The use case

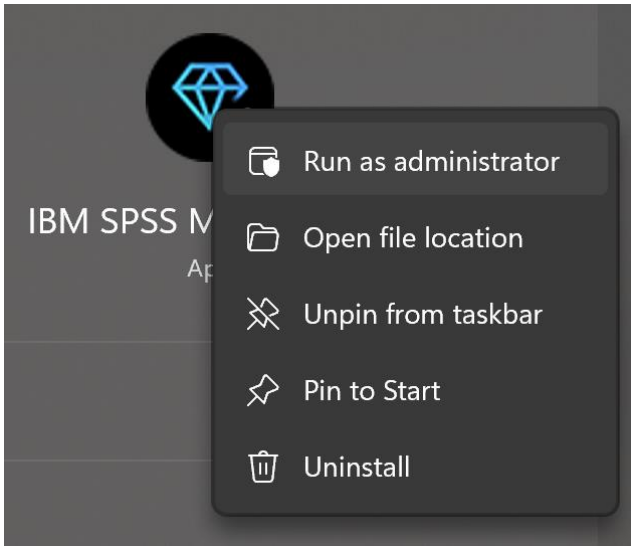
In the workshop you created a model that predicts if a customer is likely to renew their subscription. You can already use that for insight, planning, marketing, etc. What if you could generate a personal message for each customer that is based on their data and this prediction? That's what we're going to implement here!

SPSS Modeler & watsonx.ai

In this tutorial you will see how to use watsonx.ai from an application such as SPSS Modeler. It may get a bit technical at times, but we're looking at a system calling another system where the human is only involved in providing the input and evaluating the output, the rest is automated. Let's go!

Starting SPSS Modeler

On a Windows system, you need to start SPSS Modeler with admin rights to be able to install the watsonx.ai libraries, so right click the SPSS Modeler icon and choose 'Run as administrator'.



On a Mac you can start SPSS Modeler as usual, but you may be asked to allow some access later on.

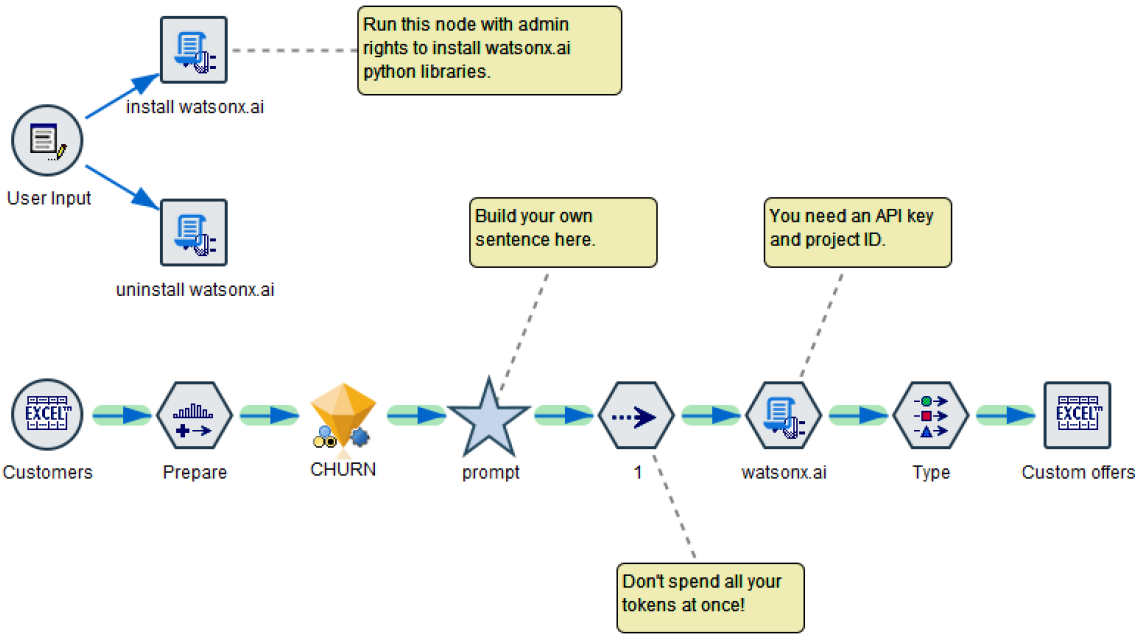
Installing watsonx.ai integration

You can find the stream file to use here:

https://github.com/dzwietering/zerotohero/blob/master/Solution/Tutorial%20SPSS%20Modeler%20-%20prompt%20template.str

If you still have the zerotohere github page open, you find the stream in the Solution Folder.

