AI Workshop: Predict Student Performance

In this AI workshop, you are going to build a model to predict student performance. The data has been collected during the 2005-2006 school year from two public schools from the Alentejo region of Portugal. We will look at their maths performance.

The dataset can be downloaded here and comes originally from the UCI Machine Learning repository site, where you can also find more information about the data:

For this workshop, we want to find out whether we could build a model to predict wether a student will pass or not, and therefore, we added a new variable “Pass”, based on the final grade of the students.

We highly recommend you visit that site and investigate what kind of data you have available.

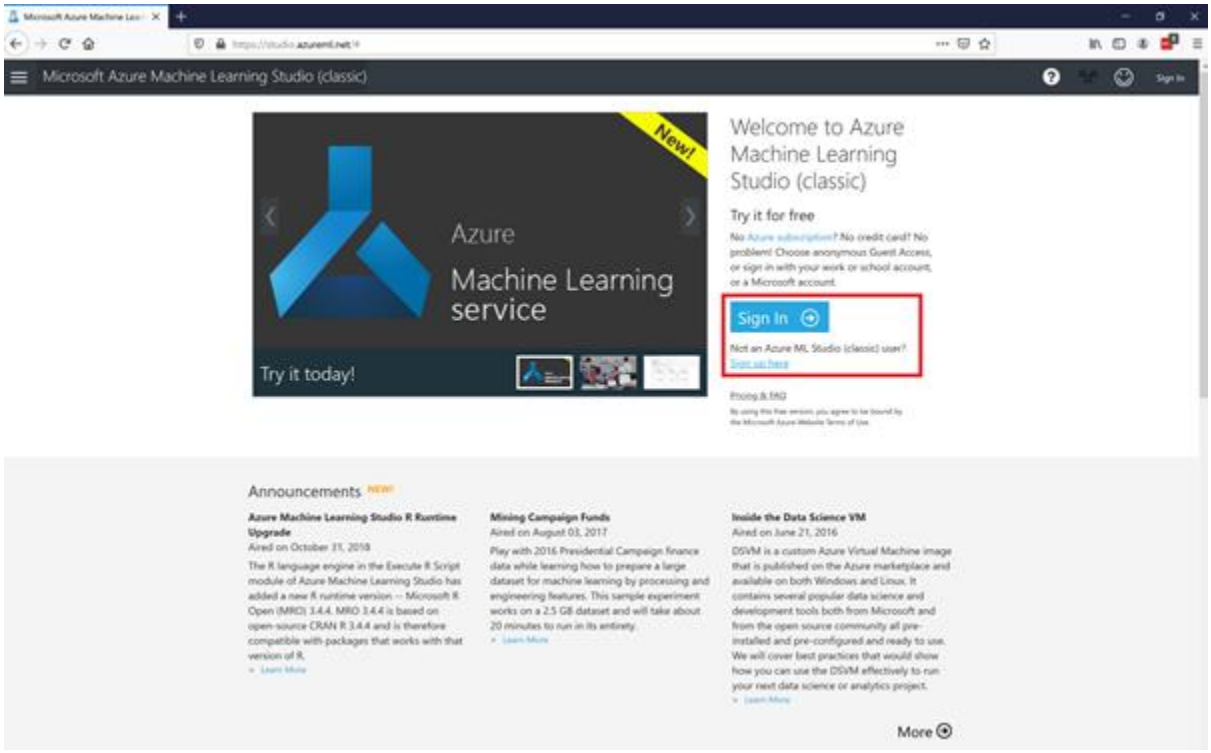
Note: this workshop is to get in touch with machine learning. We won’t pretend to build an excellent model in 1 hour. In the real world you would have to do a lot more, but this workshop does give you an idea about the steps, with the free option of the Microsoft Azure Machine Learning Studio (classic).

Steps to build the model

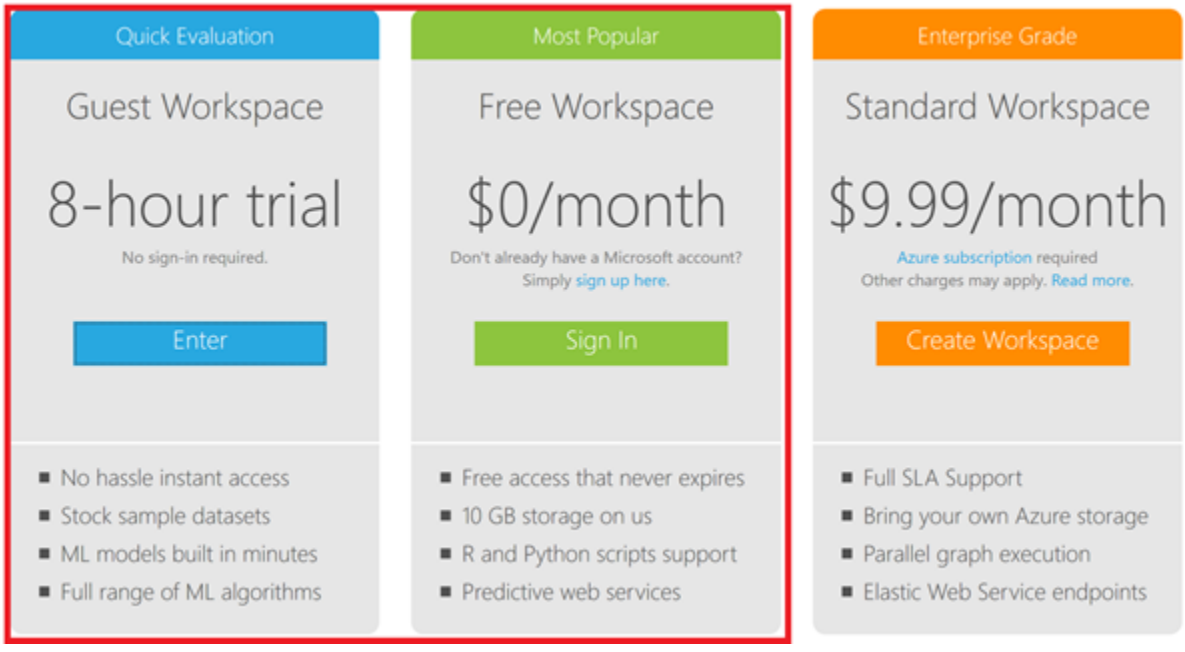
You will first open the online environment where you will build your model. Then you will upload the dataset, which you can then select and inspect the. Next, you will select the required columns and transform them if needed. Then you will split the dataset into 2 parts: 1 part to train the model with, and 1 to test the model with. You will train the model and use this model to score the test dataset. Finally, you will evaluate the model.

Step 1: Get access to the environment

Go to https://studio.azureml.net/ and select **Sign up here** for Azure ML Studio.

