Trifacta Wrangler Tutorial

Summary	In this codelab you will get a well-rounded explanation of Trifacta's transformation capabilities and walkthrough of the end to end workflow - from connecting to the data to joining datasets and creating recipes to generating output
Dataset URL	https://drive.google.com/drive/folders/11ZS06eEaAzs3wIpjE3YRbLJhdzO-wlc2?usp=sharing
Category	Trifacta
Author	Nikhil Kohli

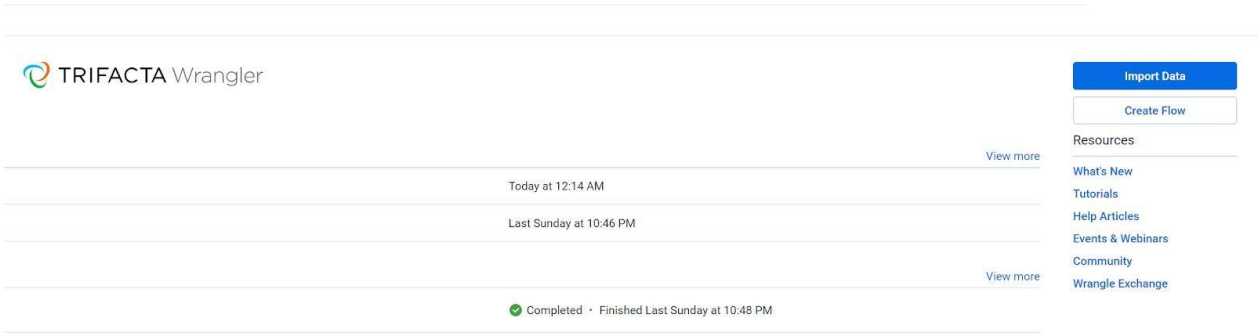
What is Trifacta Wrangler?

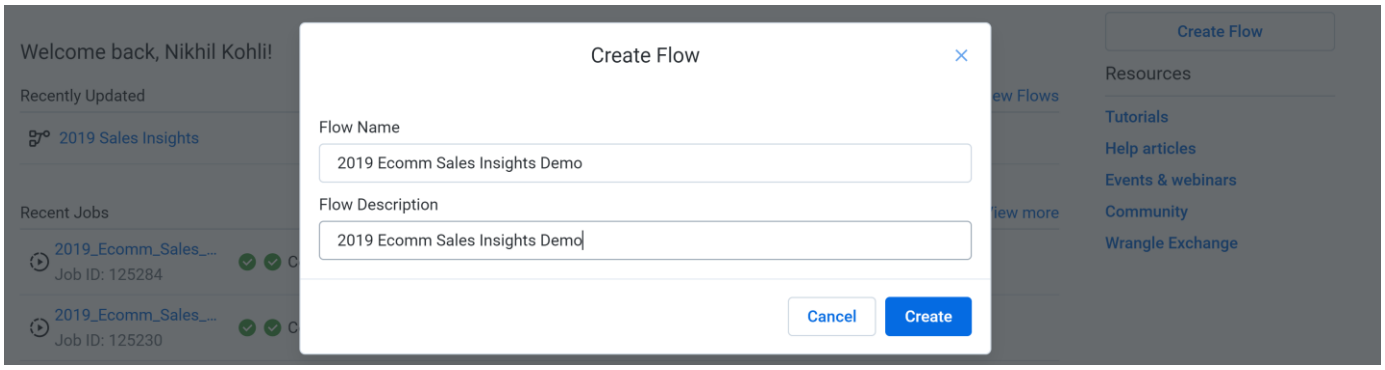
Trifacta Wrangler enables you to explore, combine, and transform diverse datasets for downstream analysis.

Demo

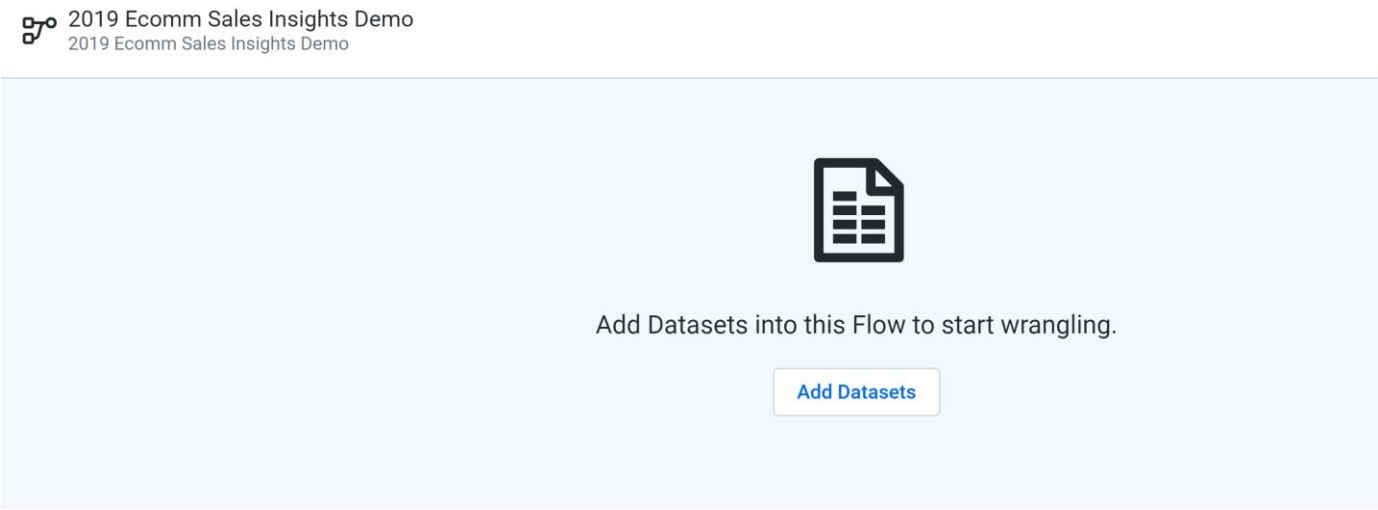
This tutorial walks through editing a sample dataset using the free tool Trifacta Wrangler, which is available for download here. The sample dataset used here is a CSV of Ecommerce 2019 Sales data and synthetically generated user data.

Once you have signed up for Trifacta Wrangler, select a new flow—this will create a new flow in which you can organize your dataset.

