Data cleaning with Azure ML

Big Scale Analytic - March 2020

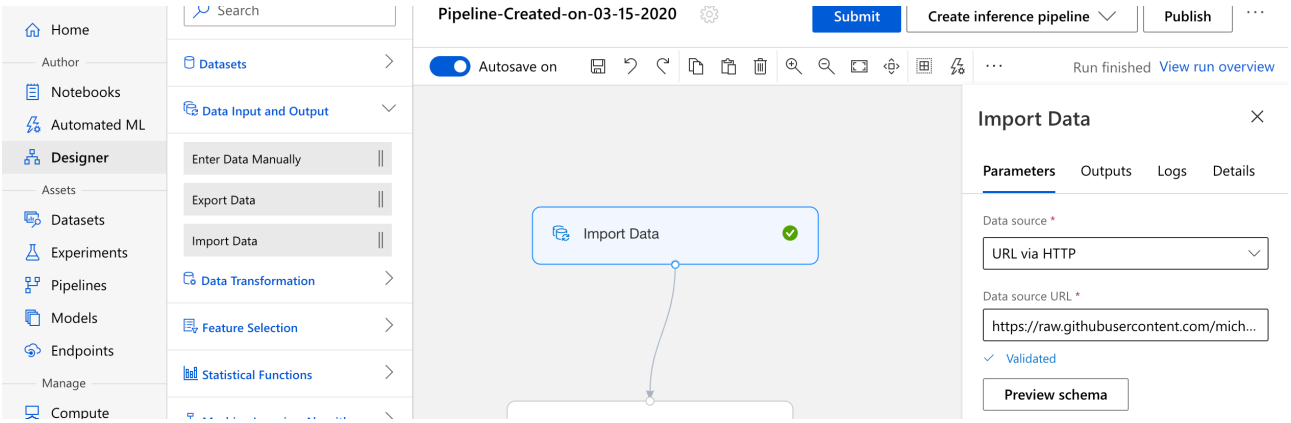
Azure ML

Azure Machine Learning is a cloud based visual tool to deploy, train, automate and track ML models. It has a visual drag and drop user interface in which you can create the whole ML pipeline from data cleaning and pre-processing to splitting data to train and test sets and training a model. It can be used for any kind of machine learning, from classical ml to deep learning, supervised, and unsupervised learning. Moreover it is not just a visual designer tool, in case you prefer to write R or Python code to create your ML pipeline, you can do it easily within Azure ML. More information on Azure ML could be found in this link:
https://docs.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-ml