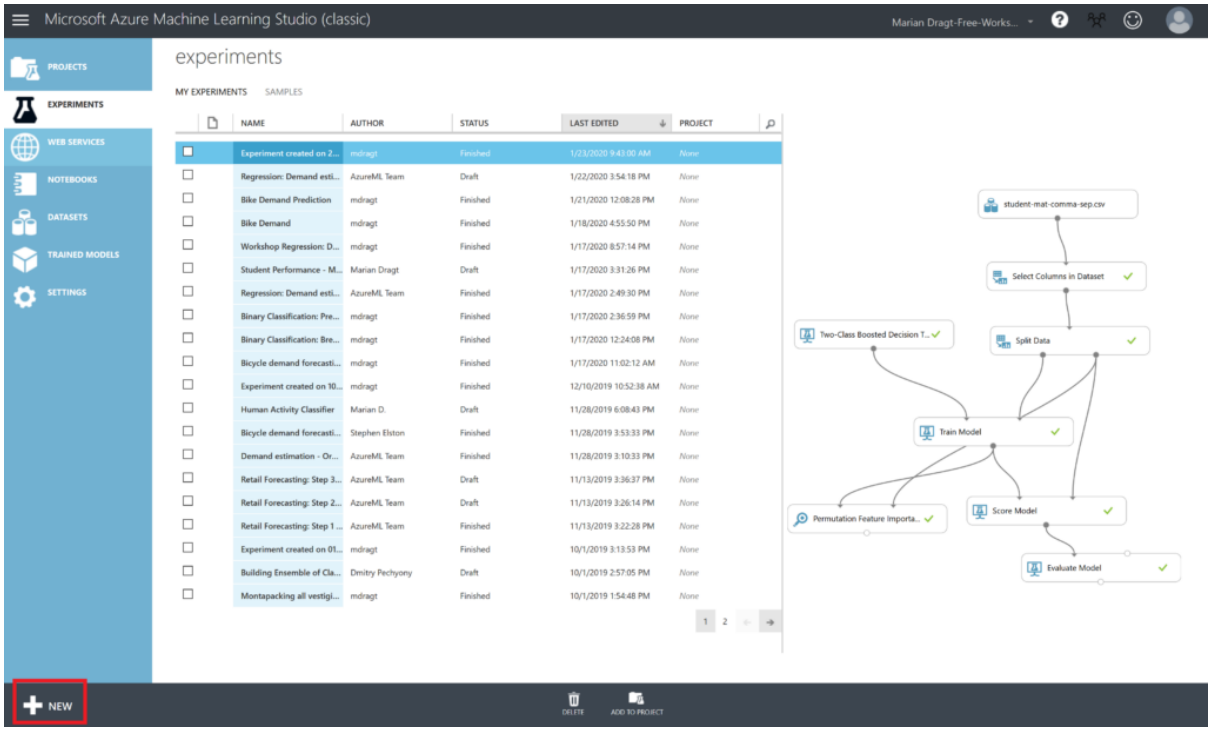


For this workshop, you can select the 8-hour trial. After entering, the Azure Machine Learning Studio environment will be opened.

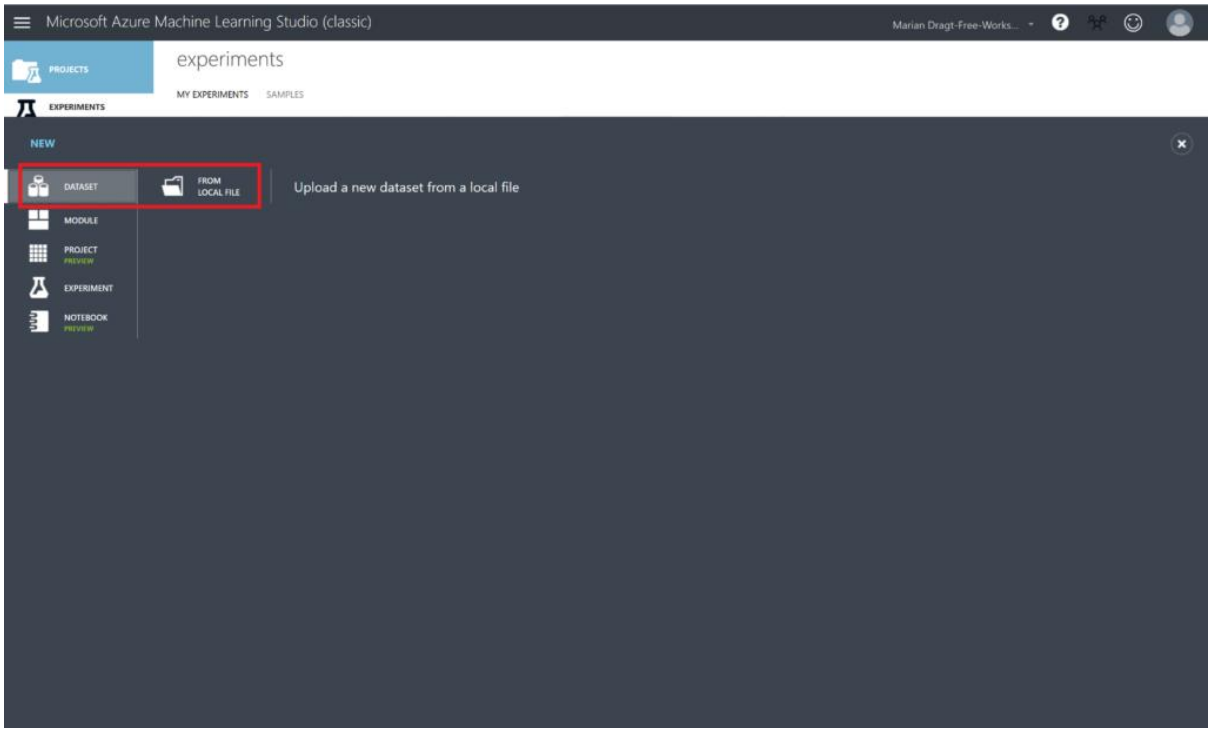


Step 2: Upload the dataset

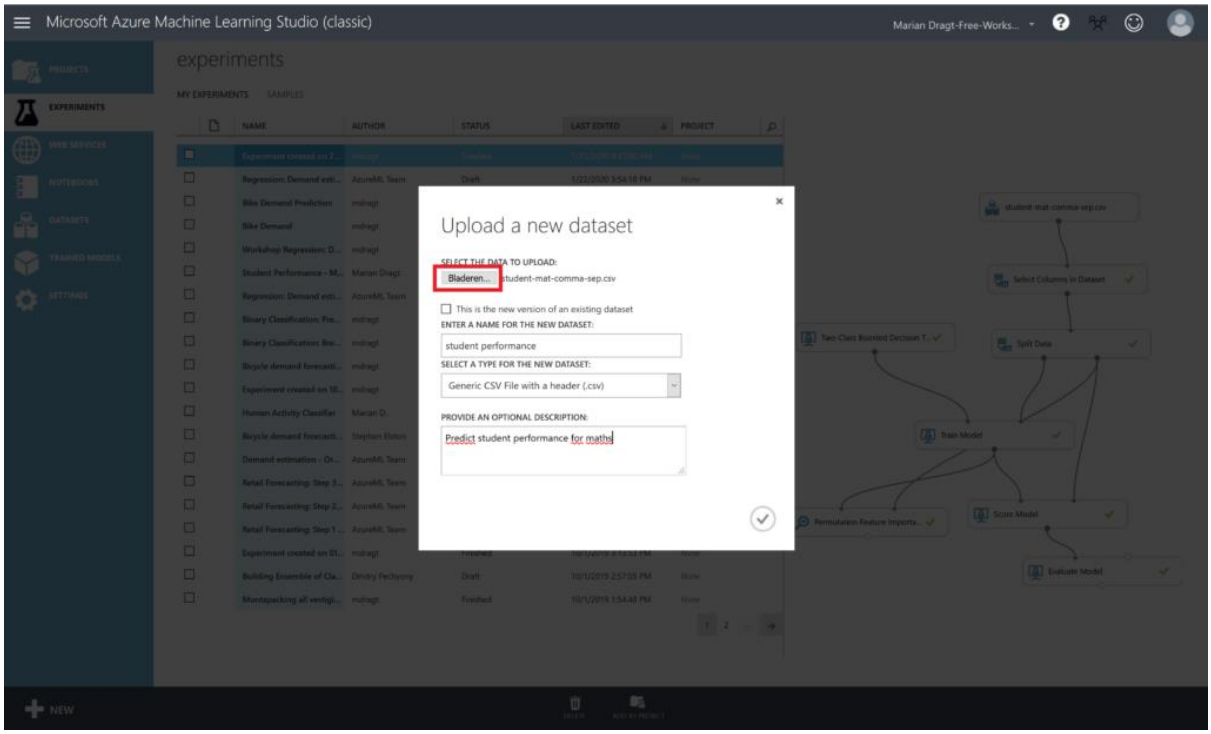
Download the dataset and store it on your local device. Next, you can upload it by creating a new dataset in your Azure Machine Learning Studio environment. First, click on the + NEW button at the bottom menu.



