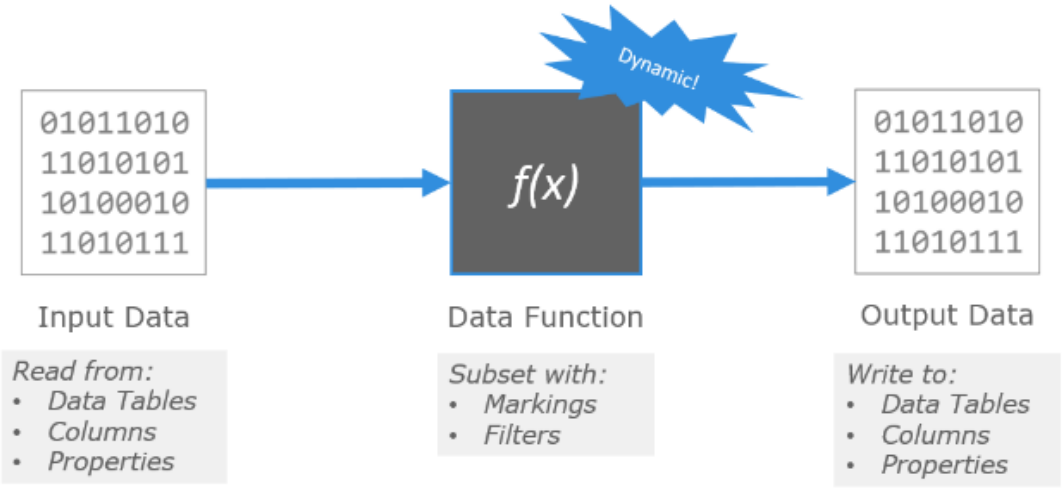
Using Python in Spotfire

Introduction

In this lab you will learn how to use **python script** in Spotfire. This will allow you to combine and extend functionality in Spotfire with advanced analytical capabilities. We will use a **spotfire data function** to accomplish this.

What is a Spotfire Data Function?
Spotfire data functions provide an interface between Visual analytics and advanced analytics. Data functions have inputs and outputs that can be mapped to columns, data tables, document properties etc, and may be responsive to the users marking and filtering. This enables creating dynamic analytical applications that use familiar concepts such as drill down, marking, filtering and drag and drop configuration of visualizations in combination with machine learning or other advanced analytics. This enables providing end users a familiar and easy to use experience powered by data science.



Data Functions may be extended to uses with other languages like MATLAB and SAS, and can be used to connect to other software directly like TIBCO Data Science, TIBCO Statistica, KNIME, and more. These concepts are excluded from this guide, but good to be aware of.

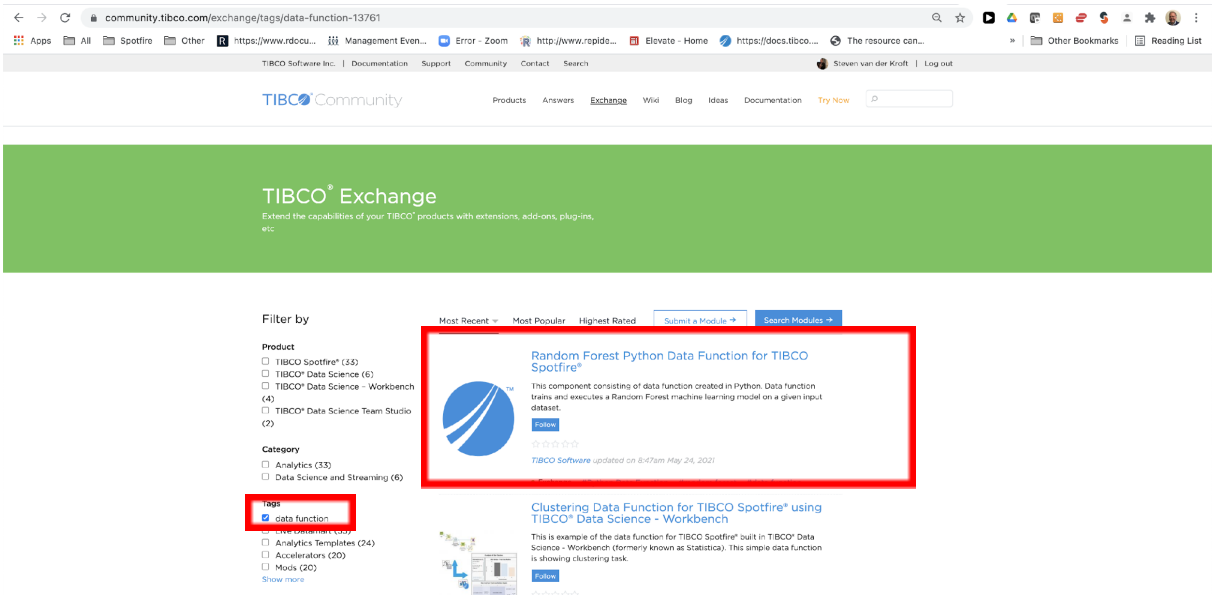
This lab consists of 2 parts. In part 1 you'll take the required actions to create a python data function that can be (re)used in various Spotfire Analysis files. In part 2 you will use the data function that we prepared in part 1 to apply the statistical model to data in Spotfire and visualize the results.

Part 1 - Adding a Python Data Function to Spotfire

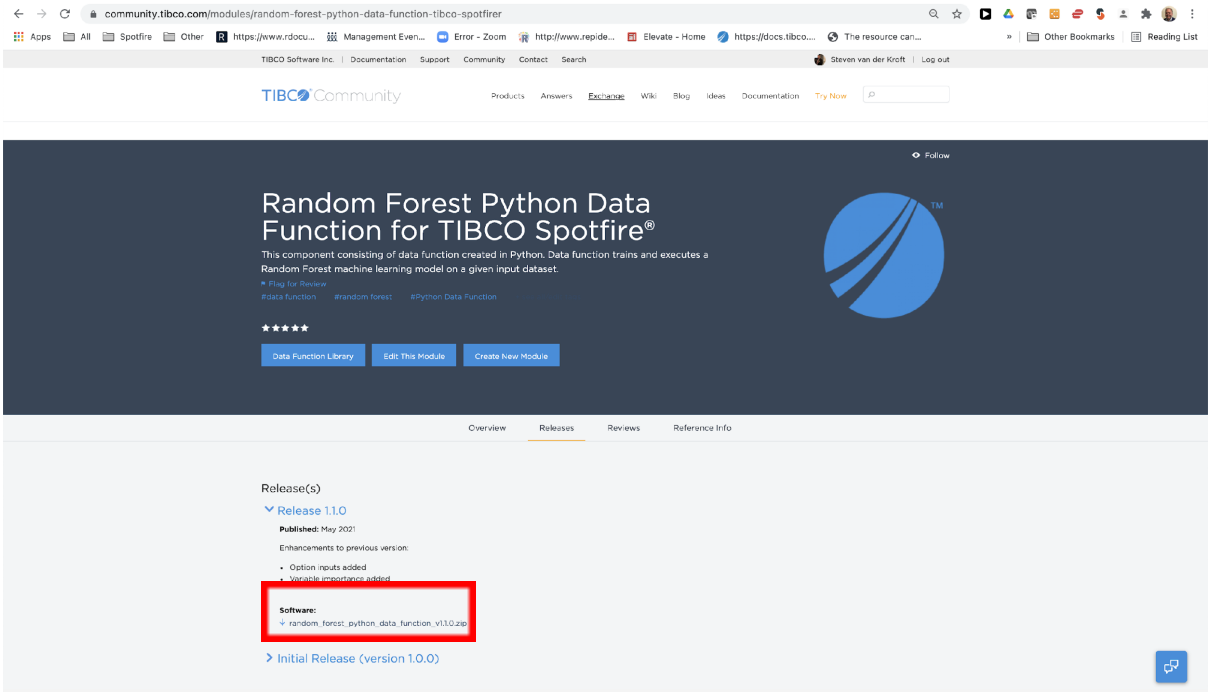
During this part of the workshop we will take you through the process of adding a data function to TIBCO Spotfire. Many companies have Data Scientists that will create Python or R scripts that can be used to, e.g. cluster, predict or prepare data elements.