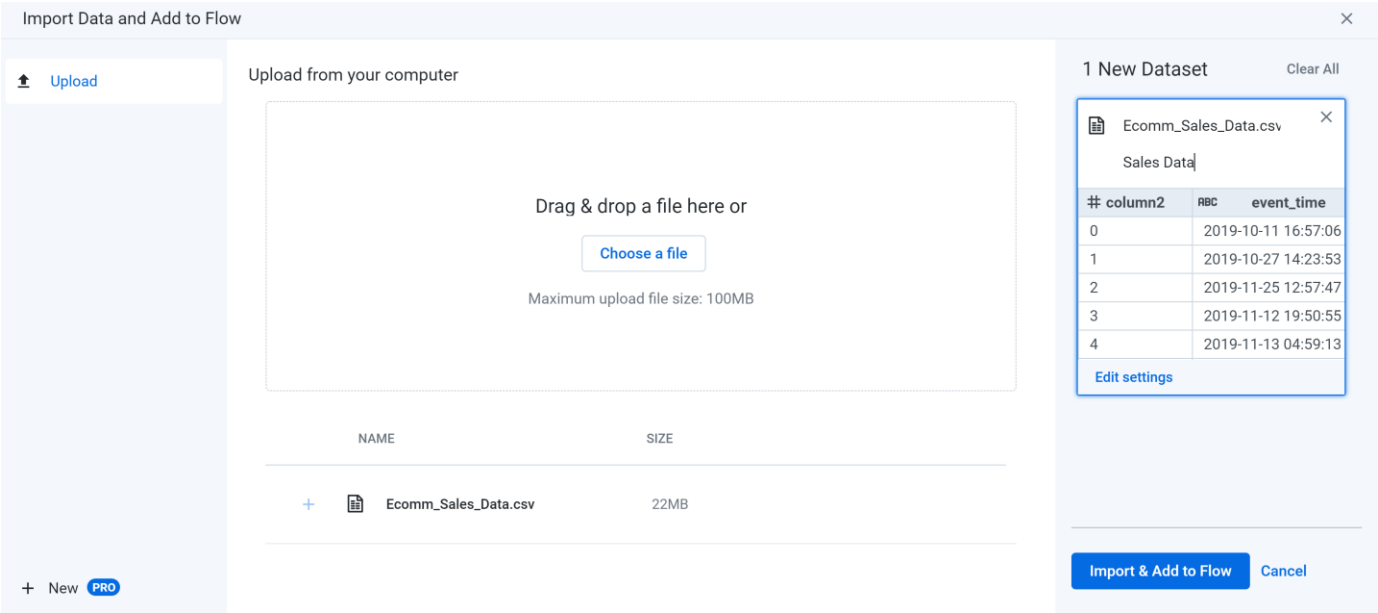




Then select add datasets to flow, and then import a dataset.

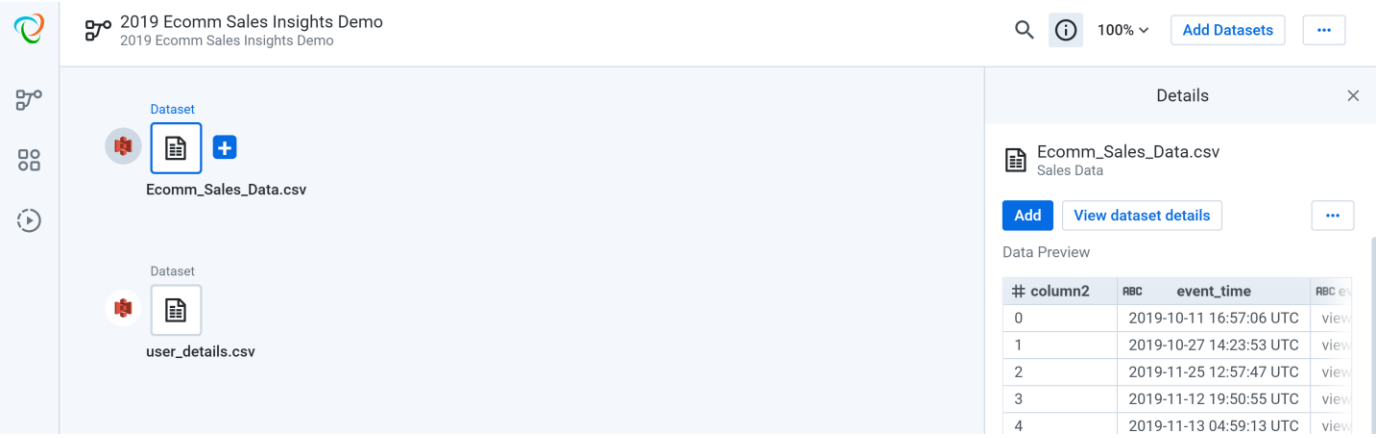


From the import dataset window, you can either drag and drop both the csv provided or choose the file from your computer. Once you have done that, make sure your dataset is selected, then click Import and add to flow.



We can connect Trifacta to many more data sources like databases and cloud storages but in the pro version.

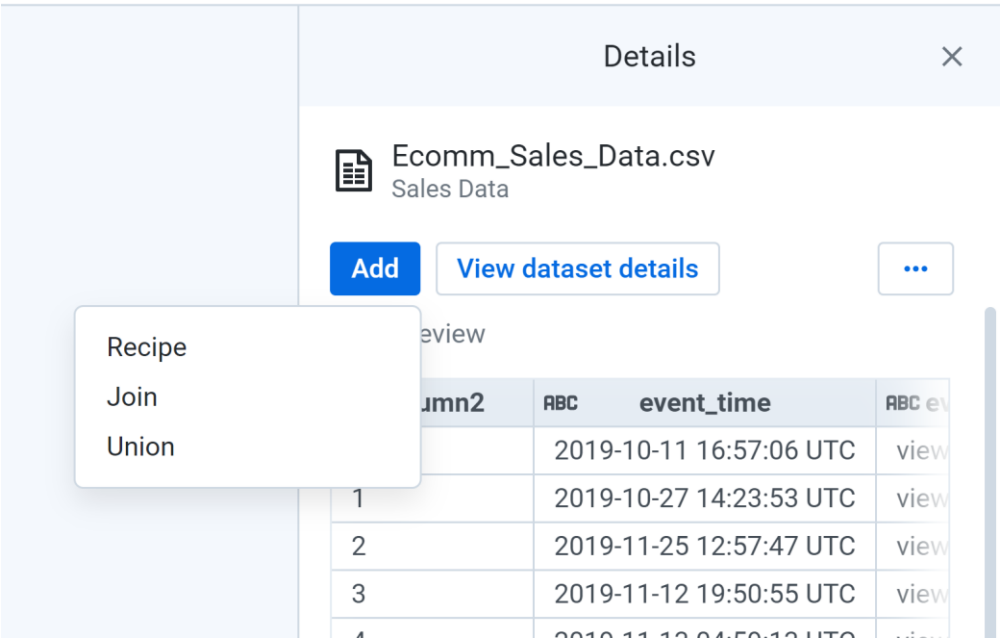
Then import the second csv too, it should look like this once you are done.