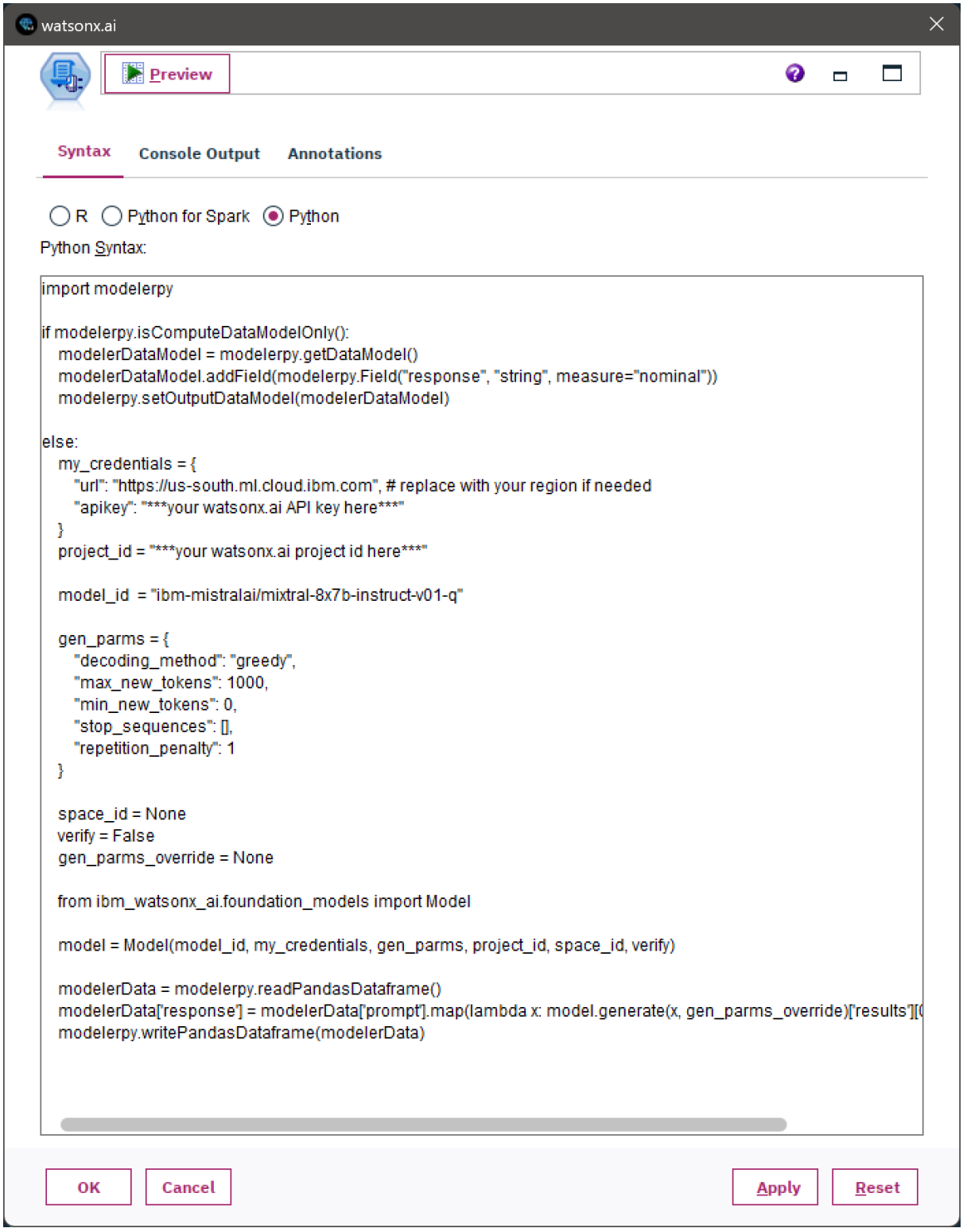
Open this file in Modeler:



The first time you use this stream, you need to install the watsonx.ai libraries. Right click the install node and choose run. This will create quite some output and should end with a 'successfully installed' message.

