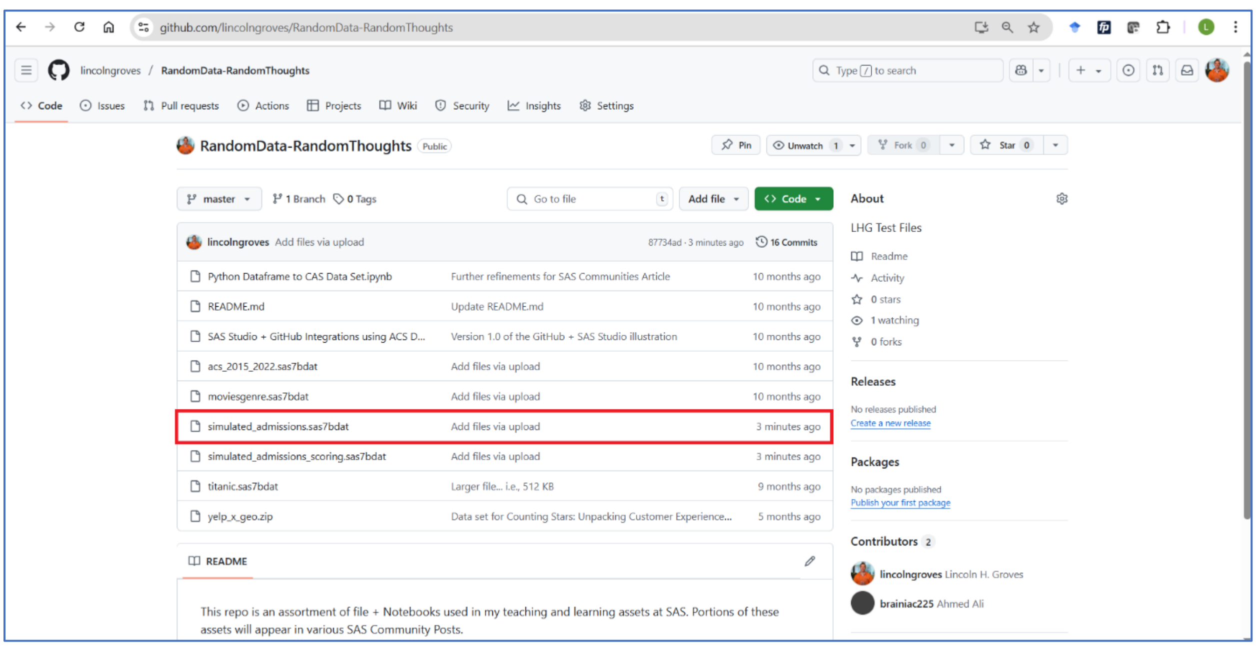


- Click that **Browse** button – and check out all those pre-built templates! When you're happy with your research, find that **Intermediate template for class target template** and select it. Then click **OK**:

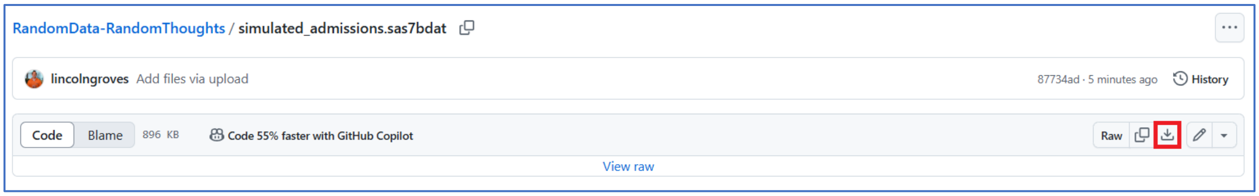


- Sidebar: why “class target”? Well, Admissions is our target variable for modeling. It has two values 1=yes, we admitted them and 0=no, we did not extend them an offer. A yes/no variables opens up a bunch of useful machine learning tools to us.

- Now let’s add some data to your **New Project**. Because we’ve gotta have data! The data are currently housed in a GitHub repo. So, follow these steps to get the original data in your hands:
  - In a browser of your choice, type or paste https://github.com/lincolngroves/RandomData-RandomThoughts
  - Find the **simulated_admissions.sas7bdat** file, here:

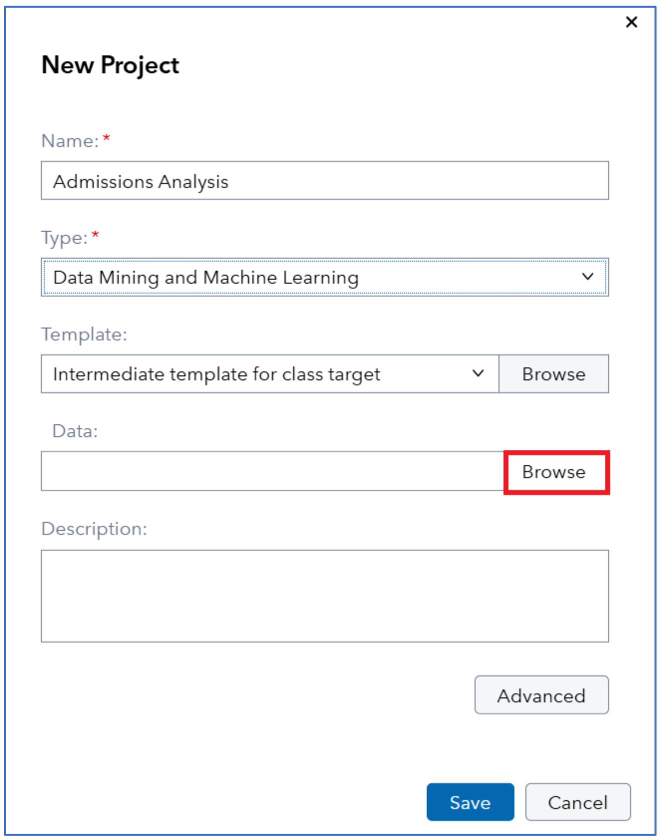


  - Click on the file to drill into the submenu. Then download it by clicking the **Download raw file** button:

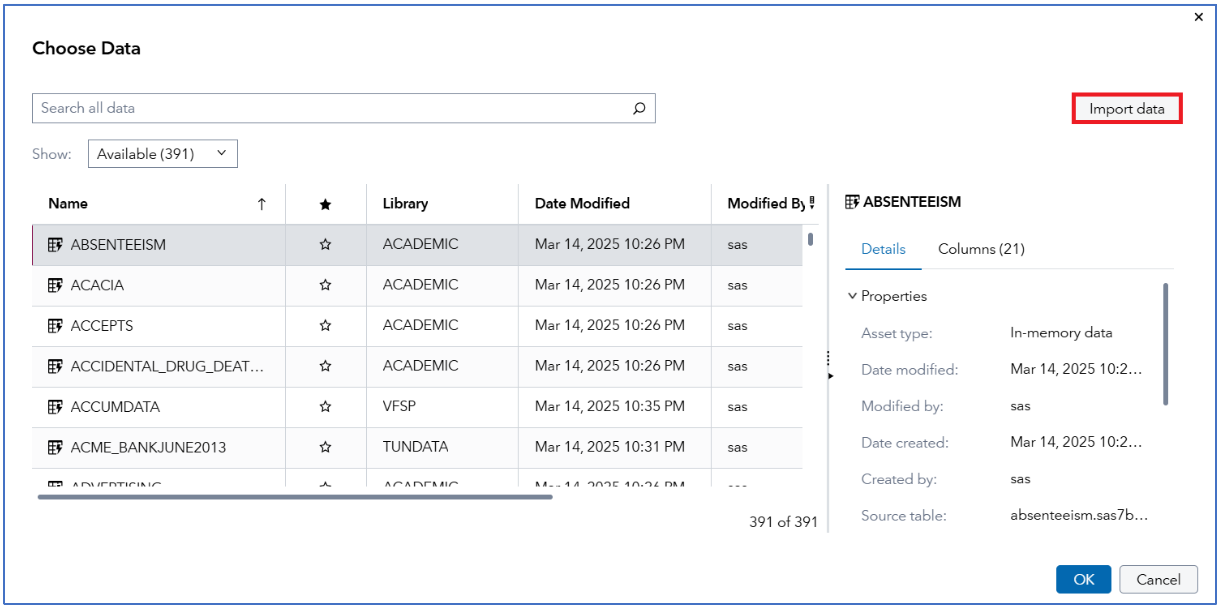


- Data are in hand! And most likely will land in the **Downloads** folder on your local drive.

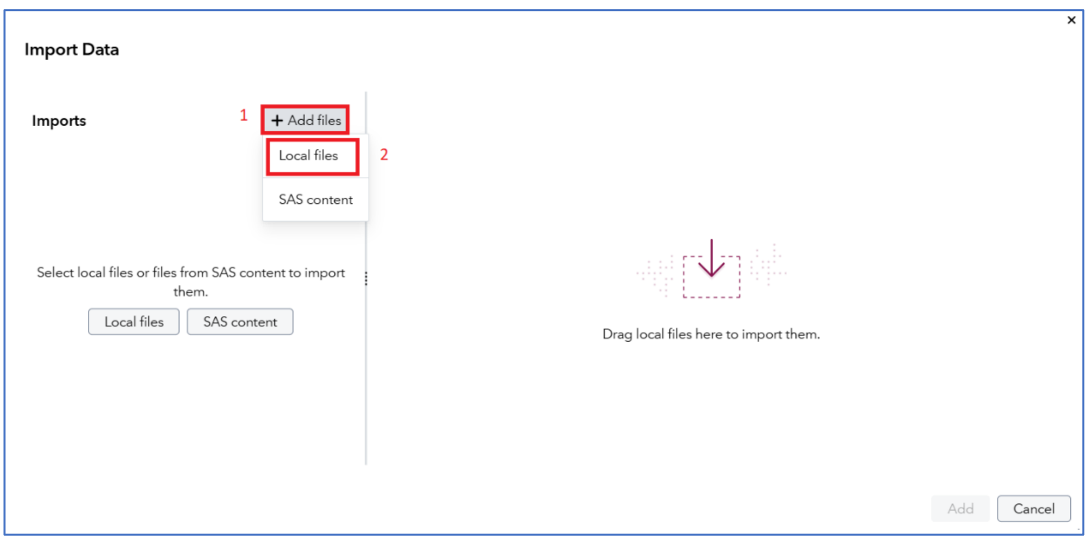
- Next, return to **SAS Model Studio** and your **New Project**. Then under that **Data** section, click that **Browse** button:



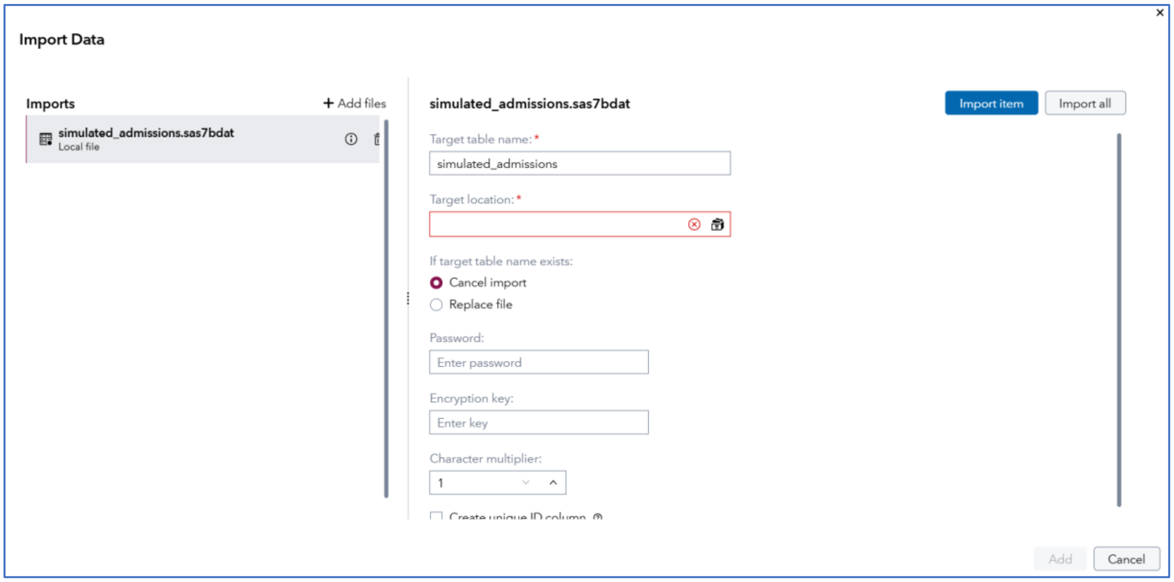
- All the in-memory tables in your environment will load. Which could be a lot! But we want to actually click that **Import data** button, here:



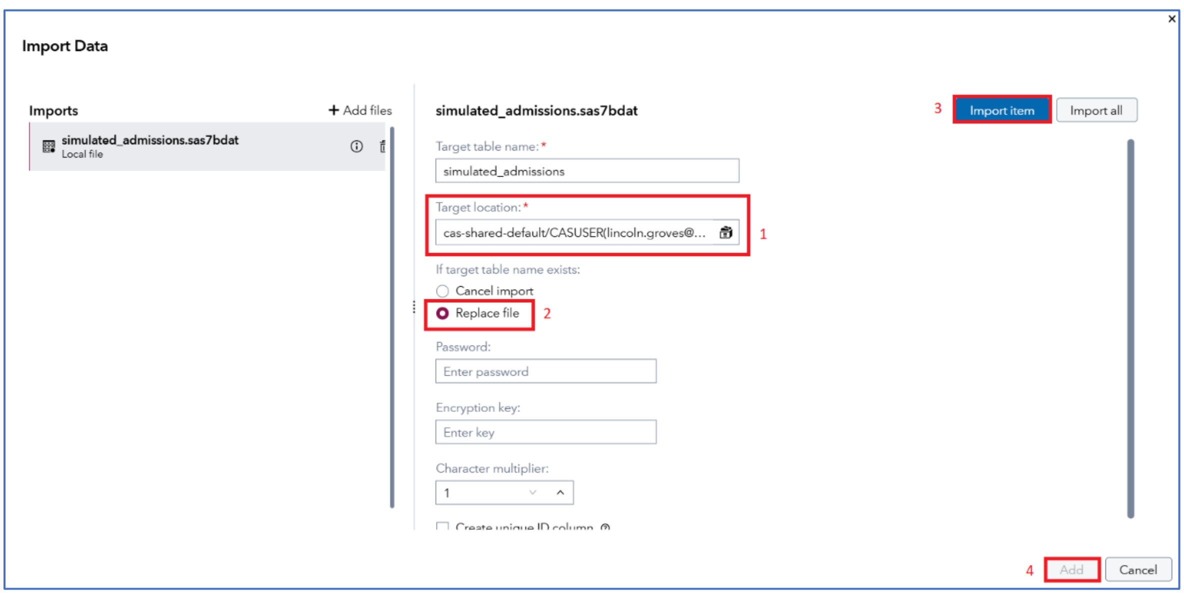
- And, yes, I'm trying to trick you into learning how to upload your own data! So, continue playing along with my trickery and click the + **Add files** and **Local files** buttons in the **Import Data** window:



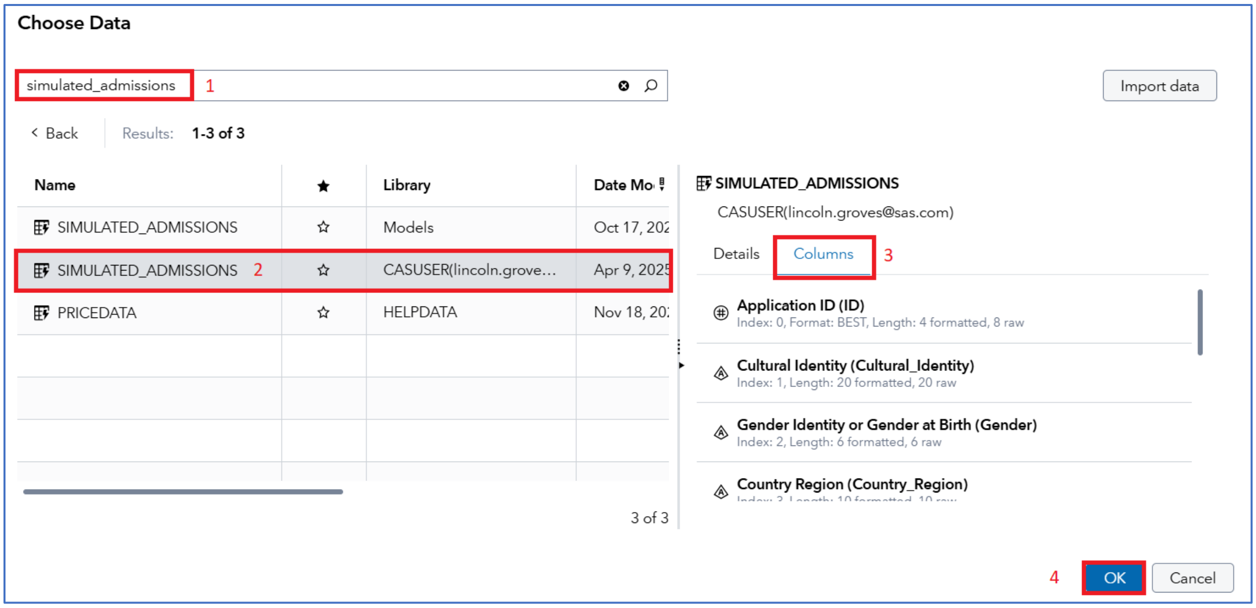
- A browser window pops up that will allow you to access the file you just downloaded. Find *Simulated_Admissions* – likely in your **Downloads** folder – and **Open** it. Success gets you to the **Import Data** window:



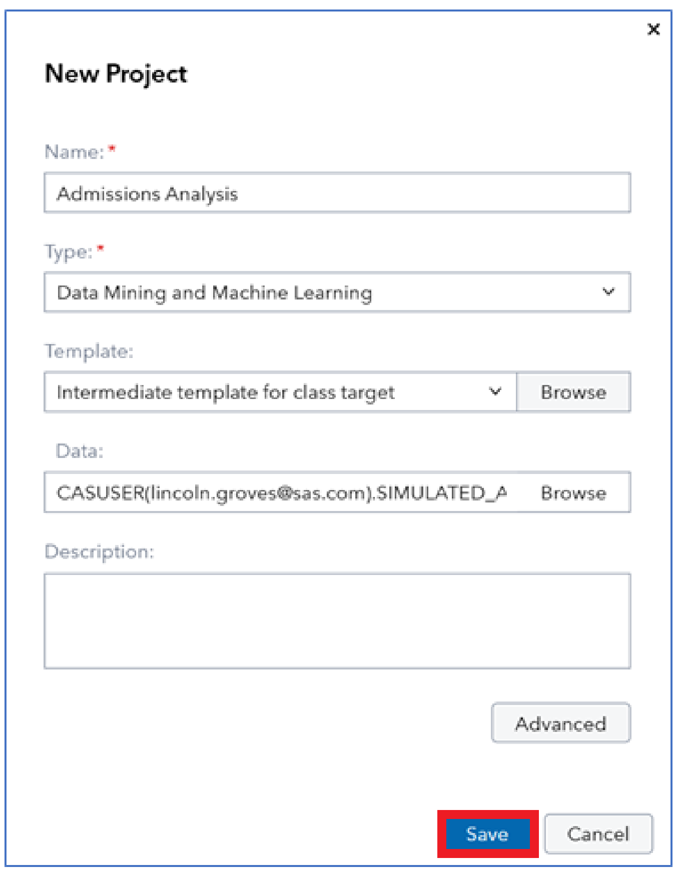
- Almost there – I promise! For **Target location**, find your **CASUSER** folder (spoiler alert: it's not lincoln.groves@sas.com). Then select **Replace file** under **If target table name exists**. Two more clicks to go! Then click **Import Item**… and once it's imported then click **Add** to add it to your project. Those cumulative clicks:



- Your data is now uploaded to the SAS Viya system as an in-memory table! (And I tricked you into learning how to upload files, so you can follow the same process for your next project.) Type **simulated_admissions** in the search bar and find your file. It will be the one in your **casuser** folder. If you'd like, also click on the **Columns** tab in the details section on the right. Here you can ensure your data matches the data dictionary outline above. And – FINALLY – click **OK**! Those steps:



- Your **New Project** details should be good to go! Click **Save** so that your new project can be created:

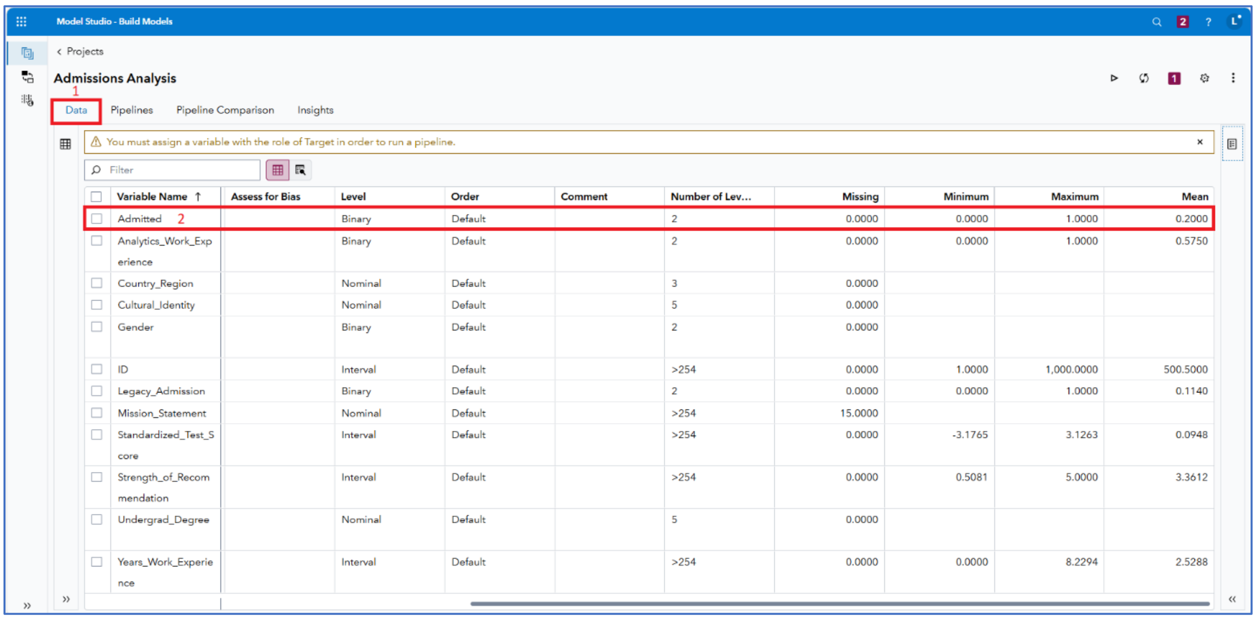


- And your first SAS Model Studio project is all set up. Great job!

Explore the Data

We resume our adventure with a SAS Model Studio project officially created. Next up is to explore the data, just to make sure we really understand what we're modeling.

- If you've never been here before, let me be the first to say: *Welcome to SAS Model Studio*! You'll love it! And you will always land on the **Data** tab. Why? Because this a great place to get to know your data a bit better – rather than just launching into the model (because, as they say: garbage in = garbage out in modeling!). You can scroll to the right to examine all the variable attributes. For example, check out the **Admitted** variable and scroll right:

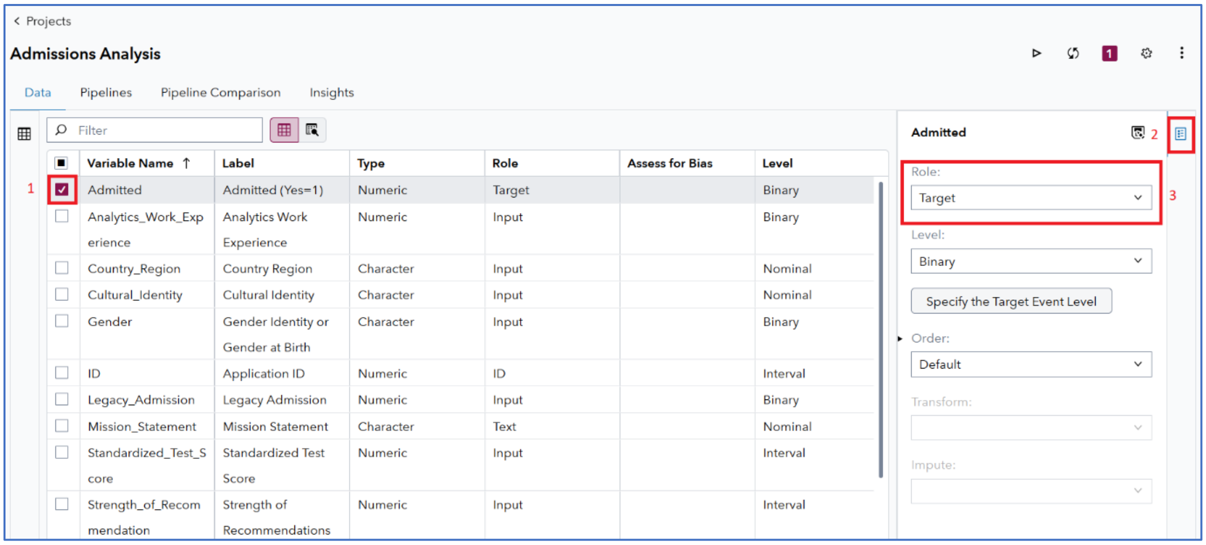


Variable Name	Assess for Bias	Level	Order	Comment	Number of Lev...	Missing	Minimum	Maximum	Mean
Admitted		Binary	Default		2	0.0000	0.0000	1.0000	0.2000
Analytics_Work_Experience		Binary	Default		2	0.0000	0.0000	1.0000	0.5750
Country_Region		Nominal	Default		3	0.0000			
Cultural_Identity		Nominal	Default		5	0.0000			
Gender		Binary	Default		2	0.0000			
ID		Interval	Default		>254	0.0000	1.0000	1,000.0000	500.5000
Legacy_Admission		Binary	Default		2	0.0000	0.0000	1.0000	0.1140
Mission_Statement		Nominal	Default		>254	15.0000			
Standardized_Test_Score		Interval	Default		>254	0.0000	-3.1765	3.1263	0.0948
Strength_of_Recommendation		Interval	Default		>254	0.0000	0.5081	5.0000	3.3612
Undergrad_Degree		Nominal	Default		5	0.0000			
Years_Work_Experience		Interval	Default		>254	0.0000	0.0000	8.2294	2.5288

  - From this succinct summary, we can see that *Admitted* has two values – which are 0 and 1 – with an average of 0.2000 or 20%. So, 20% of the applicants are admitted to the iLink University. That's helpful information!
- As noted by the warning, we must assign the outcome variable:

> ⚠ You must assign a variable with the role of Target in order to run a pipeline.