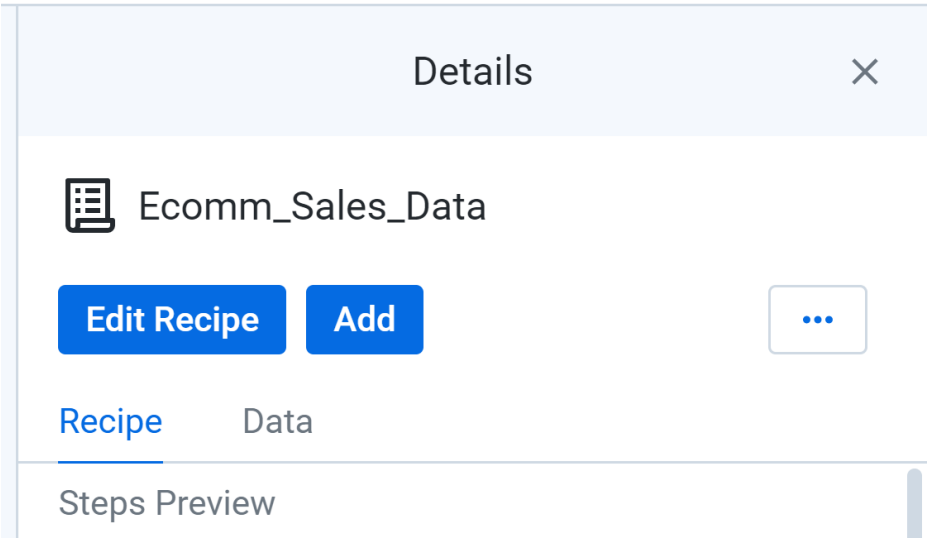
Recipes

Trifacta Wrangler works by generating *scripts* that apply *transformations* to your data and then compiling multiple scripts into a *recipe*. Multiple datasets and recipes are organized in a *flow*.

Trifacta Wrangler will automatically generate an initial recipe for your dataset that will convert it from is the original format to something Wrangler can transform. Because it is a CSV, this recipe will include steps such as converting newline characters and commas into new rows and columns. However, it will generate similar scripts from JSON files as well.

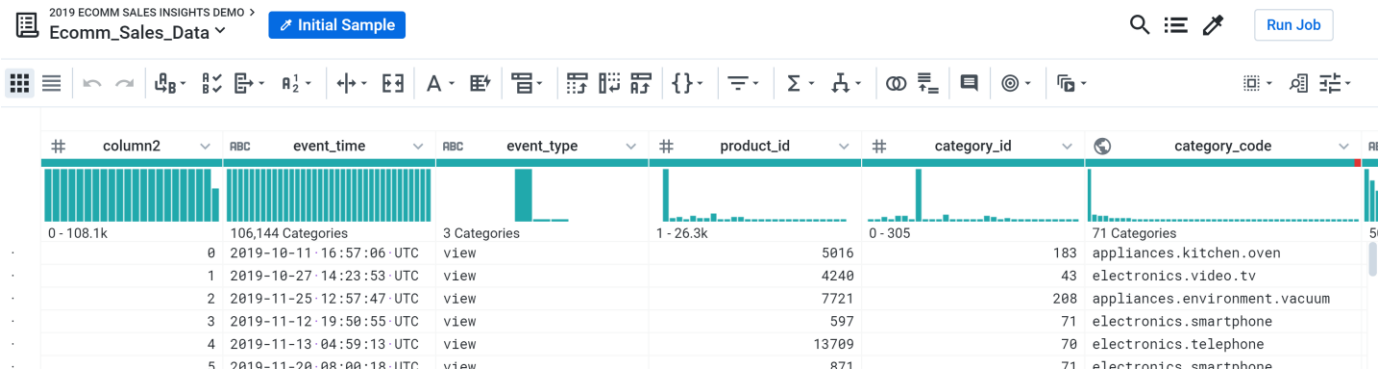


Click on Add > Recipe > Edit recipe to bring up the transform builder and a preview of the dataset. Here is a quick rundown of the editor's important features.



Data Quality indications

- For each column, Wrangler displays the percentage of the data that is *valid* (the same format as the selected or inferred data type), *invalid* (a different format), and *empty*.
- This is visible for each column directly below the column name. If the bar is completely green, the data is 100% valid values; invalid values are red and empty values are gray.