Next, select DATASET and FROM LOCAL FILE.

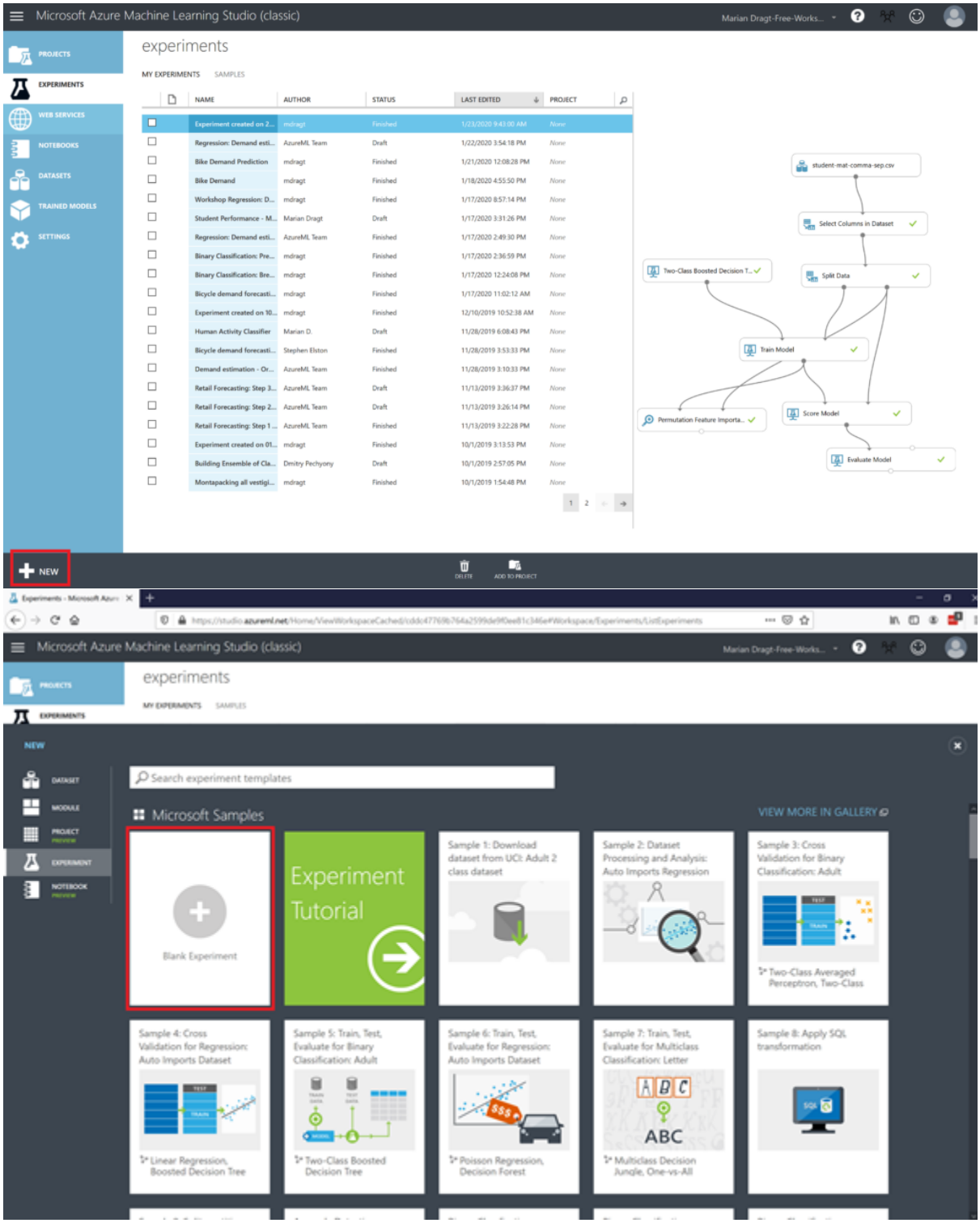


Select your locally stored file and give it a name and description.

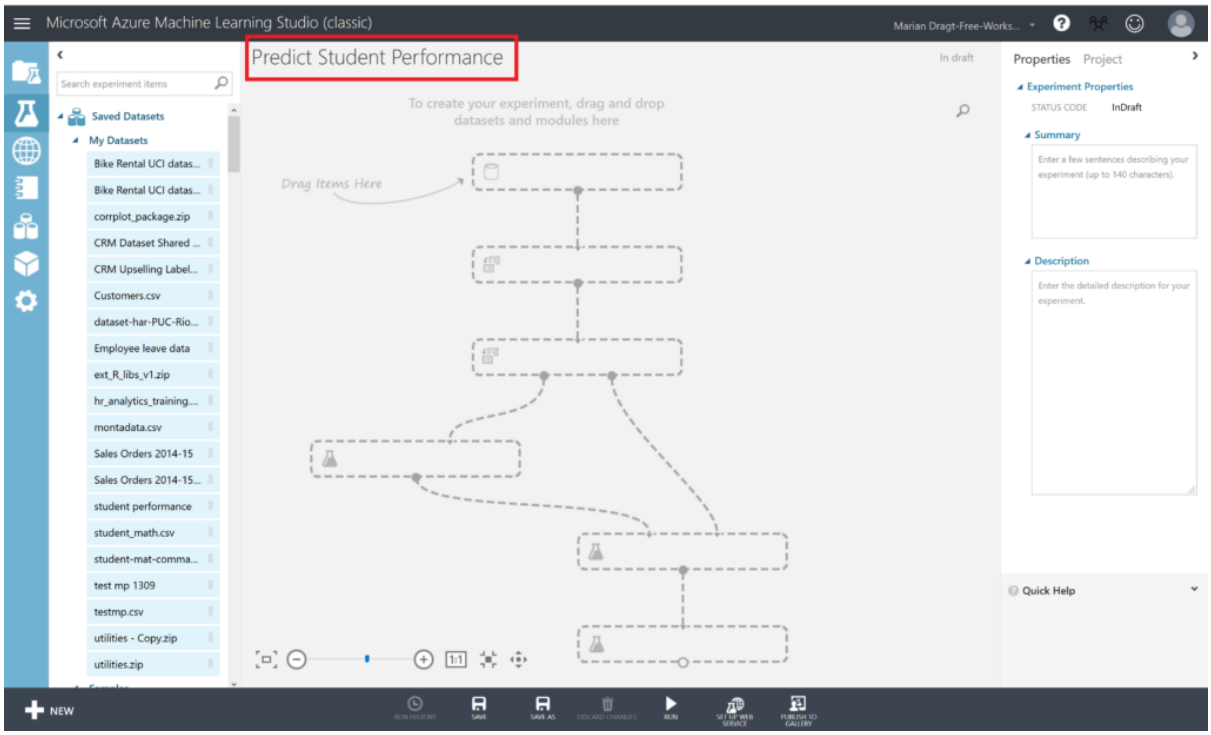


Step 3: Create a new blank Experiment and give it a name

To build your model, you first have to create a new experiment. An experiment is like an instance of your model. It will open a canvas where you can drag your modules on to build your model and run it. Create a new blank experiment by clicking on the + NEW button at the left bottom corner of the screen.



