Vantage Plugins for Dataiku Data Science Studio

Document Version 2.0

Table of Contents

1. Introduction

Dataiku Data Science Studio (DSS) is a collaborative platform that enables teams of people with different data expertise, such as data engineers, data scientists and analysts, to work together efficiently. Dataiku DSS provides a set of built-in recipes or operations that can be applied to transform or analyze a dataset. It also allows users to create their own recipes in Python, SQL or R. The DSS plugins are custom reusable recipes that can only be written in Python.

The present guide outlines installation and usage of 3 DSS plugins that enable you to interact with Teradata Vantage systems; namely, the Vantage SQLE Functions Plugin, the Vantage MLE Functions Plugin, and the Teradata Vantage SCRIPT Table Operator (STO) Plugin.

1.1. Vantage SQLE Functions Plugin

The Vantage SQLE Functions Plugin for Dataiku DSS integrates a set of analytic functions that reside in the Vantage Advanced SQL Engine, by providing a user-friendly, easy-to-use, no-SQL interface for the functions in the Dataiku DSS environment. The Vantage SQLE analytic functions can be accessed through the [`+RECIPE`] menu of the FLOW view of a Dataiku project.

In the background of the Vantage SQLE Functions Plugin user interface, the plugin essentially translates the end-user input from the plugin screens into SQL queries that are sent to the Advanced SQL Engine of a connected Vantage system via JDBC. This way, all analytic queries are executed in-database, while also all input and output managed datasets are physically located in the database of the Advanced SQL Engine on the connected Vantage system.

The plugin versioning is tied to the Vantage Advanced SQL Engine release, since the plugin is an interface to the analytic functions that come with that specific release. In that light, the plugin version `x.y.z-a` is interpreted as follows: `x.y.z` is the Vantage Advanced SQL Engine release the plugin version caters to, and `a` is the plugin release, which is a number that may increase in case of subsequent fix/feature releases. For example, the inaugural plugin version is tied to the Vantage Advanced SQL Engine release 2.0, and, per the previous, the plugin version will be 2.0-1.

1.2. Vantage MLE Functions Plugin

The Vantage MLE Functions Plugin for Dataiku DSS integrates about 180 of the Vantage Machine Learning Engine (MLE) analytic functions, by providing a user-friendly, easy-to-use, no-SQL interface for the functions in the Dataiku DSS environment. The Vantage MLE analytic functions can be accessed through the [`+RECIPE`] menu of the FLOW view of a Dataiku project, and are grouped into the following nine categories:

- Time Series, Path and Attribution Analysis
- Ensemble Methods
- Text Analysis
- Naïve Bayes
- Graph Analysis
- Association Analysis
- Statistical Analysis
- Cluster Analysis
- Data Transformation

In the background of the Vantage MLE Functions Plugin user interface, the plugin essentially translates the end-user input from the plugin screens into SQL queries that are sent to the Advanced SQL Engine of a connected Vantage system via JDBC. In the background, the target Advanced SQL Engine passes on the query to the associated Machine Learning Engine, where the query gets executed. This way, all analytic queries are executed in-database, while also all input and output managed datasets are physically located in the database of the Advanced SQL Engine on the connected Vantage system.

The plugin versioning is tied to the Vantage Machine Learning Engine release, since the plugin is an interface to the analytic functions that come with that specific release. In that light, the plugin version `x.y.z-a` is interpreted as follows: `x.y.z` is the Vantage Machine Learning Engine release the plugin version caters to, and `a` is the plugin release, which is a number that may increase in case of subsequent fix/feature releases. For example, the inaugural plugin version is tied to the Vantage Machine Learning Engine release 1.1, and, per the previous, the plugin version will be 1.1-1.

To execute Vantage Machine Learning Engine (MLE) analytic functions, note that the target Vantage system must include a Machine Learning Engine component; see also Section 2.5.

1.3. Vantage SCRIPT Table Operator Plugin

The Vantage SCRIPT TO Plugin allows the execution of R or Python scripts inside the Advanced SQL Engine Database. The plugin will take an R or Python script within a DSS notebook, or an R or Python script uploaded to the plugin and install the scripts and other related files (i.e. saved models in RDS or pickle files) onto the Advanced SQL Engine.

Similar to the Vantage SQLE and MLE Analytic Functions Plugins, the Teradata Vantage SCRIPT TO Plugin translates the user-requested tasks in the plugin into SQL queries. Queries are then sent in the background to a connected Vantage system to set up and invoke the SCRIPT Table Operator.

To execute R or Python scripts inside the Advanced SQL Engine Database with the SCRIPT Table Operator, note that the Teradata In-nodes R and Python packages must be installed in advance in the target Advanced SQL Engine; see also Section 2.5.

2. Requirements

2.1. Dataiku Data Science Studio version 8.0.2 or later

The Dataiku DSS Enterprise Edition is required to import datasets from Vantage tables. Dataiku offers both downloadable and online options which can be obtained from the Dataiku website at https://www.dataiku.com. The downloadable option can be configured to use either of the DSS Free or Enterprise editions, while the online option only comes with a free 14-day trial of the Enterprise Edition. A comparison between the two editions can be seen in the features table for Dataiku DSS Editions at https://www.dataiku.com/dss/editions.

The Vantage plugins for Dataiku DSS have been tested on Dataiku DSS version 8.0.2.

2.2. Plugin

To use any of the plugins, you will need to download and install them first; see Section 4.

