Lab: Learning to Use AI Studio for Deep Learning

In this lab, you will learn the basics of AI Studio (previously known as RapidMiner), including such things as loading and preprocessing the data, how to build and run a model, such as a logistic regression, a deep learning or a pre-trained generative model, in AI Studio. In particular, we will write a simple recommender system that takes multi-dimensional ratings of restaurants taken from TripAdvisor as the input, such as the restaurant value, atmosphere, service and food quality, and predicts the overall ratings of the restaurant as the output. As the first step, we will learn how to load and preprocess the data.

Part 1: Loading and Preprocessing Data

To start this assignment, download the dataset at trip_advisor.xlsx and open AI Studio. Then, go to the "File → Import Data" option in the upper-left-side menu, click on the "Import Data" button and import the TripAdvisor dataset into the Local Repository of AI Studio. While doing so, make sure that all columns are in the integer format, except for 'review_content' and 'rest_price', which are in the polynominal format and 'rest_rating' which is in real format ("rest" stands for "restaurant" here). Note that if spreadsheet columns are imported incorrectly into AI Studio, convert them to the correct formats by clicking on the problematic column name, selecting the "Change Type" option and clicking on the correct format.

After that, select the Design view panel and go to the "Operators" panel at the bottom left corner (and in particular to the search option there) and search for the "Retrieve" operator. After it is found, drag the Retrieve operator into the process panel and drop it in the left side of the window. Load the TripAdvisor data by setting the correct path from the Local Repository in the "Parameters" panel on the top right by clicking on the browser icon in the "Repository Entry" item and selecting the appropriate dataset ("trip_advisor" that you have just imported) from the local data repository in the "Repository Browser" window.

In the "Operator" panel at bottom left, search for the "SubProcess" operator and drag it into the process panel next to the "Retrieve" operator. Link the subprocess operator with the retrieve operator, as shown in Figure 2. Double click on the subprocess operator. As a result, a blank process panel will open on the screen. Next, search and drag the "Sample", "Select Attributes", "Numerical to Binominal", "Set Role" and "Split Data" operators into the panel and link them sequentially, as shown in Figure 1. While connecting these operators, link the "in" connector on the left edge of the panel to the "Sample" operator and link the "out" connector on the right edge of the panel to the "Spit data" operator (note that you should link the two "par" connectors in "Split Data" operator to the two

"out" connectors, as shown in Figure 1). Furthermore, set the parameters for these operators (in the Parameters panels of the corresponding operators) as follows:

Sample: sample size = 1000;

Select Attributes: set attribute "filter type" to "a subset", click on the "Select Attributes" button and select attributes rest_nb_review, rest_rating, rest_rating_neutral, rest_rating_poor, rest_rating_terrible, rest_rating_very_good, review_rating, review_rating_atmosphere, review_rating_food, review_rating_service, review_rating_value by moving them from the left to the right window.

Numerical to Binominal: set the attribute filter type to "subset", click on the "Select Attributes" button, select the "review_rating" attribute (and only it) and binarize it by setting min=0.0 and max=3.5. As a result of this step, we binarize the review rating from the 1 – 5 scale to the binary scale (like/dislike or high/low).

Set Role: enter the attribute name "review_rating" and set its target role as "label."

Split Data: Click on the "Edit Enumerations" button in the "partitions" row of the "Parameters" panel, click on "Add Entry" button and set the ratio to 0.8. Then click on "Add Entry" button again and set the second ratio to 0.2, thus creating the 80% - 20% training/testing split of the data (as a result, you should see 2 rows: the first having 0.8 and the second 0.2 in it).

Finally, click on the "Process" entry in the upper-left corner of the Process panel to return to the main panel.