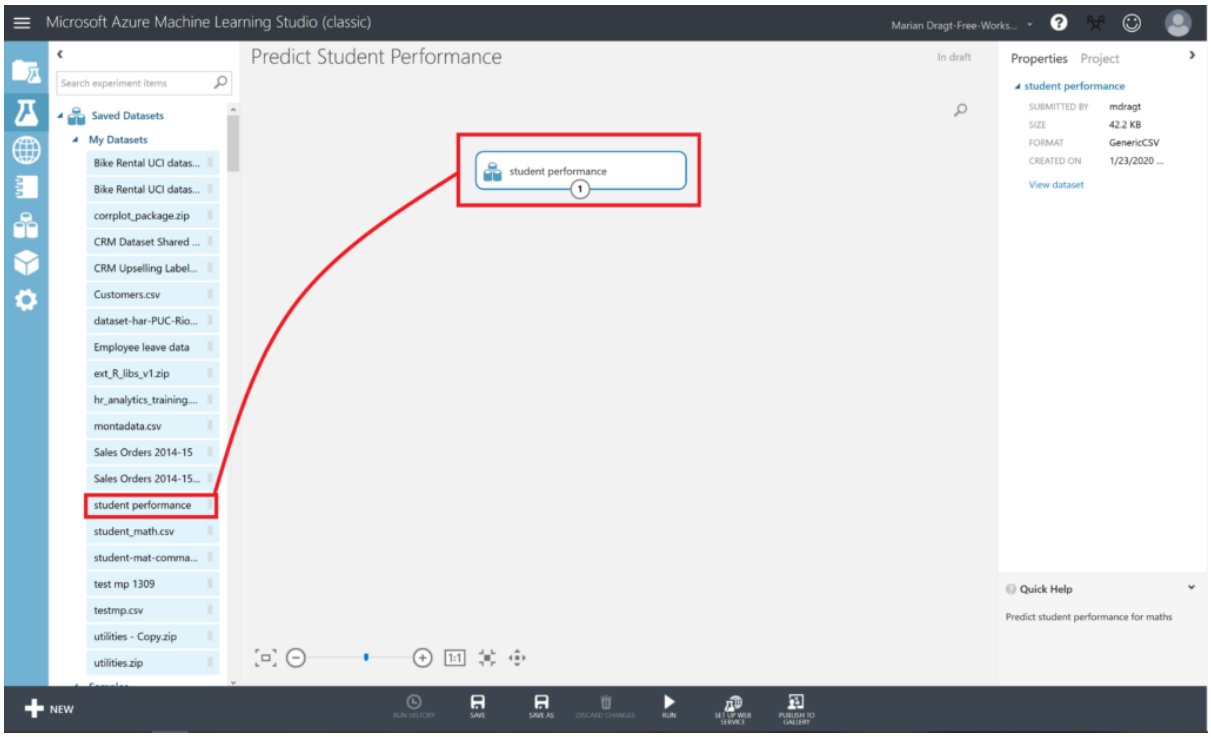
This will open a canvas where you can build your model. First give your model a name. You can select the title and change it.



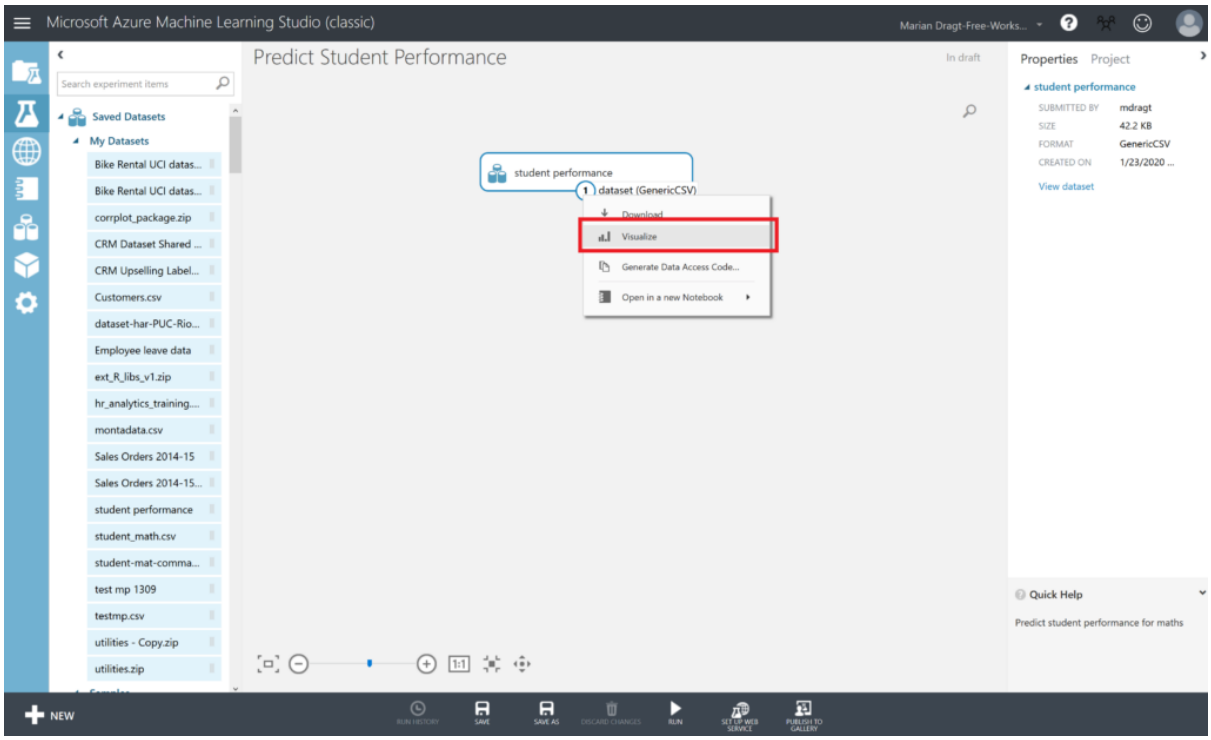
At the left, you have a menu will all kind of modules to build your model with.