2.3. Access Credentials

To use the plugins, you will need 2 different kinds of credentials, that is, one set for DSS and a second one for Vantage. Specifically:

a. Dataiku DSS user credentials allow a user to login to a DSS instance. Your DSS server administrator can provide you with these credentials.

b. Vantage credentials allow a user to connect to the Advanced SQL Engine Database of a Vantage system, and, with appropriate permissions, read and write tables into the Advanced SQL Engine. Your Vantage database administrator (DBA) can provide you with credentials and suitable permissions for one or more databases on a Vantage system.

Use your DSS user credentials to log on to a DSS instance, and then use your Vantage credentials to establish a connection between DSS and a Vantage system. Section III ("Creating A Vantage Connection") provides instructions on how to setup a DSS connection to a Vantage Advanced SQL Engine Database. It is suggested to create one connection per each database for which you intend to store output tables in.

2.4. Teradata JDBC Driver

The Teradata JDBC Driver 16.20 or later is required to establish a connection between DSS and a Vantage System.

2.5. Teradata Vantage System

All plugins require a connection to a Teradata Vantage system that minimally comprises of an Advanced SQL Engine.

2.5.1. Vantage MLE Functions Plugin

The Vantage MLE Functions Plugin further requires that your Vantage system features a Vantage Machine Learning Engine v.1.1. The Machine Learning and Graph Engines are required to completely leverage all capabilities of the Vantage MLE Functions Plugin.

2.5.2. Vantage SCRIPT Table Operator Plugin

To use the Vantage SCRIPT TO Plugin with a Vantage system Advanced SQL Engine and execute R and Python scripts in the Advanced SQL Engine nodes, the corresponding language bundles need to be installed directly on each node of the Advanced SQL Engine, per the following table:

PID	Product name	Database version
9687-2000-0120	R Interpreter and Add-on Pkg on Teradata Advanced SQL	16.20
9687-2000-0121	R Interpreter and Add-on Pkg on Teradata Database	15.10, 16.10
9687-2000-0122	Python Interpreter and Add-on Pkg on Teradata Advanced SQL	16.20
9687-2000-0124	Python Interpreter and Add-on Pkg on Teradata Database	15.10, 16.10

Moreover, your DBA must grant you in advance the additional following privileges:

- `EXECUTE` Function privilege on `TD_SYSFNLIB.SCRIPT`
  This is needed in order to invoke the SCRIPT Table Operator.

- `EXECUTE` privilege on the functions `SYSUIF.INSTALL_FILE`, `SYSUIF.REMOVE_FILE`, and `SYSUIF.REPLACE_FILE`.

3. Creating A Vantage Connection

---

1. Follow the instructions in the Dataiku Reference Document for Installing Database Drivers. In summary, one needs to execute from the command line of a DSS server:

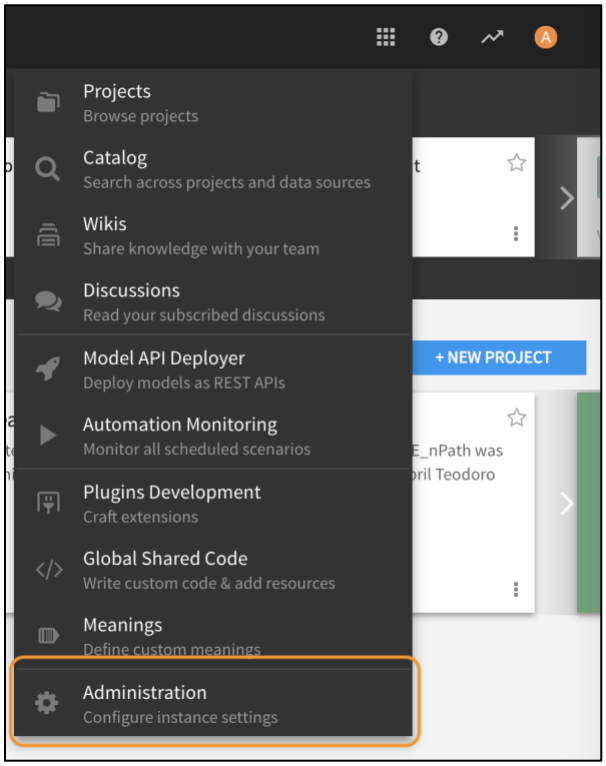
   a. Stop the Data Science Studio server, where `DATA_DIR` is the data directory where Data Science Studio is installed:
   `DATA_DIR/bin/dss stop`

   b. Copy the Teradata JDBC driver to the `DATA_DIR/lib/jdbc` directory.

   c. Restart Data Science Studio:
   `DATA_DIR/bin/dss start`

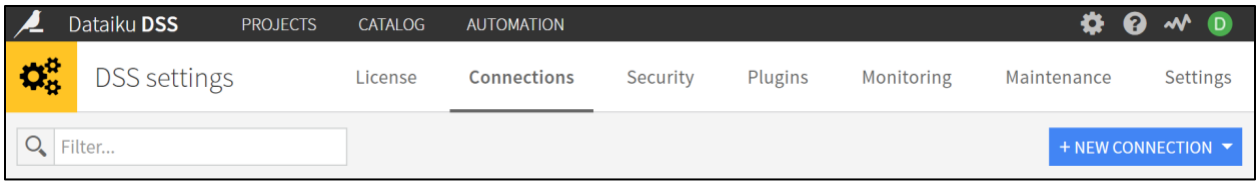
2. Access Dataiku DSS on a browser. Then, on the Dataiku DSS home page click on Apps.