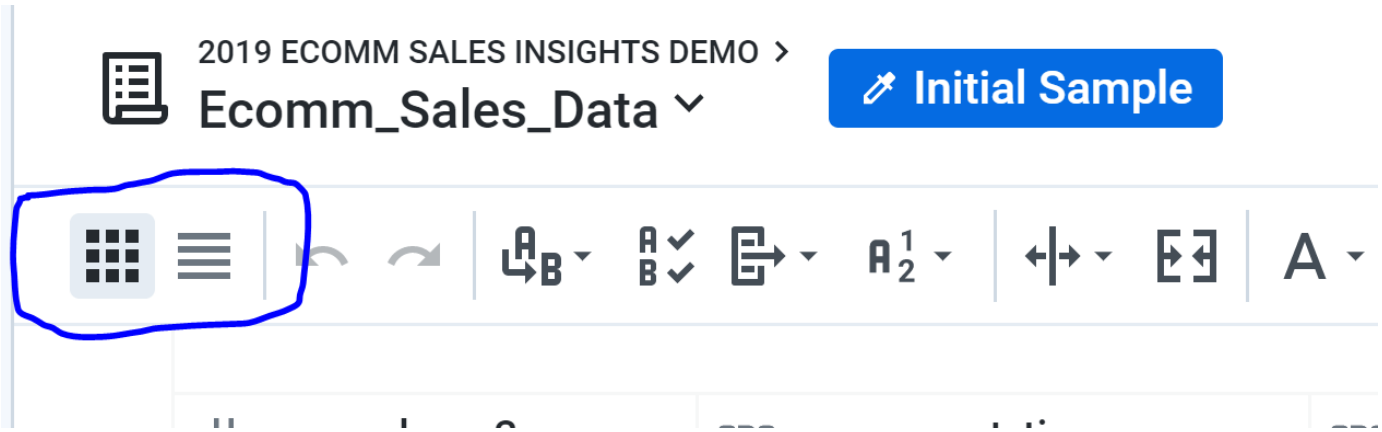


Data Types

- Wrangler will also guess at the data type of each of your column and display an icon of the data type next to the column name.
- Knowing the data type helps Wrangler suggest which transformations might be useful/applicable.
- There are many different supported types, which can be viewed, along with more on data types here.

Grid vs. Column view

- With large datasets and/or datasets with many columns, you may want to see only a few of them when writing transformations.
- Selecting *columns* will allow you to select which columns are visible in the grid view.

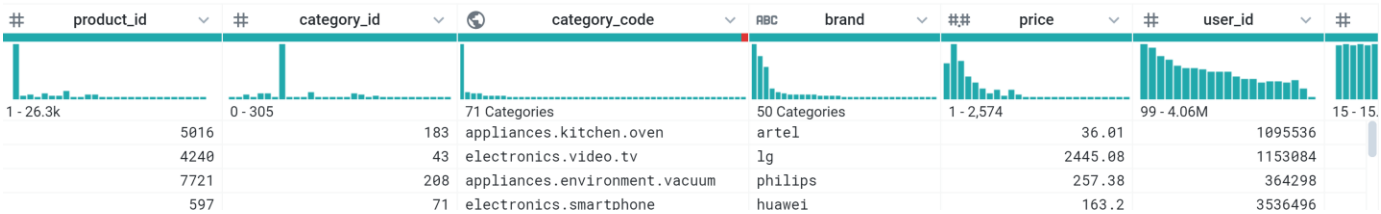


- It will also give you the data quality indicators for each column.

Transformation Builder

- This is the main feature of the tool. It allows you to choose from broad types of transformations and then customize them to edit your own data.
- Below, I'll walk through a quick tutorial of how to wrangle the sales data set you've just imported. I'll include both how to make and customize the transformations and the wrangle script that accompanies them.

First things first: matching the data types.

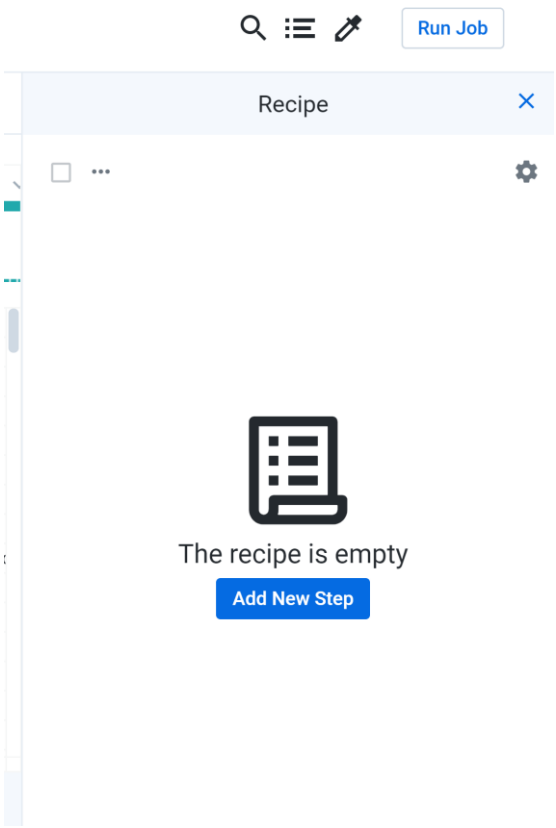


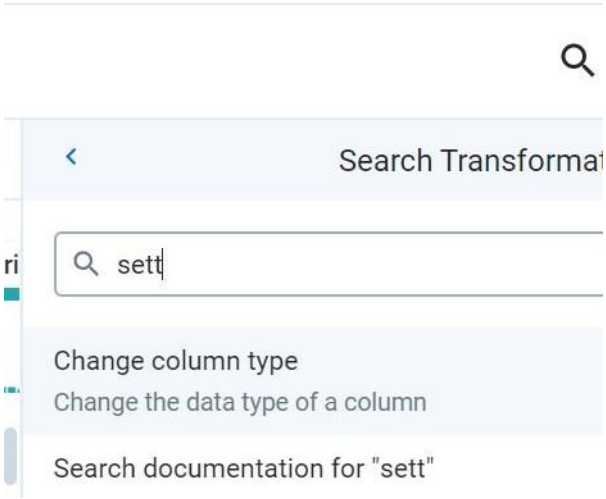
product_id	category_id	category_code	brand	price	user_id	
1 - 26.3k	0 - 305	71 Categories	50 Categories	1 - 2,574	99 - 4.06M	15 - 15.
5016	183	appliances.kitchen.oven	artel	36.01	1095536	
4240	43	electronics.video.tv	lg	2445.08	1153084	
7721	208	appliances.environment.vacuum	philips	257.38	364298	
597	71	electronics.smartphone	huawei	163.2	3536496	