Step 4: Select and visualize the data

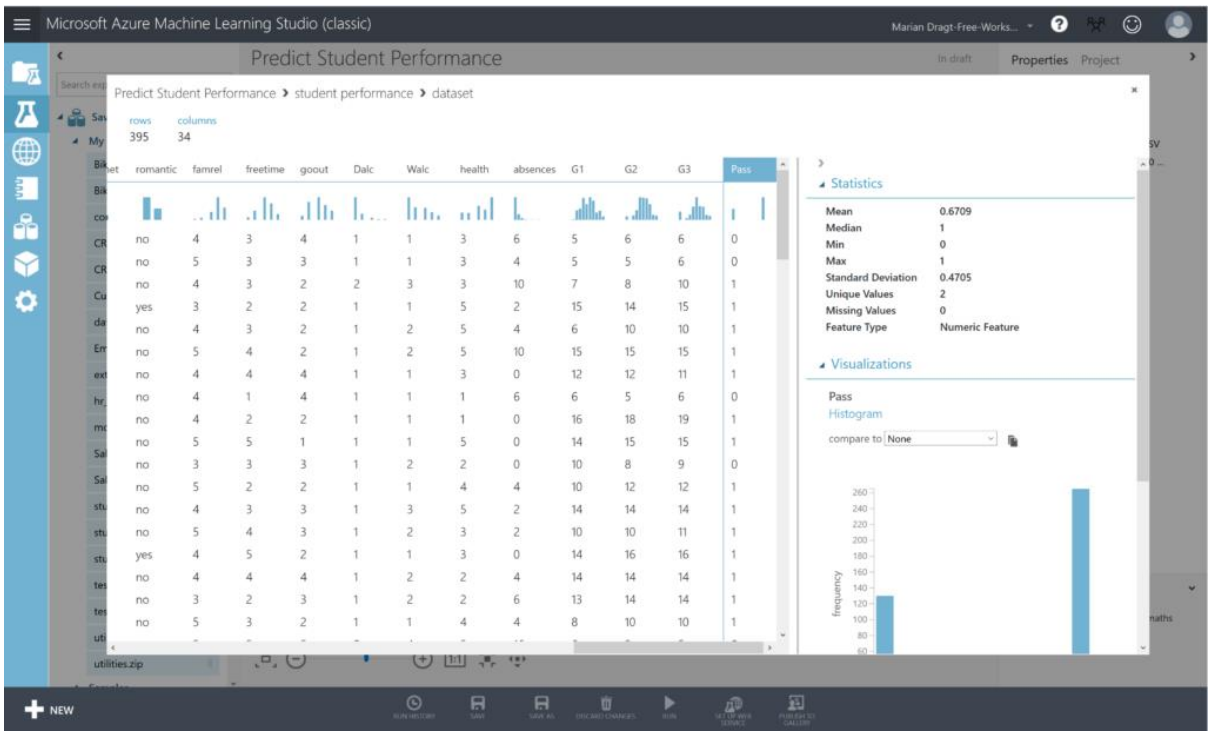
We start with selecting the dataset. You can open the My Datasets item, select the Student Performance dataset, and drag it on the canvas.



To get a quick overview of the data, you can right-click the output port and select the option Visualize. This will show you some quick insights regarding the data, like the amount of observations and variables, and the shape of the data.



You get the basic descriptive per variable and the graphical representation of the distribution.

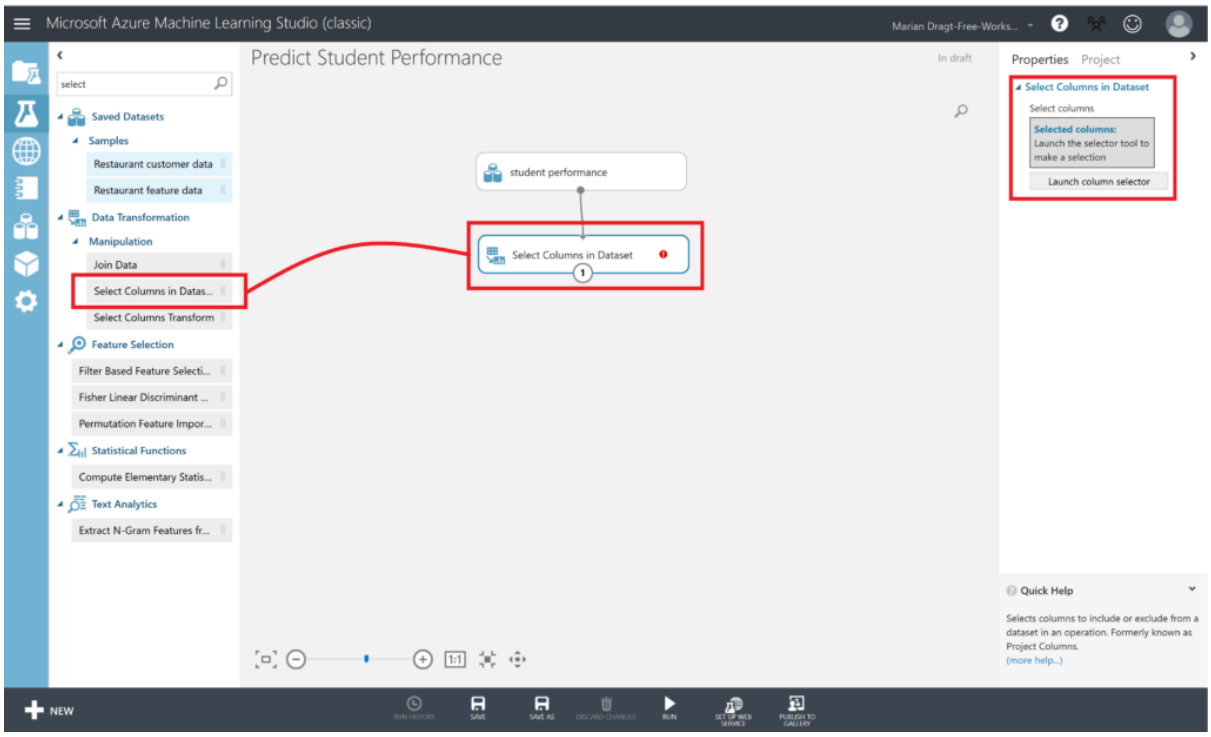


Please check the description of the variables UCI Machine Learning repository page to understand the meaning of these variables. For this model, our dependent variable is called “Pass”. A student will pass if their G3 (final grade) is 10 or higher.

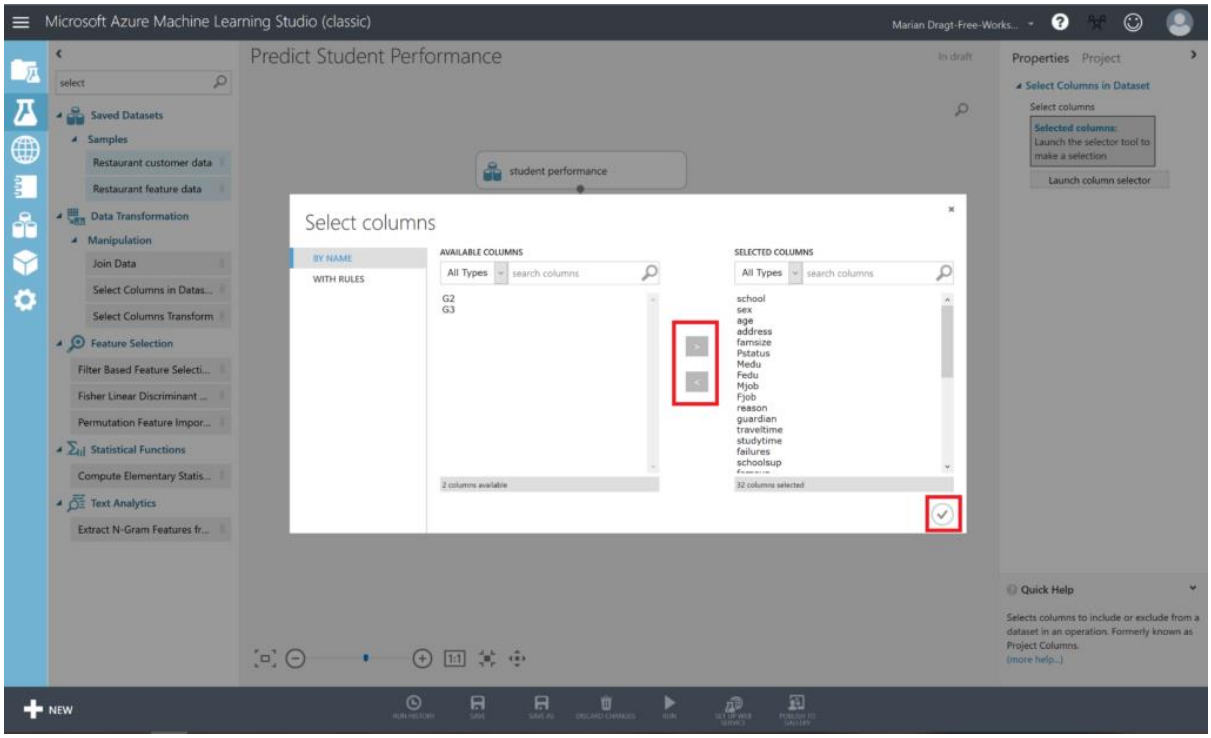
Step 5: Select the required data

To predict if a student will pass, we will use nearly all the variables, but we will exclude the grade from the second semester, as well as the final grade.