Then, on the submenu click [`Administration`] (gear icon).



Alternatively, you can go to `http://<dataikuhost>:<port>/admin/`.

3. On the DSS settings page, go to the [Connections] tab. Click on [NEW CONNECTION]. Choose [Teradata] among the options that will be presented.

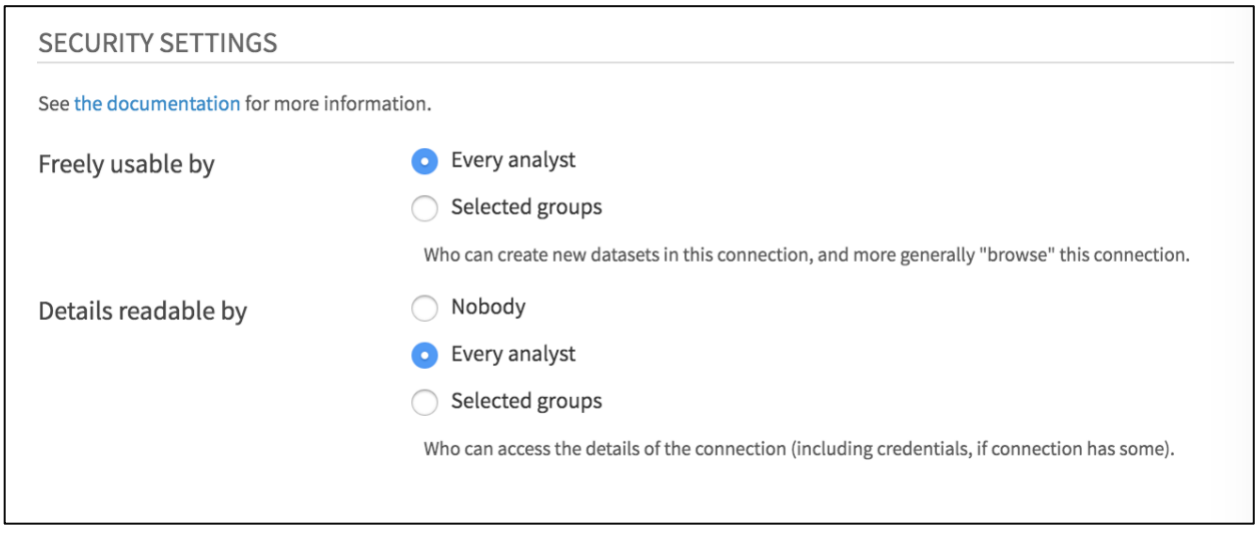


4. Fill in the fields as needed:

   Basic Params Host: <database.host.name>
   User: <your_database_username>
   Password: <your_database_user_password>
   Default Database: <default_database>
   Advanced JDBC properties:
   CHARSET: UTF8
   TMODE: TERA
   Note: If your target system connections is LDAP-based, then also specify:
   LOGMECH: LDAP
   Autocommit Mode: Check the button to enable the autocommit mode.

   All other fields can be left as-is.

5. Modify “Details readable by” to either “Every Analyst” or “Selected Groups“.



6. Click on the [Test] button to verify that connection details provided are valid.

7. Finally, click on the [Save] button.

4. Plugin Download And Installation

4.1. Get The Plugins

The Vantage Plugins for Dataiku DSS can be downloaded from the following public Github repos:

1. For the Vantage SQLE Functions plugin, visit the page:
   https://github.com/Teradata/vantage-dss-plugin-sqle-functions

2. For the Vantage MLE Functions plugin, visit the page:
   https://github.com/Teradata/vantage-dss-plugin-mle-functions

3. For the Vantage SCRIPT Table Operator plugin, visit the page:
   https://github.com/Teradata/vantage-dss-plugin-sto

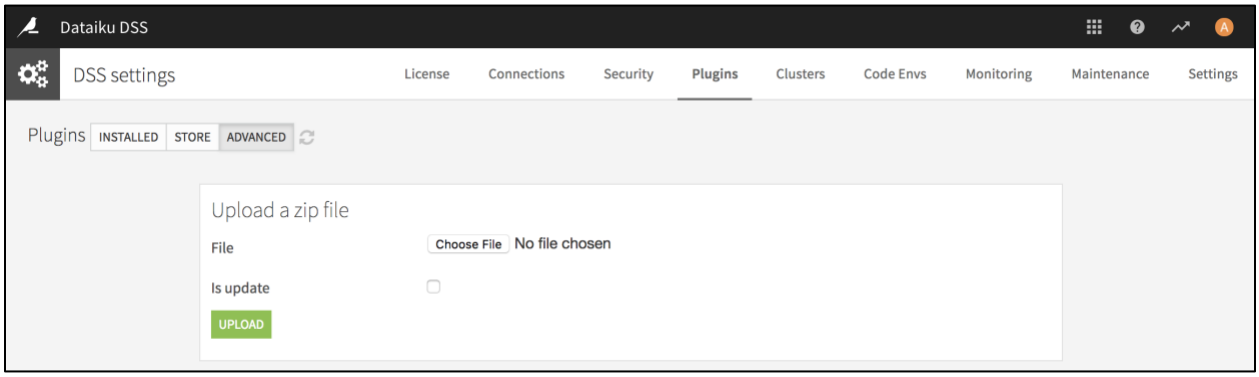
To download any of the plugins, click on the green "Code" button on the corresponding plugin page, and further select to "Download ZIP". This action will prompt you to save a compressed zip file that contains the corresponding plugin software and metadata.