Also on the TIBCO community exchange there is a large selection of data functions available. In this exercise we will use a 'Random Forest' Data Functions from the community. Let's first download the data function. In order to do so, go to http://community.tibco.com/exchange

Here you will find a large selection of templates, mods, data functions and much more. Select "data function" in the filter by area. There are currently over 30 data functions available. Look for a data function with the name "Random Forest Python Data Function for TIBCO Spotfire" and click on the name.

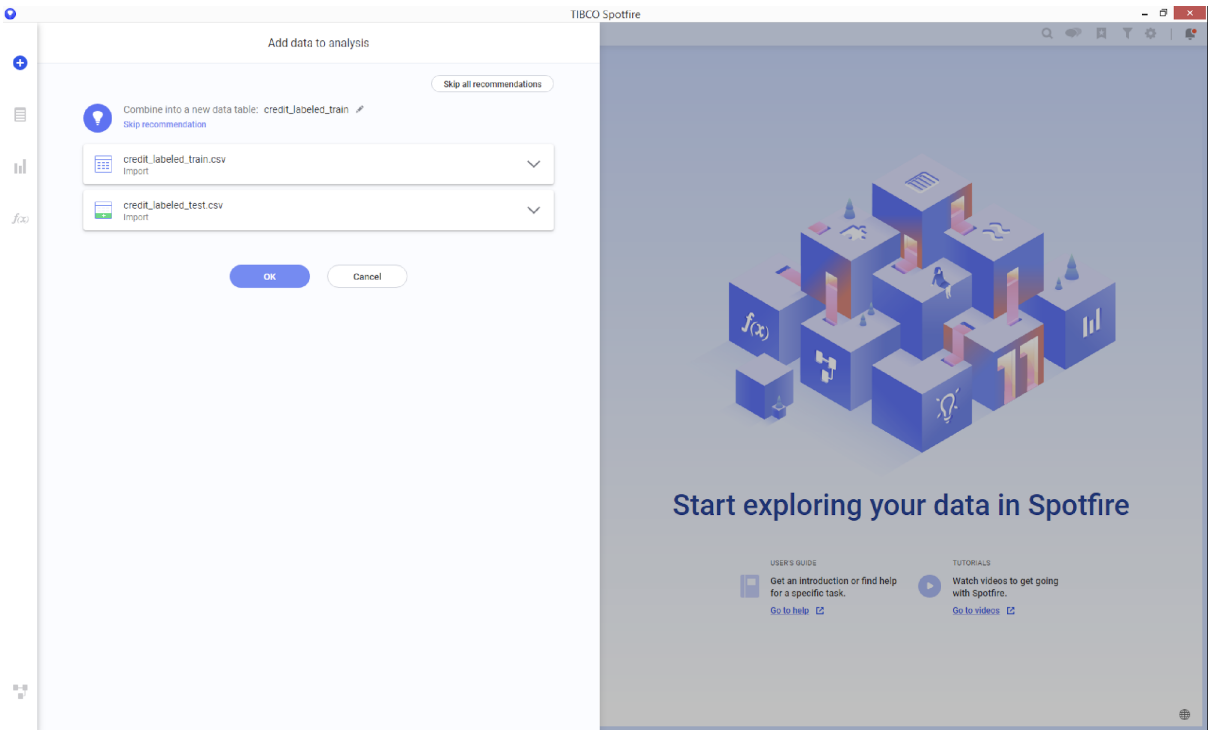


This will open the page where you can read more about the data function. Click on the releases tab and then look for the most recent release and click on the download links as shown on the screenshot below.



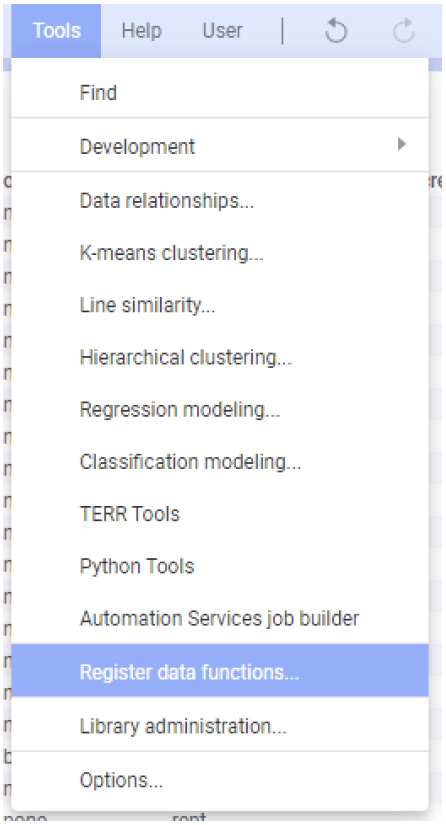
Once the zip file is downloaded unpack the zip file to a location. Keep the location in mind, as we will need this later on.

Now let's open the files that we have used in exercise 2 before. (i.e. credit_labeled_train.csv and credit_labeled_test.csv) and make sure to combine the two files in 1 table as recommended by Spotfire.

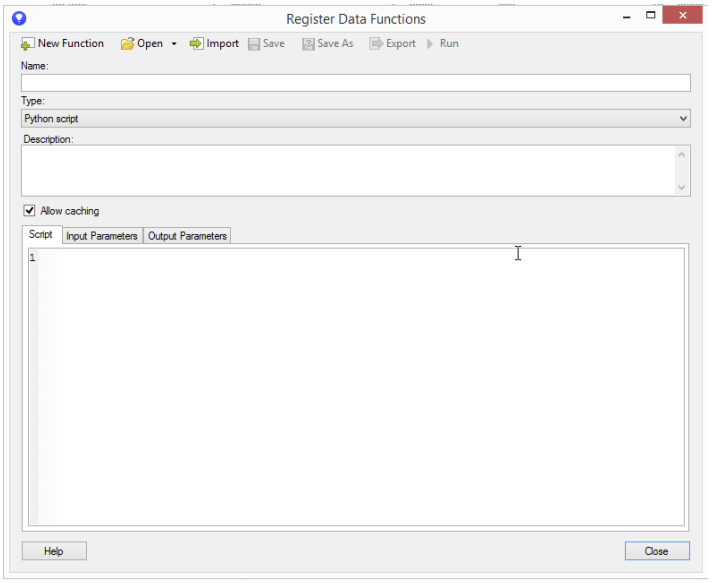


Now let's make sure we add the data function to Spotfire and make it easily reusable. In order to do so we need to add the data function to the Spotfire library and register the data function.

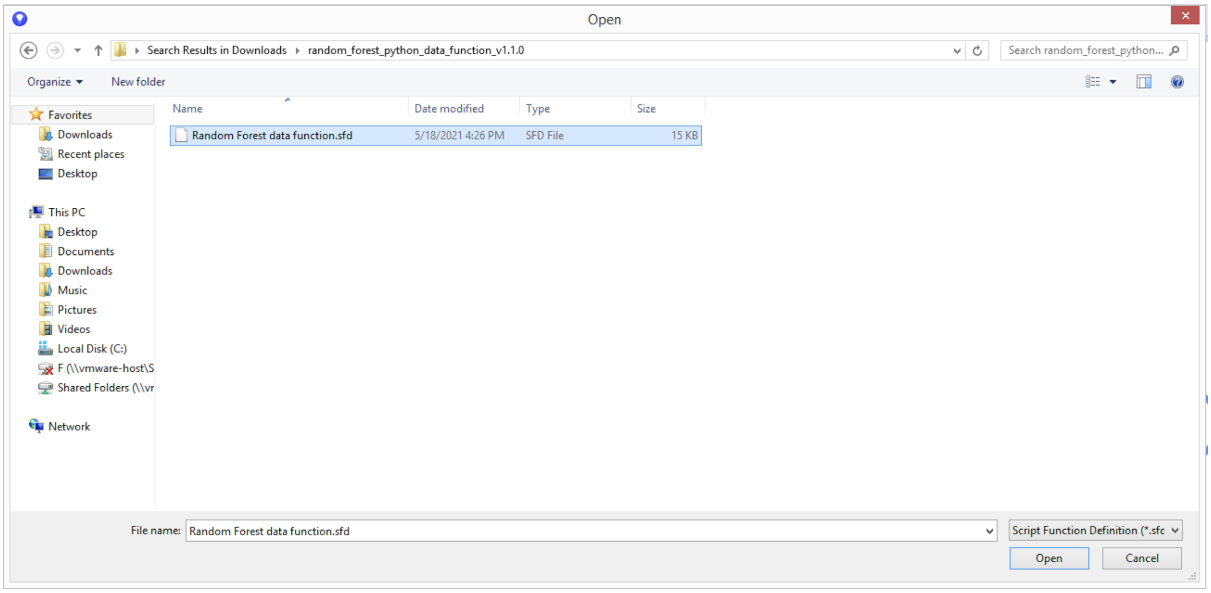
Go to the Tools menu and select Register data functions.