To select the required columns, you can use the Select Columns in Dataset module, which you can find under Manipulation in the left menu. You can connect the output port of the Student Performance dataset module with the input port of the Select Columns in Dataset module (use your mouse to draw a line between the modules). You will see a red exclamation mark, because we haven’t informed the module which variables to use. Therefore, you can open the column selector at the right side of the screen.



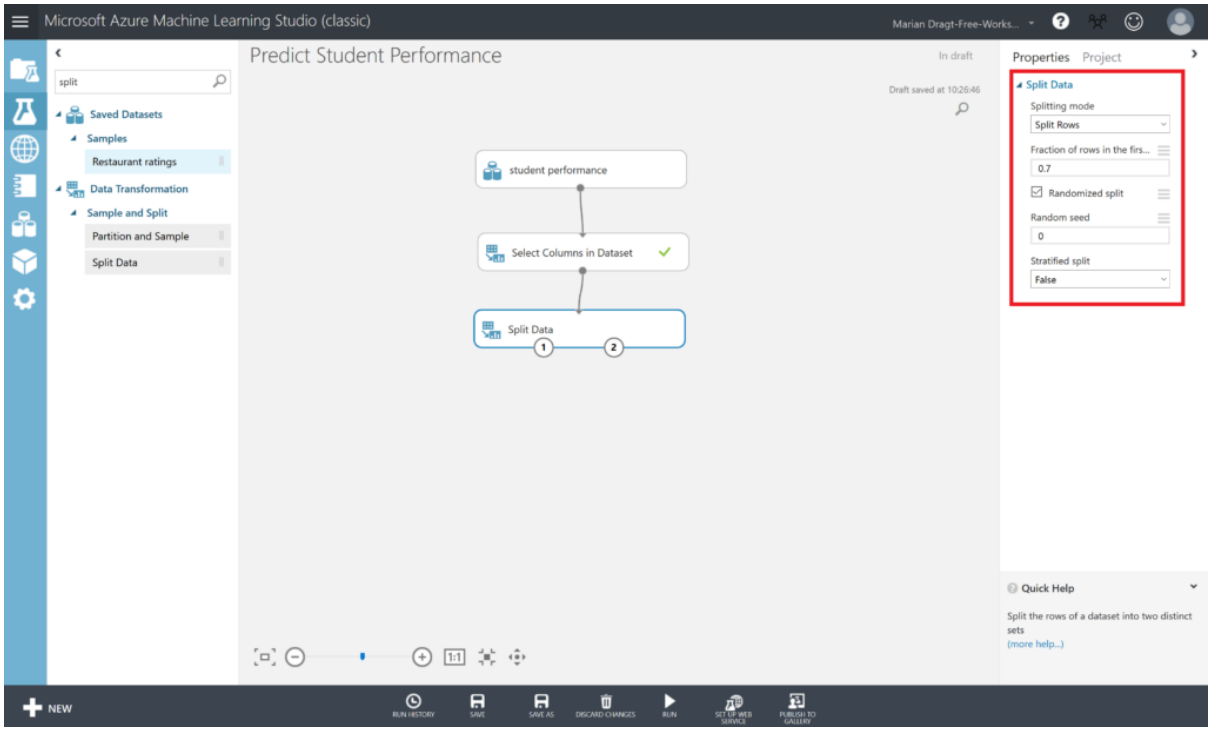
You can now select your desired variables by using the arrows. Make sure all variables except G2 and G3 are in the ‘selected column’ (see picture below). Click on the ok sign right below when you are ready.



In order to see the results, you have to **SAVE** and **RUN** the model. You can find these options at the bottom menu of the page.