4.2. Plugin Installation

The steps to install any of the Vantage SQLE Functions, Vantage MLE Functions, or the Vantage SCRIPT TO plugins are as follows:

1. Assume that the zip file of the plugin you want to install is stored in your local filesystem.

2. In DSS Settings page (accessible through the Admin Tools button), select the [`Plugins`] tab, then select the [`ADVANCED`] option.



3. Click on [Choose File] and navigate to the location of the plugin zip file in your local filesystem.

4. If a previous installation of the plugin exists, check "Is update".

5. Click on [UPLOAD] button.

6. When the upload succeeds, click on [Reload] button, or do a hard refresh (Ctrl + F5) on all open Dataiku browsers for the change to take effect.

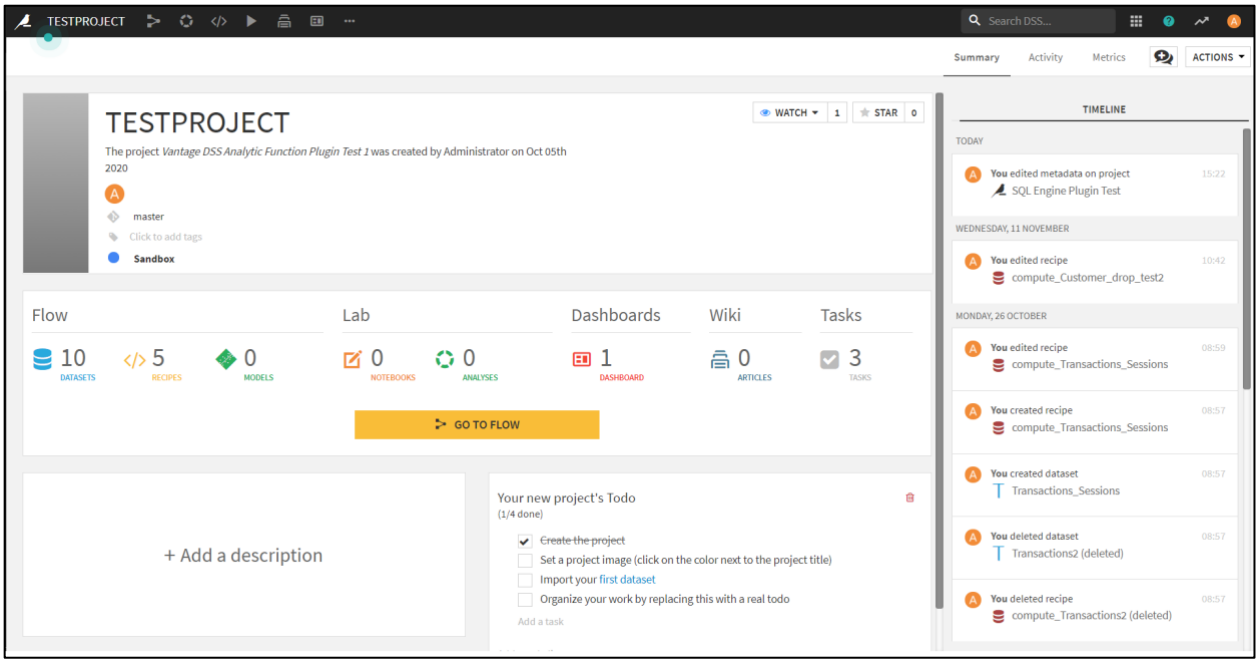
7. Repeat process, if you want to install a different plugin.

5. Using the Vantage SQLE and MLE Functions Plugins

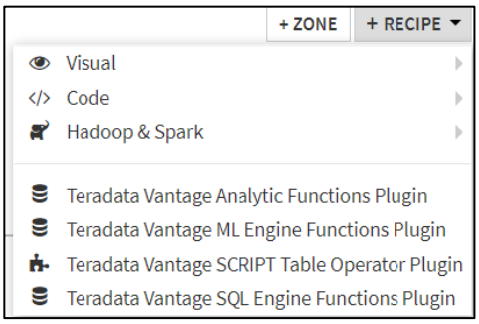
5.1. Instructions

This section assumes that a Dataiku DSS project already exists, and input datasets have already been imported. Note that recipes need a non-empty dataset as input to run.

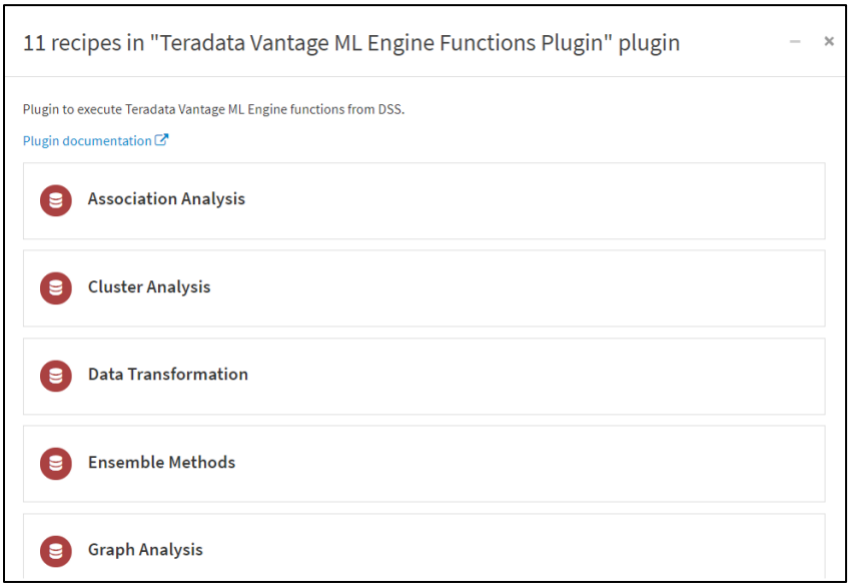
1. Go to the flow view of the DSS project, where the recipe is to be created, by clicking on the [`GO TO FLOW`] button, or by clicking on the flow icon in the project menu.