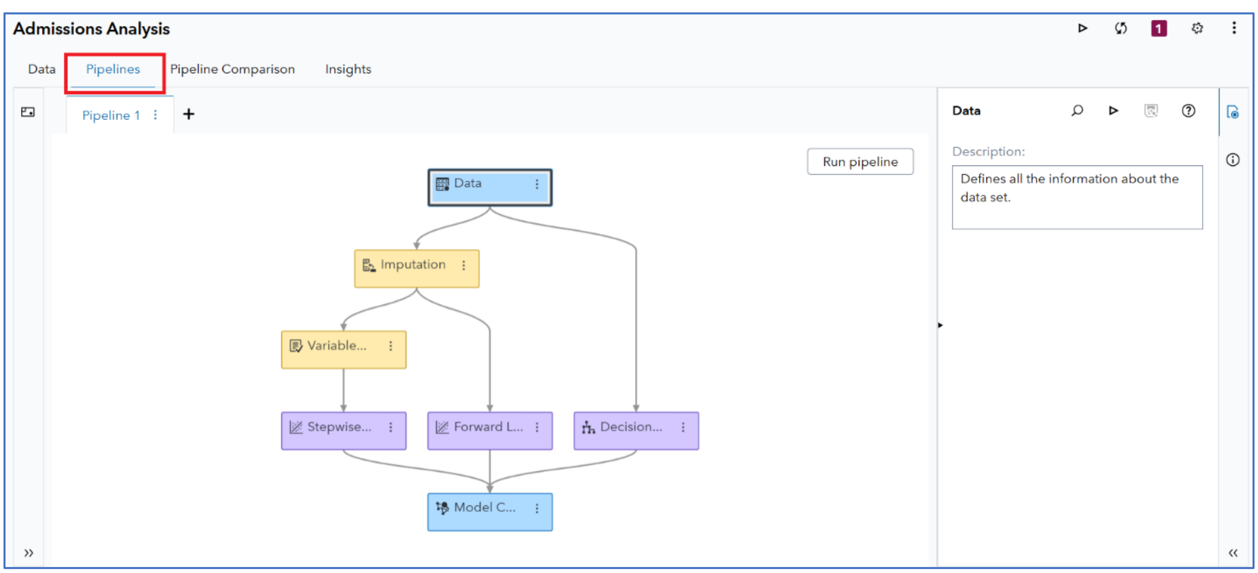
- And you know what, we're already there! Select **Admitted** and ensure that the **Properties** tab is active. Then you can change the **Role** to **Target**. Like so:



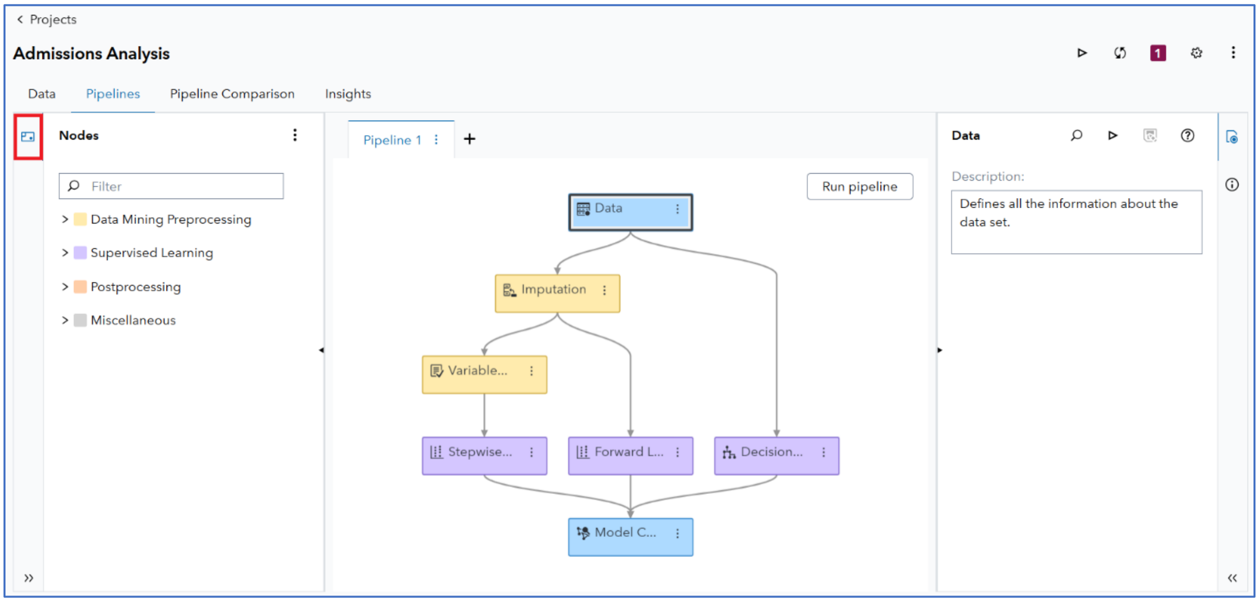
- Warning goes away! Yay! Spend a bit more time examining the variables, just so you get a little more comfortable. I don't see anything else that needs to be adjusted right now:

Variable Name	Label	Type	Role	Assess for Bias	Level	Order
Admitted	Admitted (Yes=1)	Numeric	Target		Binary	Defau
Analytics_Work_Experience	Analytics Work Experience	Numeric	Input		Binary	Defau
Country_Region	Country Region	Character	Input		Nominal	Defau
Cultural_Identity	Cultural Identity	Character	Input		Nominal	Defau
Gender	Gender Identity or Gender at Birth	Character	Input		Binary	Defau
ID	Application ID	Numeric	ID		Interval	Defau
Legacy_Admission	Legacy Admission	Numeric	Input		Binary	Defau
Mission_Statement	Mission Statement	Character	Text		Nominal	Defau
Standardized_Test_Score	Standardized Test Score	Numeric	Input		Interval	Defau
Strength_of_Recommendation	Strength of Recommendations	Numeric	Input		Interval	Defau
Undergrad_Degree	Undergraduate Degree Category	Character	Input		Nominal	Defau
Years_Work_Experience	Years Work Experience	Numeric	Input		Interval	Defau

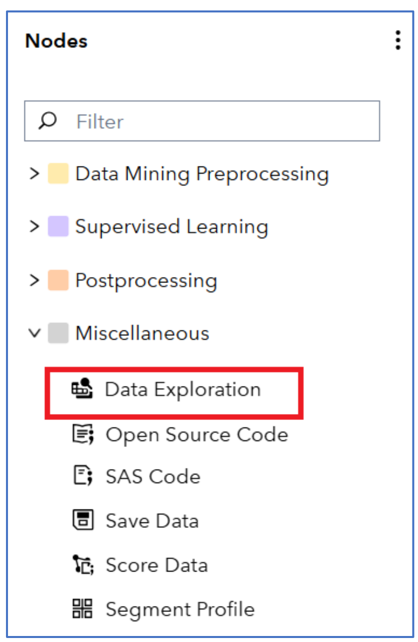
- Let's proceed to the **Pipeline** tab to unveil your first **SAS Model Studio** pipeline:



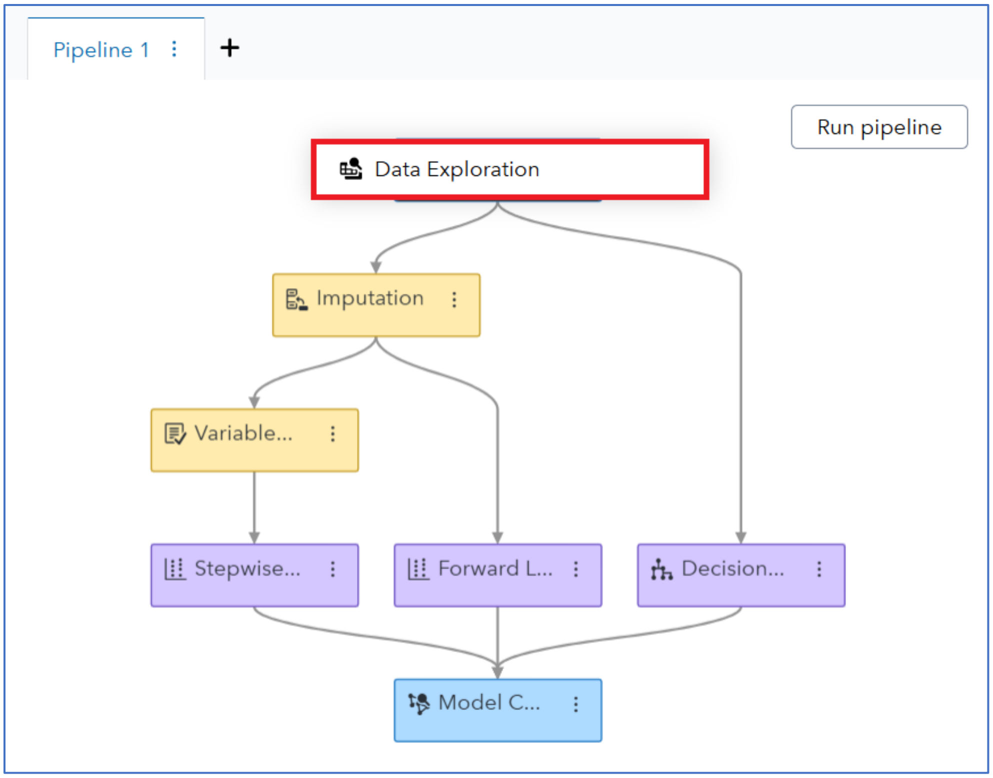
- Now we could just click **Run pipeline** and move straight on to the results from the three models from our Intermediate Template (which are Stepwise Logistic Regression, Forward Logistic Regression, and Decision Tree, respectively). But let's explore the data just a bit further. In your pipeline, find the **Nodes** icon and expand it:



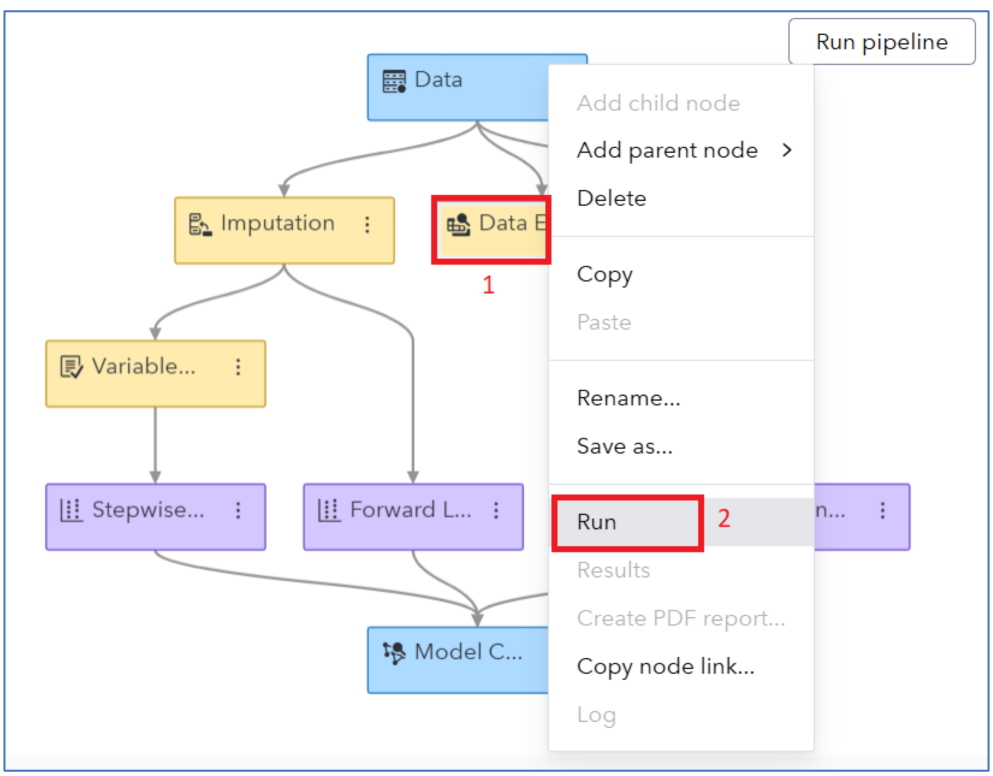
- Please explore the **Nodes** options just a little bit. Then find the **Data Exploration** node under the **Miscellaneous** section, here:



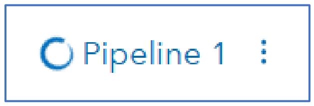
- Go ahead and drag-and-drop that **Data Exploration** node on top of the **Data** node, like so:



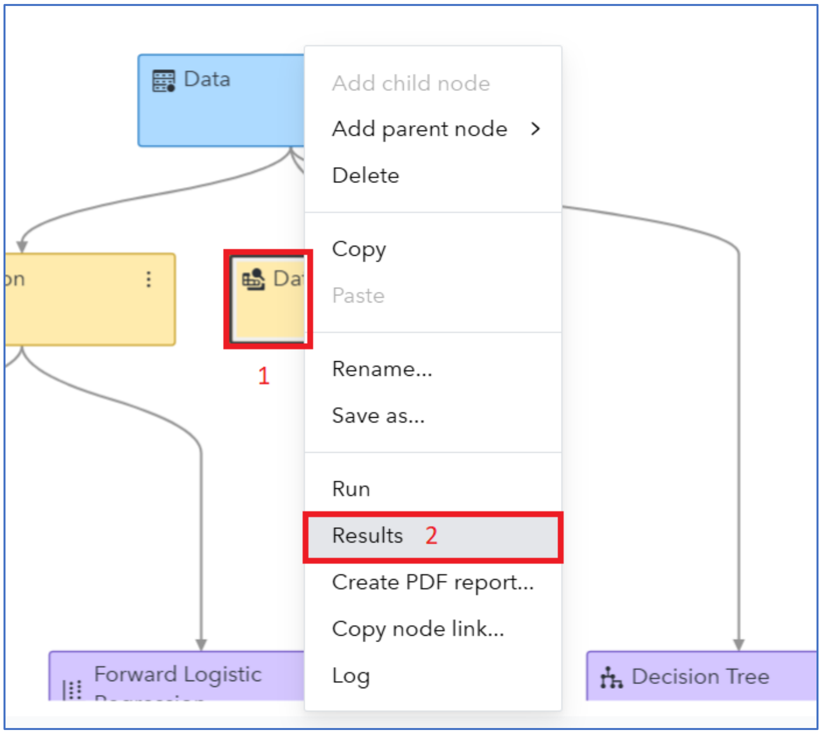
- Now look at that expanded pipeline. Right-click on the **Data Exploration** node and select **Run**:



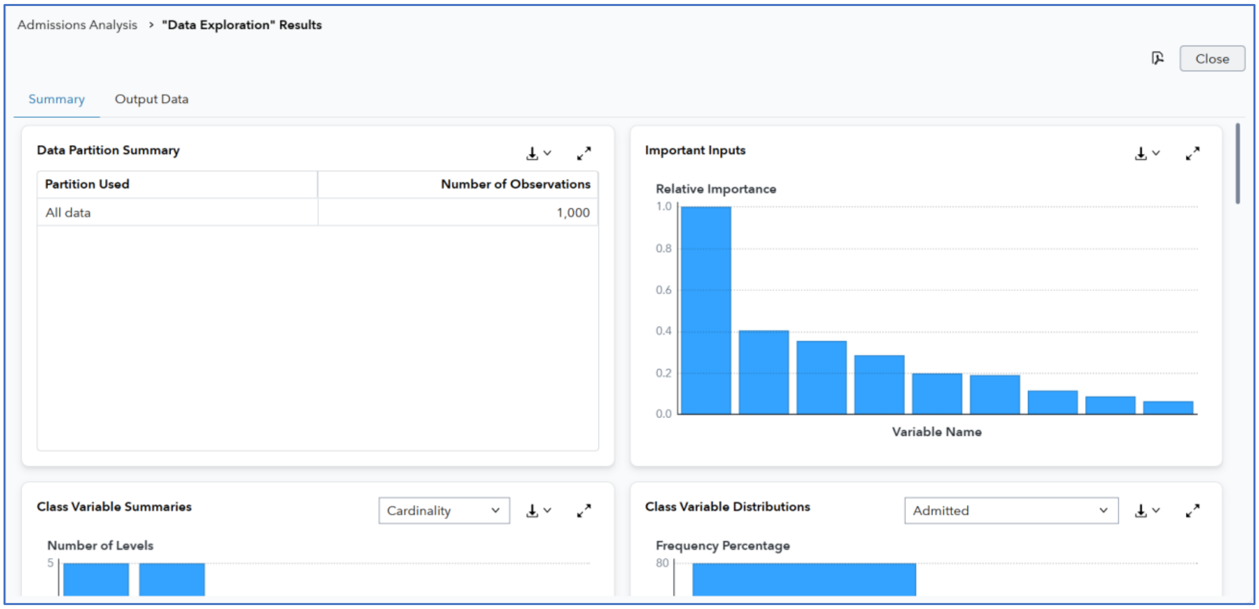
- The swirling magic means things are happening:



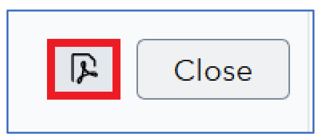
- A green checkmark means that the node has finished running. Right-click on the **Data Exploration** node when it is finished and select **Results**:



- A new window pops up with a lot of results to get you further acquainted with your data:



- Check it all out! There is a lot of really useful information in this summary, including missing values, correlations, interval moments, etc. So, spend a little bit of time here. And if you'd like to print a copy of the output to share with family, you're just a click away. Like here:



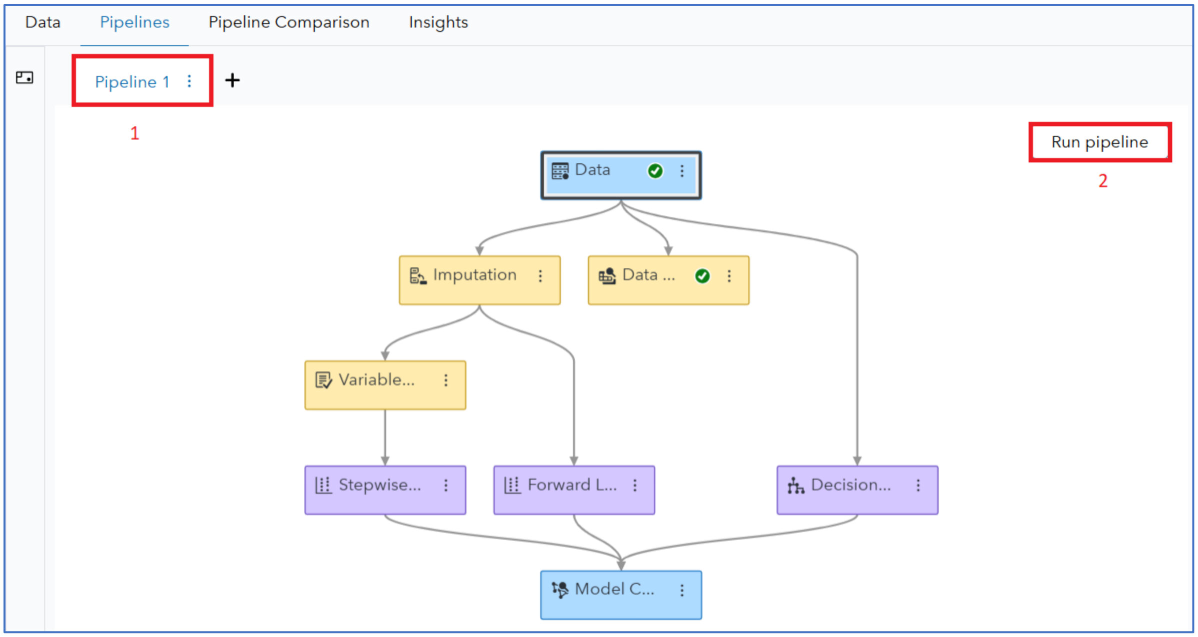
- Click that **Close** button when you're satiated. And congratulations – the **Explore the Data** section is completed!

Run Predictive Models

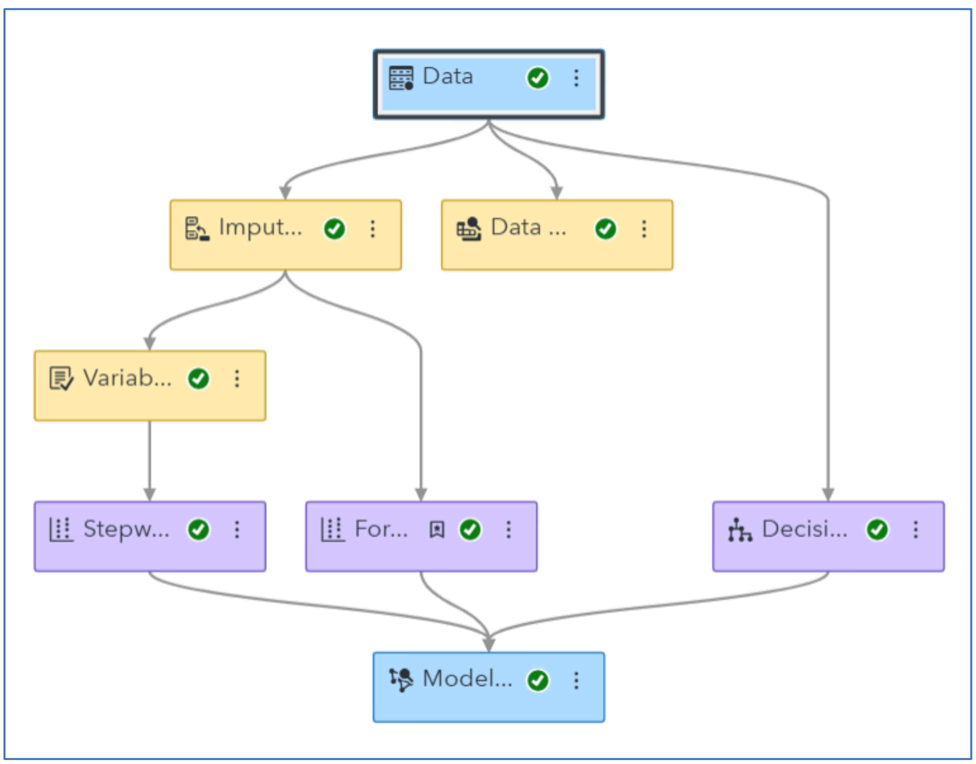
Let's be honest. Modeling is fun. Particularly with no-code tools, like SAS Model Studio. We can just select a bunch of models, hit a button, grab a coffee, and then come back to interpret the models. We can then make a couple of adjustments. Then it's run… rinse… repeat.

Let's do exactly that!

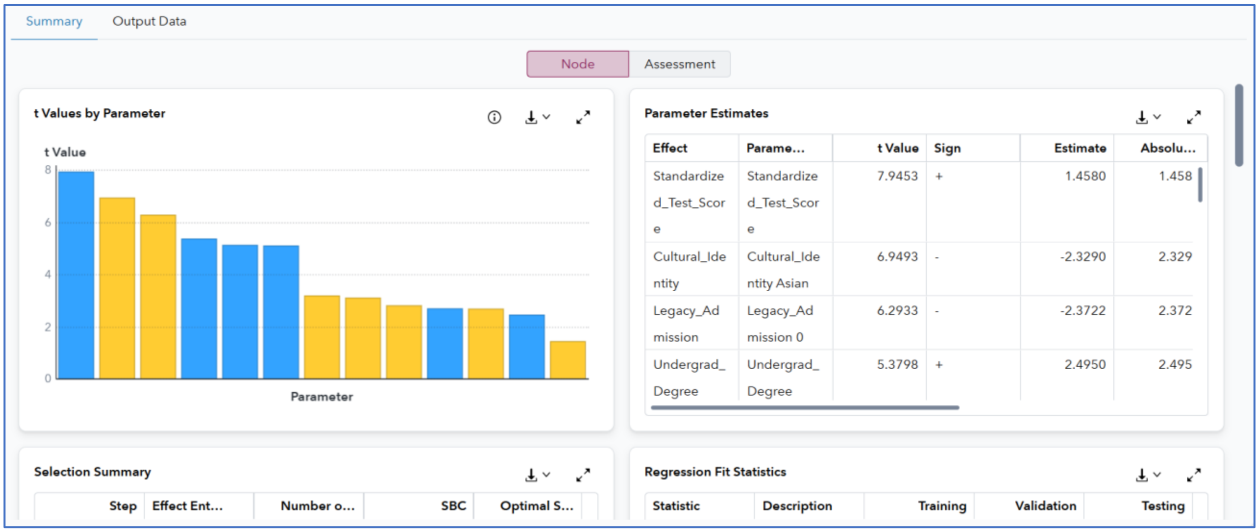
- If you've somehow navigated elsewhere, return to *Pipeline 1*. And click **Run pipeline** to run the models with their default settings:



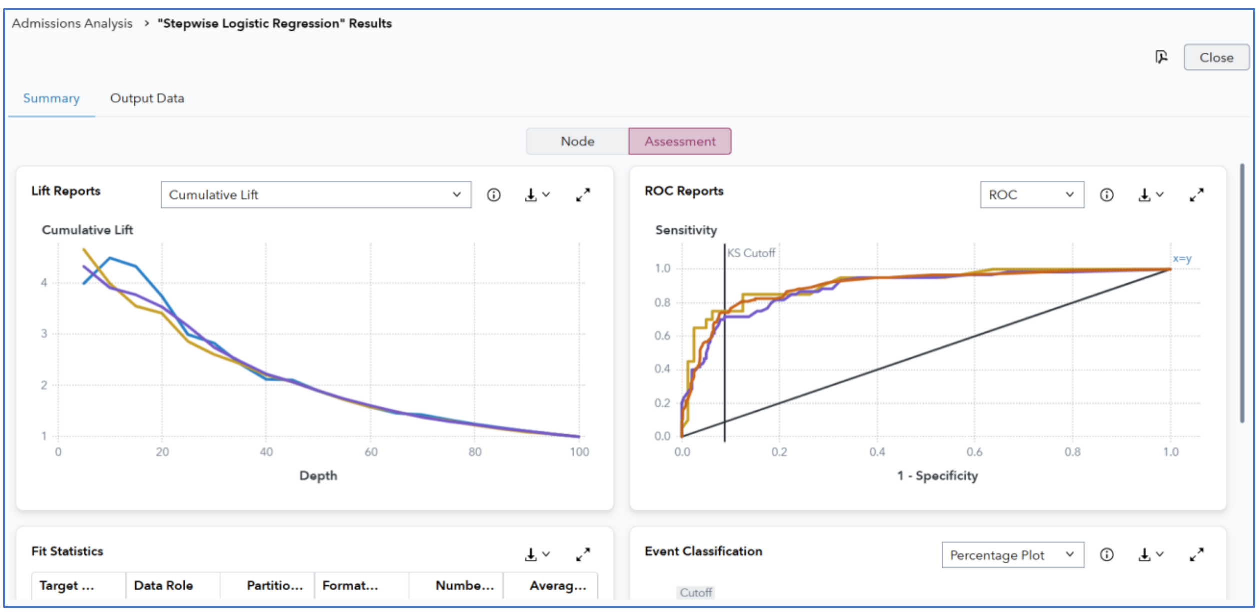
- Green checkmarks everywhere means that (modeling) life is good:



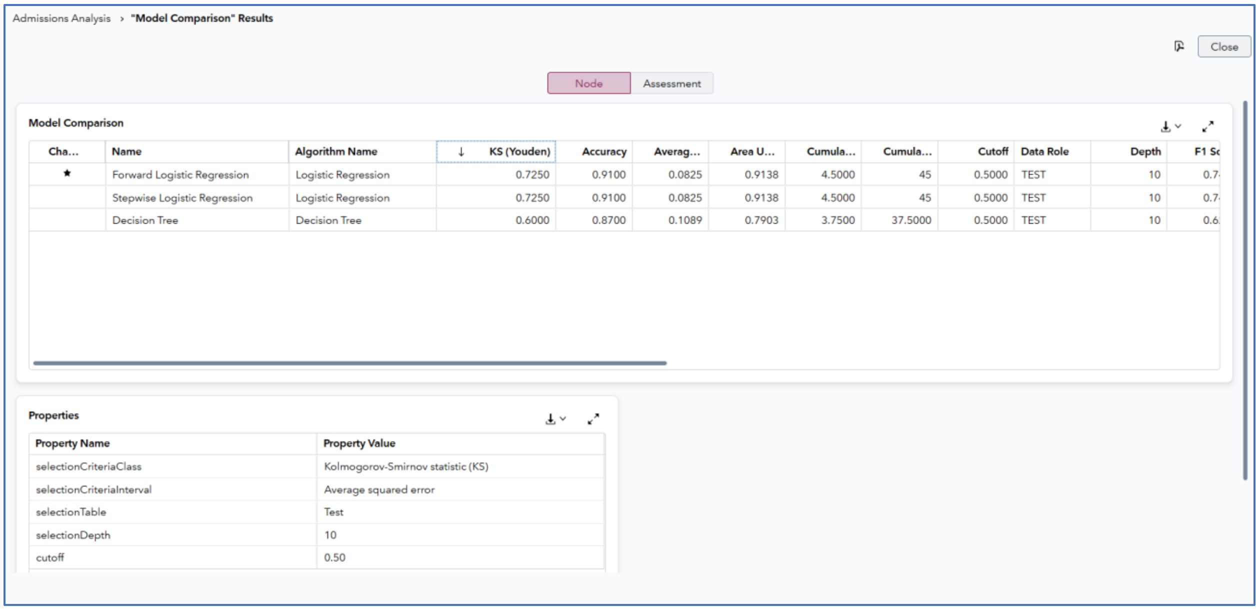
- To explore the models individually, right click on a modeling node and select **Results**. For example, here are the **Results** for the **Stepwise Logistic Regression** model:



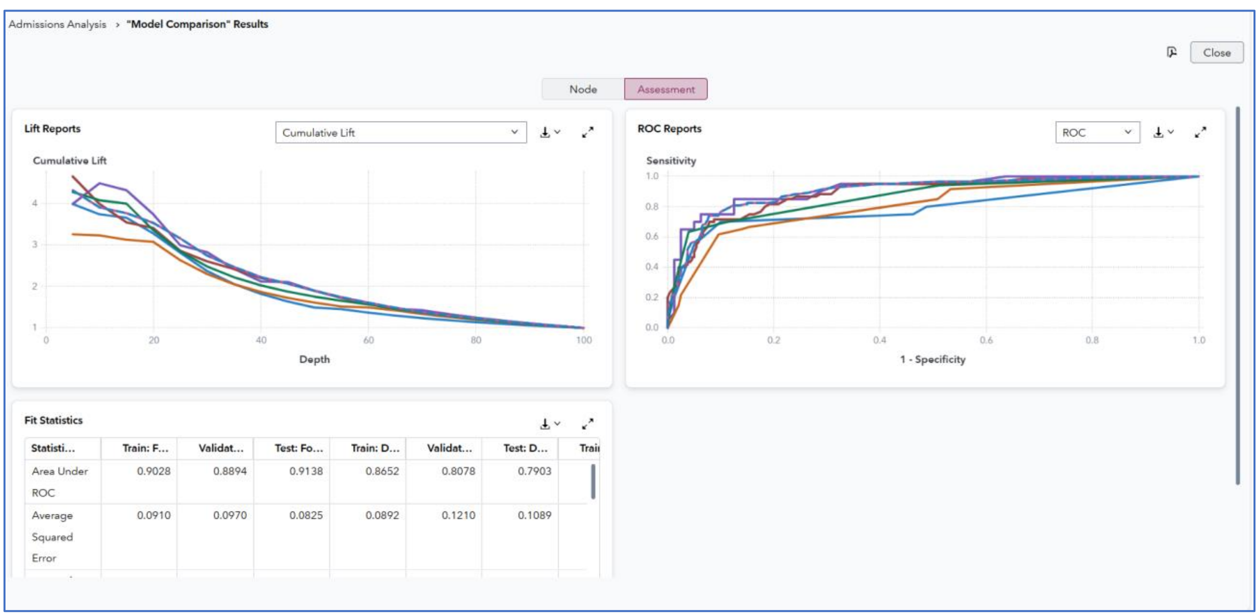
- The **Node** tab gives you traditional statistical measures like parameter estimates, selection summary (if applicable), regression fit statistics, and score code.
- Click **Assessment** for an interactive dashboard of MANY model assessment fit statistics and tools:



- And explore a bit 😊 Click **Close** when you're done with the Stepwise Logistic Regression model.
- You may think, *do I need to explore all the models individually and then choose the "best" model in this pipeline?* Nope. You sure don't. The **Model Comparison** node handles that for you. So right-click on that node and select **Results** to compare the three models in the pipeline:

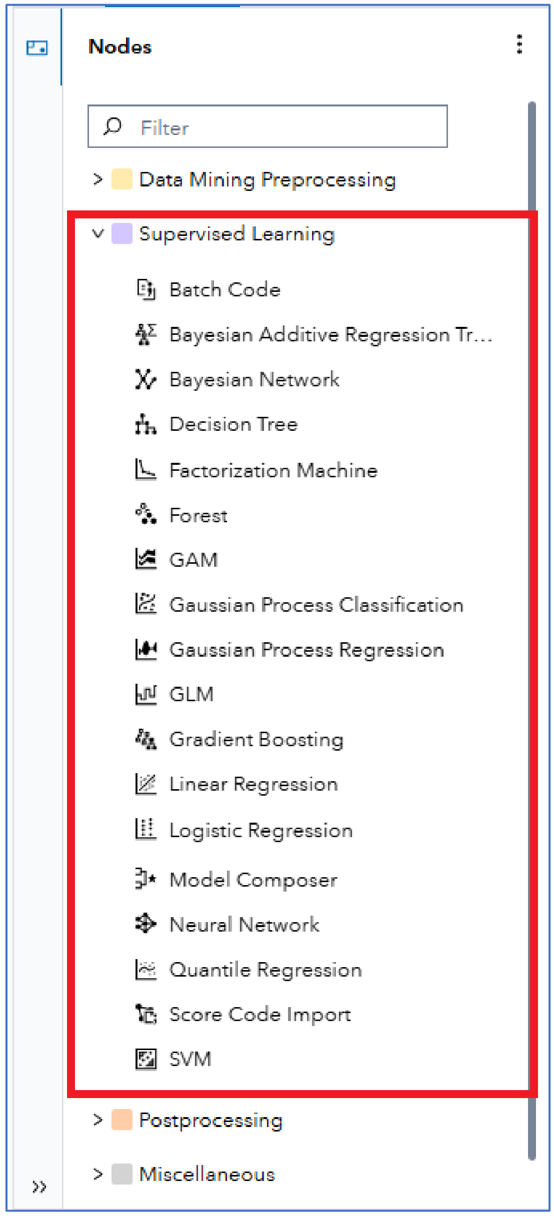


- With the **KS (Youden) statistic** as our default selection criteria for a classification target, we can see that the Forward Logistic Regression and Stepwise Regression models appear to select the same model, with a KS (Youden) statistic of 0.7250. The Decision tree performs far worse. You can also click on the **Assessment** tab to explore a variety of other model selection criteria:

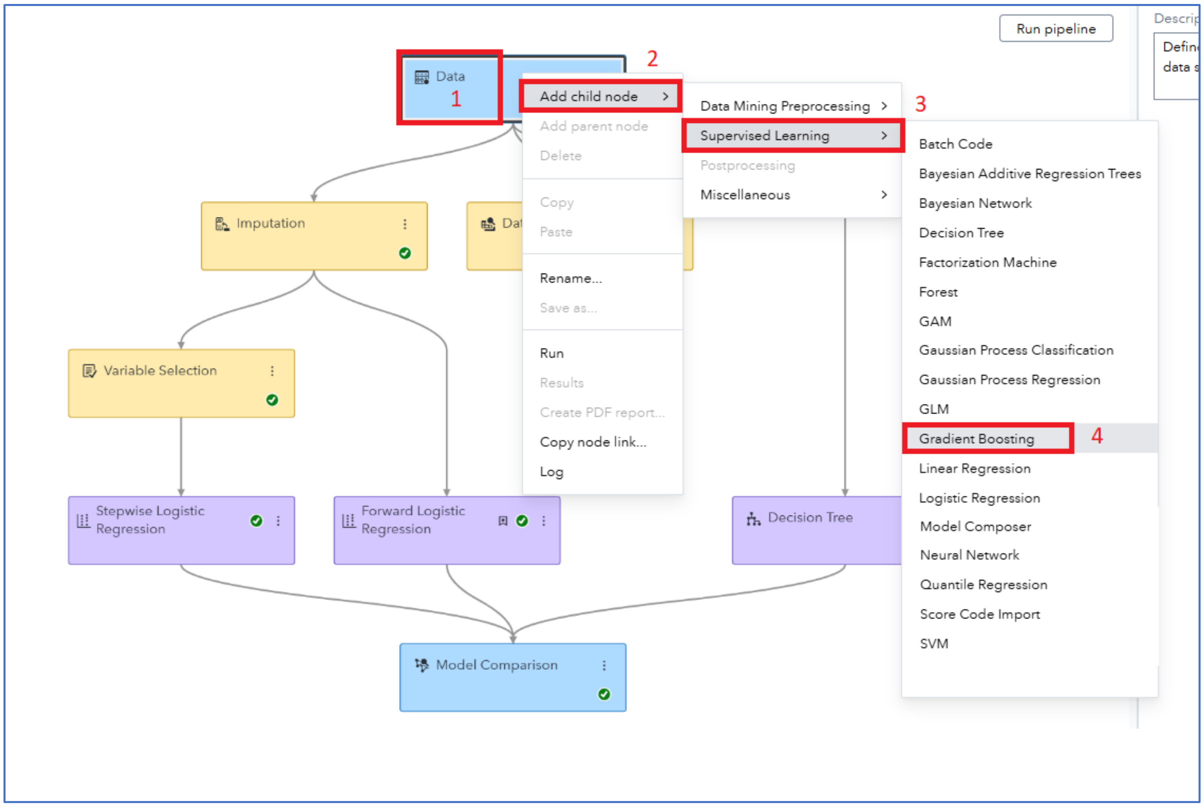


- Explore a bit, then hit **Close** when you're done.
- But… in my preamble didn't I promise machine learning models? I did. I sure did. So, let's add one true machine learning model to our pipeline. You can then tweak and add more models to your pipeline later.

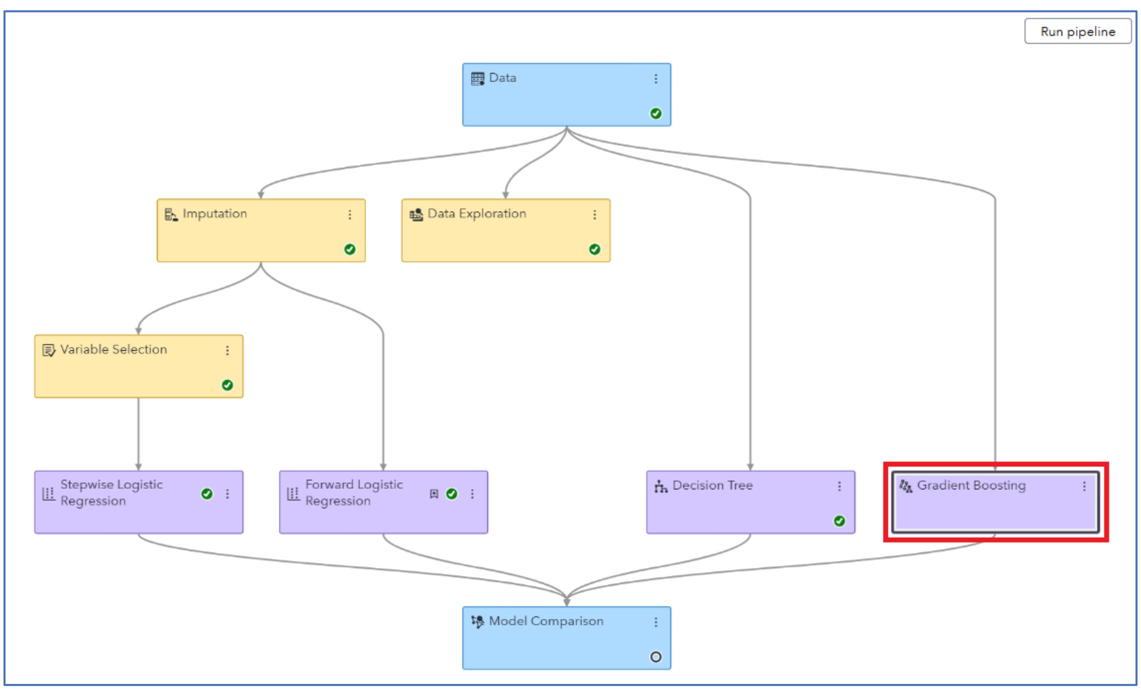
- Go back to your original pipeline. Expand the **Supervised Learning** node to explore all the modeling options available to you:



- Yup – it's a lot! Now I'll show you another way to add a node to the pipeline… one that isn't drag-and-drop (because that's not everyone's cup of tea). Find the **Data** node inside your pipeline. Right-click, select **Add child node** >> **Supervised Learning** >> **Gradient Boosting**. Like so:



- Gradient Boosting is a proper machine learning model. And shows up in the pipeline here:

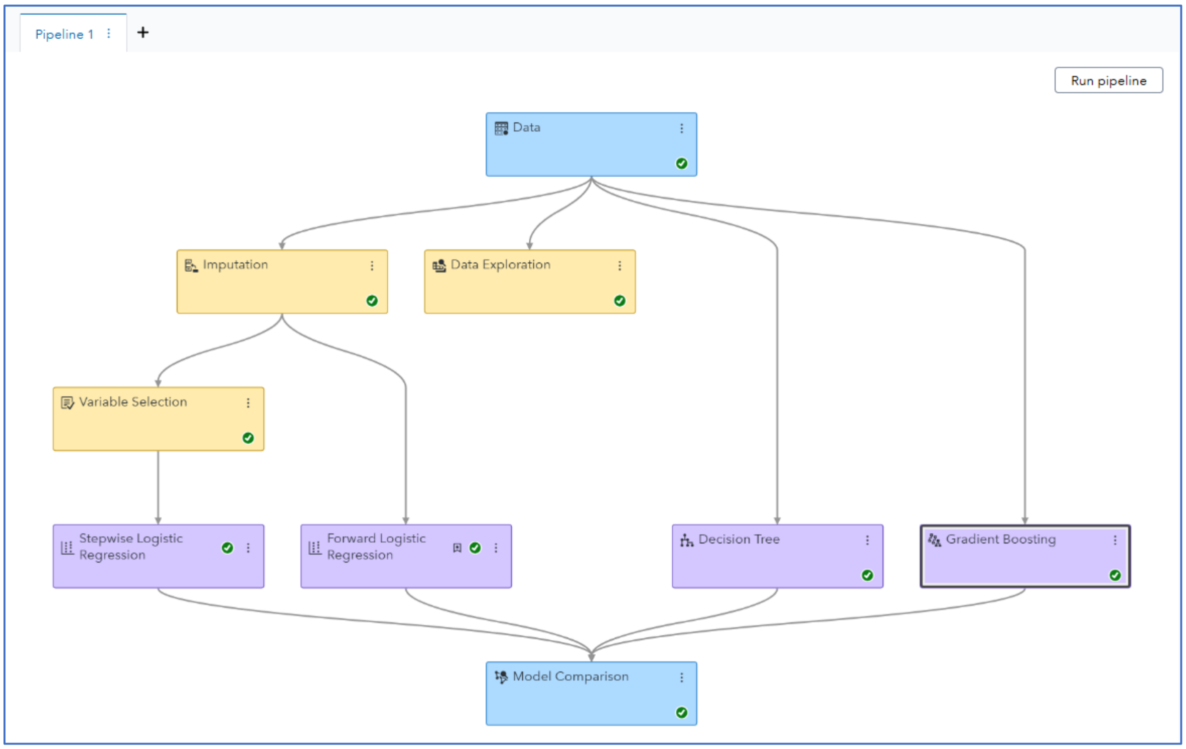


- Click **Run pipeline**. And then I'll see you in the next section. Great work!

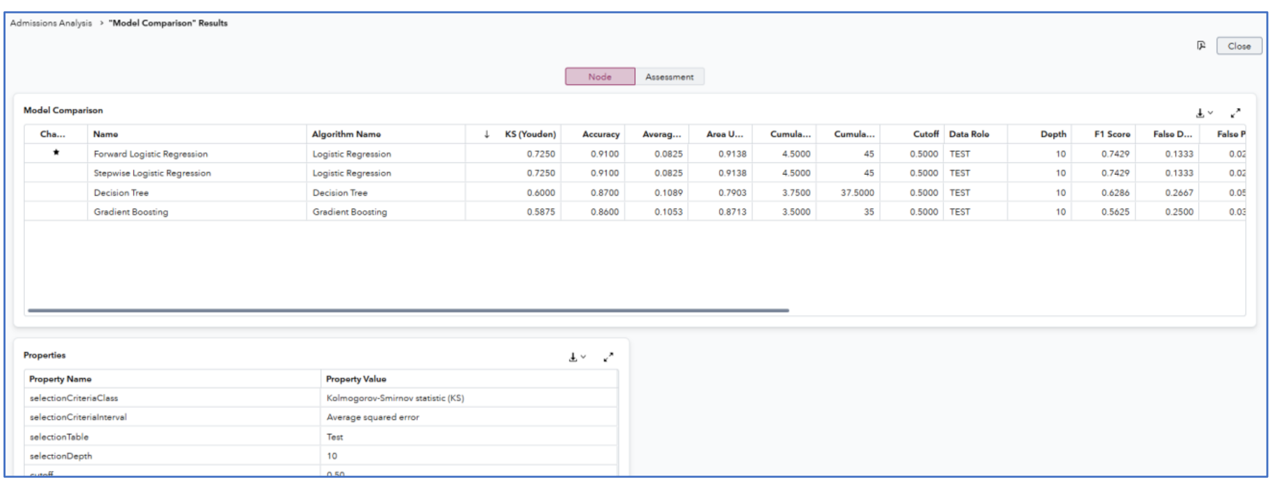
Crown a champion model

Modeling is fun. But we're not just doing modeling for modeling's sake. We're actually on a mission. And that's to choose the incoming class of 2025. Let's get on it!

- If you got lost, navigate back to *Pipeline 1*. You should have models to compare and a champion waiting under **Model Comparison**:



- A right-click and selecting **Results** on the **Model Comparison** node yields:



Model Comparison

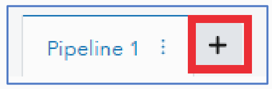
Cha...	Name	Algorithm Name	KS (Youden)	Accuracy	Averag...	Area U...	Cumula...	Cumula...	Cutoff	Data Role	Depth	F1 Score	False D...	False P
★	Forward Logistic Regression	Logistic Regression	0.7250	0.9100	0.0825	0.9138	4.5000	45	0.5000	TEST	10	0.7429	0.1333	0.02
	Stepwise Logistic Regression	Logistic Regression	0.7250	0.9100	0.0825	0.9138	4.5000	45	0.5000	TEST	10	0.7429	0.1333	0.02
	Decision Tree	Decision Tree	0.6000	0.8700	0.1089	0.7903	3.7500	37.5000	0.5000	TEST	10	0.6286	0.2667	0.05
	Gradient Boosting	Gradient Boosting	0.5875	0.8600	0.1053	0.8713	3.5000	35	0.5000	TEST	10	0.5625	0.2500	0.03

Properties

Property Name	Property Value
selectionCriteriaClass	Kolmogorov-Smirnov statistic (KS)
selectionCriteriaInterval	Average squared error
selectionTable	Test
selectionDepth	10
cutoff	0.50

- Whomp, womp. The Gradient Boosting model was, well, not good. Huh.

- Anyway, here is your homework for this section:
    - Add more models to the existing *Pipeline 1* template
    - Feeling brave? Add a **Text Mining** node as a child node to the **Data** node.
        - Does *Mission Statement* have any predictive power (hint, hint: it should).
        - And if you don't feel brave – no worries. I'll show you how to do it later.
    - Feeling even braver? Click the Add new pipeline button here:

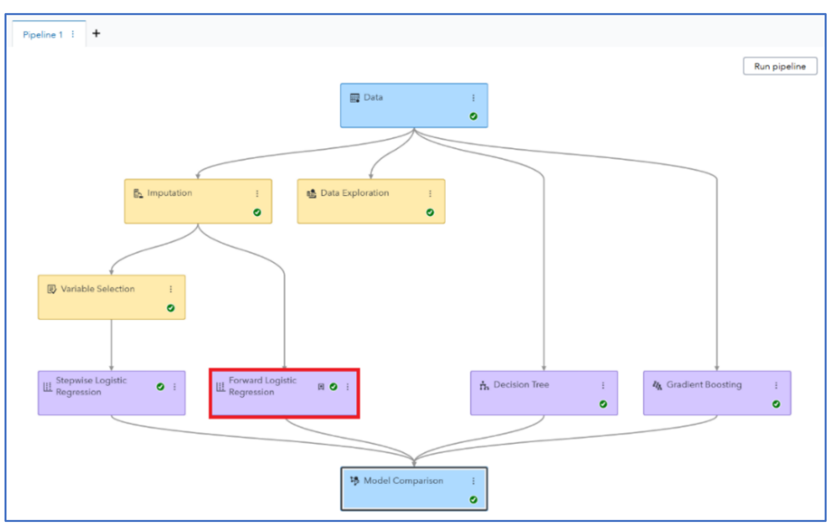


        - Give the *Advanced Template for class target* a try and further refine those models!
- When you are finished, identify your champion model – i.e., the model with the best predictive power in your modeling.
    - Welcome champion!
    - Remember this champion for the very important next section

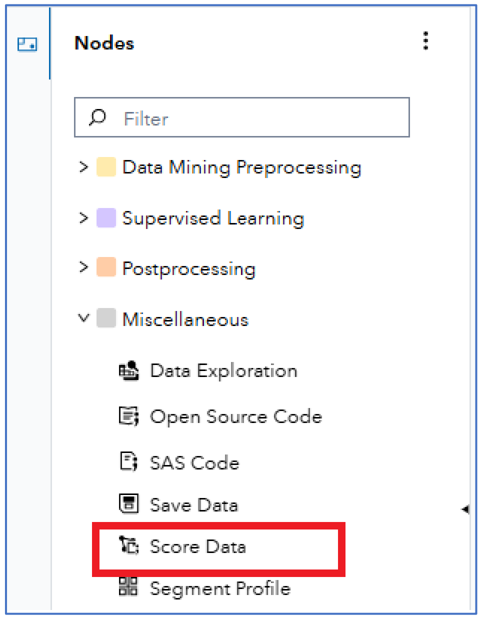
Select the Incoming Class of 2025

The big last step is here. You found a model. It predicts great. Now you've got to put it into action. That's called "scoring" in a predictive modeling sense. To select our cohort, we'll use data from the current pool of applicants to select those most similar to successful applicants of the past 5 years. And then we'll select the top 40 candidates that are most similar to our past students and then offer them admission. Let's get at it!