Configuring the integration

SPSS Modeler needs two things to call watsonx.ai. They need to be filled in the watsonx.ai node. Double click that to see what's inside:



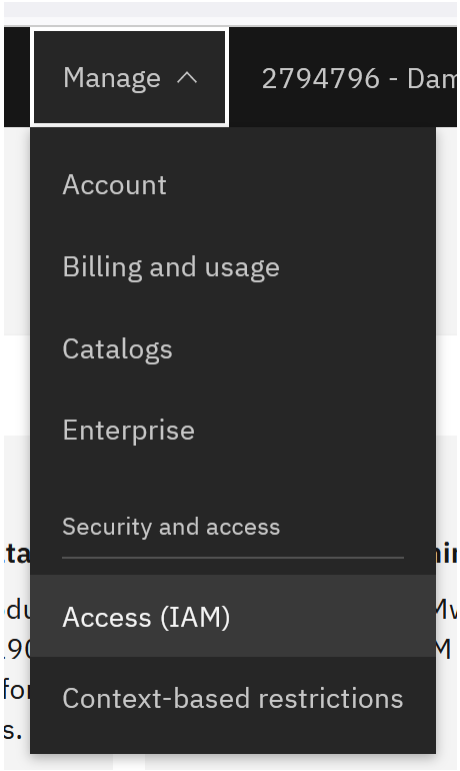
As you see, the integration is written in Python. There's a placeholder for both the API key and the Project ID.

Get your API key

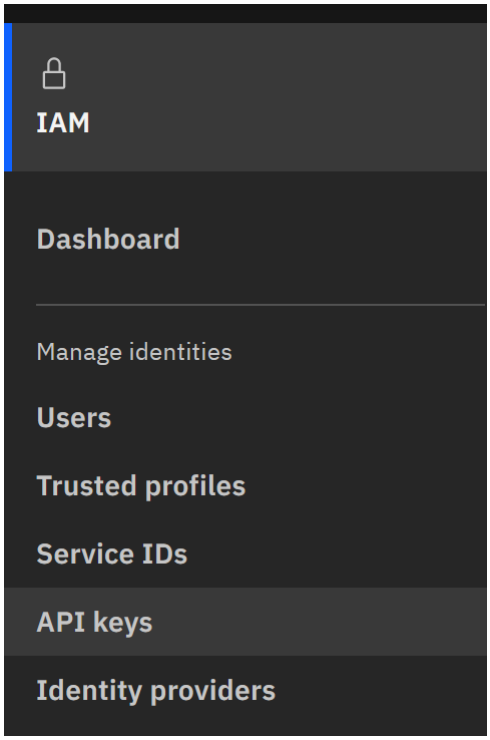
Log in to IBM cloud:

https://cloud.ibm.com/login

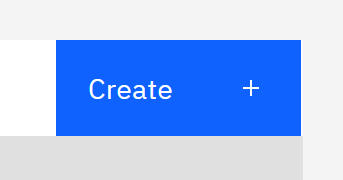
In the top bar, choose Manage, Access (IAM):



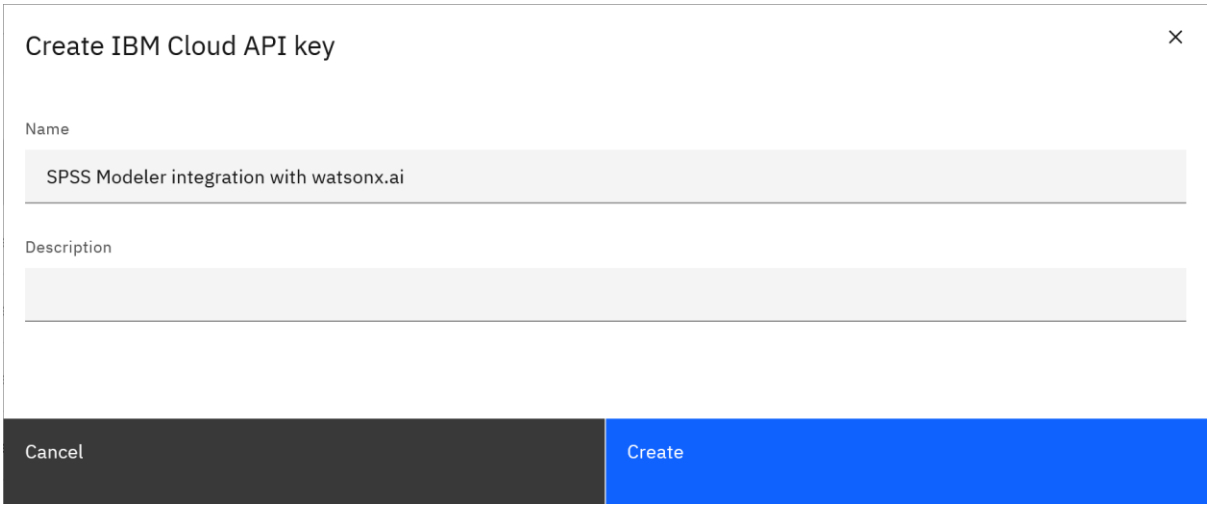
On the left, choose API keys:

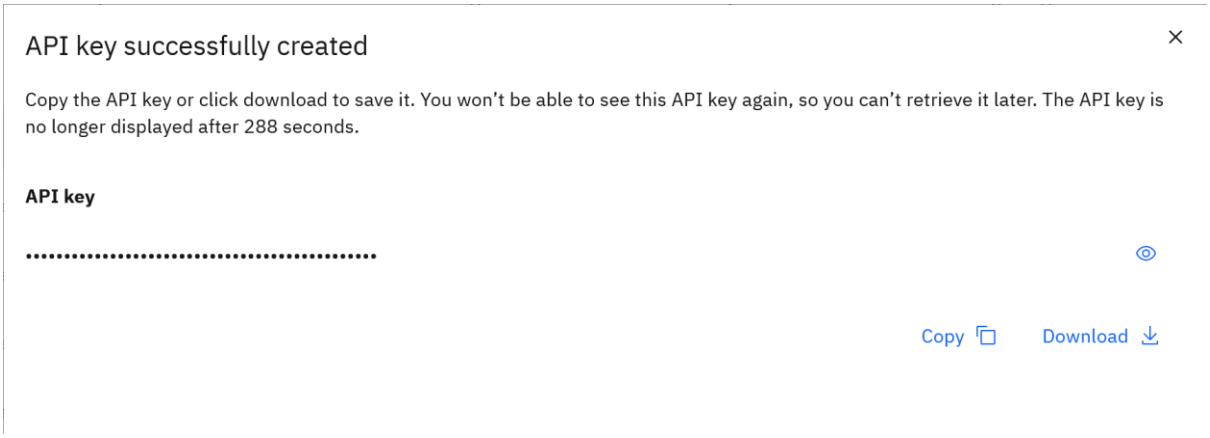


Choose Create:



Choose any name you want and clock Create:





Choose Copy here and paste the key somewhere safe (you will not be able to see it again, if you loose it you have to generate a new one).

But of course, make sure you paste it over the placeholder in Modeler so your key is between the second set of quotes:

"apikey": "***your watsonx.ai API key here***"

Get your Project ID

Login to watsonx.ai:

https://dataplatform.cloud.ibm.com/login

In the prompt engineering lab, you already created the required services and project. Open that project (it should be in the list) and go to the Manage tab. You will find the Project ID there, with a button to copy it. Past that code over the placeholder in Modeler:

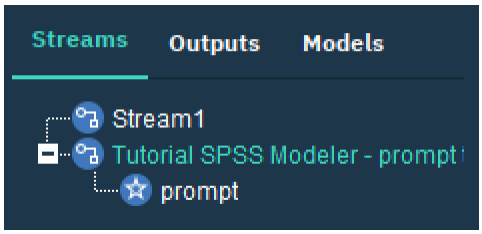
project_id = "***your watsonx.ai project id here***"

Click Ok on the window, save the stream and you're all set to start generating personalized messages!

How it works

The stream takes the familiar customer file, prepares the data, runs the CHURN model and then gets to the star node called 'prompt'. That's actually a small stream in itself. That node generates a sentence from the data, right click it and choose 'Zoom In' to have look:

We won't go into all the details here. You can easily navigate between the main stream and this substream using the Streams menu on the top right:



Right click the prompt node from the main stream and choose Preview to see what it does. The output contains all the original data (now converted to text for the message), the prediction columns and a sentence that is formed from the data, to be used as a prompt for watsonx.ai.

Next in the stream we take only the first record (because performance and capacity is limited) in the sample node that says '1' and then the magic happens in the watsonx.ai node.

Right click the last node, 'Custom offers' and choose run. It will take a couple of seconds and then start Excel with the result data. What do you think?

Run your own experiments

Of course you can increase the number of records in the sample node to see more than one result, but don't run more than a couple, it takes some time and uses a lot of resources.

It may be interesting to change the prompt sentence based on what you learned in the prompt engineering lab. To do that, enter the substream and open the prompt node just before the end there. It contains a long expression that uses the information about the customer to describe them:

"Write a call script for an update call from a telco to a " >< Status >< " " >< Age >< " years old " >< Sex >< " with " >< Children >< " " >< Child >< " who has a " >< '$XFC-CHURN_BIN' >< " probability of " >< '$XF-CHURN' >< " their subscription. They have an estimated annual income of " >< Est_Income >< " and " >< Car_Owner >< " own a car."

The best places to change are the beginning and end. For example, you could change ‘call script’ to ‘marketing email’ or add extra instructions at the end. Make sure you type your text within the quotes. Try adding ‘Use only three sentences and use a happy style.’ at the end, just before the closing quotes and see what that does. To test, go back to the main stream and run the Excel node at the end again and have look at the output.

You can change the output language by changing the beginning of the prompt into “Write a message in Dutch” for example. The mixtral model that we use here is pretty good in several languages, see what it can do in yours!