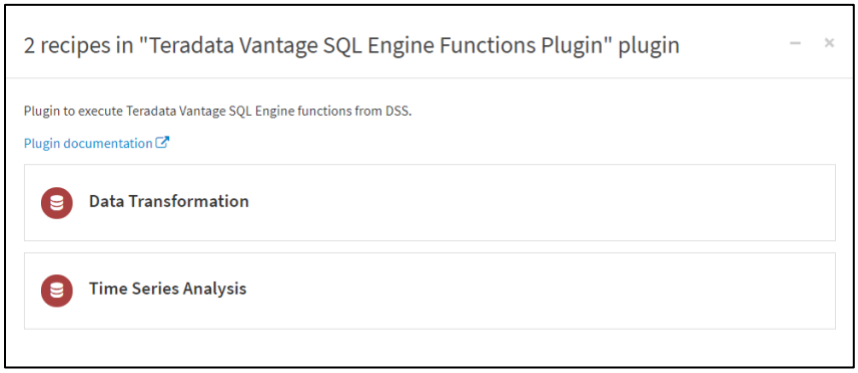


2. In the Flow view, click on the [+RECIPE] button, then select the [Vantage MLE Functions Plugin] or the [Vantage SQLE Functions Plugin] as desired.

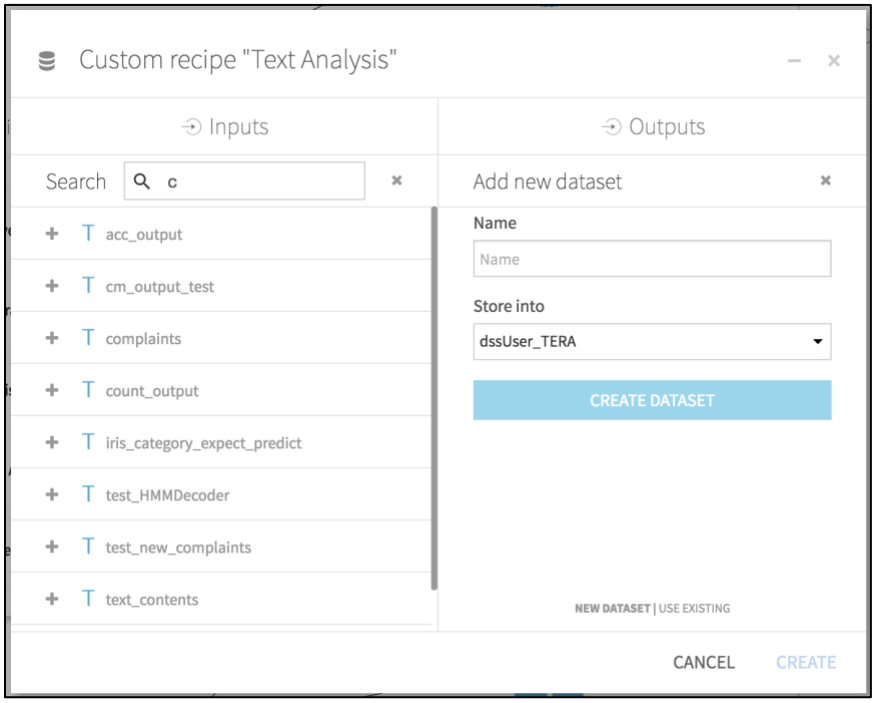


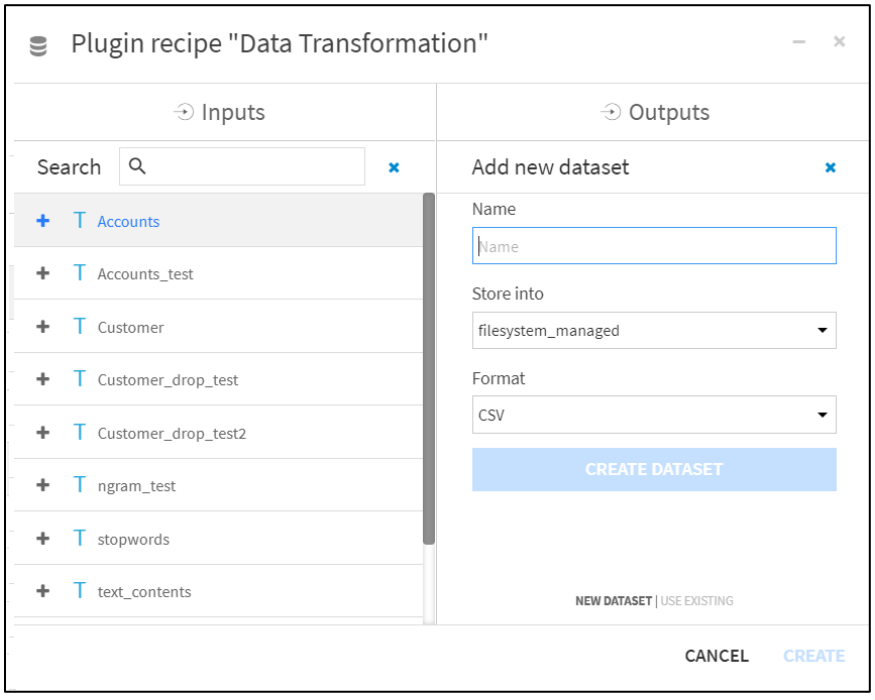
Then proceed to select the desired recipe. Available recipe names correspond to the different categories of Vantage MLE Functions or SQLE Functions, as illustrated in the following figures.