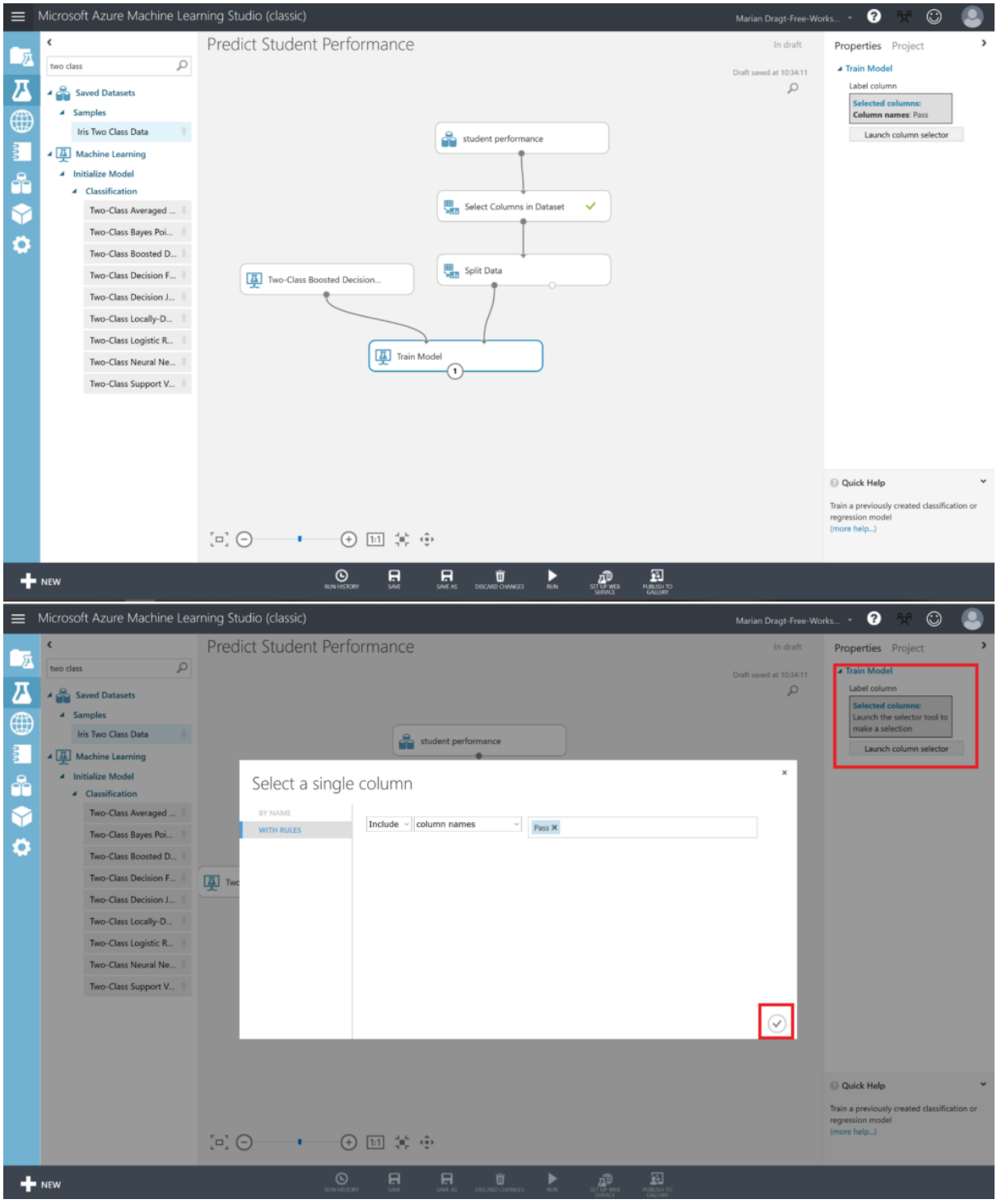
Step 6: Splitting the data

Now you are ready to split the dataset into a training dataset and a test dataset. You will train the model with 70% of the data and test the model with the remaining 30%. Select the Split Data module and drag it on the canvas. Connect the output port of the Select Columns in Dataset module to the input port of the Split Data module. At the right, you can configure this module. In this case, we are using the Split Rows splitting mode, and we select 0.7 (70%) as the fraction for our training data. **SAVE** your model and **RUN** this last step. Now you have 70% of your data in your left output port (1), and 30% of your data in your right output port (2).



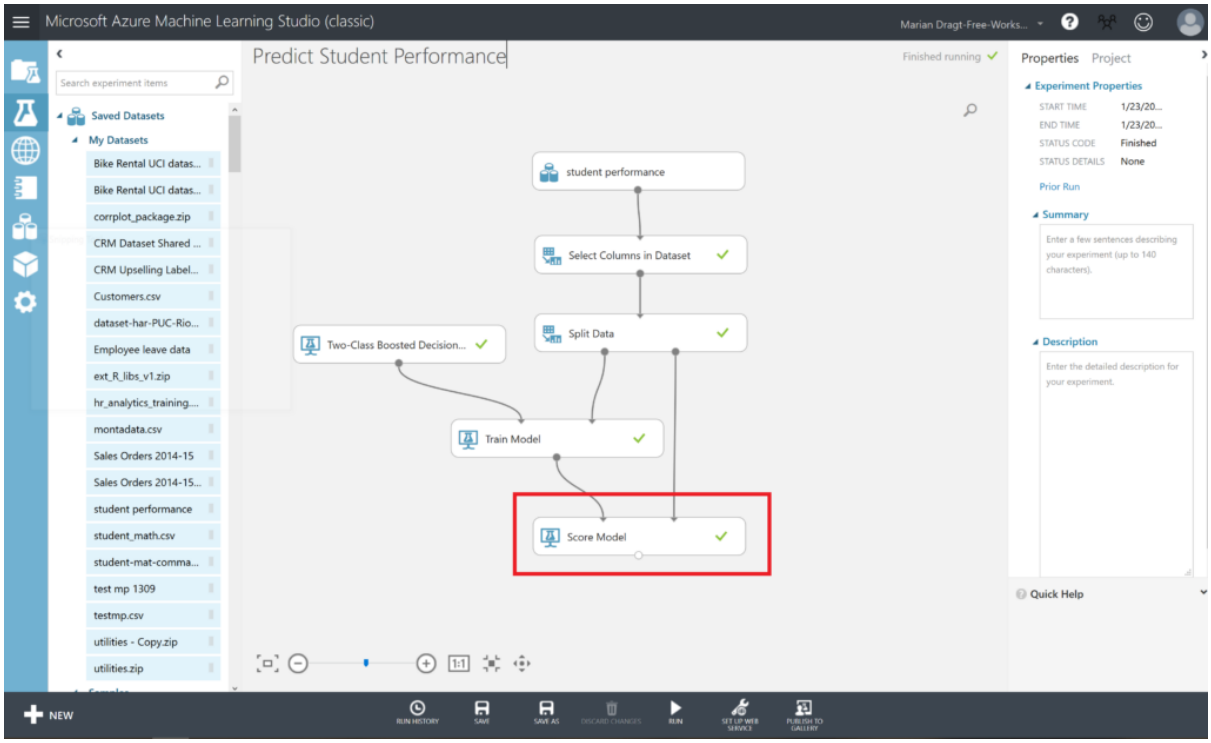
Step 7: Train the model

You are now ready to train the model. You need the **Train Model** module, an algorithm module, and the training dataset. Drag the Train Model module on the canvas and connect the training dataset to it. Besides, drag the **Two-Class Boosted Decision Tree** algorithm on the canvas and connect it to the Train Model module. You can leave the pre-set hyperparameters as they are. In the Train Model module, make sure you select the dependent variable “Pass” to train the model on. **SAVE** and **RUN** your model.

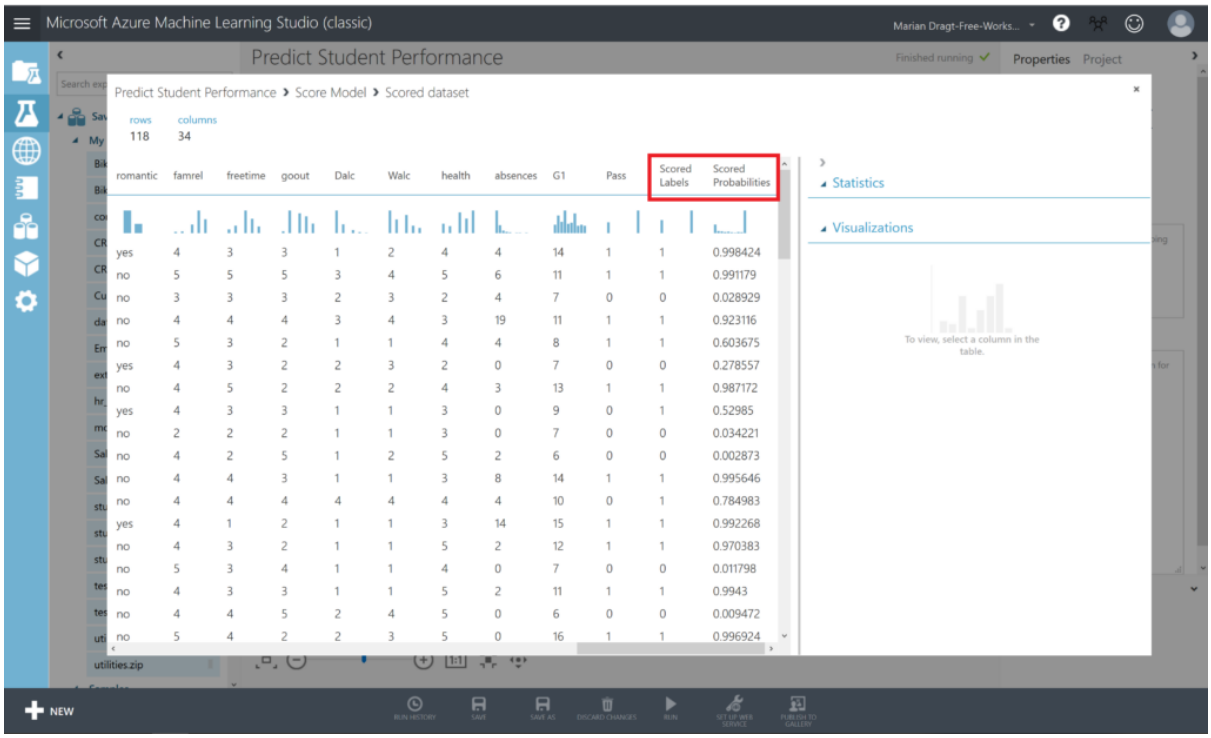


Step 8: Test the model

Now you have trained the model, and it's time to test is. You can use your model to score the test dataset by dragging the **Score Model** module on the canvas and connecting it to the test dataset. By default, the results will be appended to the dataset. **SAVE** and **RUN** your model.