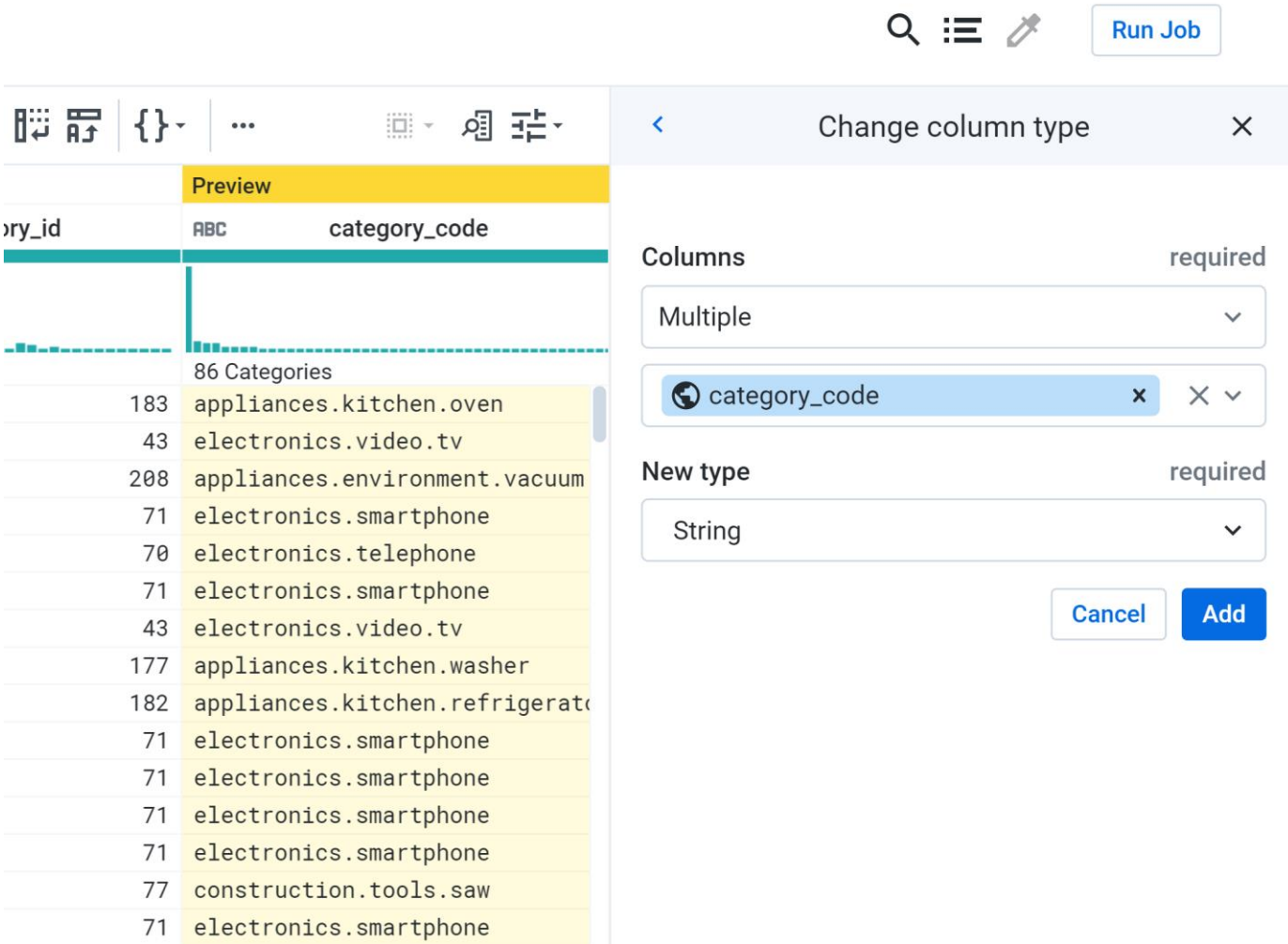
For the most part, Wrangler is good at guessing your data types, but it thinks ‘category code’ as a url.

- In the transform builder, choose the settype transform, *category code* as the column, and enter *Integer* as the new type, then select add to the recipe.

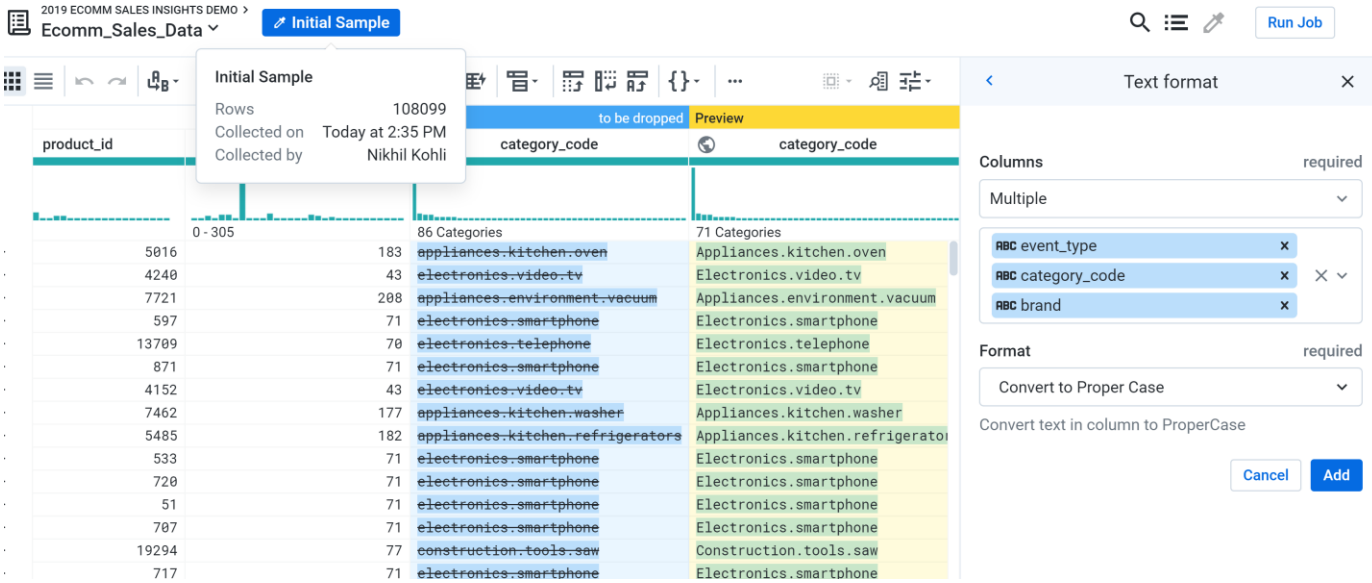
```
settype col: category_code type: 'String'
```


Run Job
Recipe
The recipe is empty
Add New Step


Search Transformat
sett
Change column type
Change the data type of a column
Search documentation for "sett"