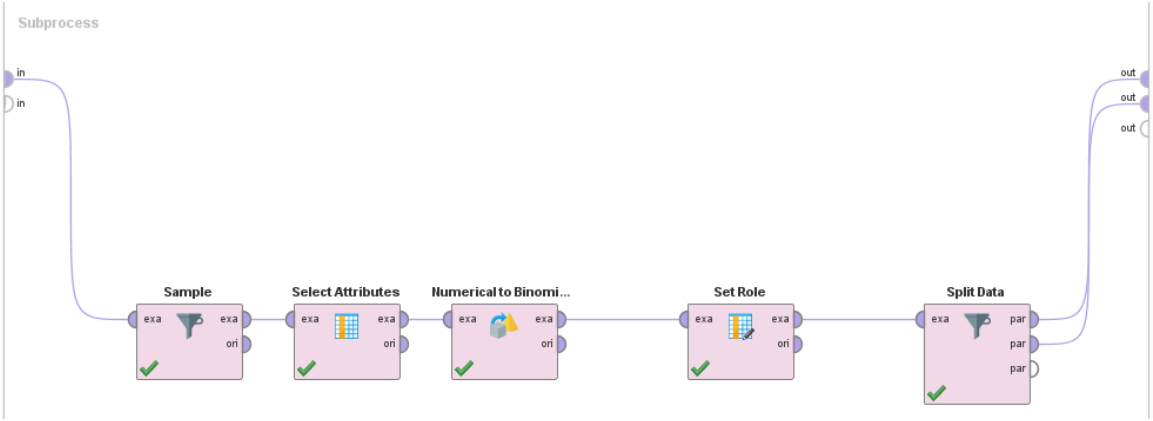

Figure 1. The flowchart of the Subprocess Operator.

Part 2: Logistic Regression

To build the Logistic Regression model for binary classification of the "overall" attribute, follow the process presented in Figure 2. We have already put and connected the "Retrieve" and "Subprocess" operators in the panel, as described in Part 1. Next, search for the "Logistic Regression" operator in the "Operator" panel at the bottom left and drag it into the process panel. Then link the "Logistic regression" operator to the "Subprocess" operator built in Part 1, as shown in Figure 2.

In the "Operator" panel at the bottom left, search for the "Apply Model" operator and drag it to the process panel. Next, link "Apply model" to the "Subprocess" operator (by connecting "out" and "unl") and to the "Logistic regression" operator (by connecting "mod" and "mod"), as shown in Figure 2.

In the "Operator" panel at the bottom left, search for the "Performance (Binominal Classification)" operator and drag it into the process panel. Then link the "Performance" operator with "Apply model" (by connecting the "lab" and "lab" connectors). Moreover, select "accuracy, classification error, AUC" measures in the "Parameter" panel.

Finally, link the "per" output in the "Performance" operator to the "res" connector on the right edge of the panel and click the "Run" button (the blue triangle button) to see the results.