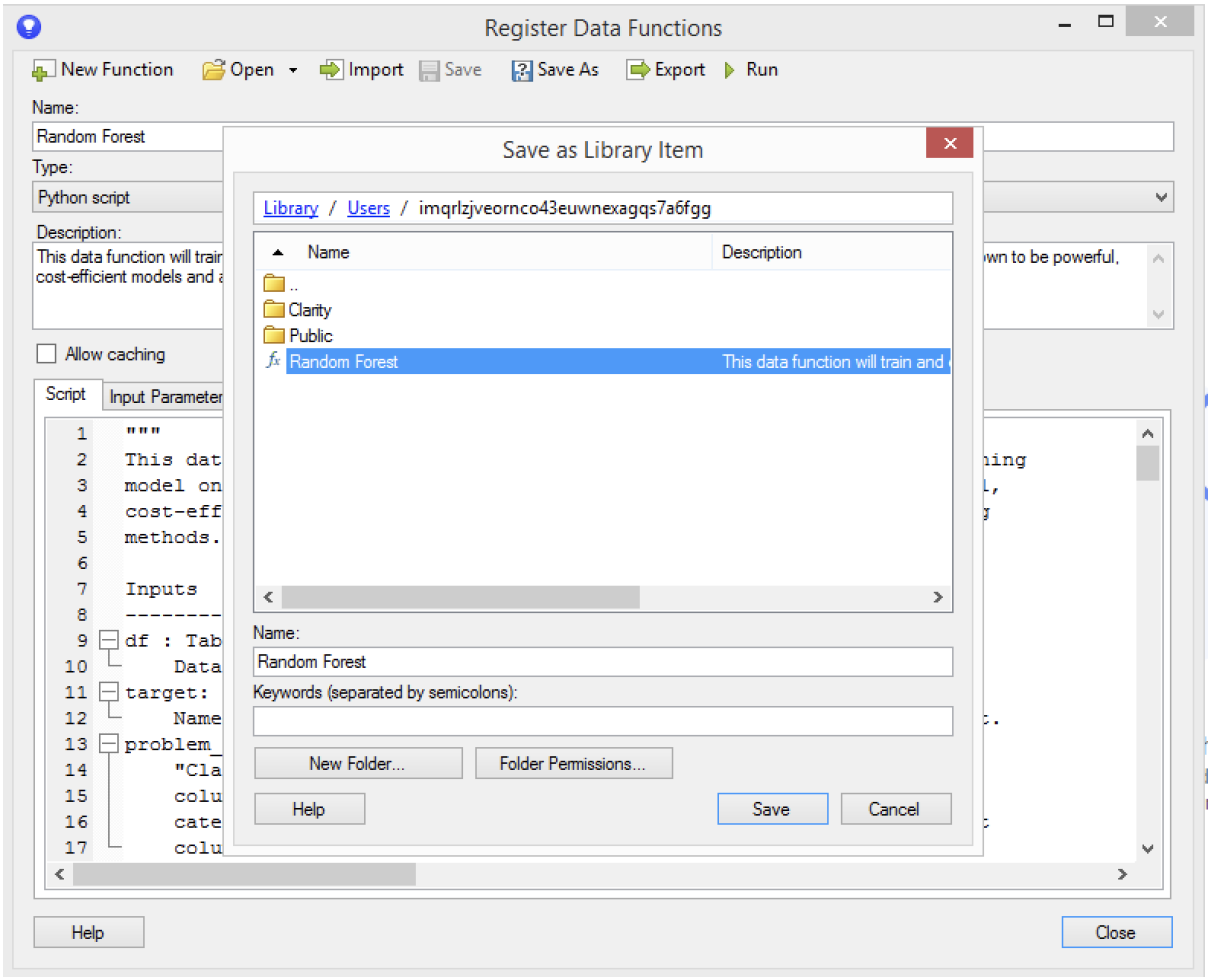
This will bring up the following screen:



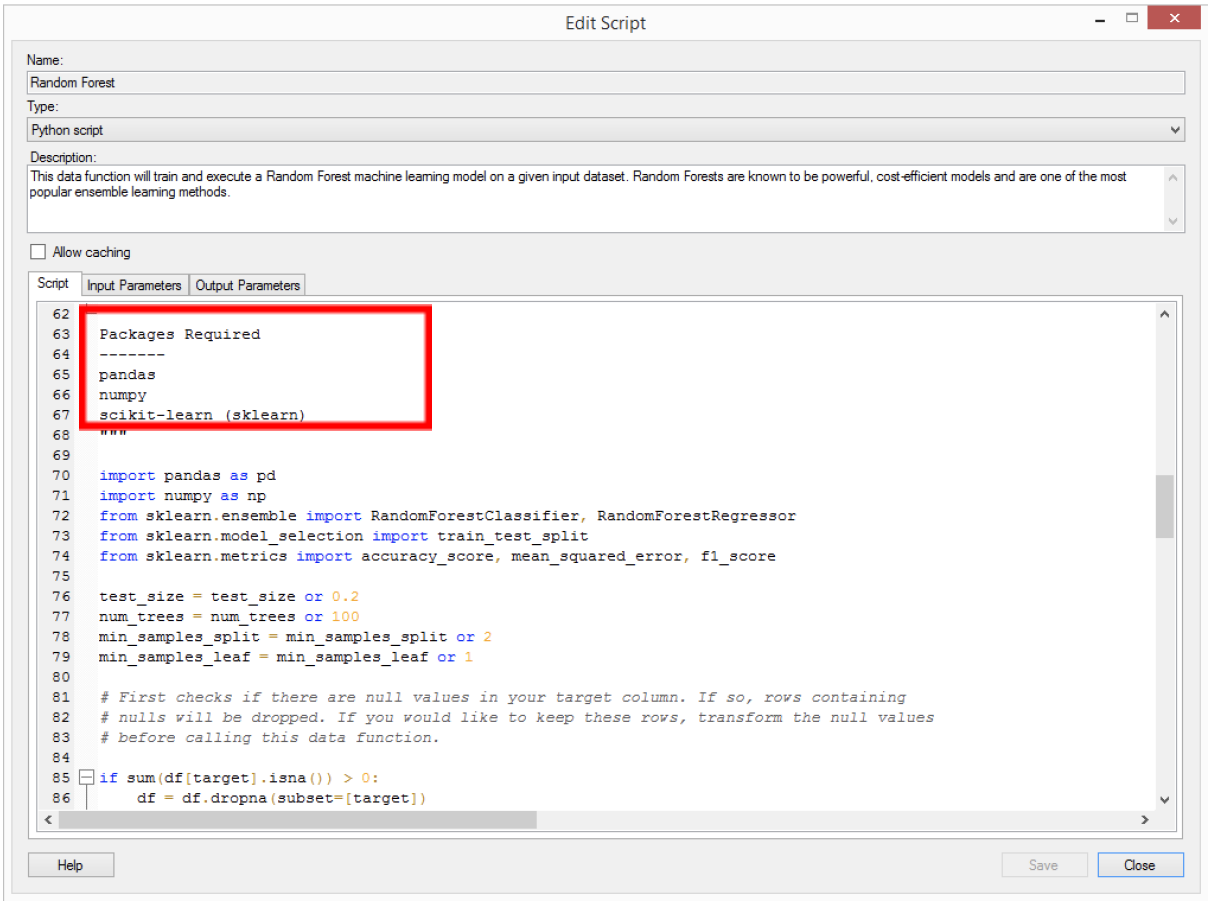
Now click on the import button and browse to the location where you extracted the zip file we downloaded from the community. Click the data function and click open.