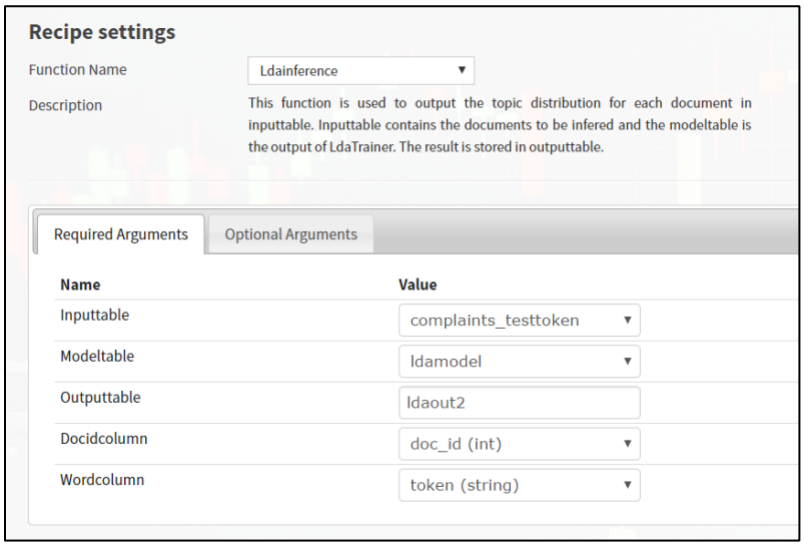


3. In the [`New custom recipe`] popup, specify the input and output datasets. There can be more than one input dataset, as in the case of multiple-input analytic functions. The same is also the case for Vantage functions with multiple output datasets. The output dataset will be stored in the database and schema corresponding to the connection selected in the [`Store into`] field. Click on [`CREATE DATASET`] button when done. The following figures illustrate examples for the Text Analysis recipe in the Vantage MLE Functions plugin, and the Data Transformation recipe in the Vantage SQLE Functions plugin.

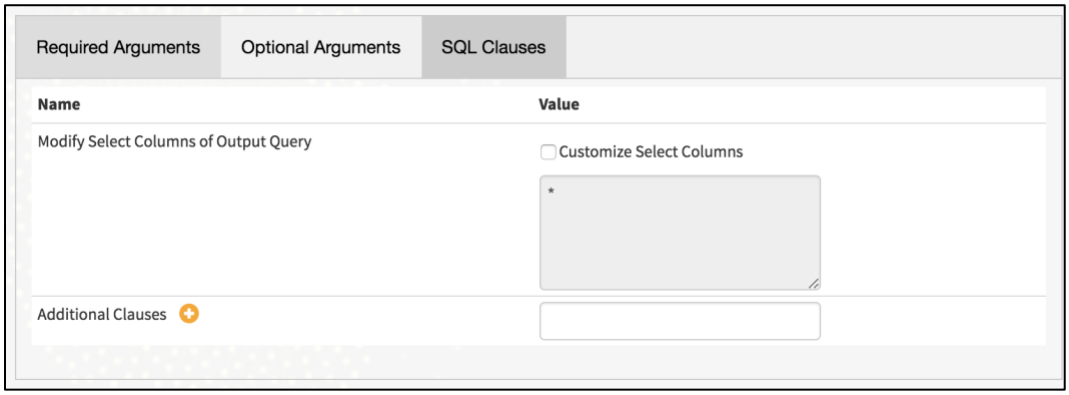




4. In the recipe settings, one can select the most suitable function for the manipulation or analysis of the input dataset. Configure the chosen analytic recipe by specifying parameters such as the input tables, partition and order attributes, and arguments. A recipe's required and optional fields are separated into different tabs.



5. The [`SQL Clauses`] tab allows the user to explicitly modify the query to be executed.



The field next to "`Modify Select Columns of Output Query`" enables the user to modify the SELECT clause of the query. The field next to "`Additional Clauses`" enables the user to append additional SQL clauses to the query such as WHERE, ORDER BY, GROUP BY, and other similar clauses. These fields have equivalent effects as if the query were modified as:

```
SELECT {modified select} FROM function_name(
    ...
    )
    {additional clauses}
```