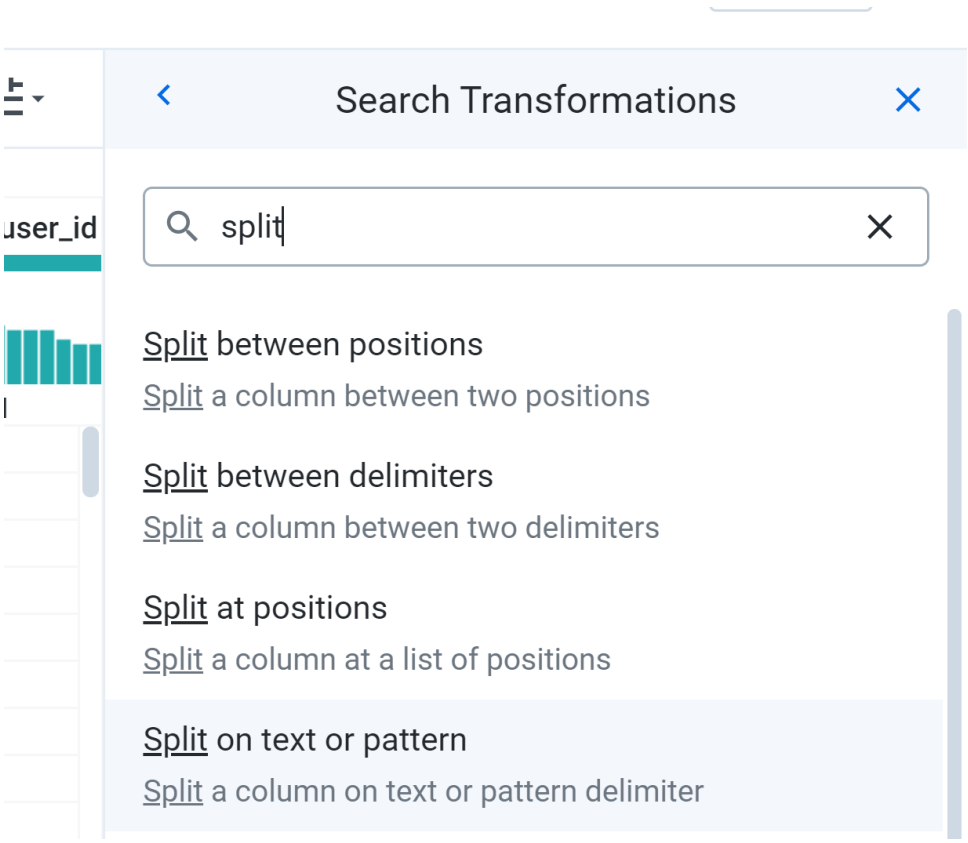


Next, Lets convert the text format to proper case for 3 columns -
Search text format and add the column names

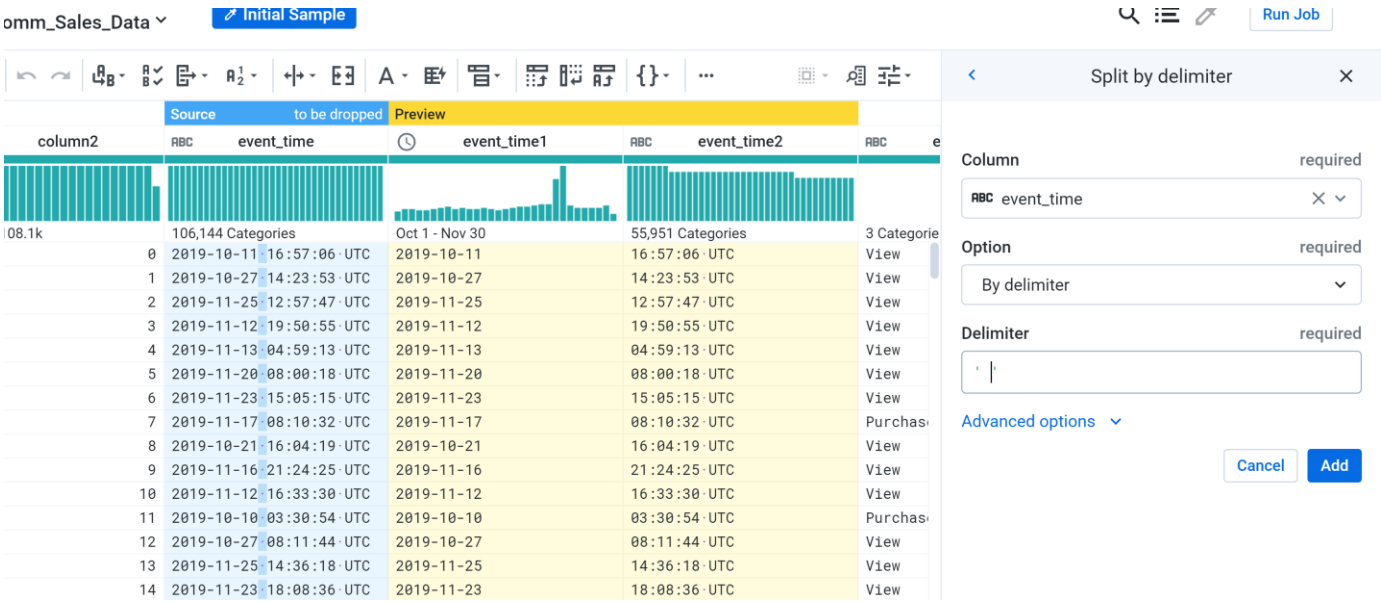


Splitting and feature Extraction

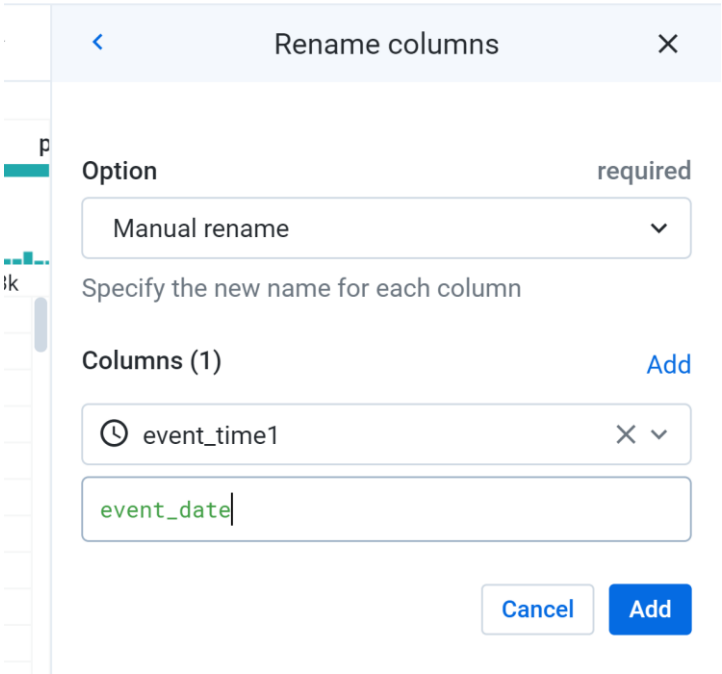
Now, we will split the date time into date and time columns so we can use the date for extracting further columns -