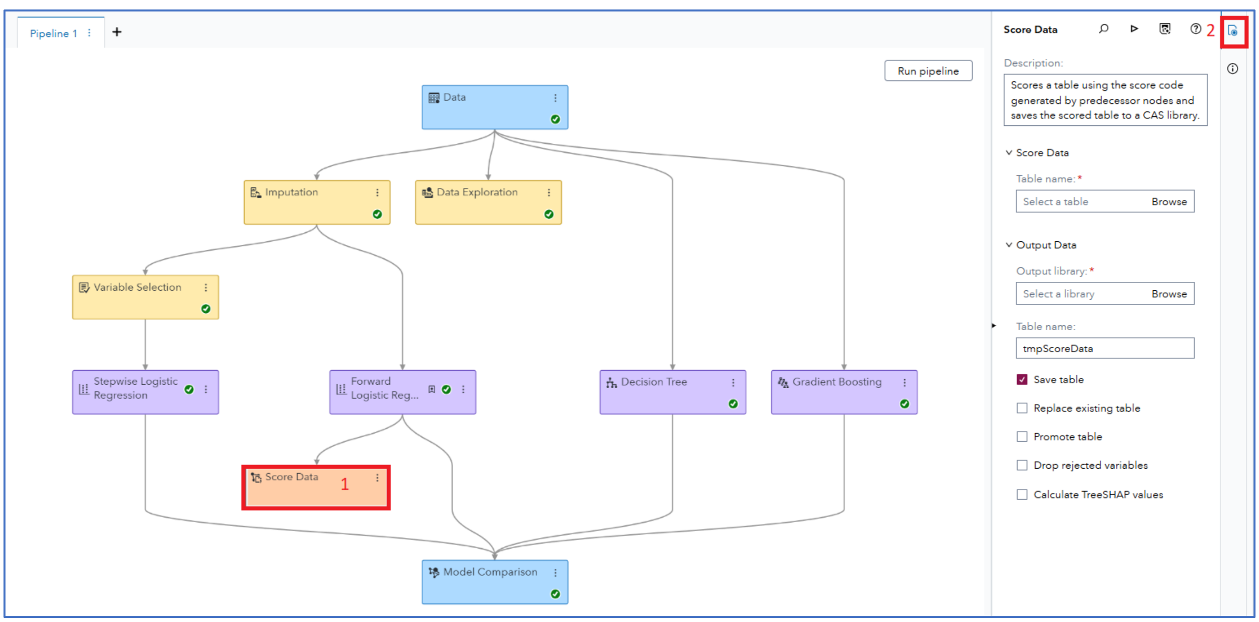
- While there are many ways to score new cases in SAS Viya, we'll keep it simple and just use the **Score Data** node in SAS Model Studio. Find your champion model in its respective pipeline. Since I'm keeping things simple, my champion is still the **Forward Logistic Regression** model, found here:



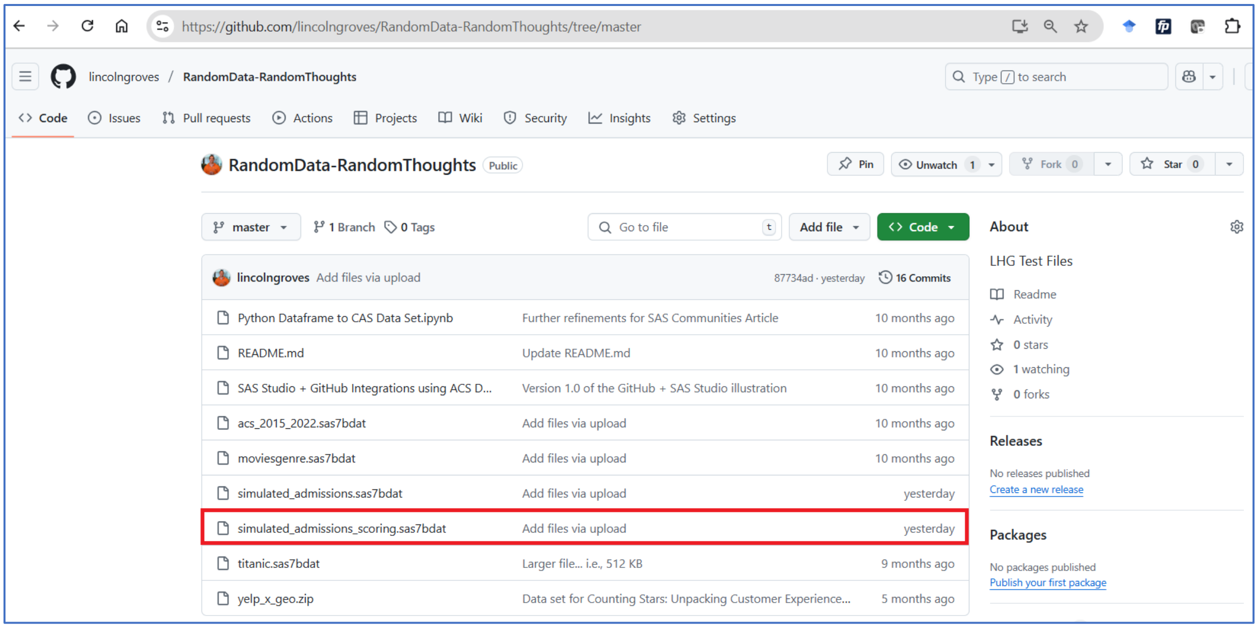
- Expand the **Miscellaneous** nodes again and find the **Score Data** node. Like here:



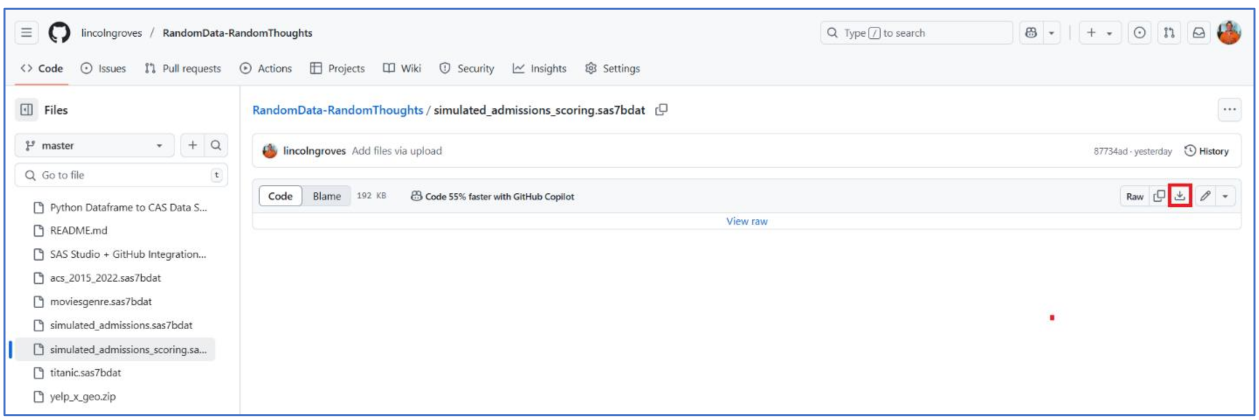
- Drag-and-drop the **Score Data** node on top of your Champion Model. Also ensure that the **Node options** are expanded for the Score Data node. Like so:



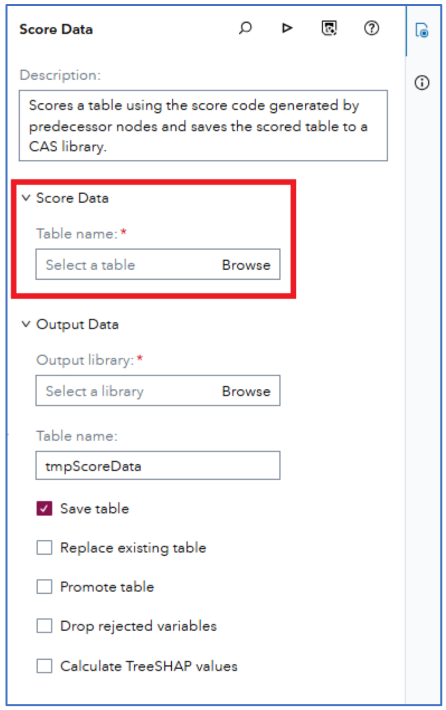
- Let's do a deep dive into those Score Data node settings. To start, we need to pull the data on the students who've applied to be part of our 2025 cohort. Data are again on GitHub, in the RandomData-RandomThoughts depo here: https://github.com/lincolngroves/RandomData-RandomThoughts
- Find – then click – the *simulated_admissions_scoring* data set:



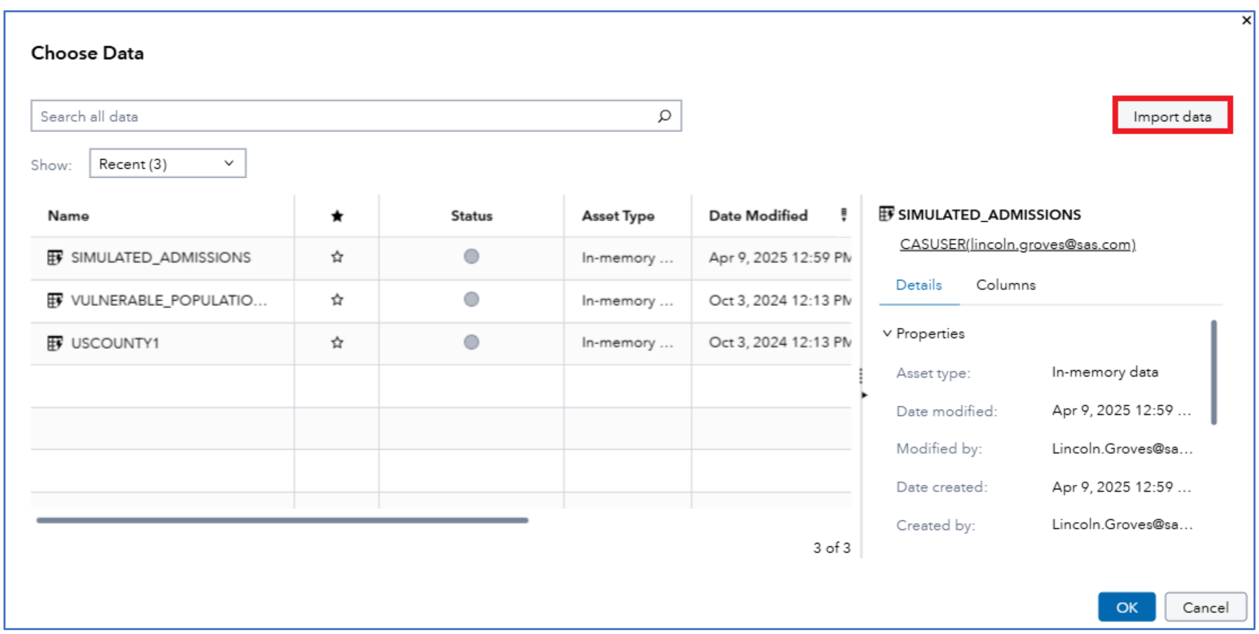
- And download the file to your local drive, again, by clicking the **Download raw file** button, here:



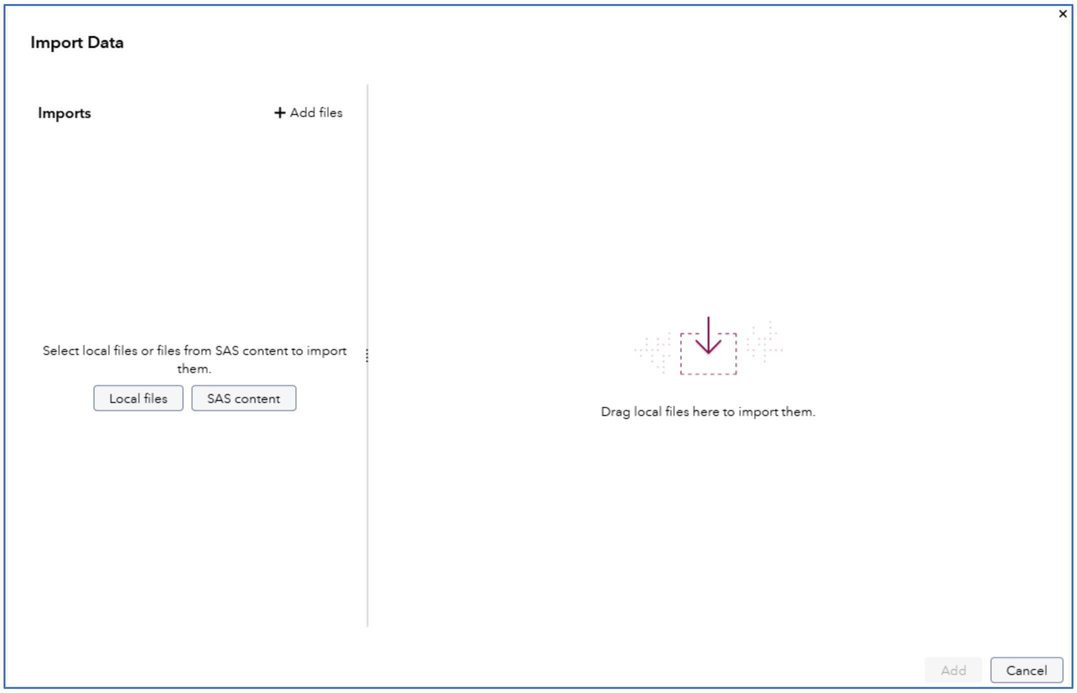
- With file in hand, we can return to SAS Model Studio in SAS Viya. Next, find the **Score Data** section in the **Node Options**, here:



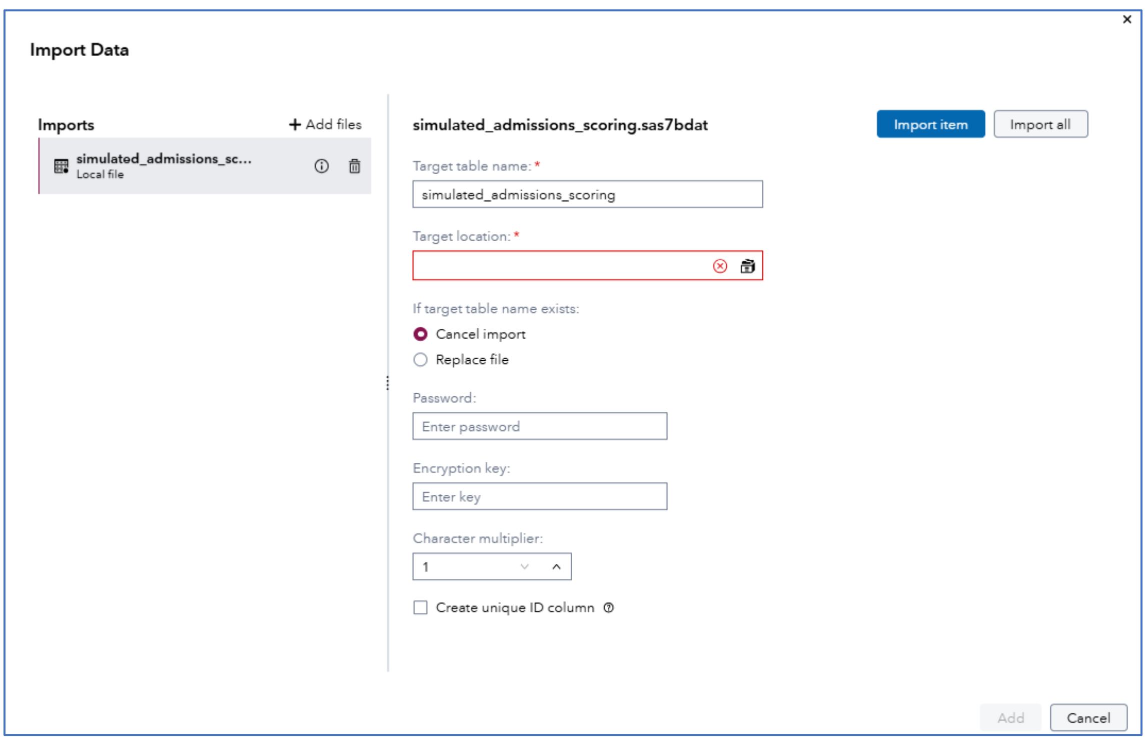
- Click that **Browse** button. It should default to your CASUSER folder, where you can then select **Import data**:



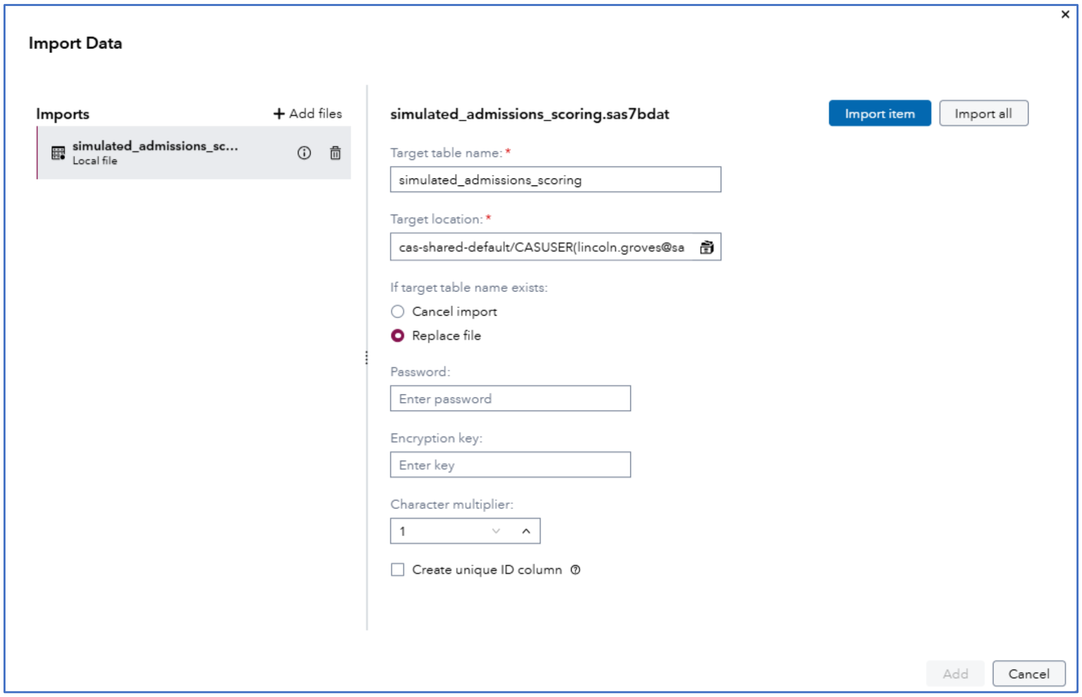
- We've been here before!



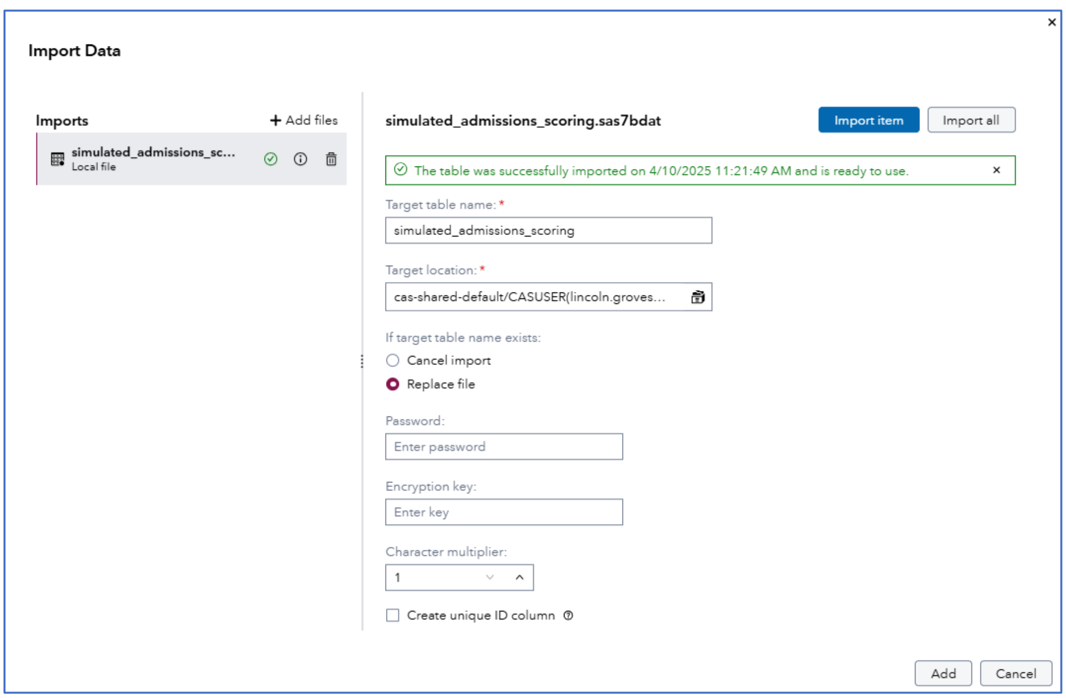
- Click **Add files** >> **Local files**. Then add *simulated_admissions_scoring.sas7bdat* – which you just downloaded. We're then here:



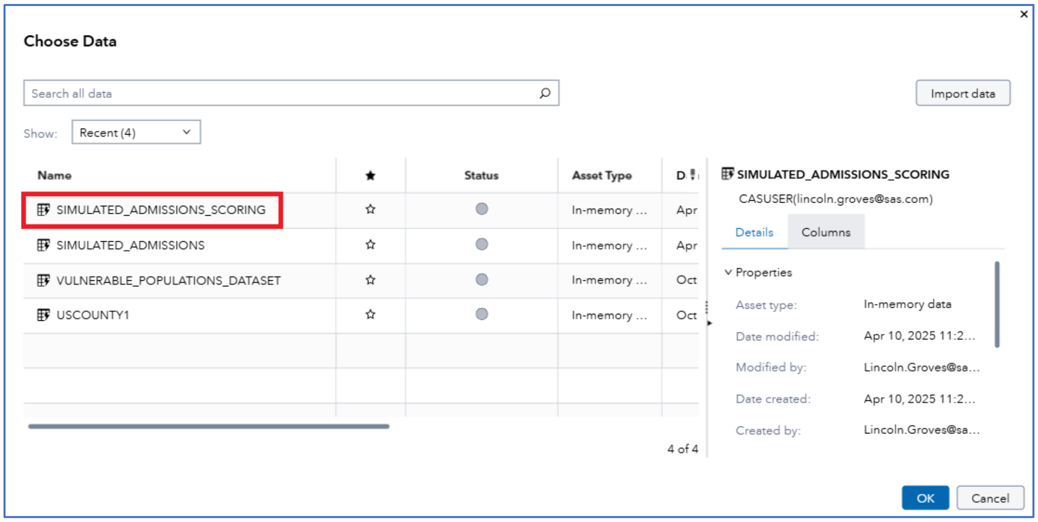
- Just a couple of clicks to go! For **Target location**, we'll again choose our CASUSER folder. Under **If target table name exists**, select **Replace file**. You'll then see:



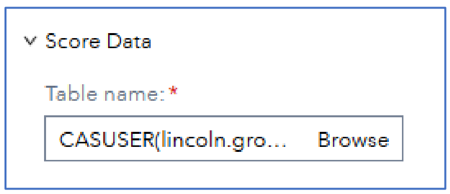
- Just two more clicks! Click **Import item**. Success is a happy green message:



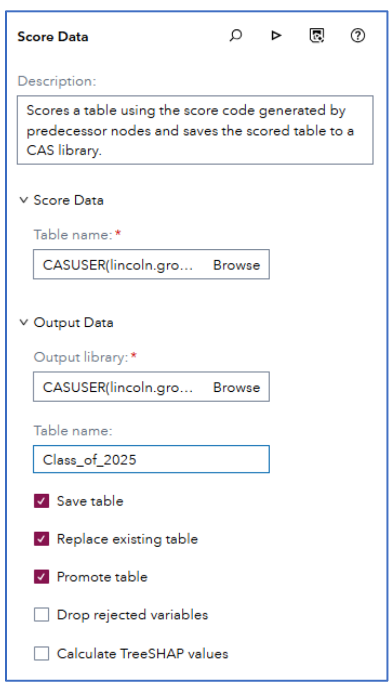
- Then click **Add**. You should now see the table in-memory in the **Choose Data** window:



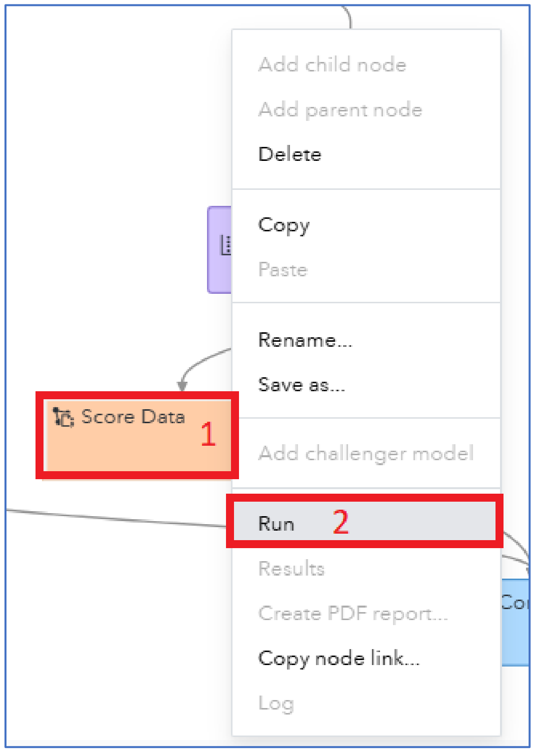
- Click on **SIMULATED_ADMISSIONS_SCORING** and then click **OK**. You've now got this year's applicants loaded into the node:



- Let's make a couple of extra changes. For **Output library**, select **Browse** and then navigate (again) to your CASUSER folder. For **Table name**, put **Class_of_2025.** Then click the options to (1) **Replace existing table** and (2) **Promote table**. All those options, succinctly summarized:

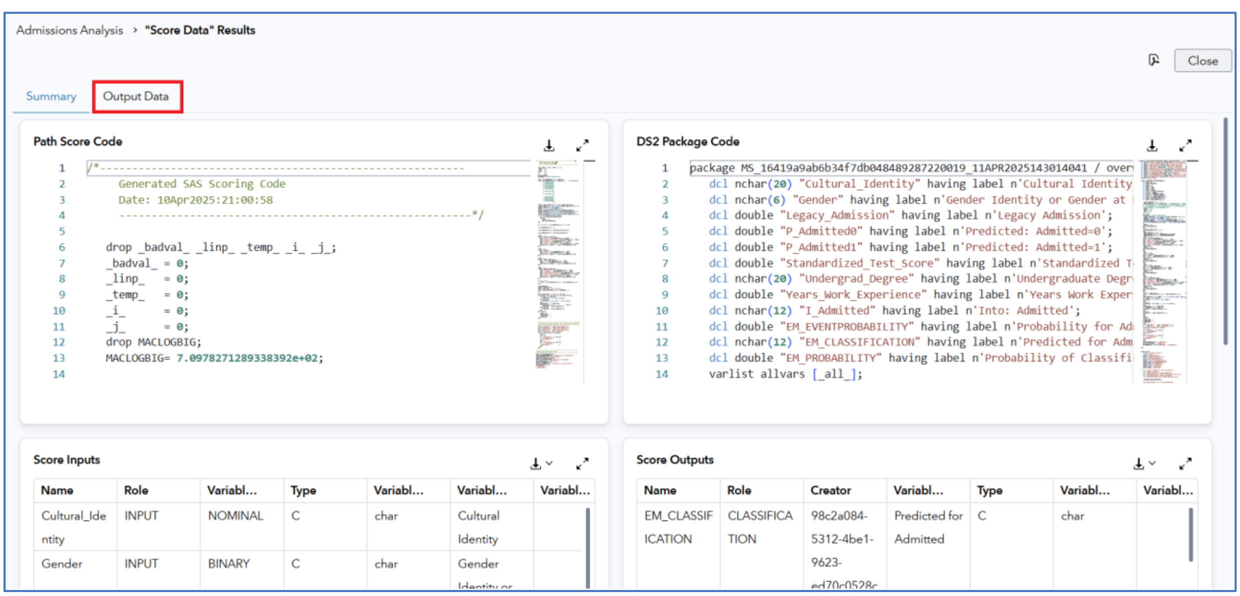


- What's next? Right-click on that **Score Data** node and select **Run**!

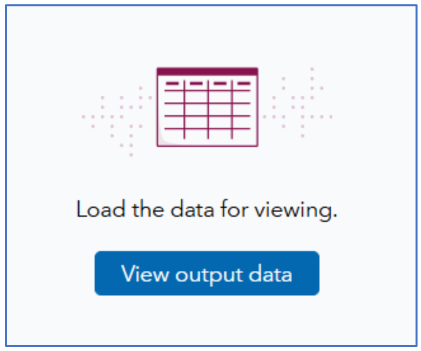


- Score code happiness looks like

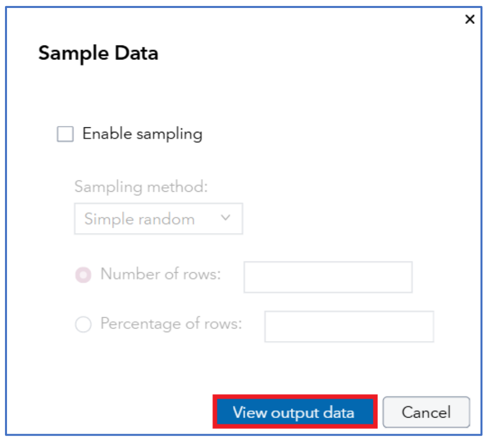
- And we have predictions! Right-click on the Score Code node and then select **Results**. The **Score Data Results** window pops up. Select the **Output Data** tab, here:



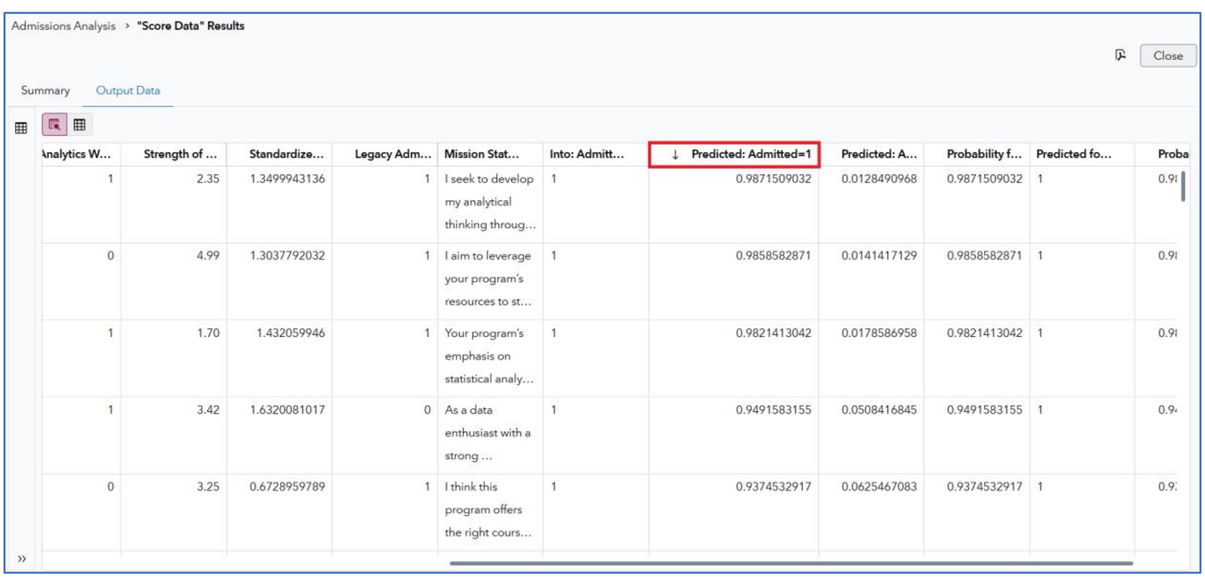
- And then you can click **View output data** to… yes… view the output data.