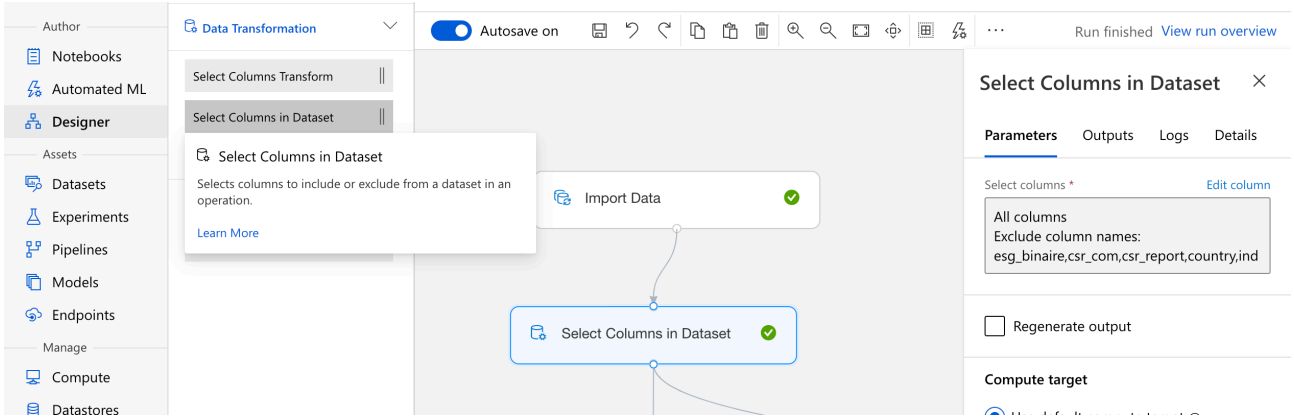
Data cleaning in Azure ML

An important step in any ML task is to clean and pre-process the data since many ML models cannot treat missing values in the data and unnormalised data properly. In this week we are going to use the visual tool of Azure ML to do data cleaning. We will use several different "Designer" tools which help us to do clean the data and also normalise numerical features if needed. You can find a brief description of some of these tools below. During the Lab we will do a walk through together to explore these tools more in details.

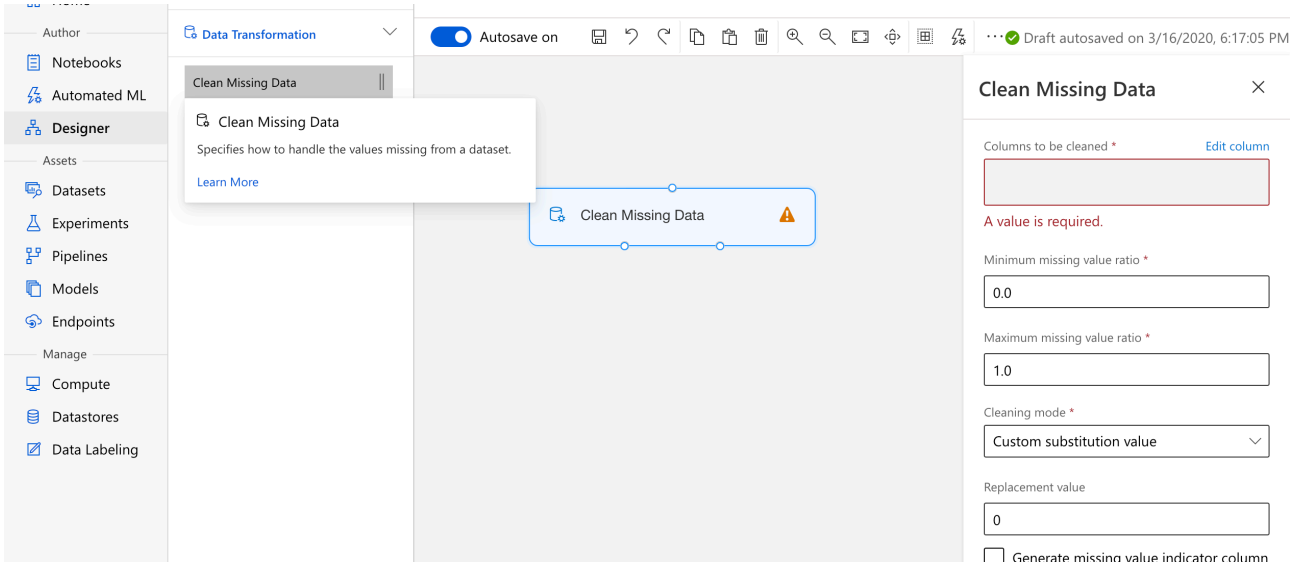
- **Import data:** The first step in making the pipeline is to import the data. You can find this tool by searching its name in the search bar. The data could be loaded from a URL or cloud storages in Azure. Once the data is loaded you can check the schema by clicking on the "preview schema" button and check the columns and their data types (and edit them if necessary)