Once opened let's save the data function in the library by clicking on the save as button.

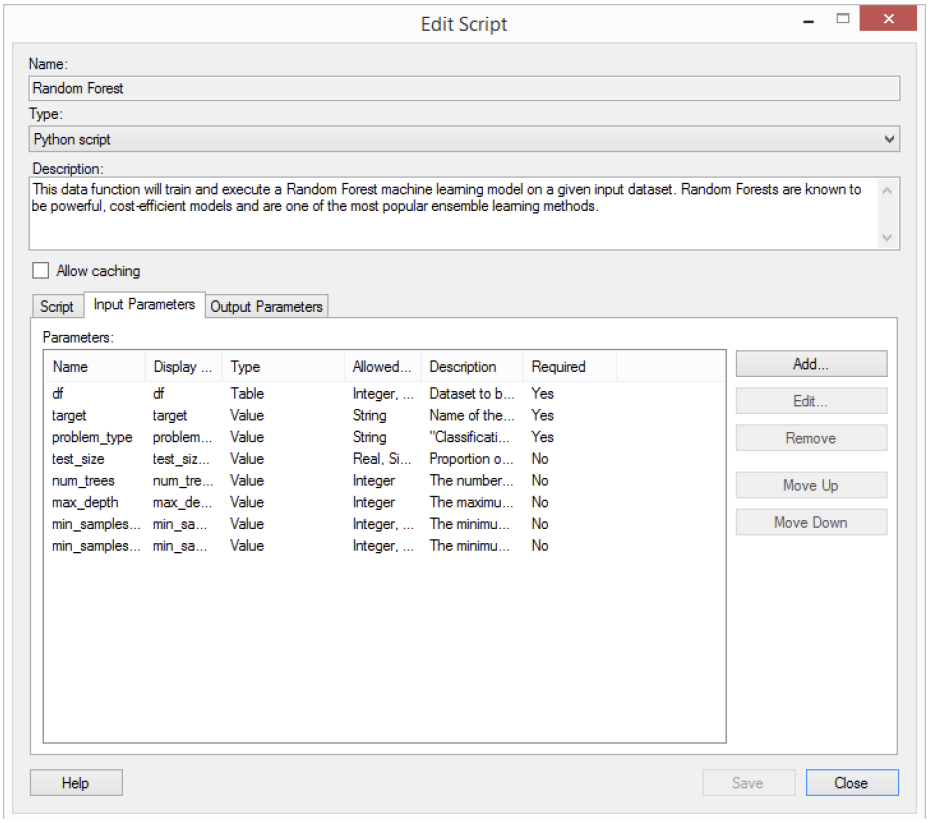


The function is now stored in the library. Before we will use it, let's have a quick look at the script. Click on the Edit Script button. This will show the contents of the script as shown below.

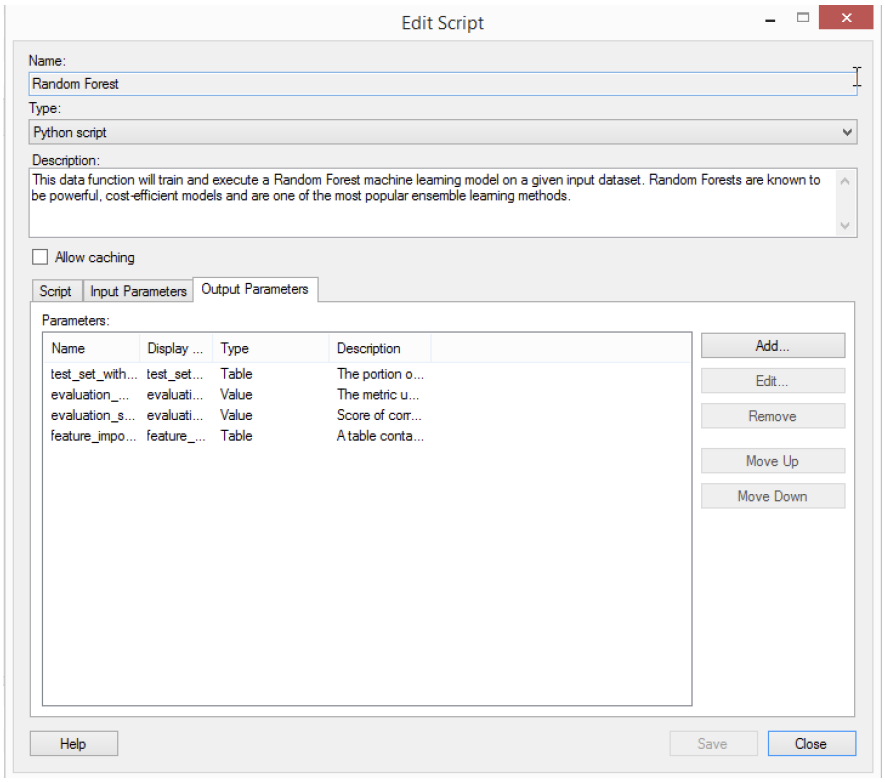


Although you do not need to understand the script itself, you can view the code. In this case comments have been added that eg. Explain what packages are required to run the data function. We need to install the packages before using the data function. We will do that in a moment.

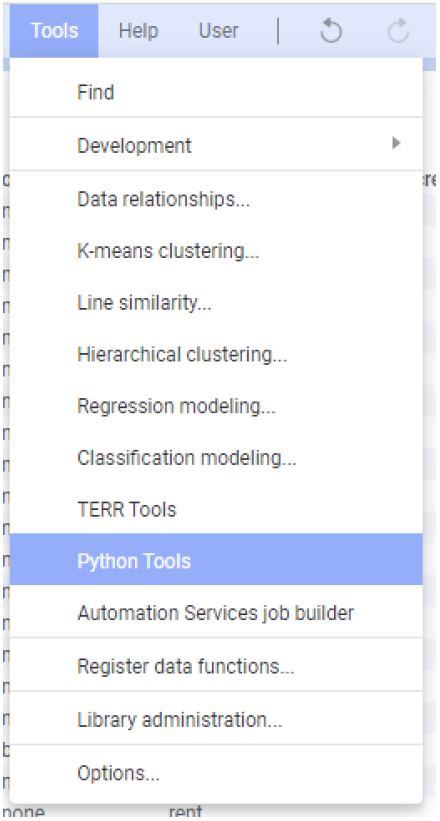
On the next tab, there are a number of input parameters defined. If you would just receive some python code you would have to add the parameters and types yourself. In this case because we downloaded a data function it is already defined.



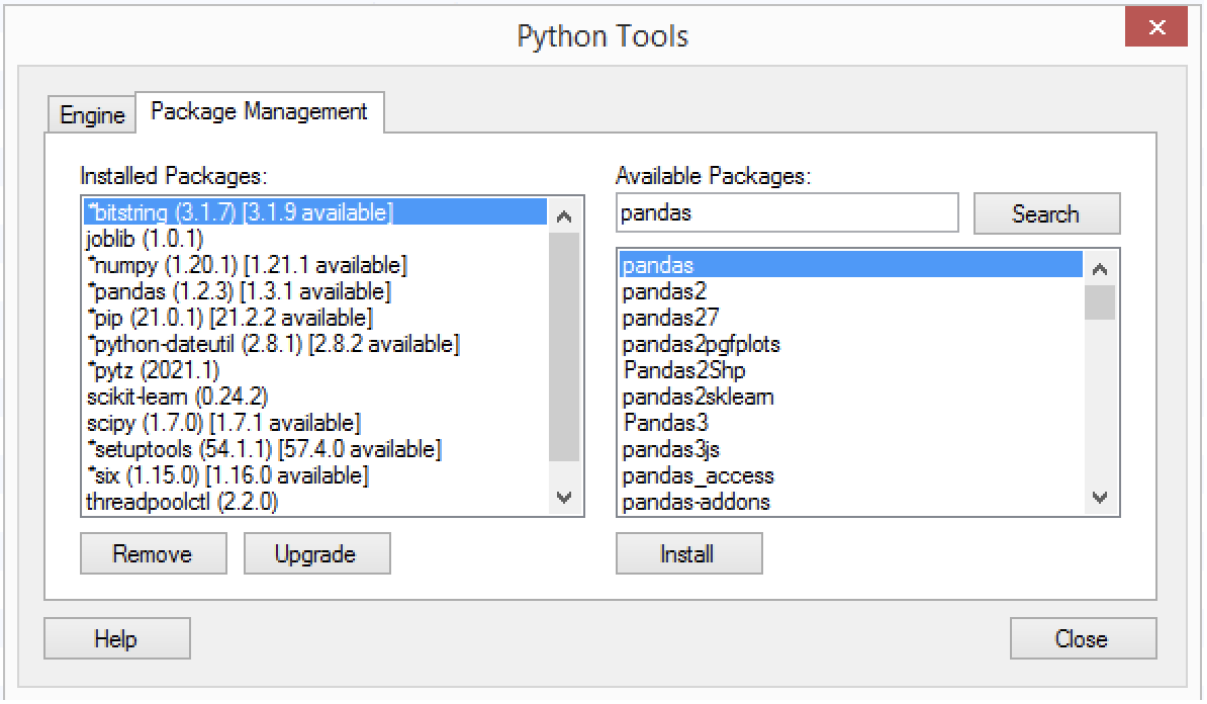
The same applies for the output parameters on the third tab.