- Almost there! A **Sample Data** window pops up. Since we had 200 applicants, let's check them all out. Click **View output data** and skip the sampling:



- Once the data are loaded, we have a sortable data set for viewing. You can expand the view and then scroll to the right. Finally, sort – descending – on the **Predicted: Admitted=1** column, here:

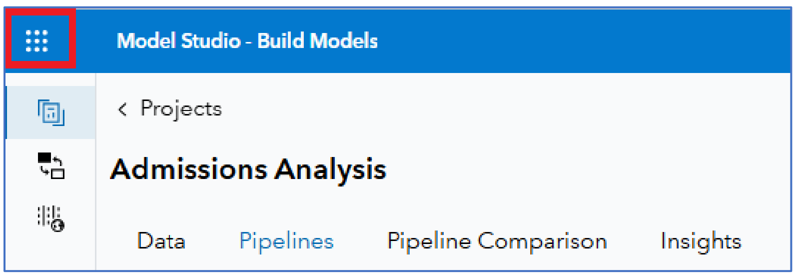


- The top 40 candidates, sorted by *Predicted: Admitted=1* will comprise our incoming Class of 2025. You did it! Let's get that list over to HR and get those letters ready!
- Click **Close** to wrap up **Part 1: Business as Usual**… or accept the challenge in the **Bonus Section**!

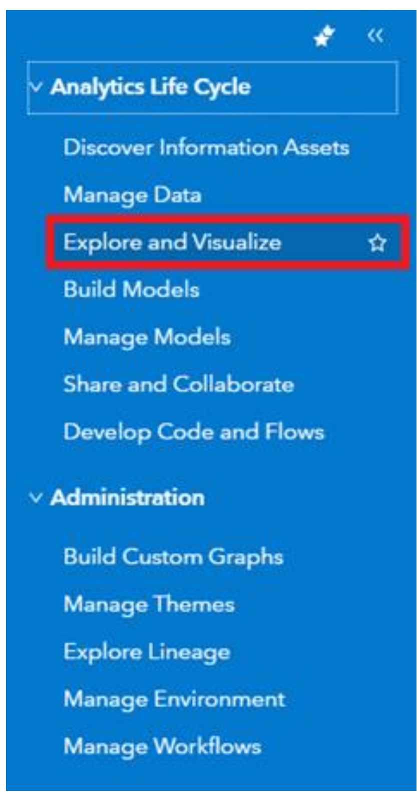
Bonus Section: Admissions Dashboard

The last section may have concluded before you were ready. You might think – "hey, what are the characteristics of the students accepted?" And that's a great question. In this section, we'll make a quick dashboard in SAS Visual Analytics to examine some demographic characteristics of the accepted students. Here's how:

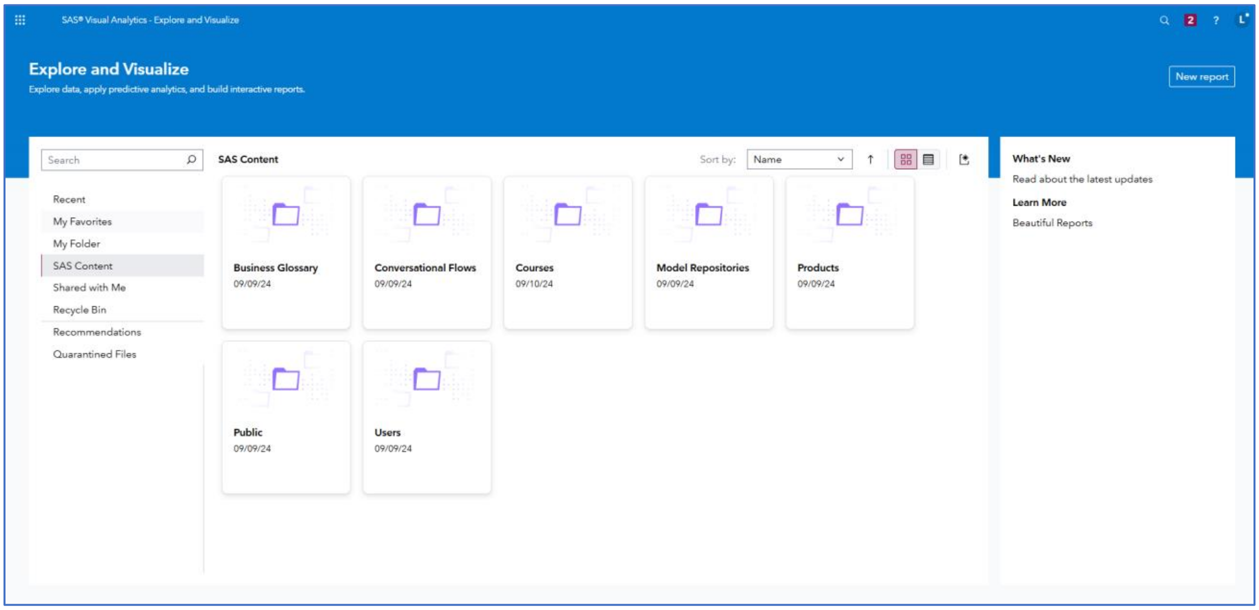
- Did you know that you already saved the dataset into memory so that it can be used in SAS Visual Analytics? No? I'll prove it. Start by locating the **Applications menu**, here:



- Click it, then select **Explore and Visualize**:

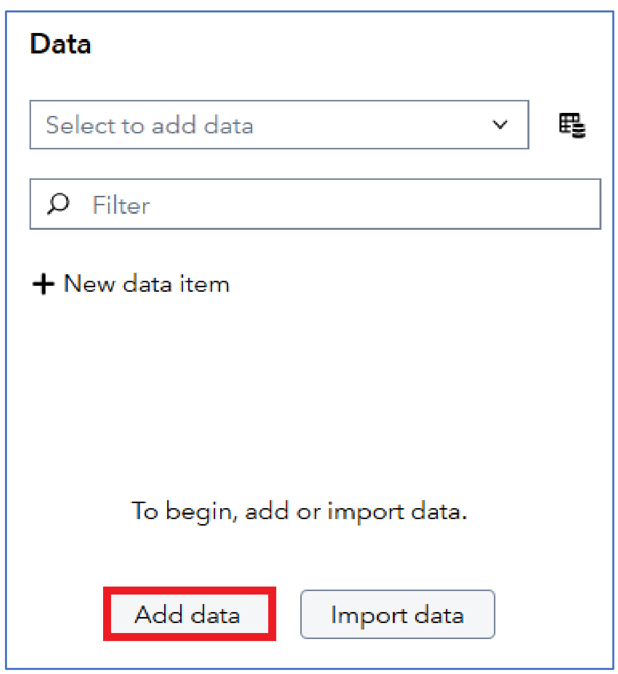


- You are taken to the **SAS Visual Analytics** home page:

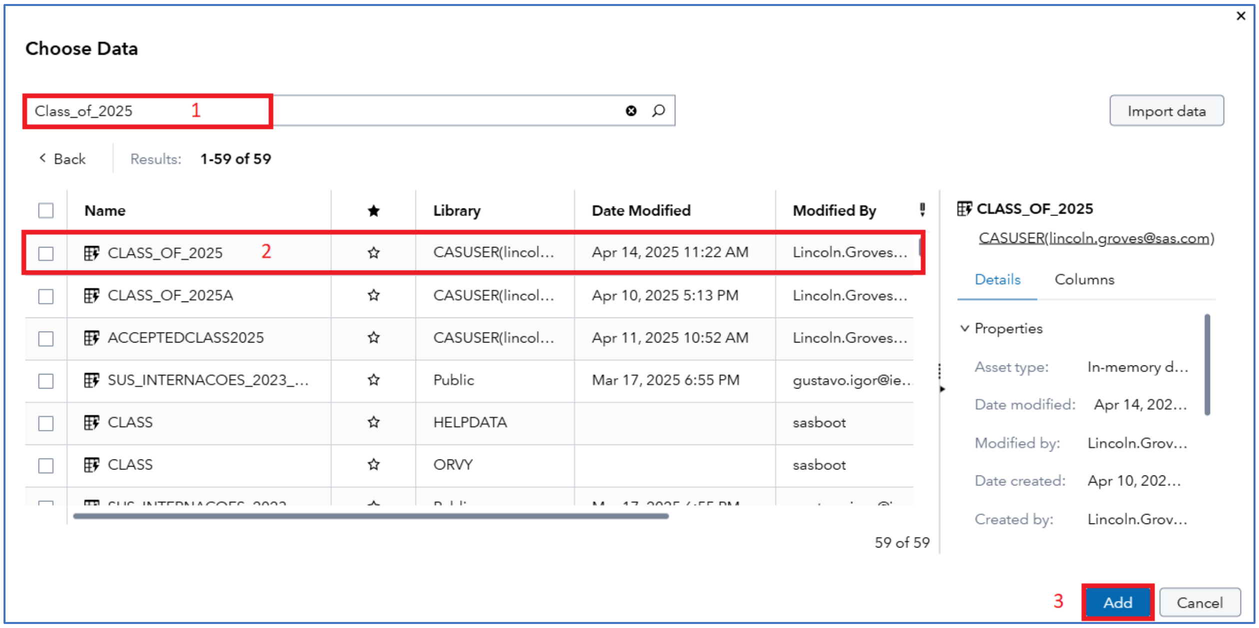


- Click that **New report** button in the upper right corner:

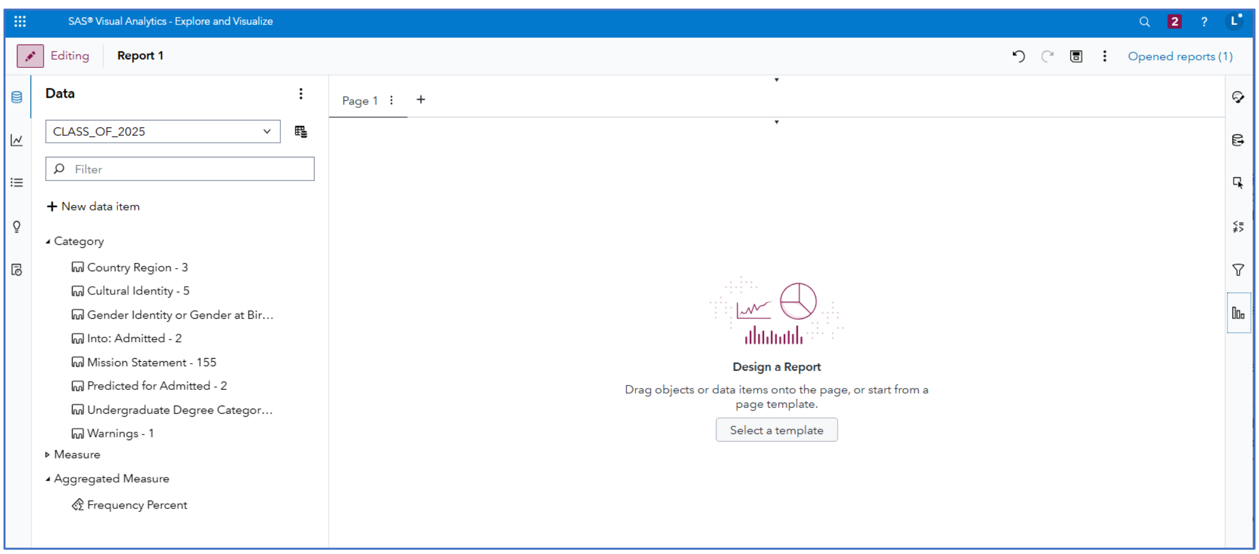
- And… in the **Data** section, you can select **Add data**:



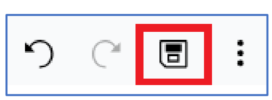
- In the **Search all data** bar, type **Class_of_2025**. There may be a lot of files to choose from, so select the right *Class_of_2025* file and then click **Add**:



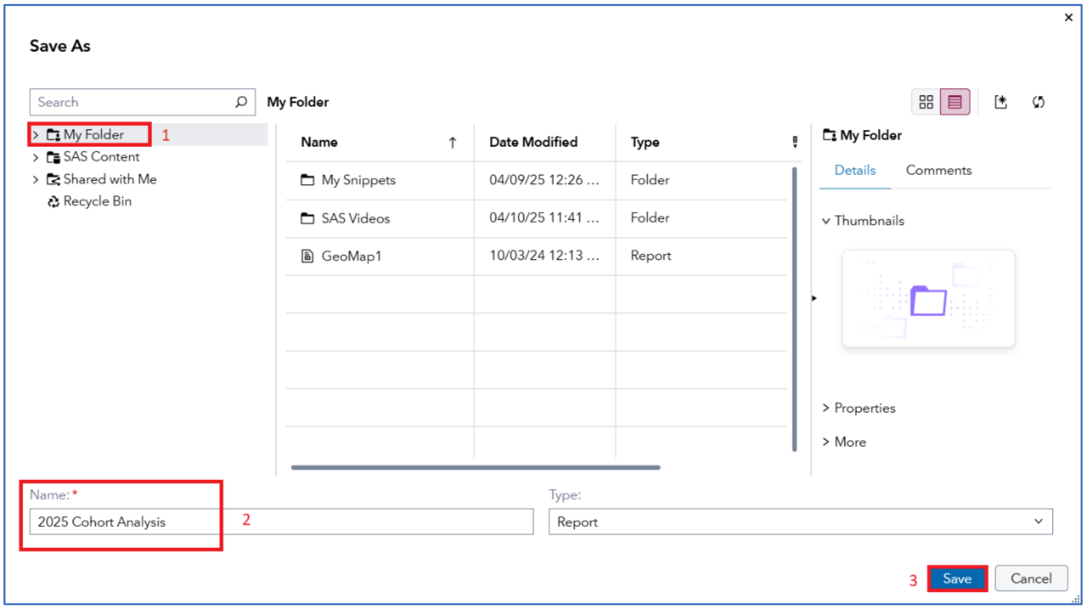
- You've got data in your report!



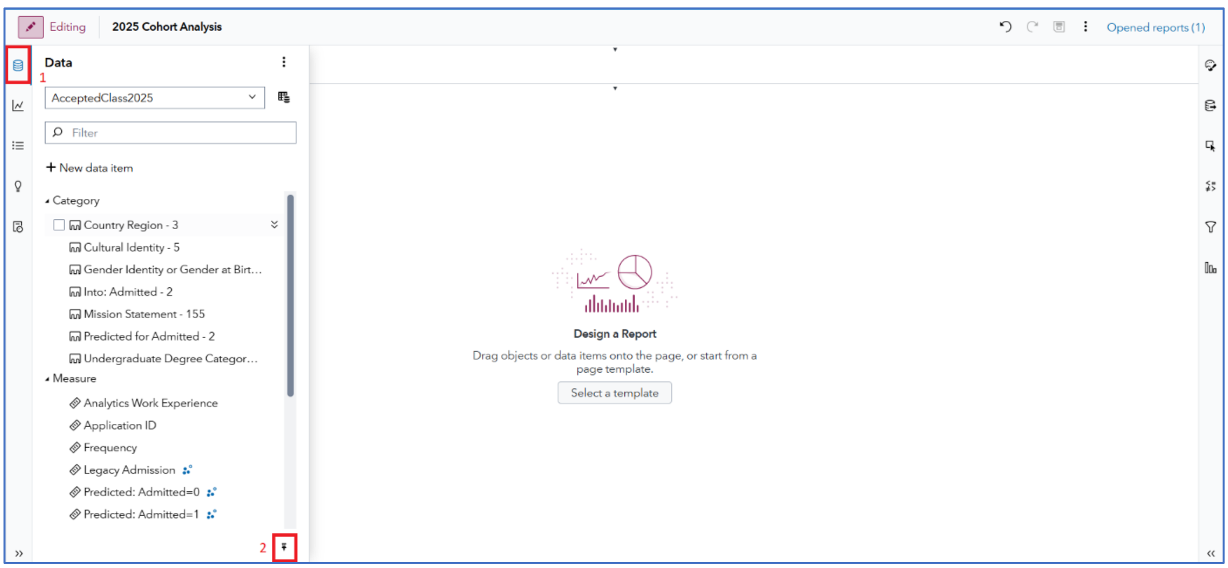
- Unlike SAS Model Studio – which has been saving our work automatically, we need to consciously save our work in SAS Visual Analytics. So, find that old school floppy disk in the upper right corner and click it:



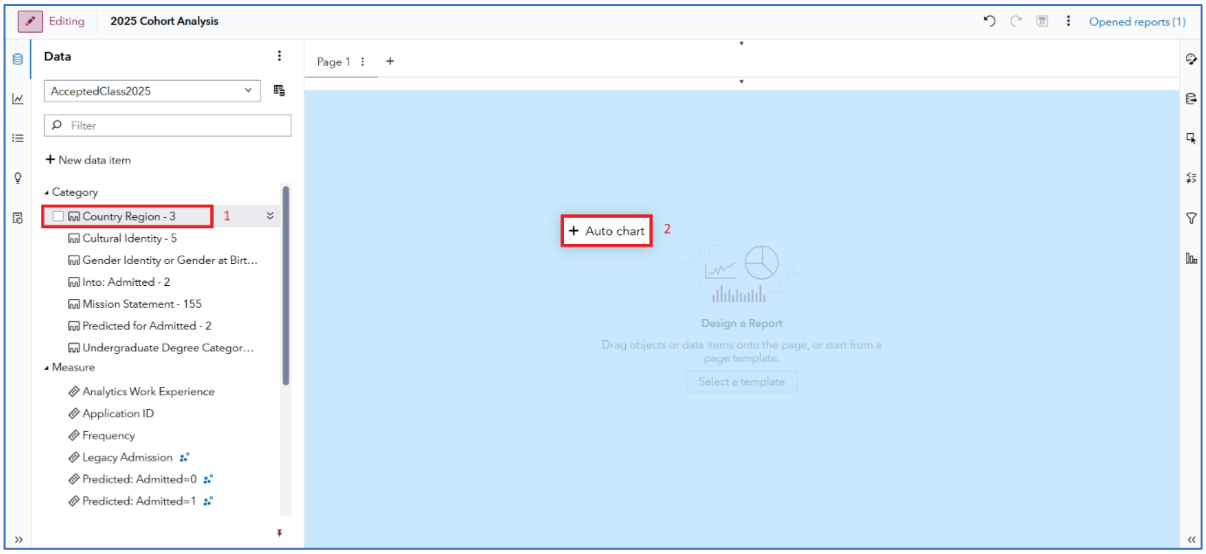
- Then save the report in **My Folder** with a name like **2025 Cohort Analysis**. Then click **Save**. Those steps:



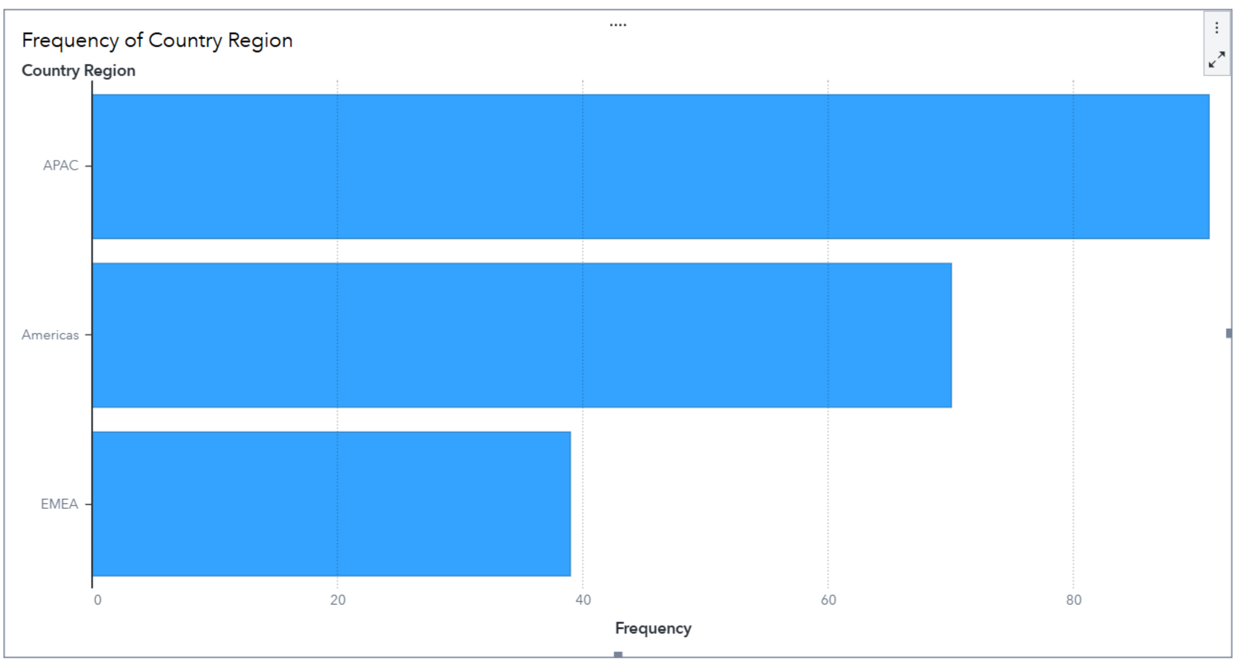
- Remember to save… and save often 😊 Now let's quickly get some items on the page. If it's not already open, activate the **Data** pane. You can also **Pin pane** with a second click:



- A wonderful feature of SAS Visual Analytics is that you can simply drag-and-drop variables on the canvas and SAS Viya will provide you with a default visualization. Let's start by taking **Country Region** and dragging and dropping it like so:

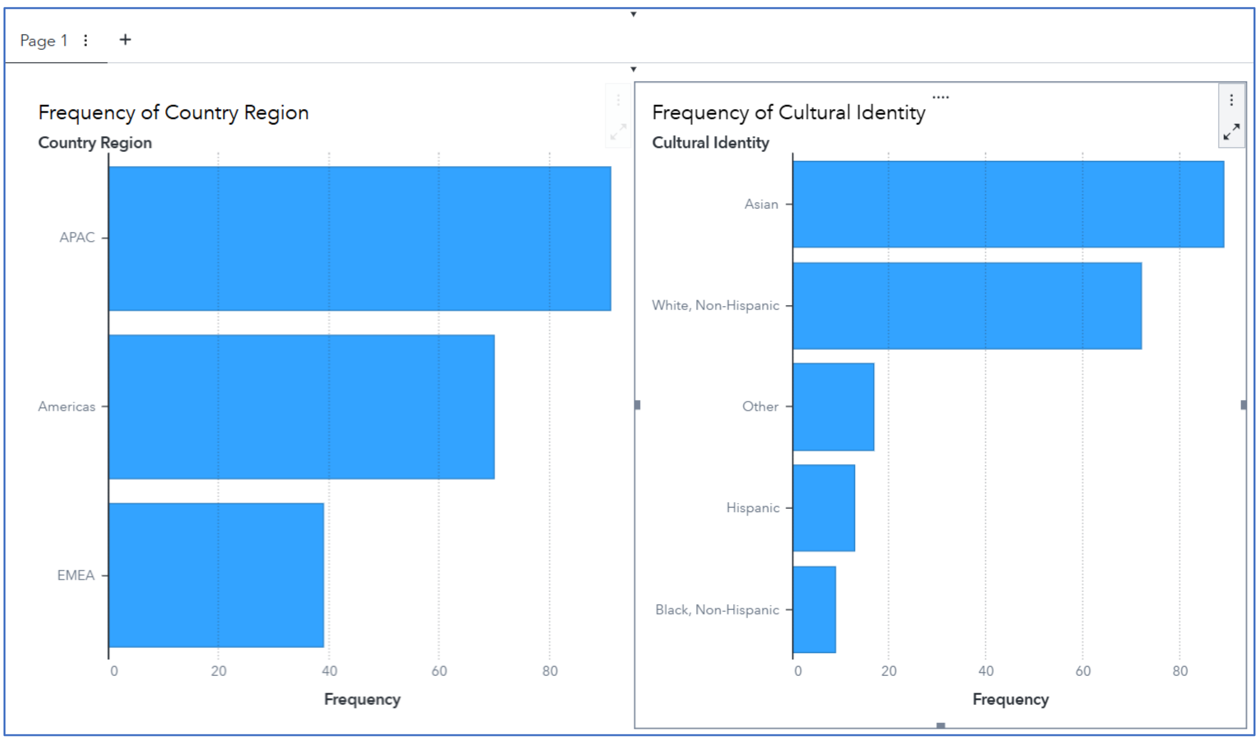


- Well, that's a nice start! This shows our 200 applicants across 3 regions:



- That's a lot of applicants from APAC! Now let's add *Cultural Identity* to the canvas. Select just that variable and drag-and-drop it to the right of the *Country Region* chart. Like here:

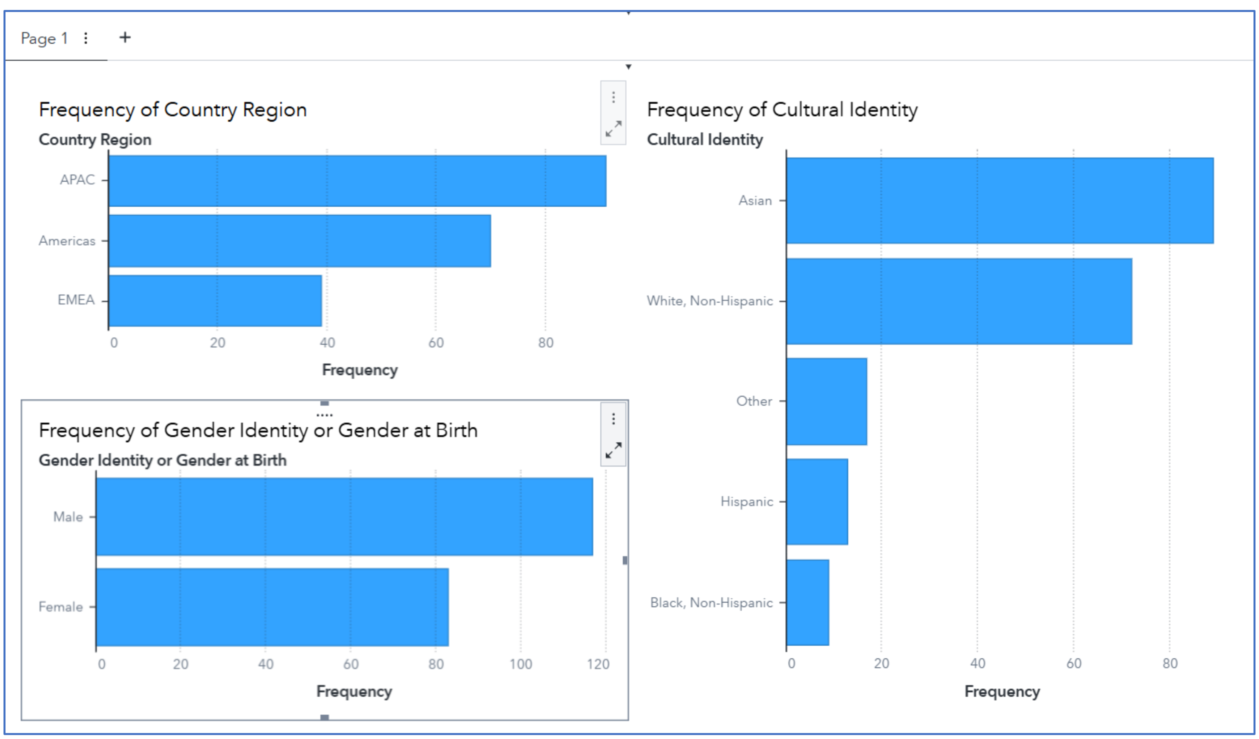
- You should see the following:



- Asian applicants had the highest number this year, followed by White, Non-Hispanic and a catchall "Other" category. And if you made a mistake in designing your dashboard, let the **Undo** button be your friend. It's found near the floppy disk here:

- And since I made you peak in the neighborhood, go ahead and **Save**. Please and thank you!
- Let's add one more "Category" variable to the canvas. Put **Gender Identity or Gender at Birth** below our Country Region analysis. Here:

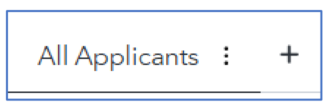
- You should now see the following:



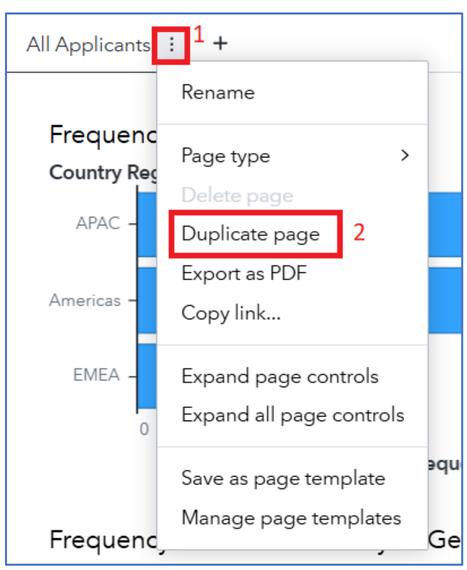
- One more to go! Let's add a "Measure"… just for fun. Find **Predicted: Admitted=1** and then drag-and-drop it below *Cultural Identity*:

- A first dashboard is nearly completed. So **Save**! Additionally, this dashboard technically has all of our 2025 applicants. So, let's make that a bit clearer in our analysis. Find the **Options** button next to the Page 1 title, here:

- Click it and then rename the Page to **All Applicants**. Success looks like:

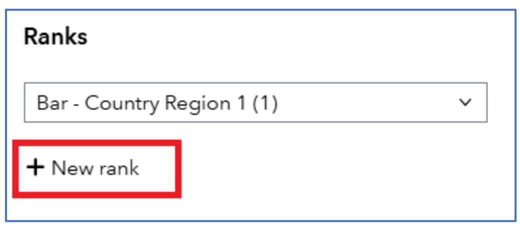


- For our final tricks, we'll want to duplicate this page and then limit the data to just those accepted. This can be handled without too much fanfare. Yay! Find that **Option** button again, and this type select **Duplicate page**:

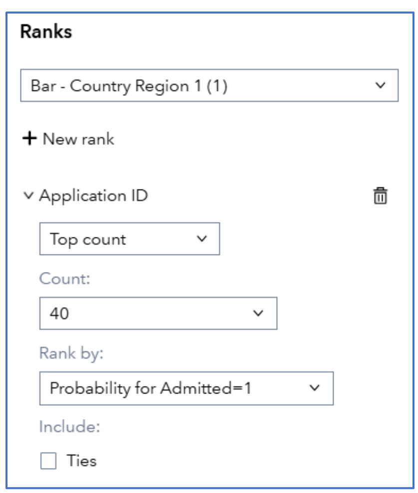


- Click that button… and check out the new page! Let's rename it right away to **Accepted Applicants**. Then activate the *Country Region* object and then click **Ranks**, like so:

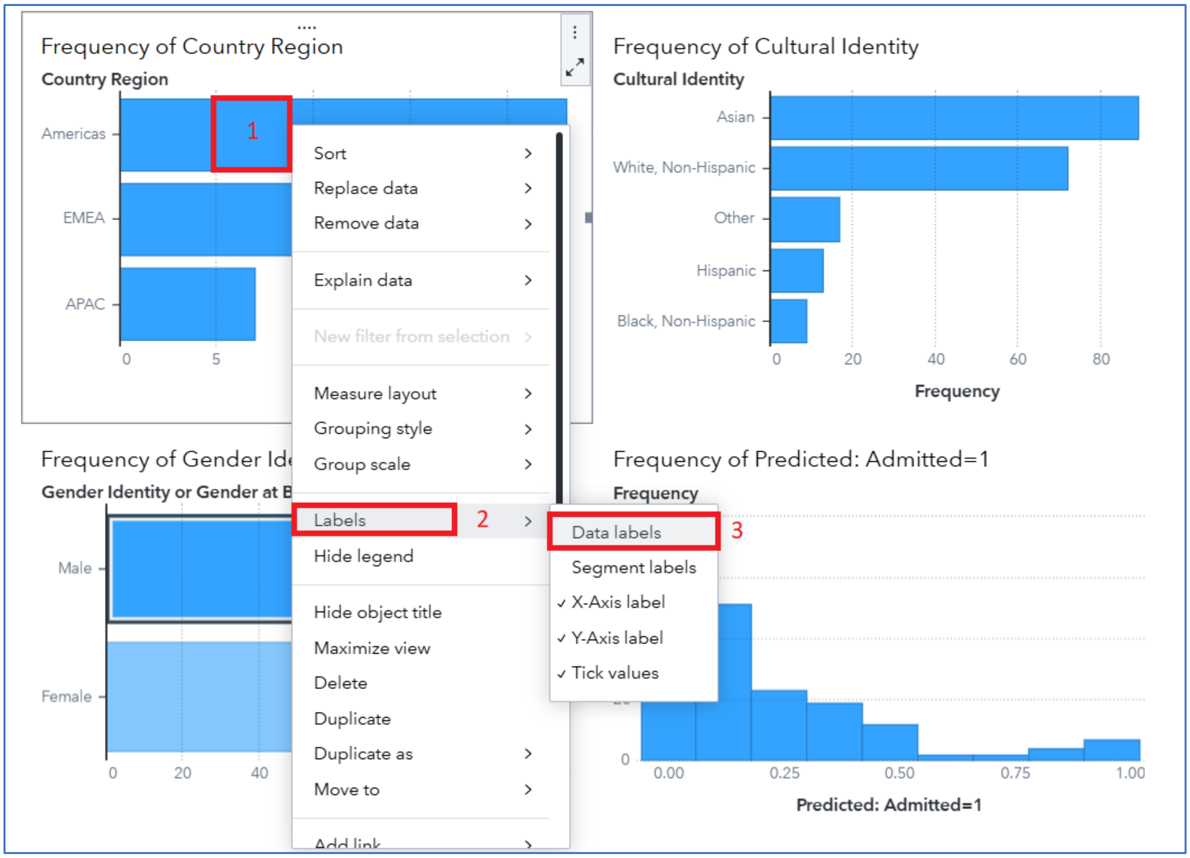
- We want to limit all the object to just the top 40 applicants – using **Probability for Admitted = 1** as our guide. So, Ranks will work very well – and we'll apply the following data filter 4 times. To start, select **New rank** on the Country Region bar chart:



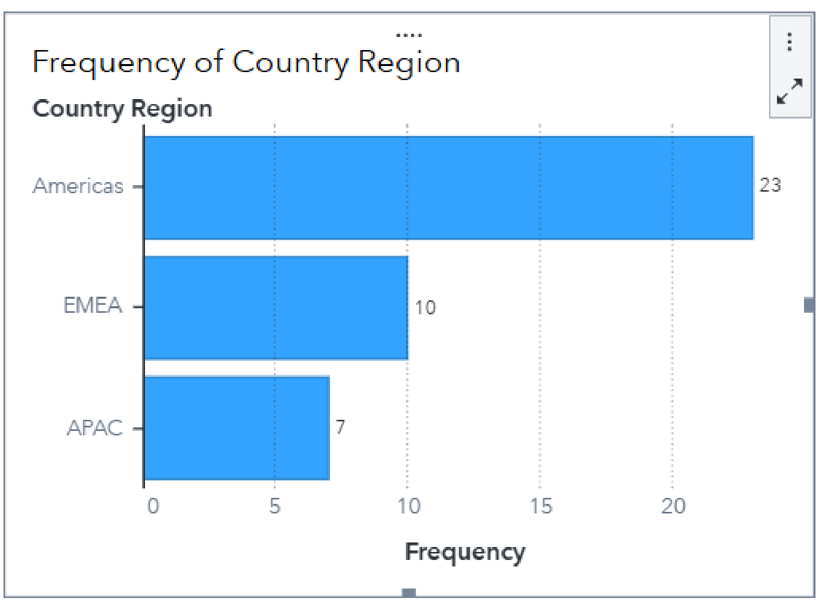
- For the variable, select **Application ID**. Then choose **Top count**, **Count** = **40** and **Rank by** = **Probability for Admitted = 1**. Those settings:



- And the Country Region bar chart should update automatically! To check if the numbers are correct, let's add the frequency counts to the Country Region chart. Right-click on one of the bars, then select **Labels** >> **Data labels**. Like this:



- Your new output. Does it add to 40? I think so!



- Now let's add the rank to the other three objects and – if you'd like, data labels to the bar charts too. My output:

- Any thoughts on that model? I see a LOT of applicants accepted from the Americas, even though APAC has a larger number of applicants. I also see a disproportionate number of White, Non-Hispanic students accepted, and – perhaps, a lower % of females accepted than the overall application pool. These are things to consider for later…
- And what's the last step in this section? All together: **SAVE**!

Part 2: Business (not) as Usual

Overview

From your graduate studies, you know that using historical data to predict new cases – particularly in the case of outcomes like admission – often perpetuates the status quo. And while this can be a good thing when it's an equitable world, it is a bad thing when it allows bias to persist.

We've recently realized that there is likely some bias in our historic "business as usual approach" but didn't really know how to address it – beyond a "gut" feeling. Fortunately, SAS has recently added some *fairness and bias statistics* to SAS Model Studio, which can allow us to statistically assess bias and address it in a more rigorous manner.

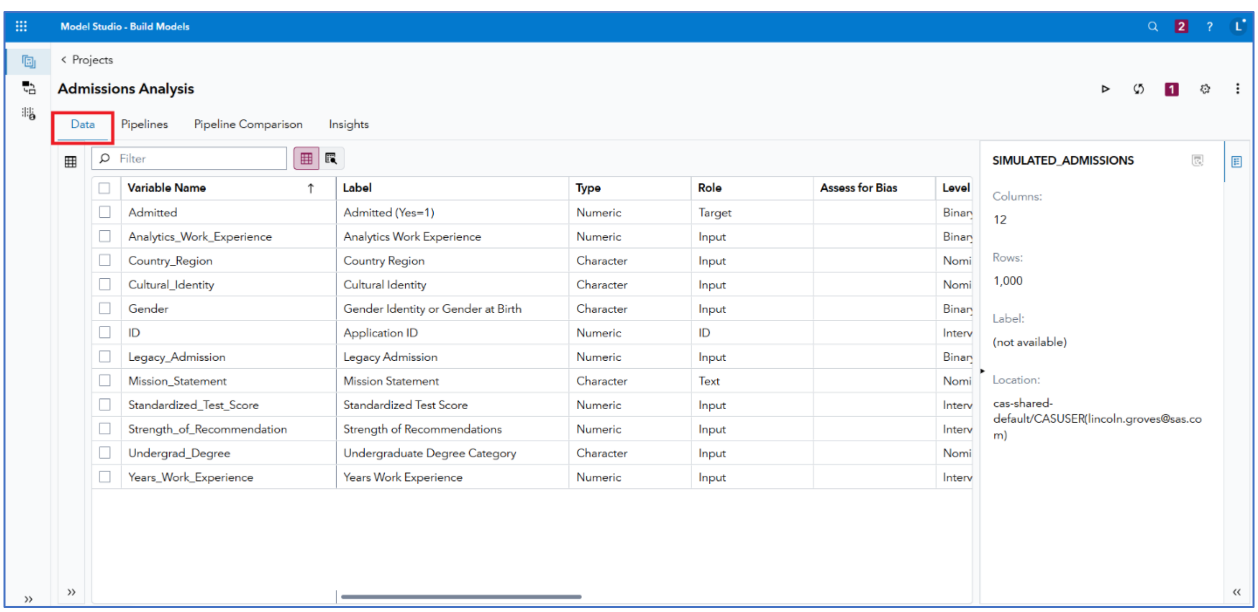
So, in this section you'll learn about the fairness and bias tools in SAS Model Studio. Then you'll be asked to update your models and then see if the incoming class of 2025 is at all different from the students selected from our historical model.

Let's learn together!

Starting from the Beginning

To use the fairness and bias tools in SAS Viya, let's go back to our **Admissions Analysis** project in SAS Model Studio and take it from the top:

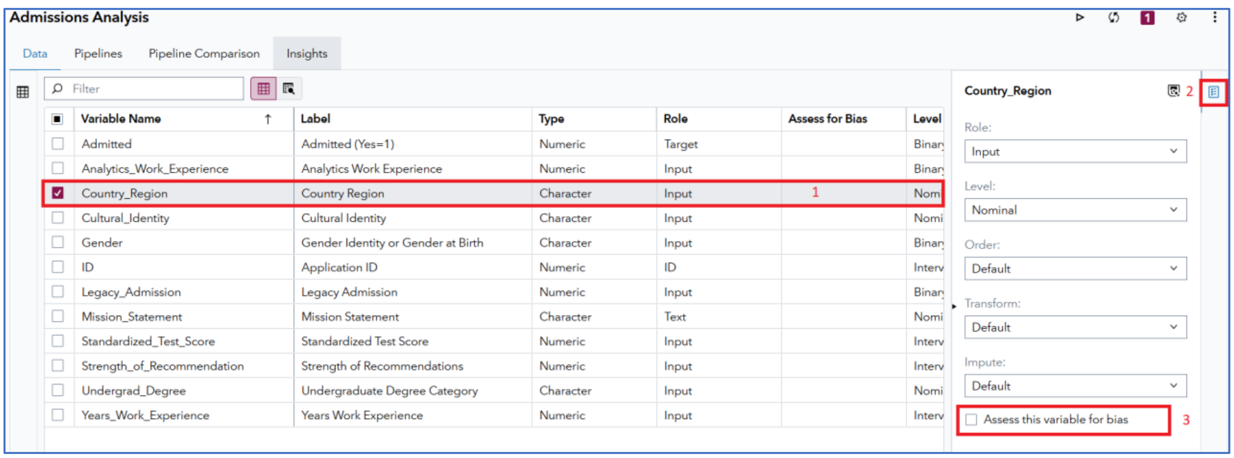
- Navigate back to the **Data** tab. It misses you:



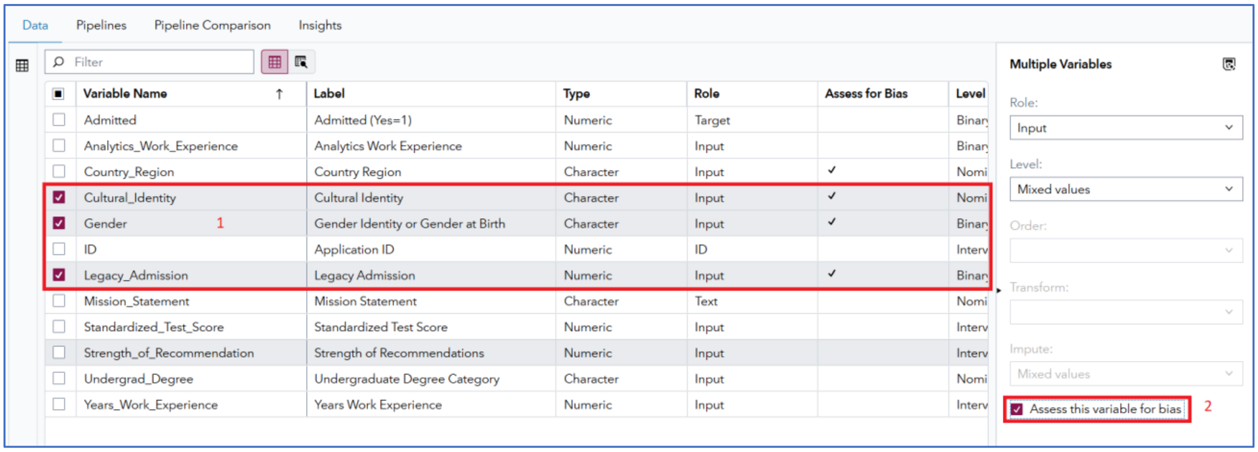
- Now perhaps your eye caught it last time… but did you see that **Assess for Bias** column? Like here:

Variable Name	Label	Type	Role	Assess for Bias	Level	Order
Admitted	Admitted (Yes=1)	Numeric	Target		Binary	Default
Analytics_Work_Experience	Analytics Work Experience	Numeric	Input		Binary	Default
Country_Region	Country Region	Character	Input		Nominal	Default
Cultural_Identity	Cultural Identity	Character	Input		Nominal	Default
Gender	Gender Identity or Gender at Birth	Character	Input		Binary	Default
ID	Application ID	Numeric	ID		Interval	Default
Legacy_Admission	Legacy Admission	Numeric	Input		Binary	Default
Mission_Statement	Mission Statement	Character	Text		Nominal	Default
Standardized_Test_Score	Standardized Test Score	Numeric	Input		Interval	Default
Strength_of_Recommendation	Strength of Recommendations	Numeric	Input		Interval	Default
Undergrad_Degree	Undergraduate Degree Category	Character	Input		Nominal	Default
Years_Work_Experience	Years Work Experience	Numeric	Input		Interval	Default

- That looks promising! But how do we activate it? Well, it's going to be via the **Properties** pane. Let's start with one example, then proceed from there. Select **County_Region**. Then ensure the **Properties** panel is open and then scroll to the bottom. Do you see the **Assess this variable for bias** option?



- Click it. It's just that easy! And, for good measure, let's select it for three other variables:
  - Cultural_Identity
  - Gender
  - Legacy Admission
- You can even get fancy and select all three variables at the same time. Then you just need to click **Assess this variable for bias** once:

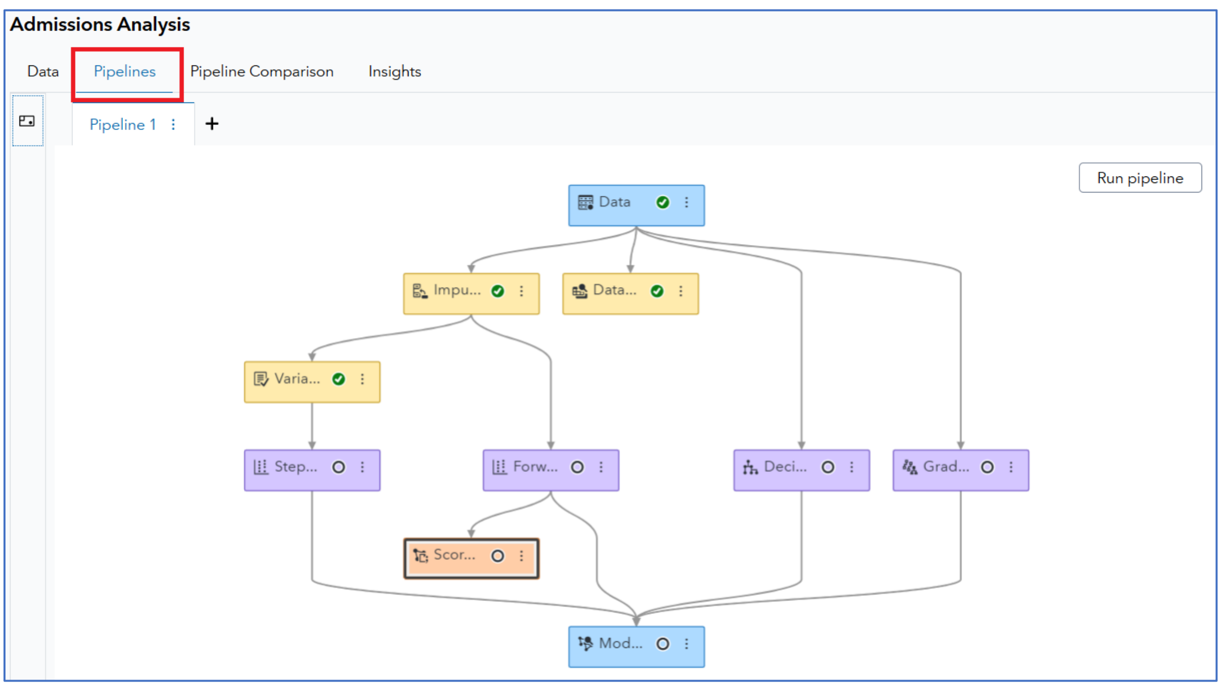


- Note that we now have four **Assess for Bias** variables selected. And so our metadata are now set.

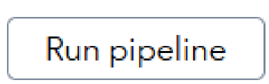
Revisit our Modeling

With our SAS Model Studio project now setup to examine bias, we can now return to the modeling. Yay!

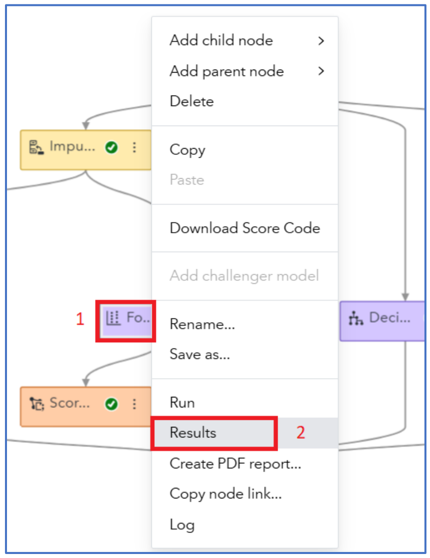
- Head on over to the **Pipelines:**



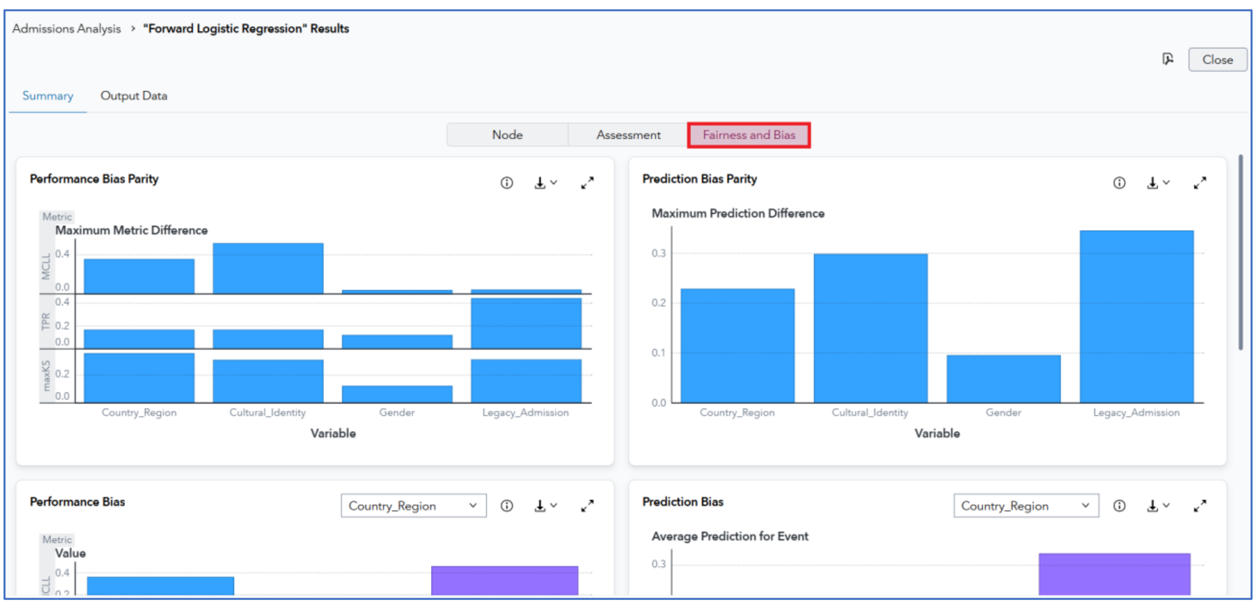
- Note that all the Supervised Learning nodes have reset themselves – because they can now incorporate the **Fairness and Bias** analyses. Click **Run pipeline** to let ‘er rip!



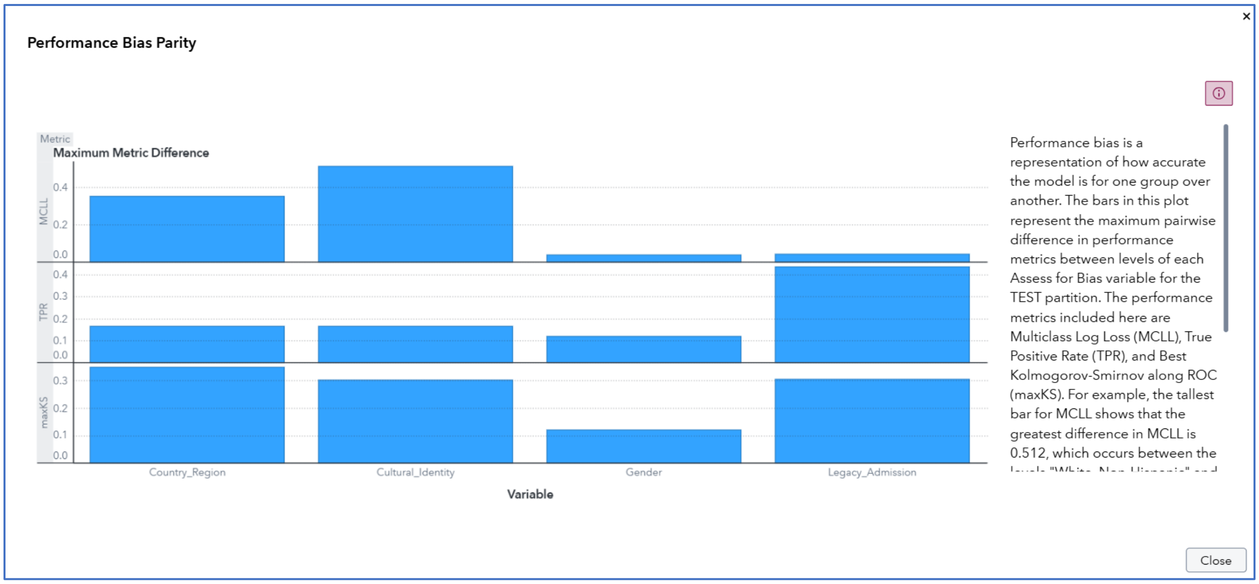
- Once the pipelines finish, let's go back to the champion model to explore the **Results**. Mine is still the Forward Logistic Regression model – but I'm hoping that you found something that fits the data a bit better. Regardless, right-click the node on a Supervised Learning node and the select **Results**:



- See what I see? Some new output here:



- Let's digest the results! And we can start with the **Performance Bias Parity** output. Expand that window and get ready to read. Because – like me – these metrics may be new to you!