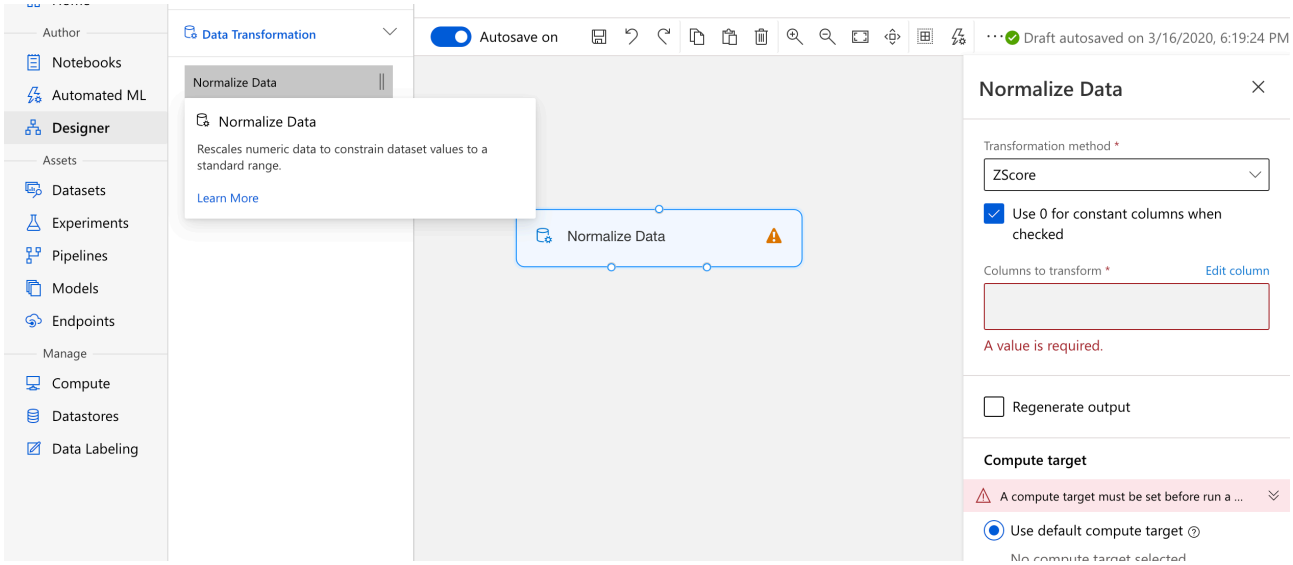
- **Select Column in Dataset:** This could be used as a feature selection step, i.e. if you want to select only a subset of the columns or in other words remove some of the columns.



- **Clean missing data:** With this tool you can deal with missing values in your data. It gives you several options to replace the missing values by median, mean etc. Also it is possible to remove any row containing missing values or remove a whole column that has many missing values.



- **Normalise data:** This tool is useful to normalise the features. There are different normalisation methods available, you can refer to the documentation to see what each of these methods does. For example the “ZScore” option which could bee see in the screen-shot below does the usual standardisation by removing the mean and dividing by the standard deviation.



- **Summarise data:** This tool gives you some statistics about each feature, eg how many unique values, what is the maximum, minimum or mean of a feature, etc. You can use this tool in the beginning of your pipeline to get some information about your data set, for instance to see which features have more missing values.
- **Export data:** By exporting the data, you can save your resulting data sets in Azure datastore. Later you can download your results to your computer or use them in another project in Azure ML.

In class assignment

In this assignment we are going to clean and preprocess ESG ratings dataset. You can find this data set under “week5/data” in Github. This dataset contains European companies included in Thompson Reuters Assets list between 2013 and 2018. European companies are chosen since

the continent contains the most different countries with available scores. The sample is delimited from 2013 to 2018 in order to only consider recent data. Only companies with an ESG Controversies Scores (ESGCS) available from at least one of the last six years analysed are kept, giving a final sample of 1096 companies.
More information ESG ratings can be found here.

In Azure ML create a new pipeline and follow these steps:

1. Import the data using the URL from Github
2. Check for the data types in the schema and correct them if necessary
3. Drop the following columns: “esg_binaire”, “csr_com”, “csr_report” and “ind”
4. Drop the columns which have more than a 1000 missing values
5. Drop the rows which contain at least one NaN value
6. Normalise the numerical columns to be between 0 and 1 (except for “esg_score” and “Year”)
7. Export the result and save it in the datastore.

Compare the number of rows and columns in the beginning and after applying the above steps.