Now let's add the required packages. TIBCO Spotfire provides package management for Python as well as R (TERR). In order to access this we go to tools and select Python Tools from the menu.



Here we can search for the packages and install them. First let's search for pandas and install it.

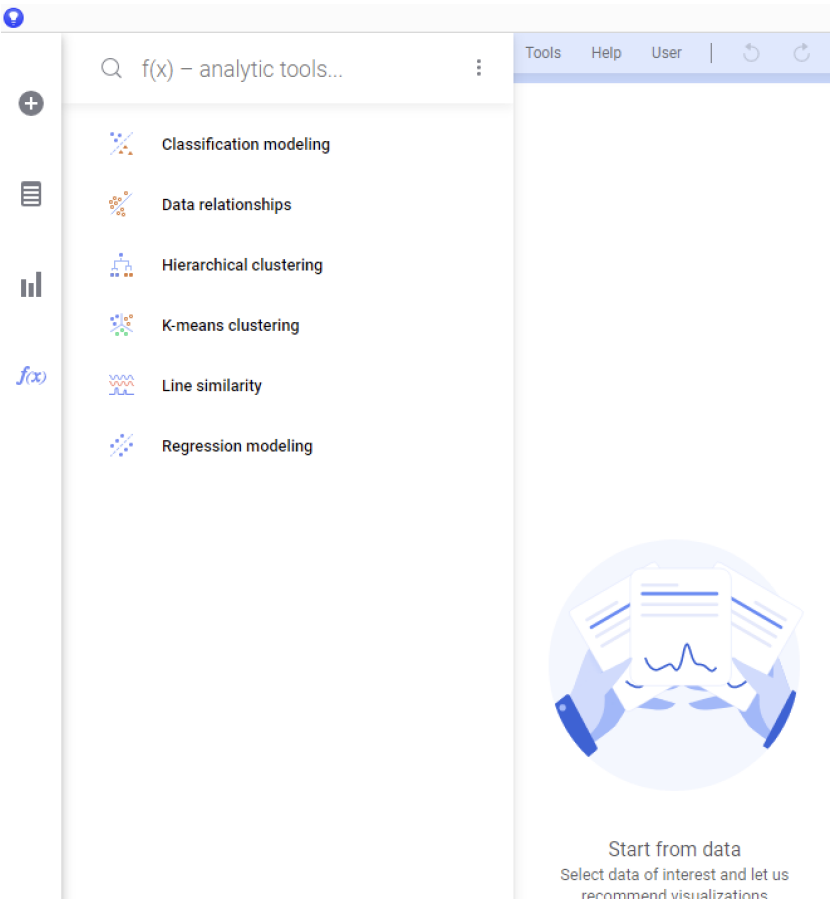


Then repeat this for the two other packages, i.e.
numpy
scikit-learn

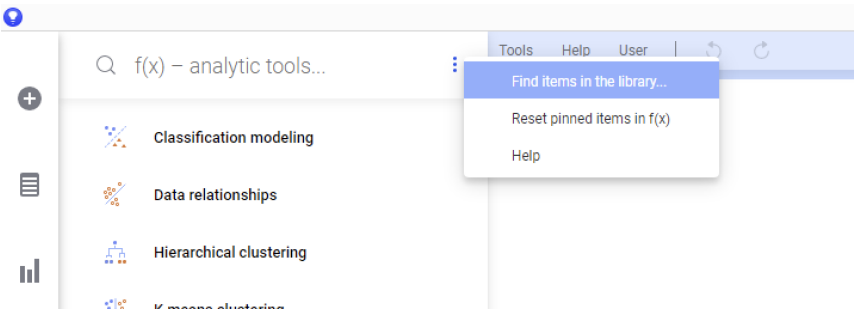
Now the environment is set up for usage.

Part 2 - Using the Data Function in Spotfire

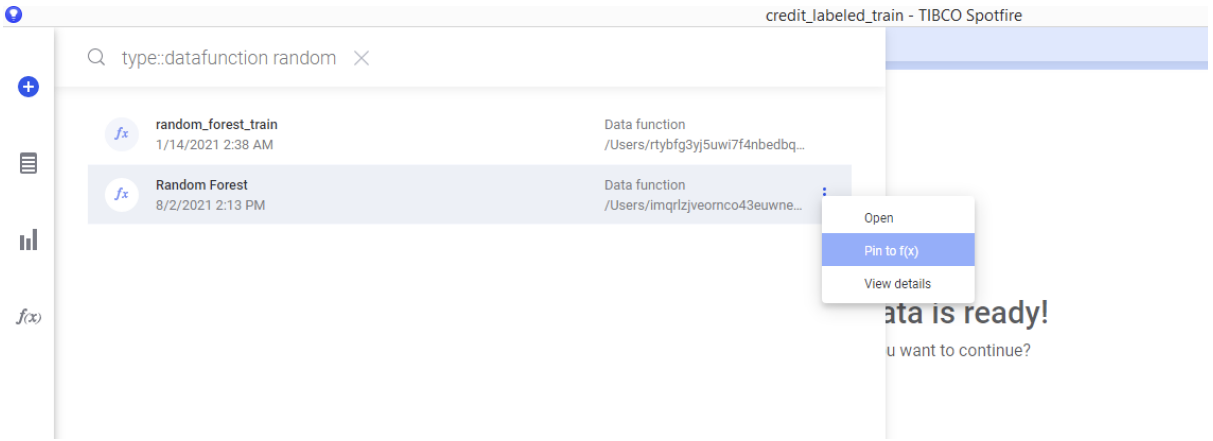
One of the latest additions to Spotfire is the f(x) fly-out. The is accessible from the menu on the left by clicking on the f(x) symbol. This will open the fly-out.



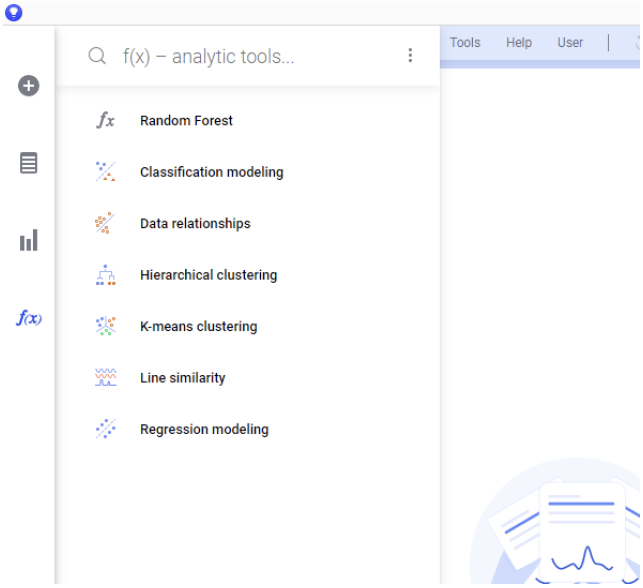
As we have previously added the data function to the library we can access it from the fly-out. In order to do so, click on the three stacked dots and click “Find items in library…”



Then browse to the data function or type part of the name in the search window.