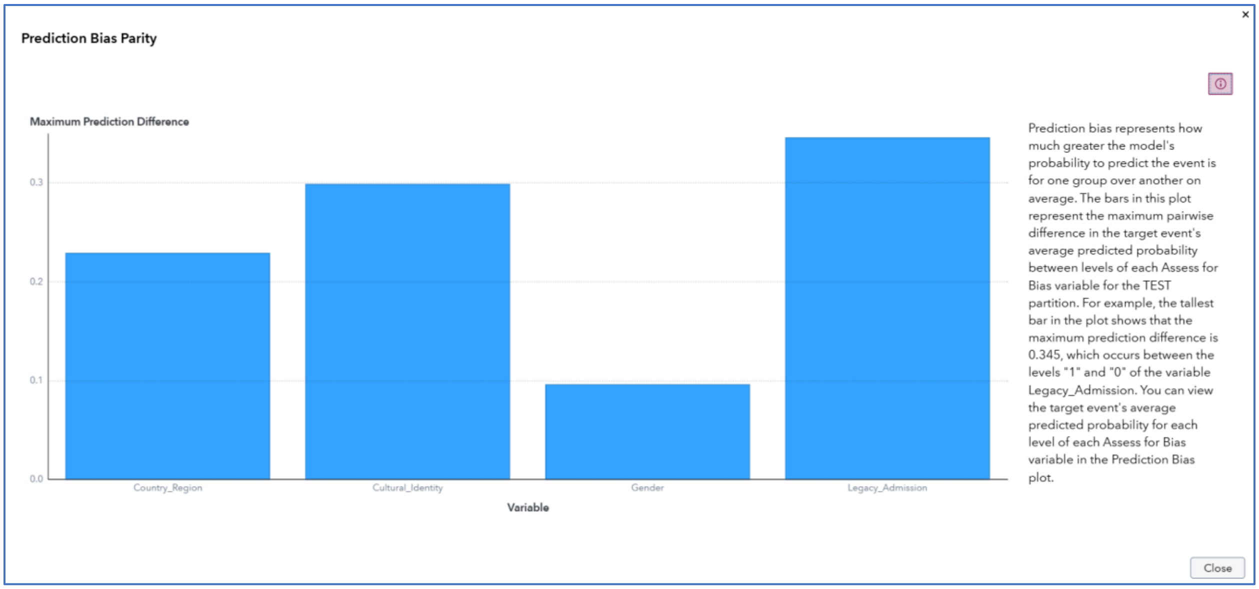


- I find the note on the right here really helpful to help me with interpretations. So helpful, in fact, that I'll repeat the note below:

  Performance bias is a representation of how accurate the model is for one group over another. The bars in this plot represent the maximum pairwise difference in performance metrics between levels of each Assess for Bias variable for the TEST partition. The performance metrics included here are Multiclass Log Loss (MCLL), True Positive Rate (TPR), and Best Kolmogorov-Smirnov along ROC (maxKS). For example, the tallest bar for MCLL shows that the greatest difference in MCLL is 0.512, which occurs between the levels "White, Non-Hispanic" and "Black, Non-Hispanic" of the variable Cultural_Identity. You can view the values of the performance metrics for each level of each Assess for Bias variable in the Performance Bias plot.

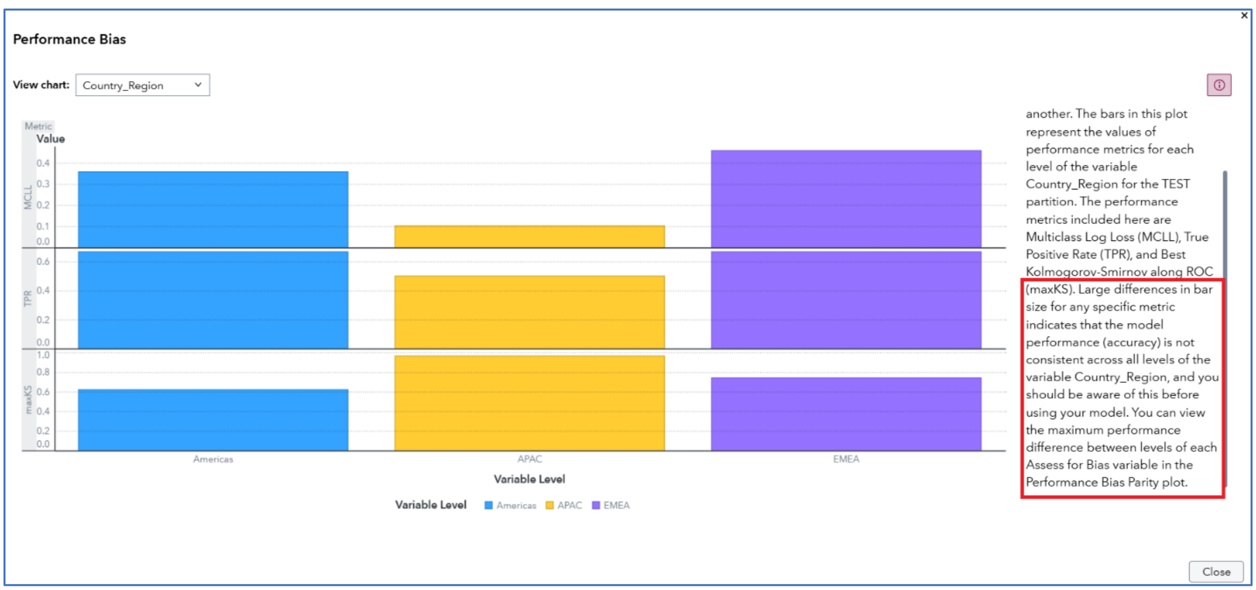
  (Note: The maximum pairwise difference in TPR is also known as "Equal Opportunity".)

- So… what are your interpretation of that interpretation?
  - I'm finding that *County Region* and *Legacy Admission* is incredibly important in driving admissions selection. So, we could be unconsciously weighting those factors a little too strongly in our admittance algorithm.
- Let's check out **Prediction Bias Parity**. Expand the object and then let's read together:



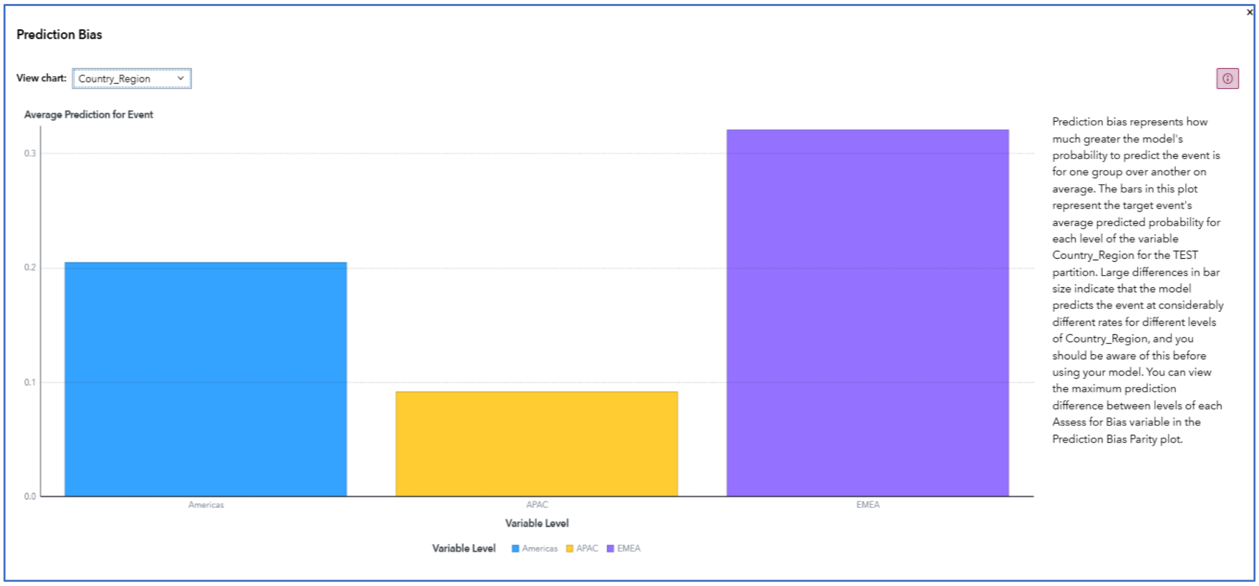
- Yikes. While we can't tell the direction, *Legacy_Admissions* and *Cultural_Identity* have very large values of around 0.3. We can interpret that to mean – on a prediction bound between 0 and 1 – that those variables lead to a swing of nearly 30 percentage points in terms of an acceptance rate. That's BIG.

- Let's keep learning. Expand the **Performance Bias** object, which allows us to do a deeper dive into bias at the variable level (it was a bit masked in **Performance Bias Parity**). We start with **Country_Region**. And I'll highlight a very important part of the narrative:



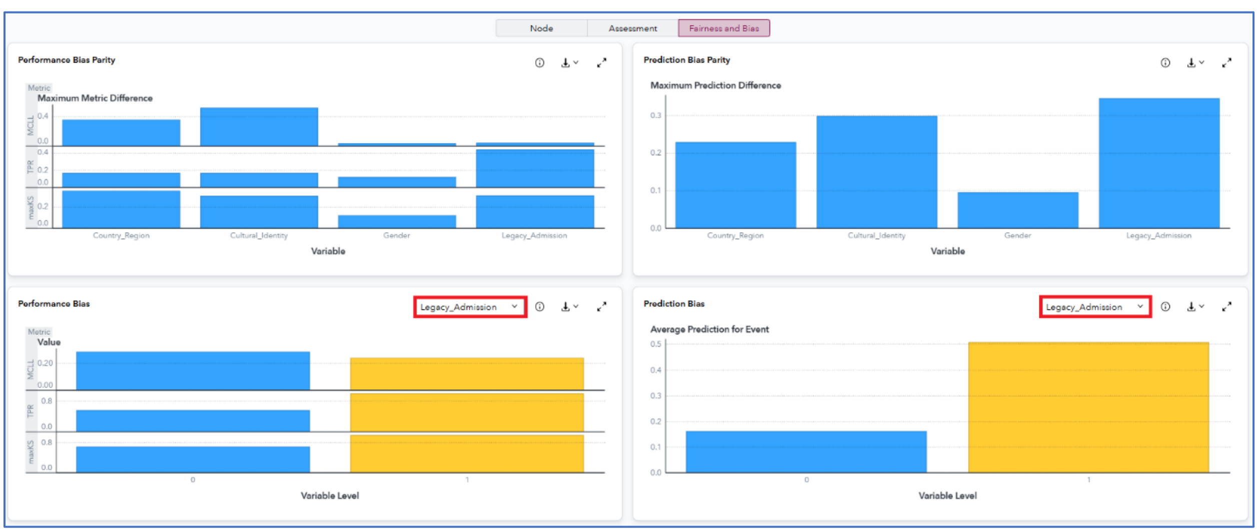
- From the region analysis, it looks like students from APAC are being systematically treated differently than students from the other regions. But, I can't quite tell you how yet.

- And… this is where the **Prediction Bias** object is useful. Expand it:



- Oh boy. Students from EMEA have acceptance rates around 32%, while APAC students are accepted at less than 10%. While that could be explained by other variables, we actually know that many of our applicants from APAC are top-notched.
- So, we may be unconsciously biasing our admission process towards students from EMEA and the Americas, at the expense of APAC. For no other reason than students from the past have come predominantly from EMEA and the Americas. Yikes. We should drop *Country_Region* from the analysis because it will just perpetuate historical trends in acceptance, rather than the aptitude of the students. And we want the best students.

- It is very easy to examine other variables in the **Fairness and Bias** output. Simply use the dropdown boxes to change the view, like so:



- And this is where the investigative journalist in you can take over. Spend a bit of time exploring **the Fairness and Bias** results and consider:
    - Which variables are just perpetuating historical trends in admissions – rather than helping us accept the best students?
    - And, as such, which variables should we drop to create a better admissions process?
- Click **Close** when you're done digesting the **Results**. And head over to another model if you like. But definitely take a break and grab a coffee. You deserve it!

Rerunning the entire analysis

I'll start this section with an obvious insight: modeling is an iterative process. We RARELY get the model right the first time – and each model can build on the subsequent insights from the last. And I love that about the empirical process.

But you may be thinking, "iterating is going to take a lot of work".

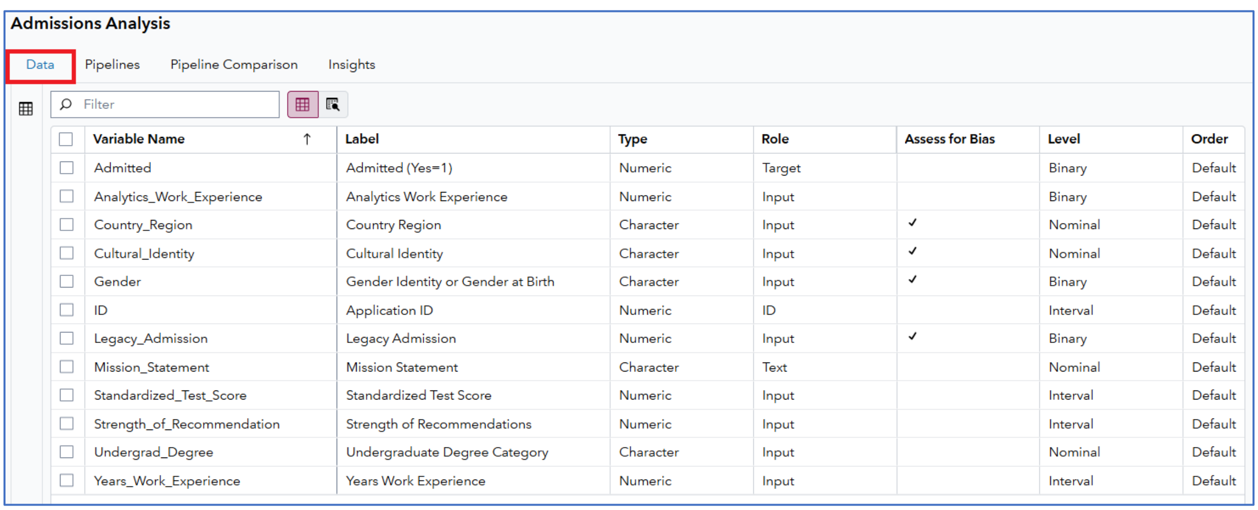
Nope!

One of the things that makes SAS Viya special is that it automates a LOT of the modeling process for you. I'll share three ways in this sections that SAS Viya can make it easier to iterate and iterate.

It's easy to adjust your metadata

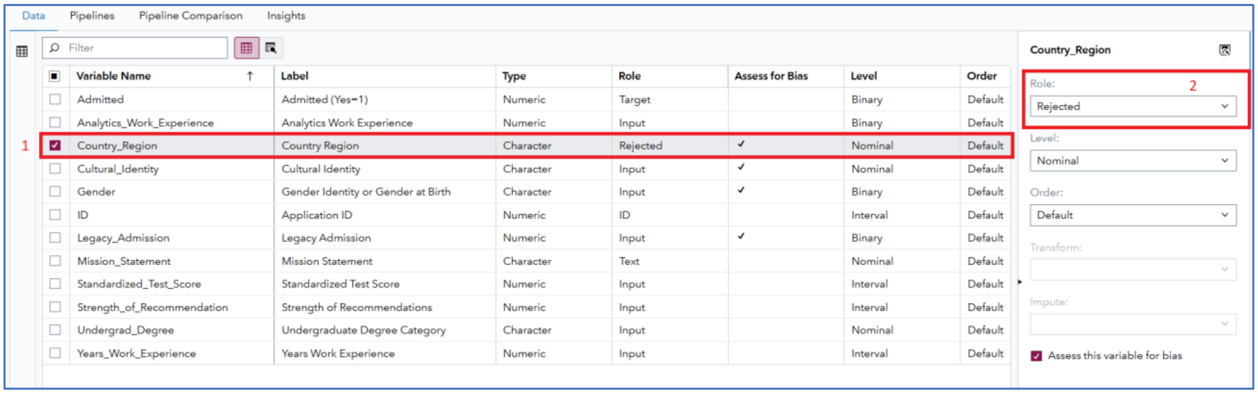
Updating your metadata – and all the models that flow from that – is literally a few clicks away. I'll show you how to adjust your metadata for one variable in this section.

- Return to our friend, the **Data** tab:

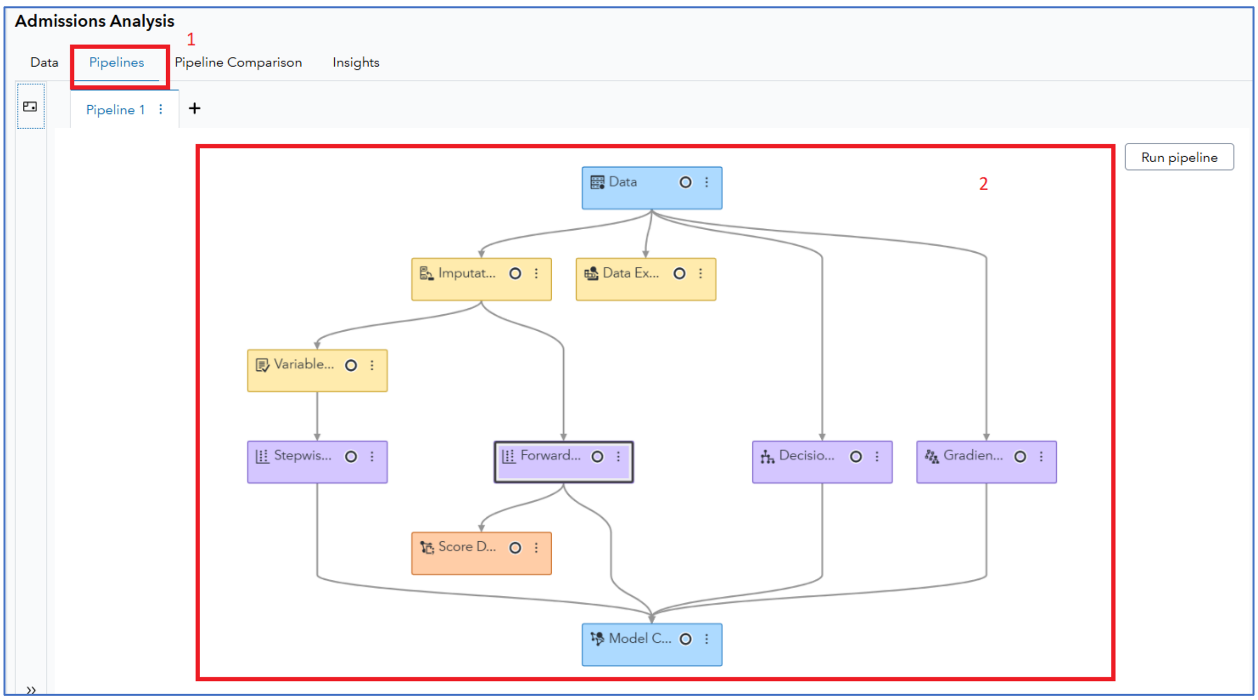


Variable Name	Label	Type	Role	Assess for Bias	Level	Order
Admitted	Admitted (Yes=1)	Numeric	Target		Binary	Default
Analytics_Work_Experience	Analytics Work Experience	Numeric	Input		Binary	Default
Country_Region	Country Region	Character	Input	✓	Nominal	Default
Cultural_Identity	Cultural Identity	Character	Input	✓	Nominal	Default
Gender	Gender Identity or Gender at Birth	Character	Input	✓	Binary	Default
ID	Application ID	Numeric	ID		Interval	Default
Legacy_Admission	Legacy Admission	Numeric	Input	✓	Binary	Default
Mission_Statement	Mission Statement	Character	Text		Nominal	Default
Standardized_Test_Score	Standardized Test Score	Numeric	Input		Interval	Default
Strength_of_Recommendation	Strength of Recommendations	Numeric	Input		Interval	Default
Undergrad_Degree	Undergraduate Degree Category	Character	Input		Nominal	Default
Years_Work_Experience	Years Work Experience	Numeric	Input		Interval	Default

- County_Region appears to be introducing some significant bias into our model. To exclude it, click on the **County_Region** variable and then change the **Role** from Input to **Rejected**. That's it! Two clicks:



- Once a variable is rejected (or, in general, changed), the pipeline will reset and needs to be run again. Want proof? Click on **Pipelines**:



- That's essentially a new pipeline for you to run. And new adventures to explore!

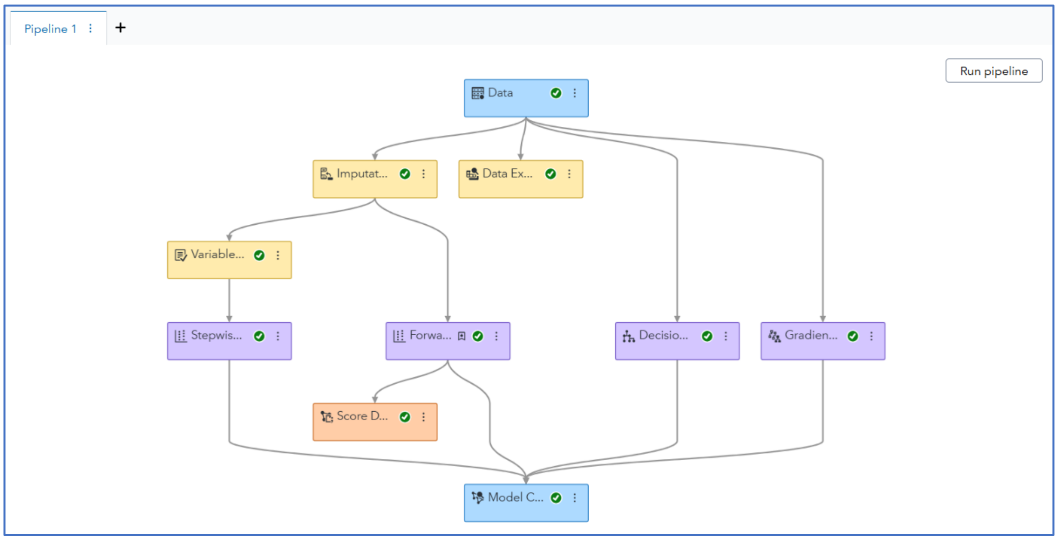
It's easy to try new models

You may think "why don't I drop nearly all the variables and just base the admission decision on a single item like standardized test score". Well… you *could* do that. But then we lose the value of the data from our historical application pool – which, btw, our alumni are terrific! Moreover, test scores aren't everything – so we really should try to capture as many factors as possible that make a dynamic candidate dynamic. And that means finding a way to weigh several factors into a decision process.

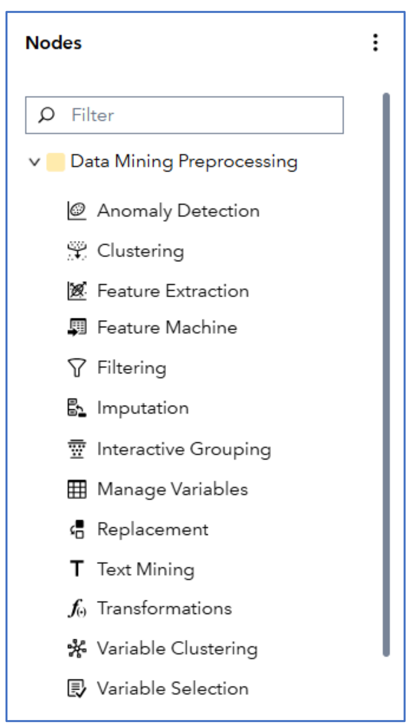
Fear not – we can always run more models and keep trying to maximize the KS (Youden) fit statistics in our quest to crown a single champion model. Now you did this in Part 1, but if you've dropped variables then the data have changed. And you may need to work a little harder to find information for the algorithms to use to predict new cases.

Don't fret. I've got you:

- Return to your **Pipelines**. I've still got only one pipeline, so I'll remain here:

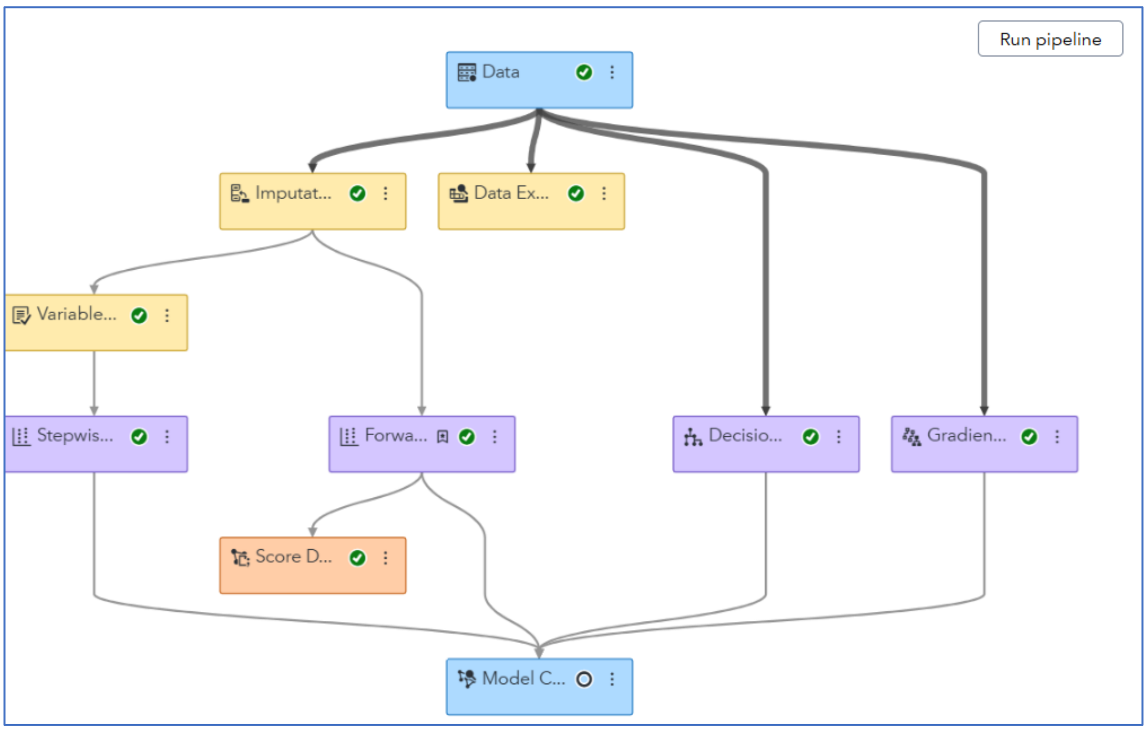


- It is very easy to add **Data Mining Preprocessing** nodes to your pipeline! Now, now, what do I mean by preprocessing? Well, I mean all these options:

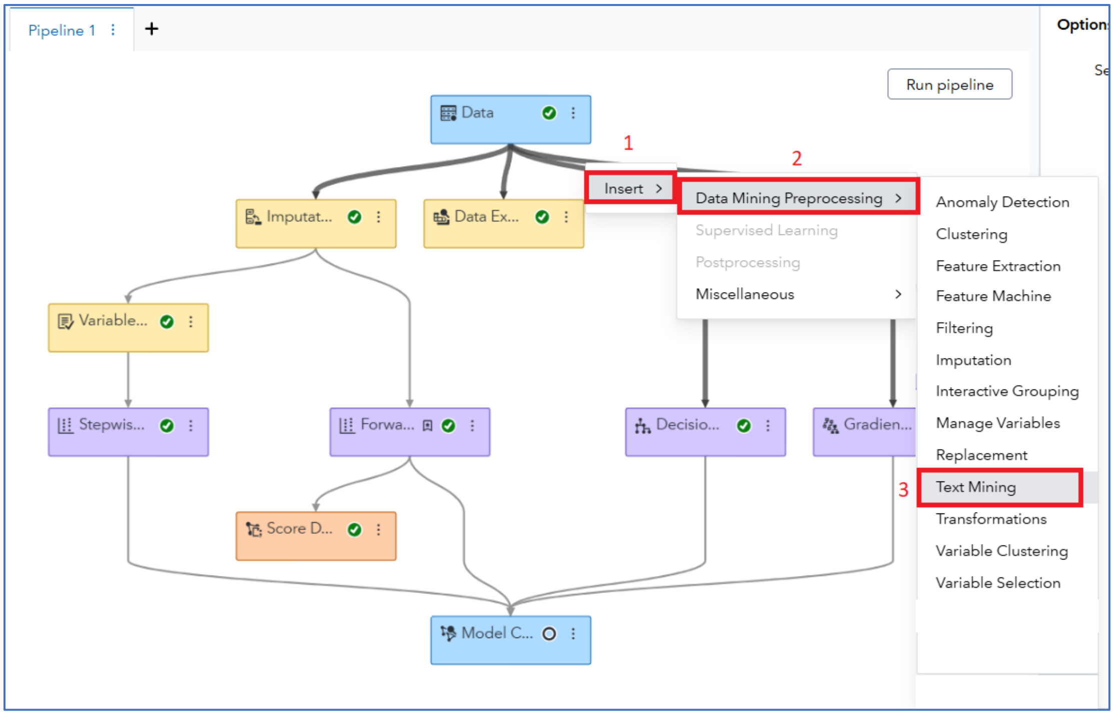


- Why would we want to do this? Well, we have a limited amount of information on each candidate – but want to select the best overall candidates. Towards that end, we can create new variables from existing variables – to account for things like interactions between factors (like standardized test scores and work experience) or we could actually use *Mission Statement*, which is currently being ignored.