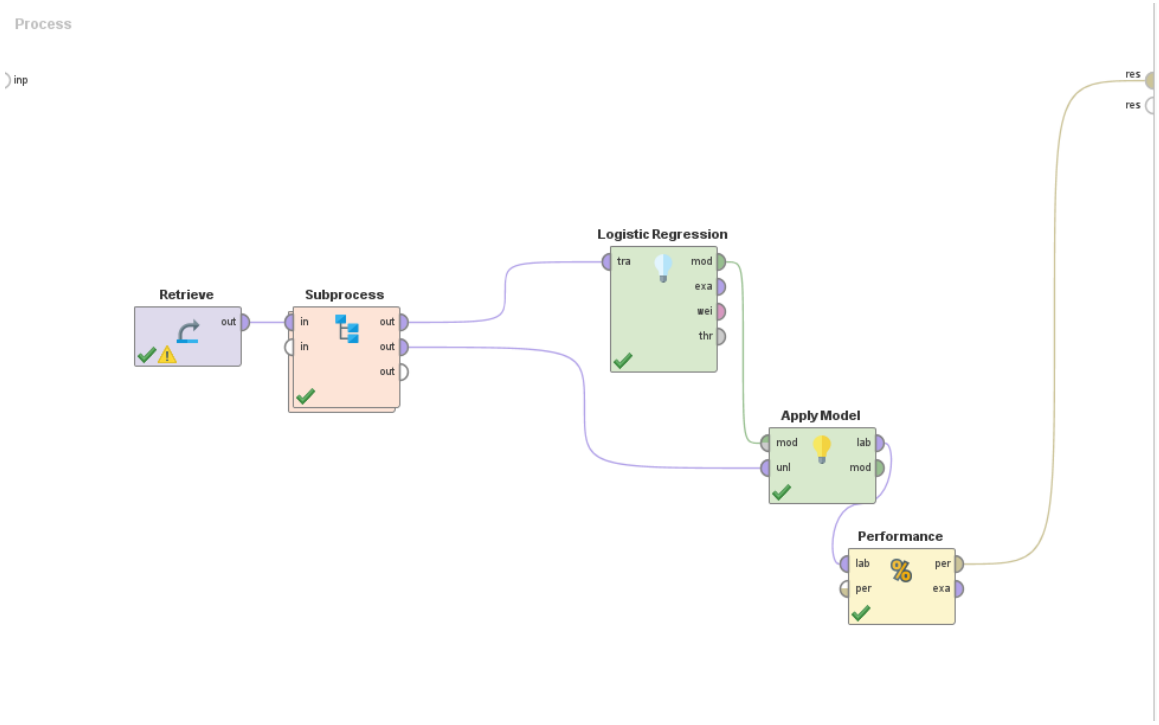

Figure 2. The flowchart of the process of building a Logistic Regression model.

Part 3: Deep Learning

The flowchart of the process of building a Deep Learning model is presented in Figure 3. Note that for Deep Learning, the "Retrieve" process is the same as the one presented in Part 1. The "SubProcess" operator, however, is somewhat different than the one shown in Figure 2. In particular, we only need the "Select Attributes", "Set Role" and "Split Data" operators this time. You should link these three operators, as shown in Figure 4 and set the parameters accordingly.

In the "Operator" panel at the bottom left, search for the "Deep Learning" operator (note that it is located on the "Extensions/Deep Learning/Modeling/Deep Learning" path!) and drag the operator into the process panel. Link the "Deep Learning" operator with the "Subprocess" operator, as shown in Figure 3.

Double click on the "Deep Learning" operator and add two fully-connected layer operators, followed by the output layer, as shown in Figure 5 (search for the "Add Fully-Connected Layer" and the "Add Output Layer" operators in the Operator panel and drag them to the Process panel). Then link all of them, as shown in Figure 5.

Next, specify the hyperparameters of the two Fully Connected layers by selecting Number_of_neurons = 32 and Activation_Function = "ReLU" in the Parameters panel. For the Output Layer, select "output type" as FullyConnected, "loss function" as "Mean Squared Error," and "activation function" as ReLU in the Parameters panel. Then go back to the main panel by clicking on the "Process" entry in the Process panel and specify the hyperparameters of the "Deep Learning" operator in the Parameters panel as *epochs = 50, updater = "Adam," learning rate = 1.0 E-6* and *weight initialization = "Normal"* (note that we discussed some of these parameters of the NN before in class).

After finishing the Deep Learning setup, search for "Apply Model" in the "Operator" panel and drag the "Apply Model" operator into the process panel. Then link the three operators ("SubProcess", "Deep Learning" and "Apply Model"), as shown in Figure 3. Next, in the "Operator" panel, search for the "Performance (Regression)" operator and drag it into the process panel. Link the "Performance" operator with the "Apply Model" operator and the right end of the panel, as shown in Figure 3. Select "root mean squared error" and "absolute error" measures in the "Parameters" panel of the Performance operator.

Finally, click the "Run" button to run the deep learning model. Please, record the running time of your model and the performance measures that it produced by clicking on the Results tab located next to the Design tab on the upper part of the screen.