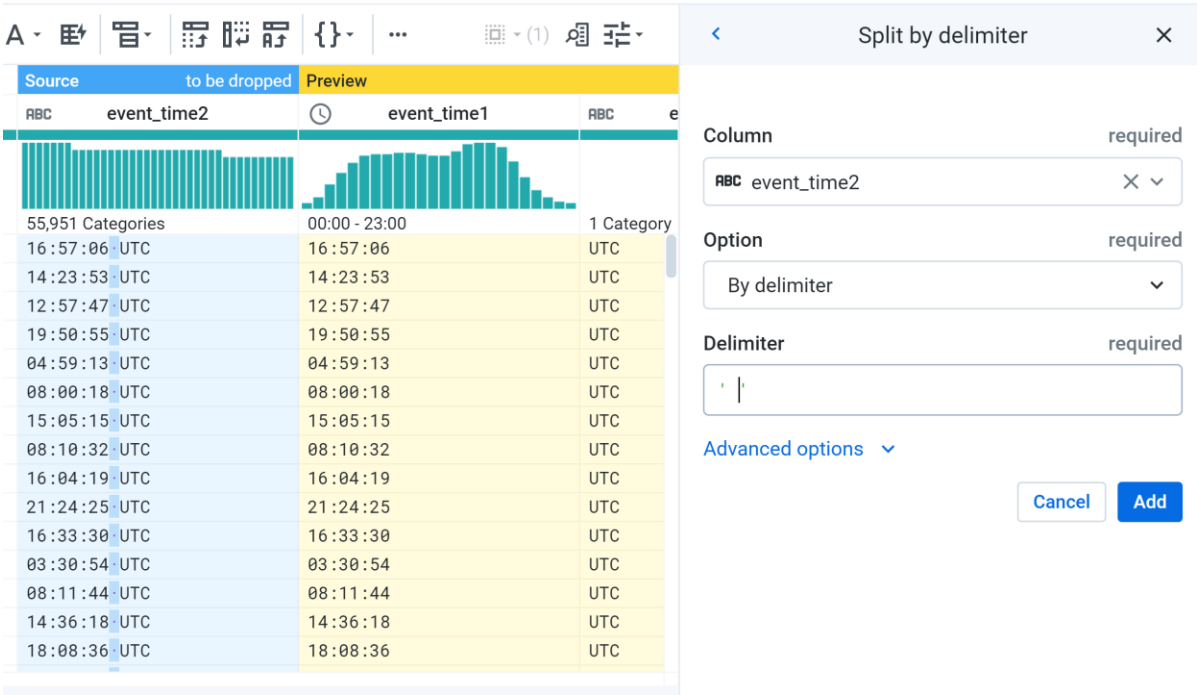
Search - Split in the transformations and select - 'split between delimiters'



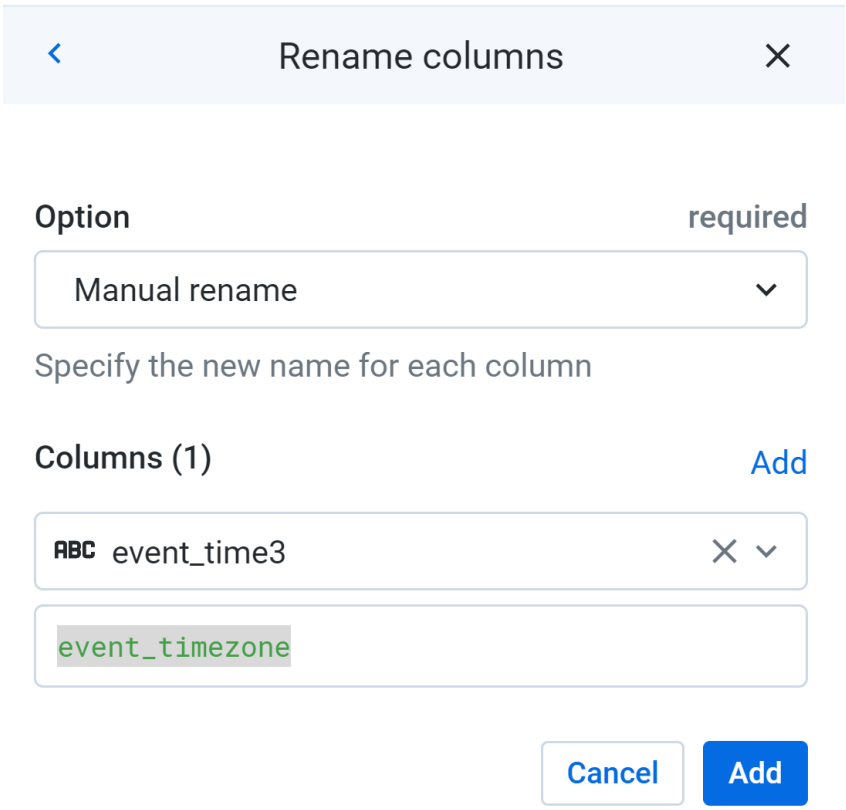
This will create 2 new columns with date and time separately. Now we can rename these 2 columns.