Once found, click on the name and then again on the three stacked dots to the right to pin the data function to the fly-out so it is easily accessible in future.



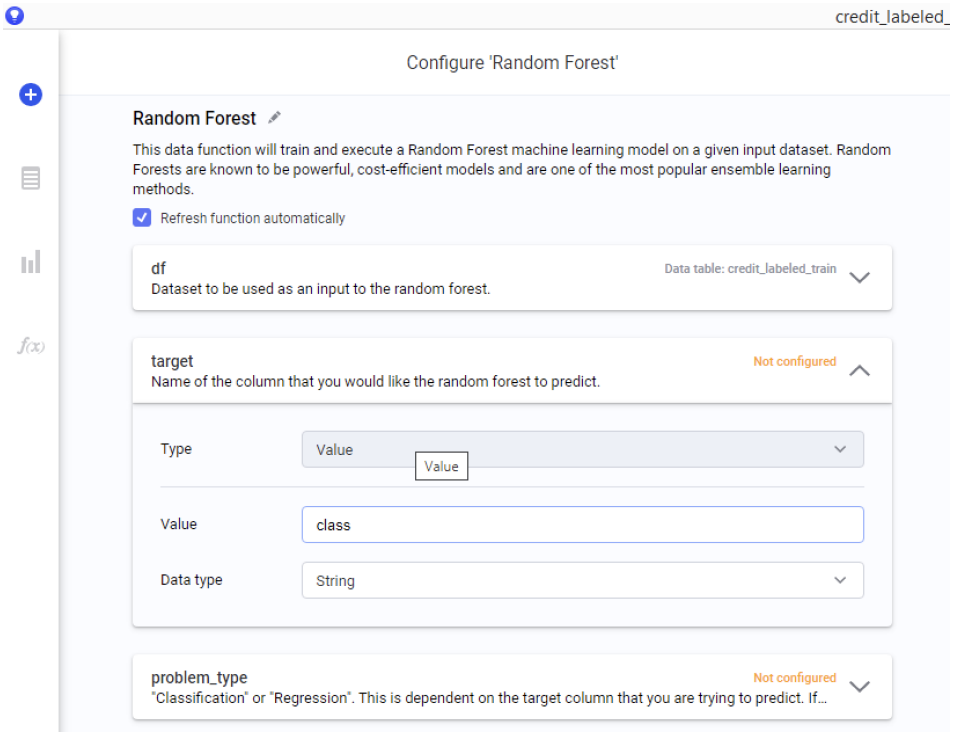
As you can see on the screenshot above, the data function is now shown on the f(x) flyout. When we select it from there we need to tell Spotfire what input we want to use and how we want to deal with the output from the data function.

As we can see there are a couple of required input fields and a couple of optional input fields. We want to use the random forest data function in this case to analyse the data table we have currently loaded and create a model that predicts whether the customers have a good or bad credit rating (class). This type of problem requires a classification model.

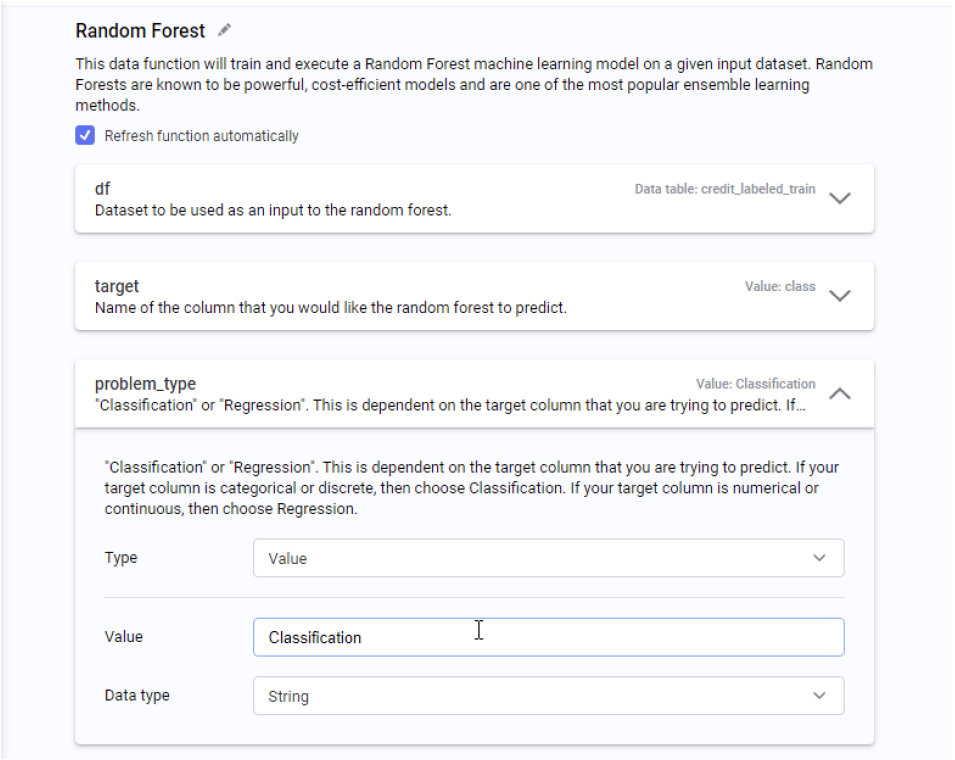
Let's feed this into the configuration. First let's check the Refresh function automatically button, so spotfire will execute the function when needed. Then expand the df section by clicking on the arrow on the right.

You will notice that as we have only one dataset loaded, Spotfire will automatically select the dataset as soon as we expand the df element in the menu.

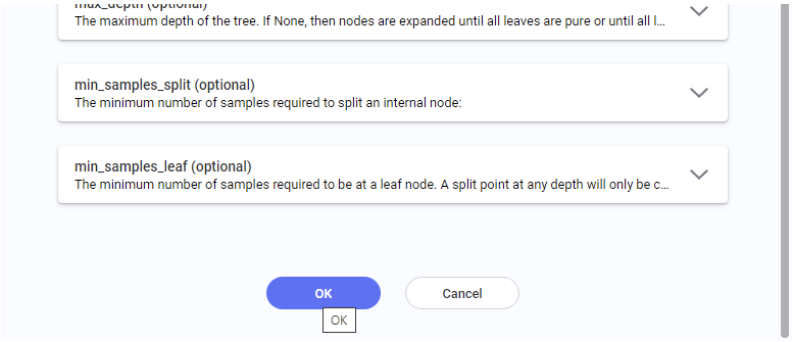
Next expand the target and specify the column from the dataset we are trying to predict (class).



