

ZERO SHOT

With reference to Atomic Data Services from the RAG system as Context, please suggest suitable services to address each of the steps below.

Given

Datasource 1: CSV file with 31 columns, including a row ID, timestamp, 28 temperature values, and X axis displacement values from a sensorized spindle on a working center machine.

I need services to perform the following steps.

STEP 1: A Collect service for reading the content of Datasource 1 by excluding the row ID column.

Based on the provided context information, there is a service called "CSV data retrieval" which can be used for step 1. This service has a path `"/read-csv"` with a get request that allows for reading the content of a CSV file while excluding specific columns by using the query parameter `"projection_attributes"`. In this case, we would set `"projection_attributes"` to exclude the row ID column.

STEP 2: A Dispatch service for selecting a subset of the 28 temperature features on the dataset obtained from STEP 1 by applying Factor Analysis. Produce as output only the columns regarding the temperature features selected by the algorithm and displacements.

Based on the context provided, the "Factor Analysis Service" described in the JSON file ``/content/data/factor_analysis.json`` can be used for STEP 2. This service has a POST request at the `"/execute_analysis"` endpoint that accepts a Pandas DataFrame and allows for specifying the variables upon which the Factor Analysis has to be performed.

STEP 3: A Monitor service for building a multiple linear regression model on the dataset obtained from STEP 2. Consider the temperature features selected in STEP 2 as input for the model and the displacements feature as target. Produce as output only the two columns regarding the real and the predicted displacements.

Based on the context provided, the "Multiple Linear Regression Service" described in the first JSON file (``/content/data/mlra_algorithm.json``) can be used for STEP 3. This service has a POST request at the `"/multiple_linear_regression"` endpoint that accepts a Pandas DataFrame and allows for specifying the independent variables (temperature features in this case) and dependent variable (displacements feature).

Since the CSV data from STEP 2 is already available as a Pandas DataFrame, you can use it directly to make the API call. The required fields are "dataframe", "independent_variables" (the temperature features), and "dependent_variable" (the displacements feature). In addition, since we only want the output columns for real and predicted displacements, set the "projection_attributes" parameter to include these two columns.

STEP 4: A Monitor service for comparing the real and predicted displacements on the dataset obtained from STEP 3.

Based on the context provided, the "Model vs. Real Data Comparison Service" described in the first JSON file (`/content/data/comparison_output_algorithm_with_real_data.json`) can be used for STEP 4. This service has a POST request at the `/compare_model_real_data` endpoint that accepts the real and predicted displacements as input datasets, which are assumed to have the same structure (i.e., identical columns) since they only contain these two columns.