6. Click on the [RUN] button or save the recipe settings for later use.

5.2. Usage Notes

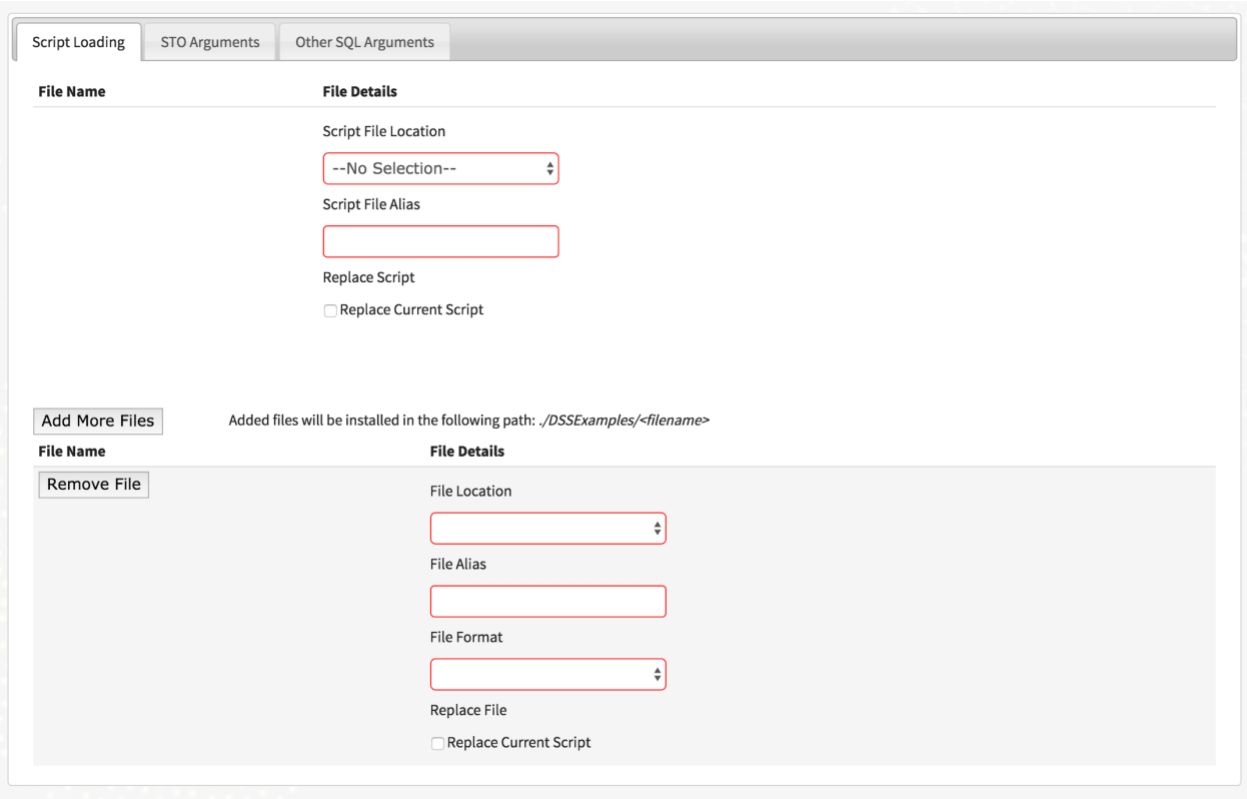
A function with multiple output datasets will typically require an output dataset for the function's output message/result, in addition to any other output tables/datasets specified in the recipe. Please note that the output dataset(s) name(s) should also match the name within the recipe's settings.

6. Using the Vantage SCRIPT Table Operator Plugin

This section assumes that a Dataiku DSS project already exists and input datasets have already been imported. Note that recipes need a non-empty dataset as input to run.

There are three (3) main tabs containing arguments used to install/replace the script files on the Advanced SQL Engine Database and/or invoke the SCRIPT Table Operator (STO).

6.1. Script Loading



- Script File Name
  - The name of the script file to be uploaded.
  - This is the main script used in the SCRIPT Table Operator.
  - Depending on the selected Script File Location this input changes:
    - If the script is on the Vantage Server – A text input field is provided to enter a file name.
    - If the script is in the DSS Managed Folders and DSS Notebooks – A drop-down box containing a list of the files under their respective locations is provided.
  - The Script File Name will not appear until the Script File Location is selected.

- Script File Location
    - The location of the script to be installed, either on the Vantage server, a DSS Jupyter Notebook, or a DSS Managed Folder

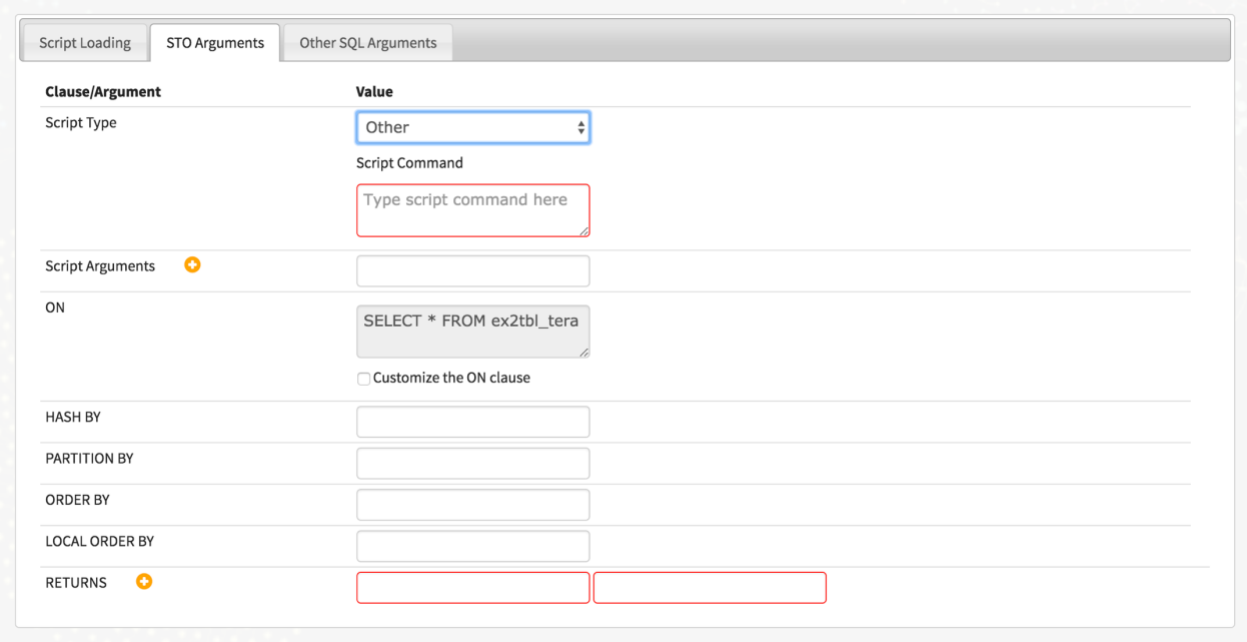
- Script File Alias
    - The file alias to be used in the SQL statement
    - This is mainly used by the SCRIPT Installation/Replace process in the metadata tables.