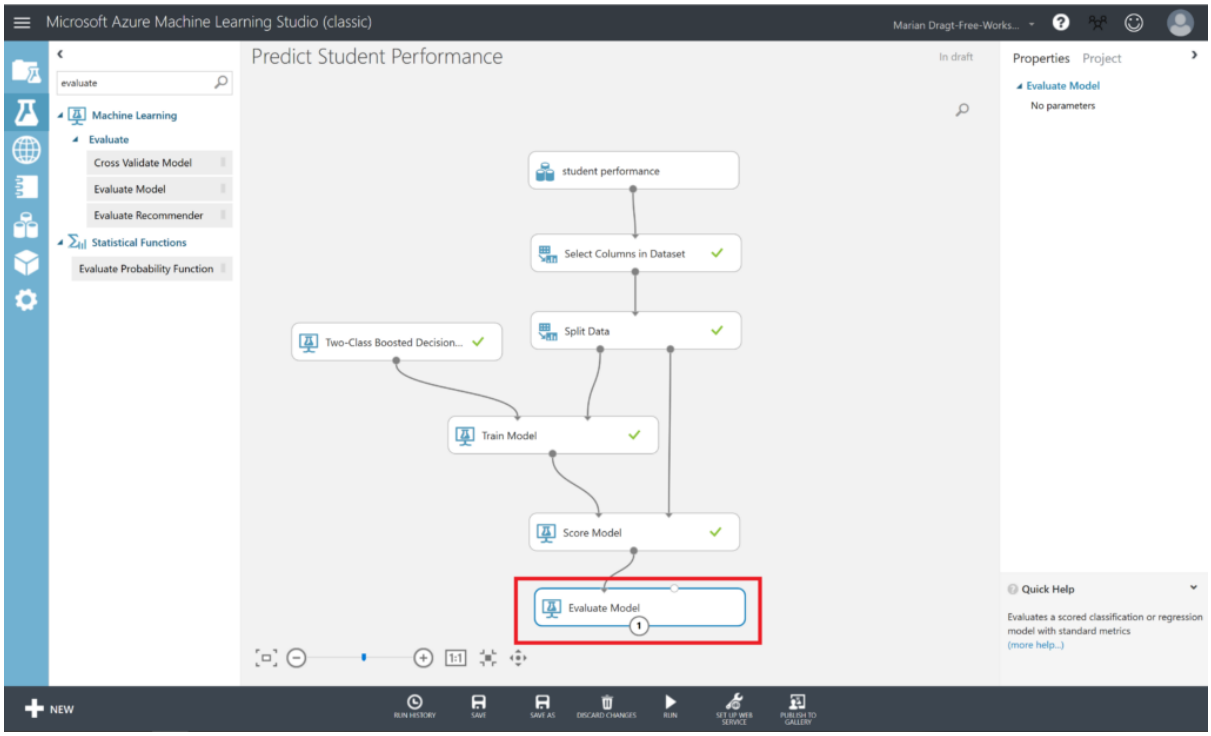


If you inspect the output of the Score Model module, by righ-clicking on the output port and selecting Visualize, you will see that there are 2 extra column in your dataset, named Scored Labels and Scored Probabilities. Scored Lables contains the prediction whether a student will pass or not, and is based on the Scored Probabilities value, where 0.5 is the cut-off: from 0.5 a student will pass.

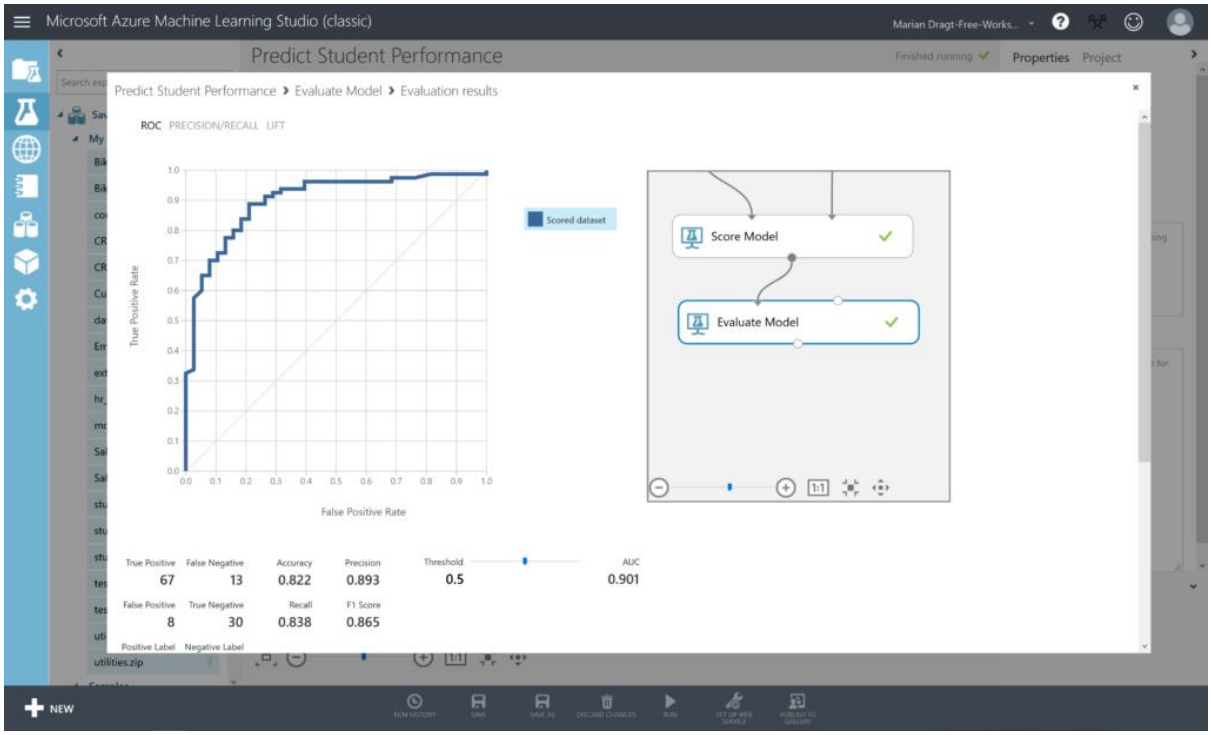


Step 9: Evaluate the model

As we also have the real scores of the students, we can make the evaluation of the model. Luckily for us, there is a module that does the trick. Drag the **Evaluate Model** module on the canvas, connect it, and run your model. The Evaluate Model module has 2 input ports so you can compare models with each other. As we have only one model, make sure you connect it to the left input port.



After you have ran the model, you can inspect the results by right-clicking on the output port and choosing the Visualize options.



With this model, you are 83% accurate in predicting whether a student will pass of fail. Is that good enough? Well, that depends….

Step 10: Be sharp!

Although this model seems to be quite good, we have to be critical. Looking at the collected data, we miss information about at what moment of time during the year the data has been collected. If you want to do this prediction i.e. at mid term, you would also need the variables at that moment of time. An example is the variable “absences”: is this the total number of absences during the complete year?

We hope you have enjoyed this workshop and hopefully it inspired you to build your own models. If you want to take your models into production, then please use another environment: https://ml.azure.com/

Here you can find a similar interface, called Designer, but with this interface, you can also deploy and manage your models.