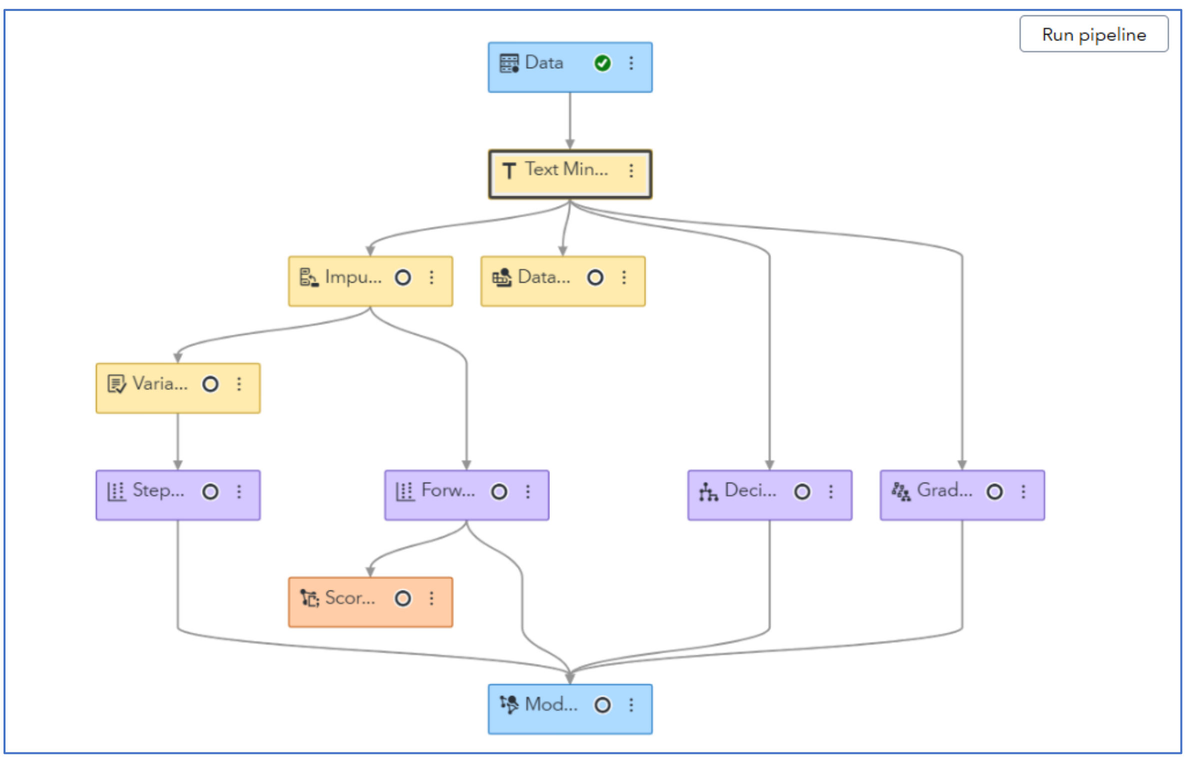
- I vote yes on adding a **Text mining** node to generate new variables from *Mission Statement*. And since I want to use it in several models, I need to set it up in a very particular way:
    - **Ctrl-Click** the pipelines that you want to be connected to the **Text mining** node. Like this example, where 4 pathways are selected:



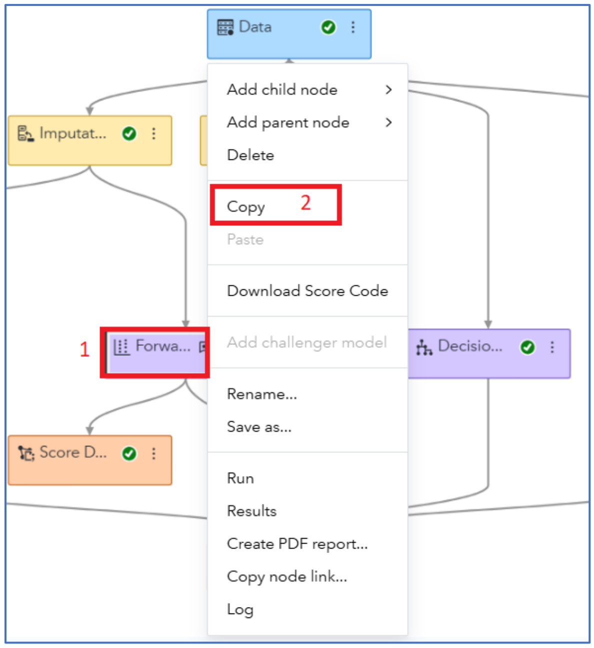
    - With those pathways identified, right-click, select **Insert** >> **Data Mining Preprocessing** >> **Text Mining**



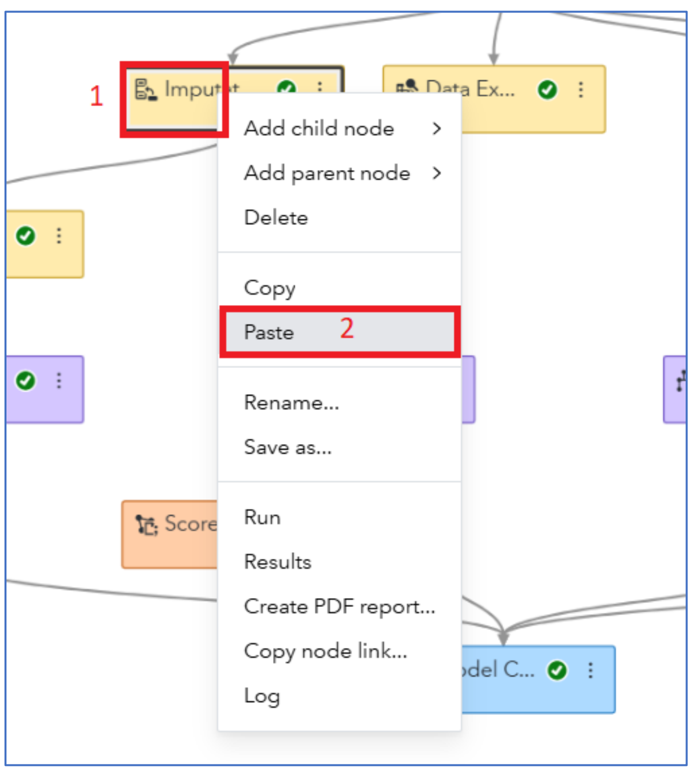
- Text mining on the text variables listed in the **Data** tab is now part of your analysis. And it's just that simple 😊 See:



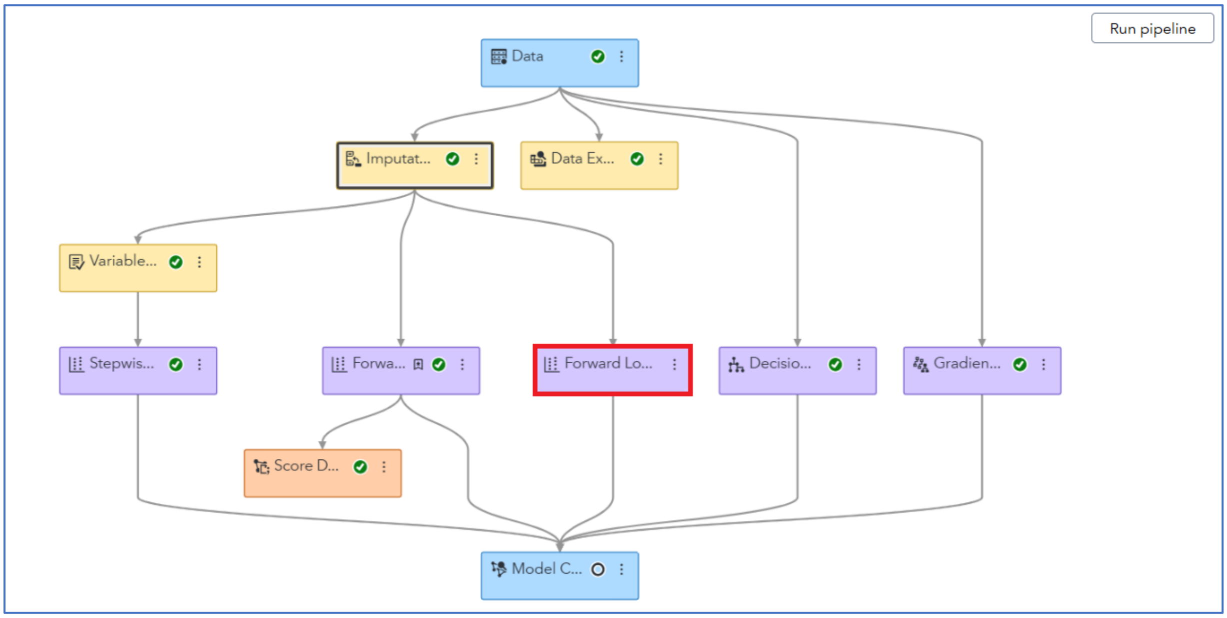
- All of those empty circles mean that you need to rerun the models with the new text variables. But before getting there – and as Part 2 of "try new models" – did you know that you can copy-and-paste nodes? Are you like "I just got here… why would I know that?" Well… that would be a fair response. But, regardless, you can! I'm going to copy my champion model and see if I can do better. To start, I right-click on the *Forward Logistic Regression* model and select **Copy**. Like so:



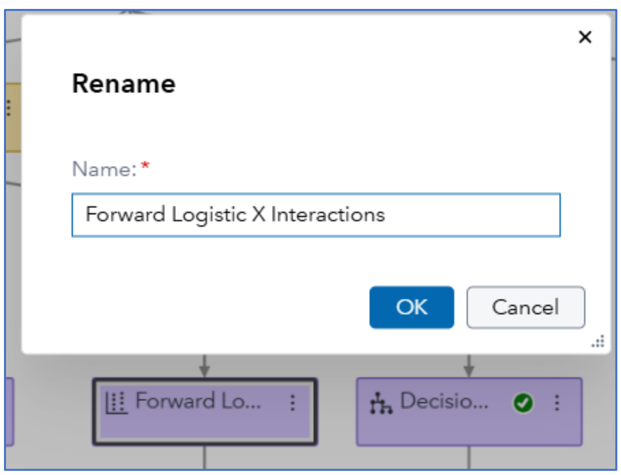
- I can them paste it up the pipeline – here on the **Imputation** node. So, right-click on the **Imputation** node and then select **Paste**. Like this:



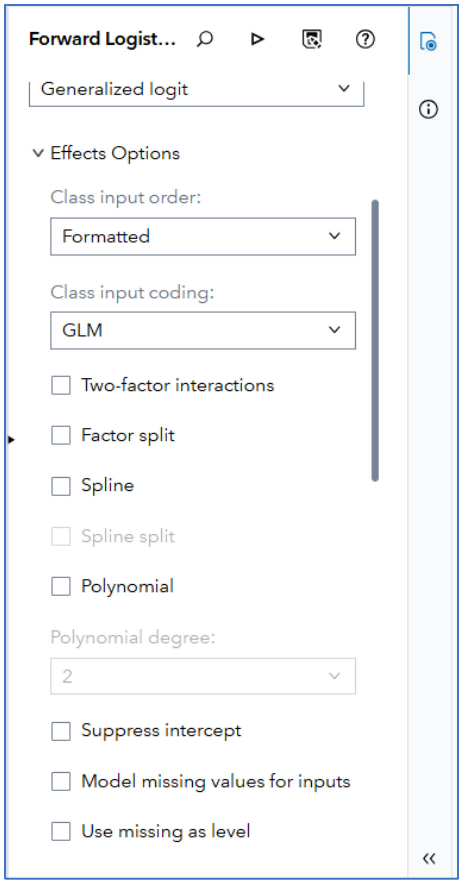
- You should see a new model to run:



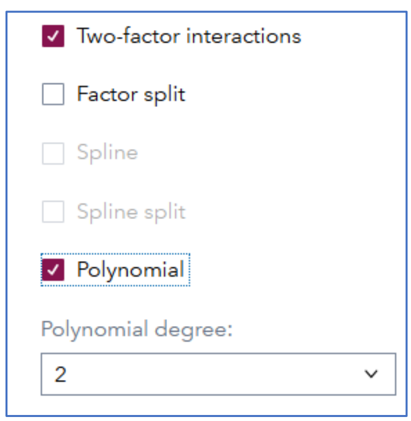
- I'm going to right-click on the new node and select **Rename**. Then I'll use Forward Logistic X Interactions as the name, which is a preview of things to come:



- For the **Node options** on the *Forward Logistic X Interactions* model, let's expand the **Effects Options**. Here:



- These options will allow us to better account for some non-linear relationships in (and between) our variables. Choose your own adventure, but here's what I'd like to explore:



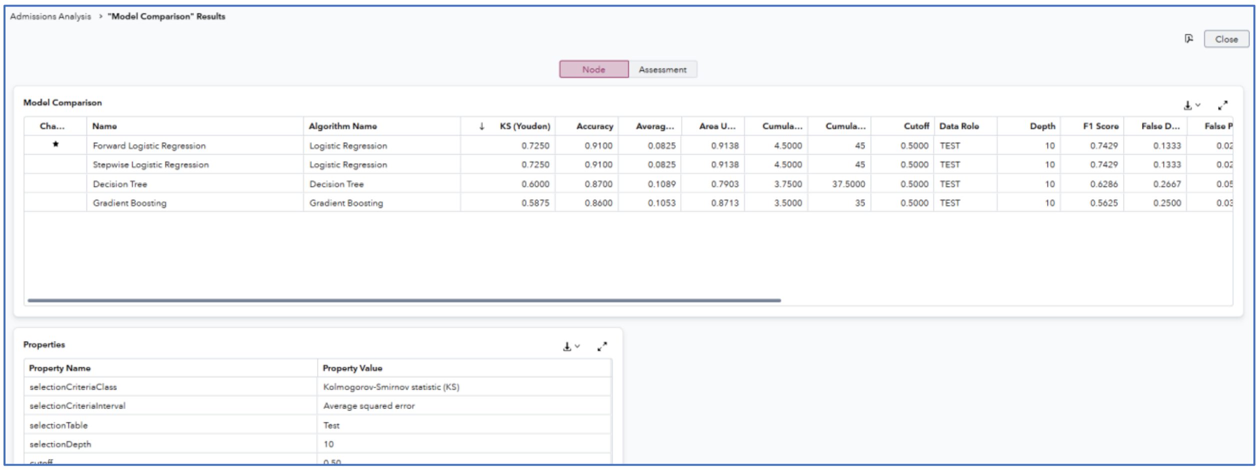
- With those settings locked in, let's run the entire pipeline! This pipeline now has (1) some text analytics variables and (2) a new model that includes interactions and allows for non-linear relationships for variables like work experience. Moreover, I've excluded some questionable demographic variables that may have unwittingly introduced bias into my modeling.

- After it runs, let's check out that **Model Comparison**:

Model Comparison

Cha…	Name	Algorithm Name	↓ KS (Youden)
★	Decision Tree	Decision Tree	0.7750
	Forward Logistic Regression	Logistic Regression	0.7500
	Stepwise Logistic Regression	Logistic Regression	0.7500
	Gradient Boosting	Gradient Boosting	0.6625
	Forward Logistic X Interactions	Logistic Regression	0.3125

  - How did that compare to our last models? Well, here are my results from the *Part 1: Business as Usual* analysis:


Admissions Analysis > "Model Comparison" Results

Node | Assessment | Close

Model Comparison

Cha…	Name	Algorithm Name	↓ KS (Youden)	Accuracy	Averag…	Area U…	Cumula…	Cumula…	Cutoff	Data Role	Depth	F1 Score	False D…	False P
★	Forward Logistic Regression	Logistic Regression	0.7250	0.9100	0.0825	0.9138	4.5000	45	0.5000	TEST	10	0.7429	0.1333	0.02
	Stepwise Logistic Regression	Logistic Regression	0.7250	0.9100	0.0825	0.9138	4.5000	45	0.5000	TEST	10	0.7429	0.1333	0.02
	Decision Tree	Decision Tree	0.6000	0.8700	0.1089	0.7903	3.7500	37.5000	0.5000	TEST	10	0.6286	0.2667	0.05
	Gradient Boosting	Gradient Boosting	0.5875	0.8600	0.1053	0.8713	3.5000	35	0.5000	TEST	10	0.5625	0.2500	0.03

Properties

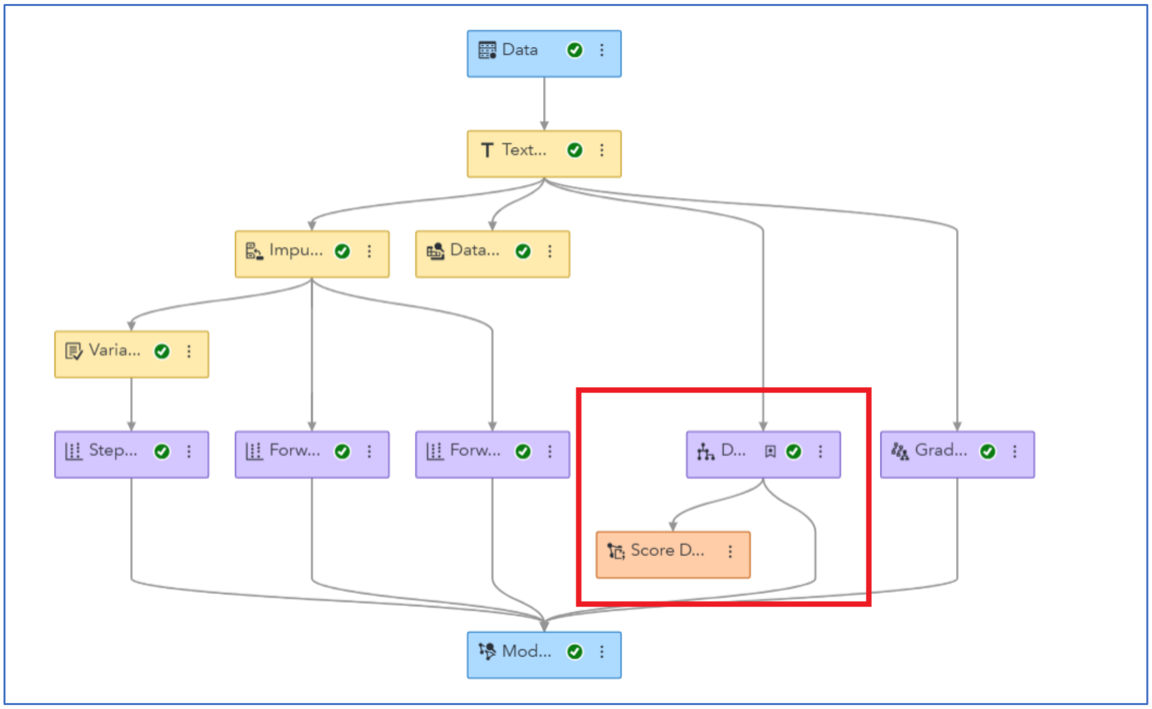
Property Name	Property Value
selectionCriteriaClass	Kolmogorov-Smirnov statistic (KS)
selectionCriteriaInterval	Average squared error
selectionTable	Test
selectionDepth	10
cutoff	0.50

  - So… it looks like my added efforts were worth it. Check your results and know there is some more modeling work ahead. And with that promise, exhale again for a bit.

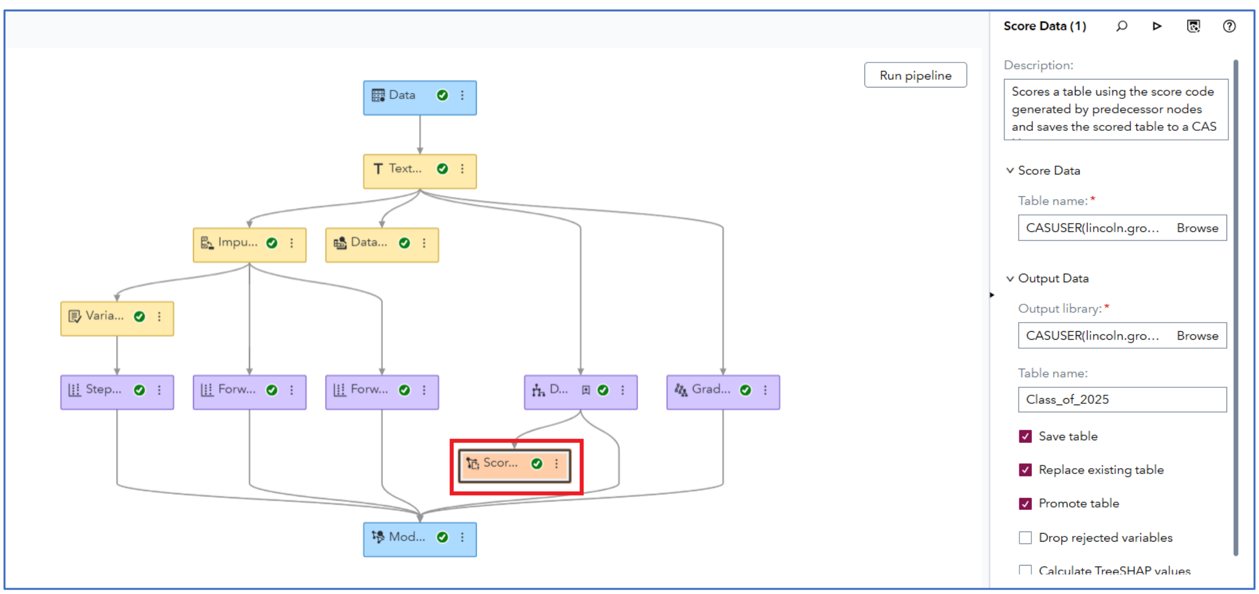
Bonus, Revisited: Linked output updates automatically in SAS Viya!

And by linked output – I'm really talking about our VA report here. Remember that one? That has the list of the 40 students we're going to admit? I know you didn't forget 😊 Let's see how easily that analysis can be updated in SAS Viya:

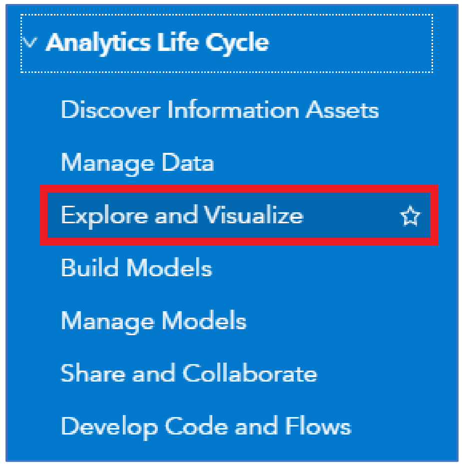
- In SAS Model Studio, find your champion pipeline. Move the scoring node to the champion model and keep the settings the same (I recommend that you copy-and-paste… and then delete the old node). For example, after excluding some potentially troublesome demographic variables – and creating new one – my champion model is a decision tree:



- Run the pipeline to score the new champion model. Check marks the spot:



- Now let's return to your VA report, aptly named **2025 Cohort Analysis**. If you forgot how to get there, it's via the **Applications menu**, here:



- Now the first page, **All Applicants** should look the same. Because it's still the same summary of everyone who has applied. But, slide over to the **Accepted Applicants** tab. See what I see:

  - Whoa: that's a significantly different incoming 2025 cohort! With a more merit-based approach – one that disregards demographic variables and status – we see significantly more students accepted from APAC. Males are still accepted at a 2-1 ratio than females, but that's more in line with overall application trends.
- There is a lot more to unpack, but I'll stop there for now. Just know that the dashboard you built will be automatically updated each time you update the **Class_of_2025** data set from SAS Model Studio. Welcome to some significant time savings!

Your takeaway Tasks

Let’s recap what we accomplished in our time together today:

- You got a new job (nice!)
- You grabbed historical data on admissions and stuck it into our model, just like always
- The algorithm gave us a list of applicants… and we could have just run with it.
- But… we instead used Fairness and Bias tools in SAS Model Studio to create a more equitable admissions process – one that better seeks to accept the absolute best candidates rather than the candidate that mirrored the successful applicants of our recent past.

That was a lot of learning – so well done! If nothing else, this exercise hopefully highlights an important point:

> *Algorithms just do what they are told (i.e., maximizing or minimizing some fit statistic). It’s up to us – as humans in the loop – to determine whether the model is being used properly!*

In other words, algorithms are not implicitly evil – but if we feed all the applicant data into the model – particularly a historical model - it will find a way to maximize the differences between the groups in order to meet the objectives of the model fit statistics.

And now I leave the modeling in your capable hands. With two tasks:

- Please continue to address the bias that was unwittingly included in our old admissions approach. Fix them and make the model better.
- Finally, please think about the trade-off between model fit and using potentially biased variables. In other words, how much of a difference did using the biased variables make in leading to a better KS-Youden fit statistics? Were you able to find a better model by not using the more questionable variables – and using more advanced statistical procedures + techniques?
    - Or, perhaps, were you able to create new variables from existing data? Examples: text mining or non-linear variable interactions.