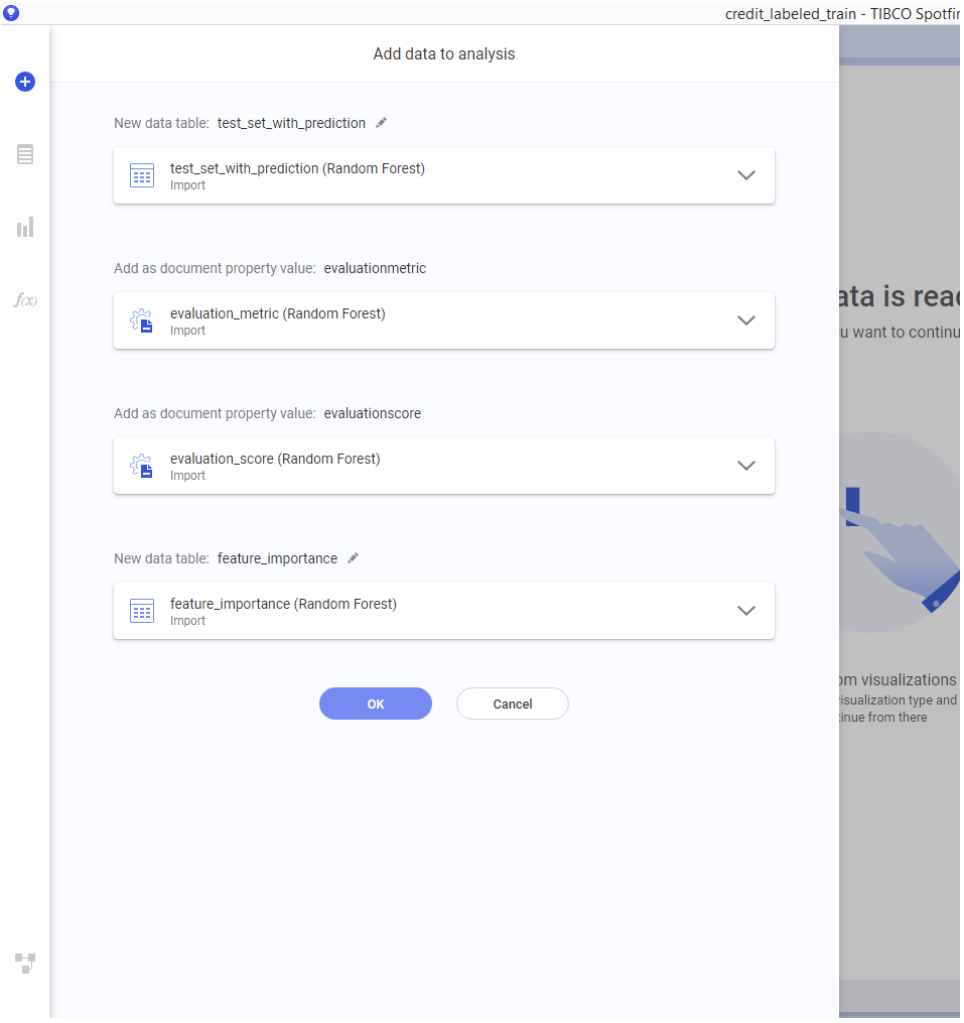
Finally we need to specify the problem type, by typing "Classification" in the problem type value box.



Now scroll down to the bottom and you will see the OK button in blue. If the button is not available, make sure you have pressed tab after typing Classification.

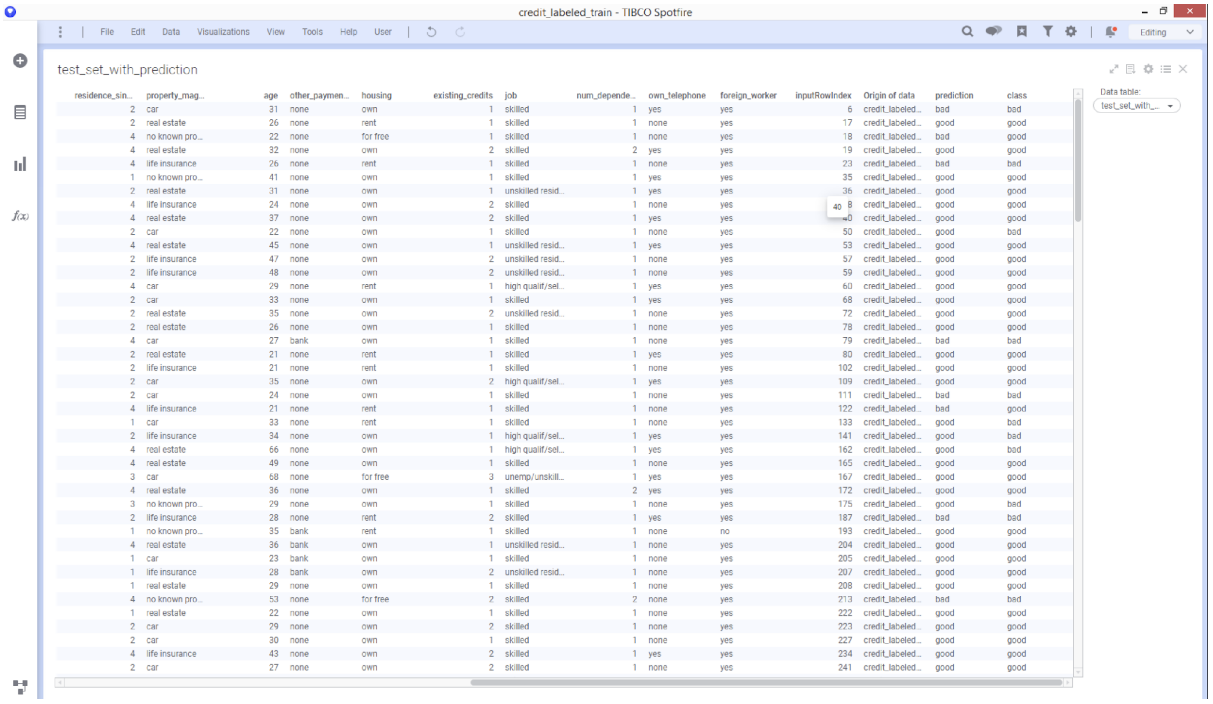


Clicking OK will bring you to the next screen that specifies the output elements. By default Spotfire will create the settings for you, but you could change this by changing eg. The table name or property names that have been created by default.



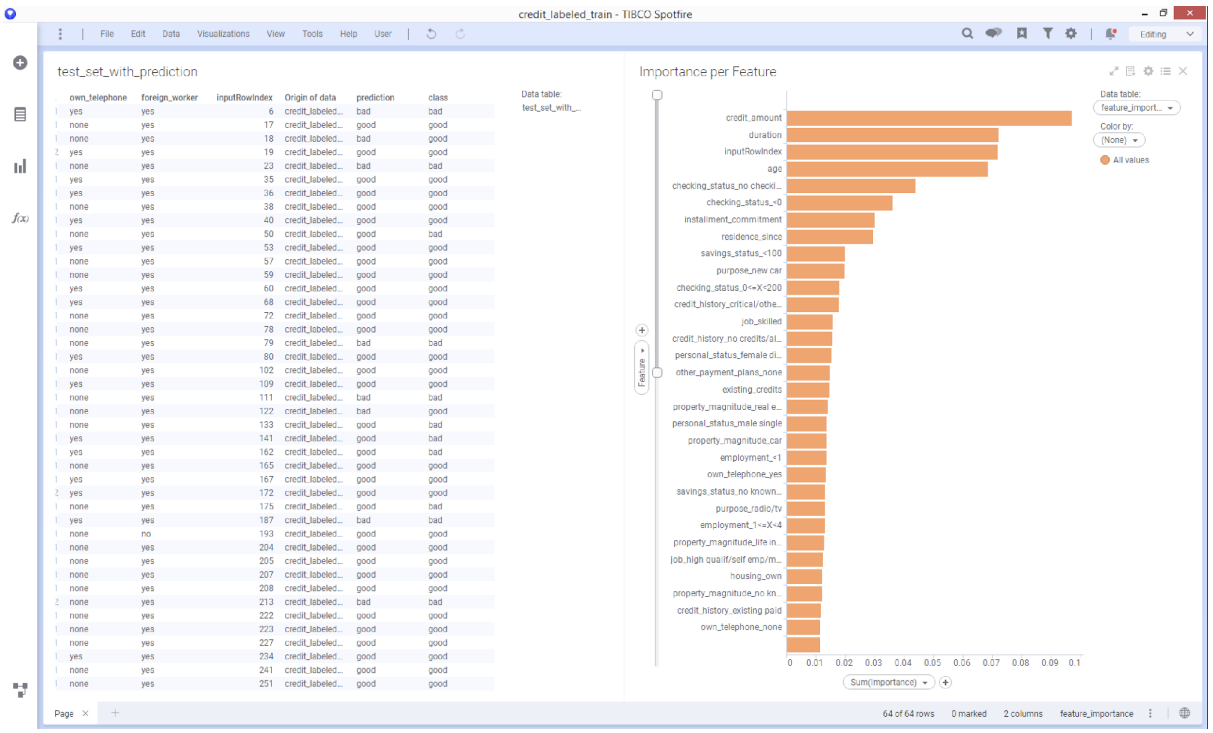
The data function will add a new table with the test set, including a prediction as well as a table with the variable importance. Finally two properties are created with an evaluation metric and an evaluation score.