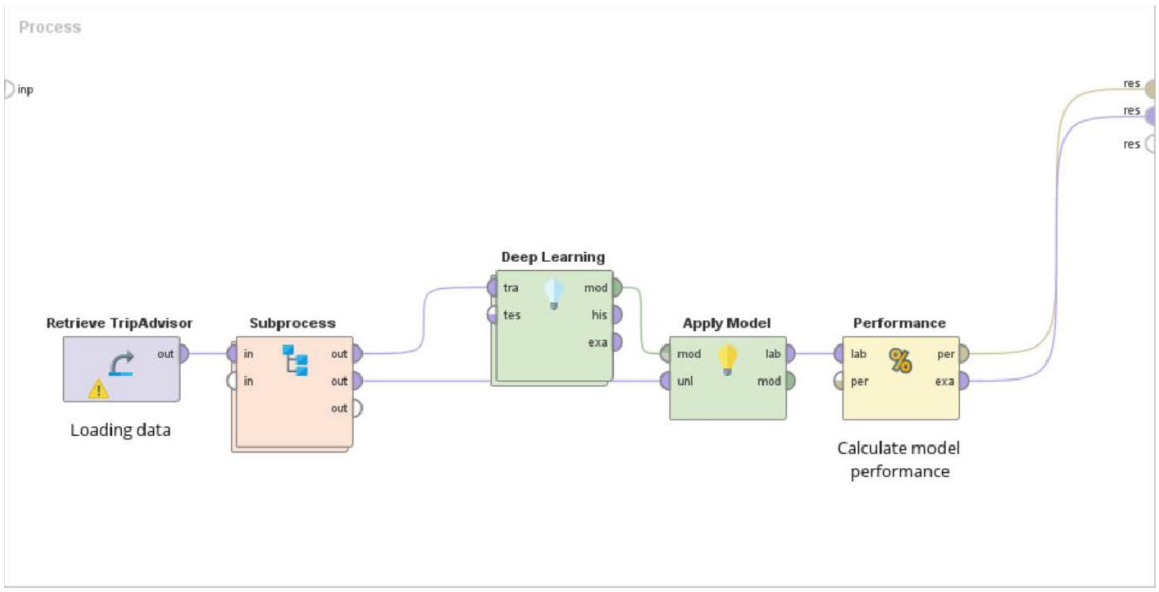

Figure 3. The flowchart of the process of building a Deep Learning model.

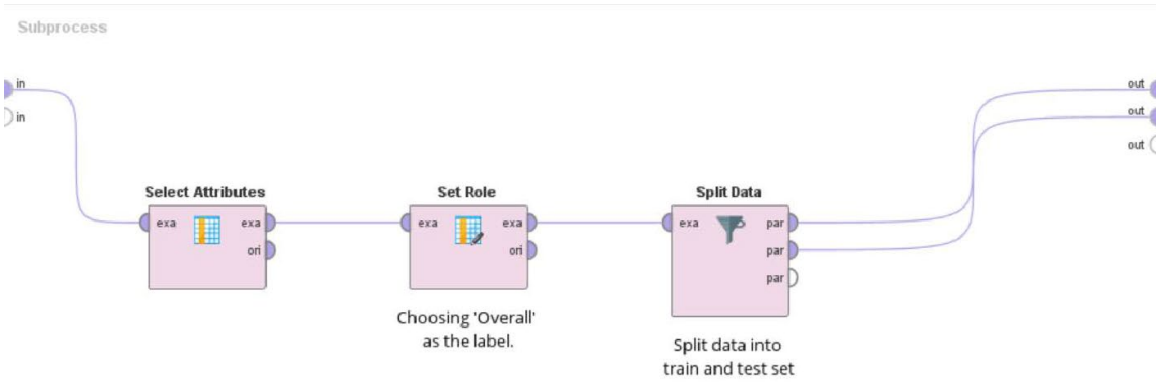

Figure 4. The flowchart of the Subprocess Operator for the Deep Learning Model.

Figure 5. The flowchart of the Deep Learning Operator.

Part 4: Generative Models

The flowchart of the process of building a Generative model is presented in Figure 6. Note that for the Generative model, the "Retrieve" process is the same as the one presented in Part 1. The "SubProcess" operator, however, is somewhat different than the one shown in Figure 2. In particular, we only need the "Sample" operators this time. You should link the operator, as shown in Figure 7, and set the parameters as:

Sample: absolute; sample size = 100;