- Script File Address
    - The fully qualified file location on the Vantage Server
    - This only appears if the selected option for Script File Location is “Vantage Server”

- Add More Files
    - This button allows the user to have additional files installed in the Vantage Advanced SQL Engine.
    - There is a file path specified to the right of the button in which the additional files are installed.
        - This may normally be used in instances where the user’s main script references an additional file.

- Additional Files:
    - File Name
        - This is the file name of an additional file.
        - Similar to the Script File Name it is a Text Field for files located in the Vantage Advanced SQL Engine and a drop-down box if DSS Managed Folder is selected as the File Location
    - File Location
        - The location of the file to be installed, either on the Vantage server or a DSS Managed Folder
    - File Address
        - The fully qualified file location on the Vantage server
        - Similar to the Script File Address this only appears when "Vantage Server" is selected as the file location.
    - File Format
        - Specifies whether the additional file to be installed is a BINARY or TEXT file.

6.2. SCRIPT Table Operator Arguments



- Script Type
  - The type of script to be used typically Python or R'
  - Script Command
    - This is a Text area where the user can enter a custom Script Command.
    - This argument only appears if the selected Script type is “Other”.

- Script Arguments
  - The arguments for the script, place one argument per box. Click on the (+) button to add more arguments'

- ON
  - The ON Clause used as the input data for the script
  - If UNMODIFIED the clause defaults to *“SELECT * FROM {input_table}”*

- Customize the ON clause
  - A checkbox which specifies whether the ON clause should be modified.

- HASH BY
  - A HASH BY clause will cause the rows in the ON clause to be redistributed to AMPs based on the hash value of the column(s) specified'

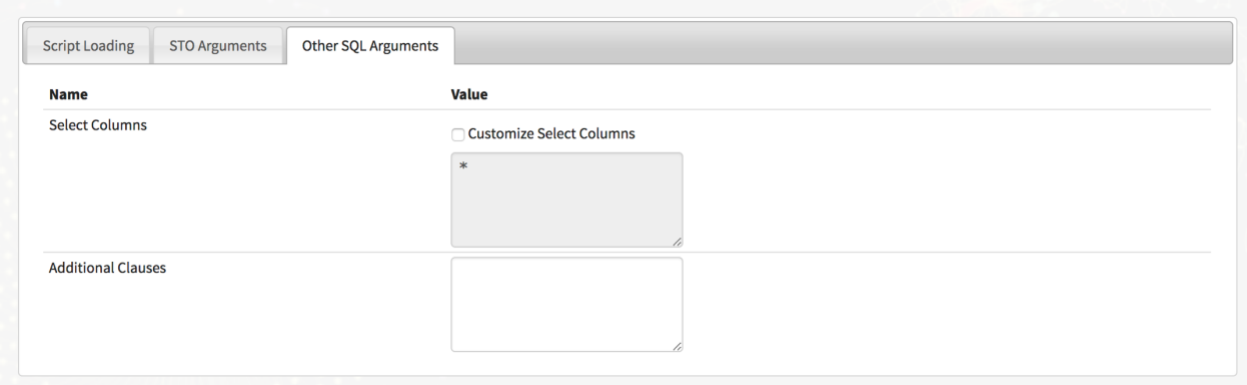
- PARTITION BY
  - A PARTITION BY clause will cause the STO to execute against specific groups (partitions) based on the column(s) specified

- ORDER BY
  - 'An ORDER BY clause specifies the order in which values in a group (partition) are sorted

- LOCAL ORDER BY
  - A LOCAL ORDER BY clause orders the rows qualified on each AMP

- RETURNS
  - RETURNS NAME
    - The first column under returns
    - Specifies the name of the column(s) to be returned by the STO'
  - RETURNS TYPE
    - The second column under returns
    - Specifies the data type of the column(s) to be returned by the STO

6.3. Other SQL Arguments



- Select Columns
  - Specifies the contents of a user customized SELECT statement (data to be returned by the query)
  - Default is to SELECT all column(s) in the RETURNS clause

- Customize Select Columns Checkbox
  - Determines whether the SELECT (output) columns (data to be returned by the query) should be modified.

- Additional Clauses
  - Specifies any additional clauses to the output such as a HAVING or QUALIFY clause

6.4. Running the Vantage SCRIPT Table Operator Plugin

After setting up the arguments, click on the [RUN] button to run the SCRIPT Table Operator.