Let’s inspect the table that is created with the predictions for the test set.

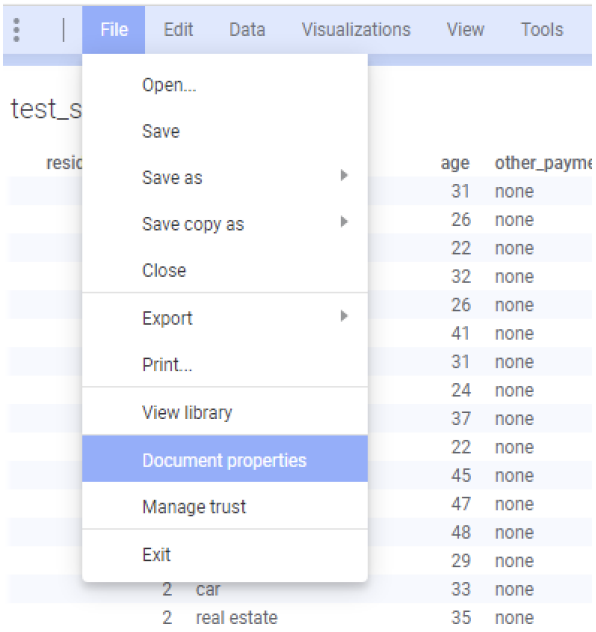


residence_sin...	property_mag...	age	other_paymen...	housing	existing_credits	job	num_depende...	own_telephone	foreign_worker	inputRowIndex	Origin of data	prediction	class
2	car	31	none	own	1	skilled	1	yes	yes	6	credit_labeled...	bad	bad
2	real estate	26	none	rent	1	skilled	1	none	yes	17	credit_labeled...	good	good
4	no known pro...	22	none	for free	1	skilled	1	none	yes	18	credit_labeled...	bad	good
4	real estate	32	none	own	2	skilled	2	yes	yes	19	credit_labeled...	good	good
4	life insurance	26	none	rent	1	skilled	1	none	yes	23	credit_labeled...	bad	bad
1	no known pro...	41	none	own	1	skilled	1	yes	yes	35	credit_labeled...	good	good
2	real estate	31	none	own	1	unskilled resid...	1	yes	yes	36	credit_labeled...	good	good
4	life insurance	24	none	own	2	skilled	1	none	yes	40	credit_labeled...	good	good
4	real estate	37	none	own	2	skilled	1	yes	yes	40	credit_labeled...	good	good
2	car	22	none	own	1	skilled	1	none	yes	50	credit_labeled...	good	bad
4	real estate	45	none	own	1	unskilled resid...	1	yes	yes	53	credit_labeled...	good	good
2	life insurance	47	none	own	2	unskilled resid...	1	none	yes	57	credit_labeled...	good	good
2	life insurance	48	none	own	2	unskilled resid...	1	none	yes	59	credit_labeled...	good	good
4	car	29	none	rent	1	high qualif/sel...	1	yes	yes	60	credit_labeled...	good	good
2	car	33	none	own	1	skilled	1	yes	yes	68	credit_labeled...	good	good
2	real estate	35	none	own	2	unskilled resid...	1	none	yes	72	credit_labeled...	good	good
2	real estate	26	none	own	1	skilled	1	none	yes	78	credit_labeled...	good	good
4	car	27	bank	own	1	skilled	1	none	yes	79	credit_labeled...	bad	bad
2	real estate	21	none	rent	1	skilled	1	yes	yes	80	credit_labeled...	good	good
2	life insurance	21	none	rent	1	skilled	1	none	yes	102	credit_labeled...	good	good
2	car	35	none	own	2	high qualif/sel...	1	yes	yes	109	credit_labeled...	good	good
2	car	24	none	own	1	skilled	1	none	yes	111	credit_labeled...	bad	bad
4	life insurance	21	none	rent	1	skilled	1	none	yes	122	credit_labeled...	bad	good
1	car	33	none	rent	1	skilled	1	none	yes	133	credit_labeled...	good	bad
2	life insurance	34	none	own	1	high qualif/sel...	1	yes	yes	141	credit_labeled...	good	bad
4	real estate	66	none	own	1	high qualif/sel...	1	yes	yes	162	credit_labeled...	good	bad
4	real estate	49	none	own	1	skilled	1	none	yes	165	credit_labeled...	good	good
3	car	68	none	for free	3	unemp/unskill...	1	yes	yes	167	credit_labeled...	good	good
4	real estate	36	none	own	1	skilled	2	yes	yes	172	credit_labeled...	good	good
3	no known pro...	29	none	own	1	skilled	1	none	yes	175	credit_labeled...	good	bad
2	life insurance	28	none	rent	2	skilled	1	yes	yes	187	credit_labeled...	bad	bad
1	no known pro...	35	bank	rent	1	skilled	1	none	no	193	credit_labeled...	good	good
4	real estate	36	bank	own	1	unskilled resid...	1	none	yes	204	credit_labeled...	good	good
1	car	23	bank	own	1	skilled	1	none	yes	205	credit_labeled...	good	good
1	life insurance	28	bank	own	2	unskilled resid...	1	none	yes	207	credit_labeled...	good	good
1	real estate	29	none	own	1	skilled	1	none	yes	208	credit_labeled...	good	good
4	no known pro...	53	none	for free	2	skilled	2	none	yes	213	credit_labeled...	bad	bad
1	real estate	22	none	own	1	skilled	1	none	yes	222	credit_labeled...	good	good
2	car	29	none	own	2	skilled	1	none	yes	223	credit_labeled...	good	good
2	car	30	none	own	1	skilled	1	none	yes	227	credit_labeled...	good	good
4	life insurance	43	none	own	2	skilled	1	yes	yes	234	credit_labeled...	good	good
2	car	27	none	own	2	skilled	1	none	yes	241	credit_labeled...	good	good

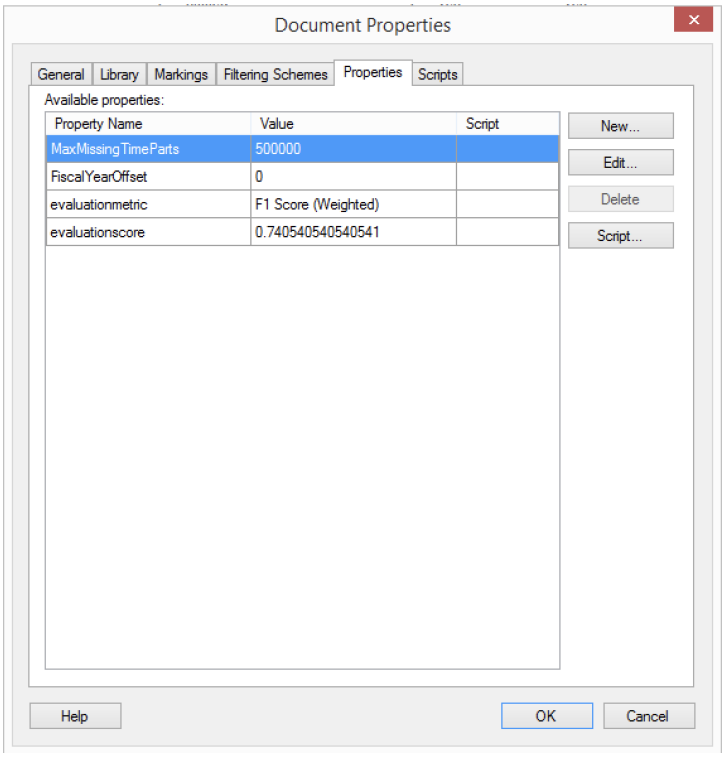
Also add a bar chart with the variable importance. As you have done the first part of the workshop we will let you work out how the configure the bar chart to look like the one on the screenshot below:



Finally let's check the document properties that have been created. Go to file > document properties and select "Document Properties"



You will see the values in the list on the Properties Tab.



If you are ready and there is time left, see if you can visualize the document properties in the analysis file, besides the table and the bar chart (hint: use a text area and property controls to do this).

You have now added a python data function to Spotfire, configured it and executed it successfully. You also made it easy to find by pinning it to the f(x) fly-out.