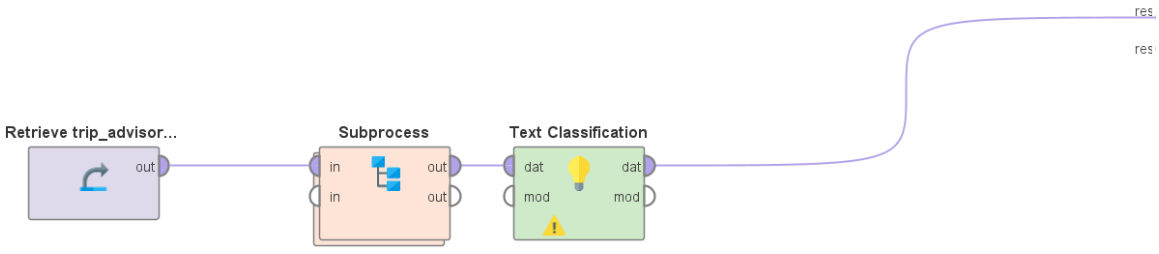

Figure 6. The flowchart of the process of building a Generative model.

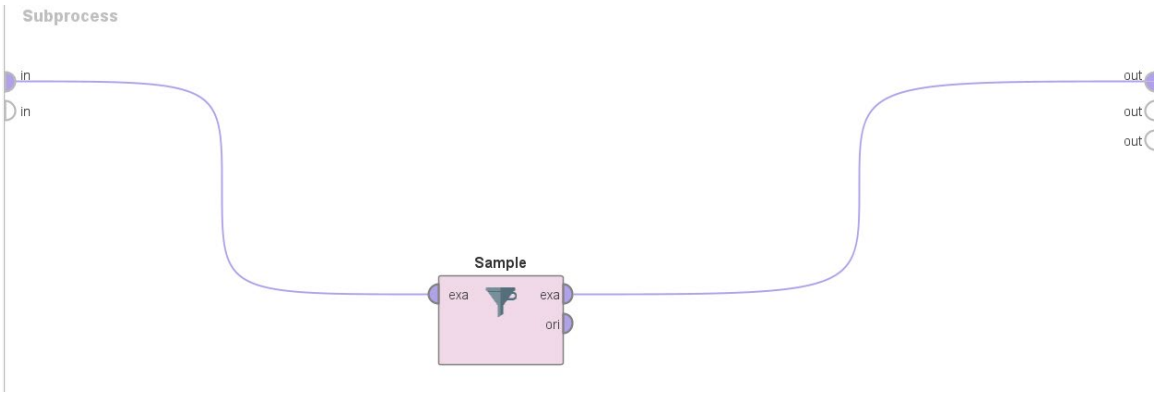

Figure 7. The flowchart of the Subprocess Operator for the Generative Model.

In the “Operator” panel at the bottom left, search for the “Text Classification” operator (note that it is located on the "Extensions/GenerativeModels/Models/Huggingface/Tasks" path!) and drag the operator into the process panel. Link the “Text Classification” operator with the “Subprocess” operator, as shown in Figure 6.

Next, specify the hyperparameters of the “Text Classification” operator in the Parameters panel as model = nlptown/bert-base-multilingual-uncased-sentiment (note that this is the name of a pretrained sentiment analysis model from the Hugging Face platform), name = pred_rating (this is the name of the predictive variable – in the “pred_rating” column of the dataset), device= CPU/GPU (depending on whether your laptop contains GPU or not). For the prompt, click on “Edit Text” and type [[review_content]] to specify the text input column (we will be using the reviews located in the “review_content” column of the dataset as prompts for the aforementioned Hugging Face sentiment analysis model to estimate the rating of a review) and click on the “Apply” button.

Finally, click the “Run” button to run the aforementioned pretrained sentiment analysis model. Please, record the running time of your model and the performance measures that it produced by clicking on the Results tab located next to the Design tab on the upper part of the screen. Your predicted rating will be shown in column “pred_rating”, as indicated with the parameter “name” in the previous step.

Note that the predicted restaurant ratings have values of “1 star”, “2 stars”, ..., “5 stars”, instead of the numeric ratings of 1 to 5 since the pre-trained model “nlptown/bert-base-multilingual-uncased-sentiment” was trained for a similar sentiment analysis task on a different dataset with the target label having text-based values of “1 star”, “2 stars”, ... “5 stars”, which leads to the predictions that are incompatible with our dataset. Hence, we cannot do the direct comparison of our models with this Hugging Face model [note that we will discuss the Hugging Face platform and its generative models later in the course].

Congratulations on completing this Lab!