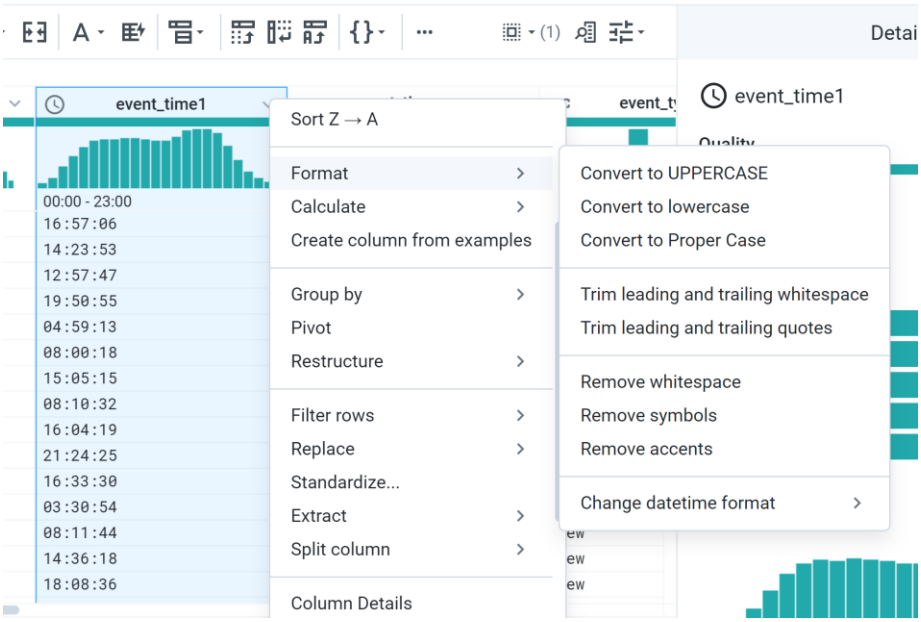
Split event_time2 again to fetch Time zone and time.



Rename the column to event_timezone



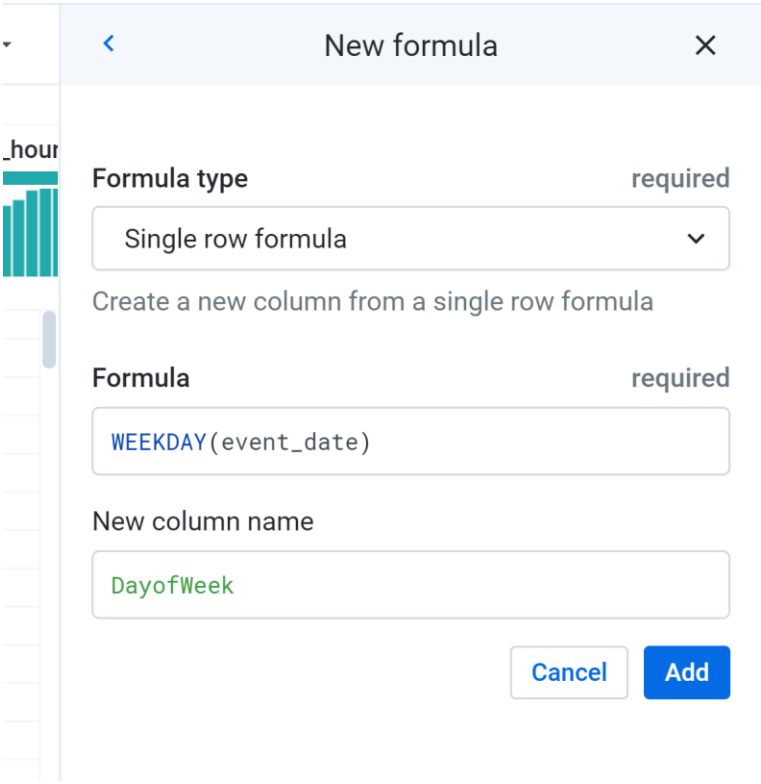
Extract hour from event_time



Extract > Datetime > Hours

Extract Day of week column from event date

Extract > Datetime > Day of Week

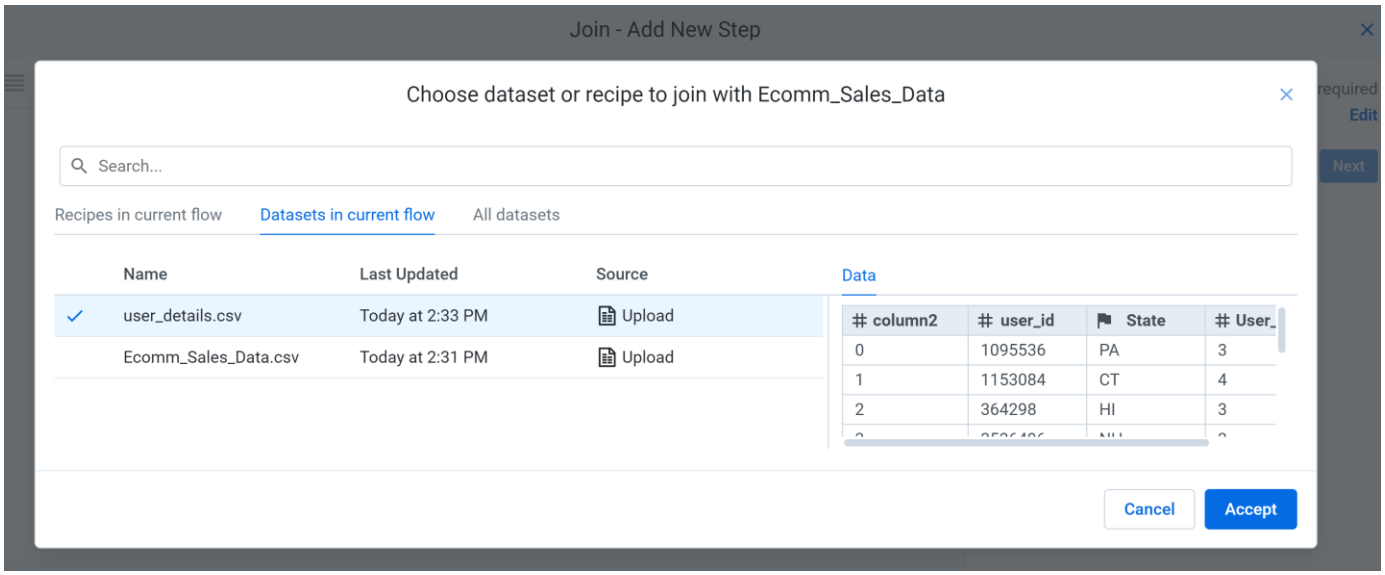


Now lets use if conditions to update the day of week

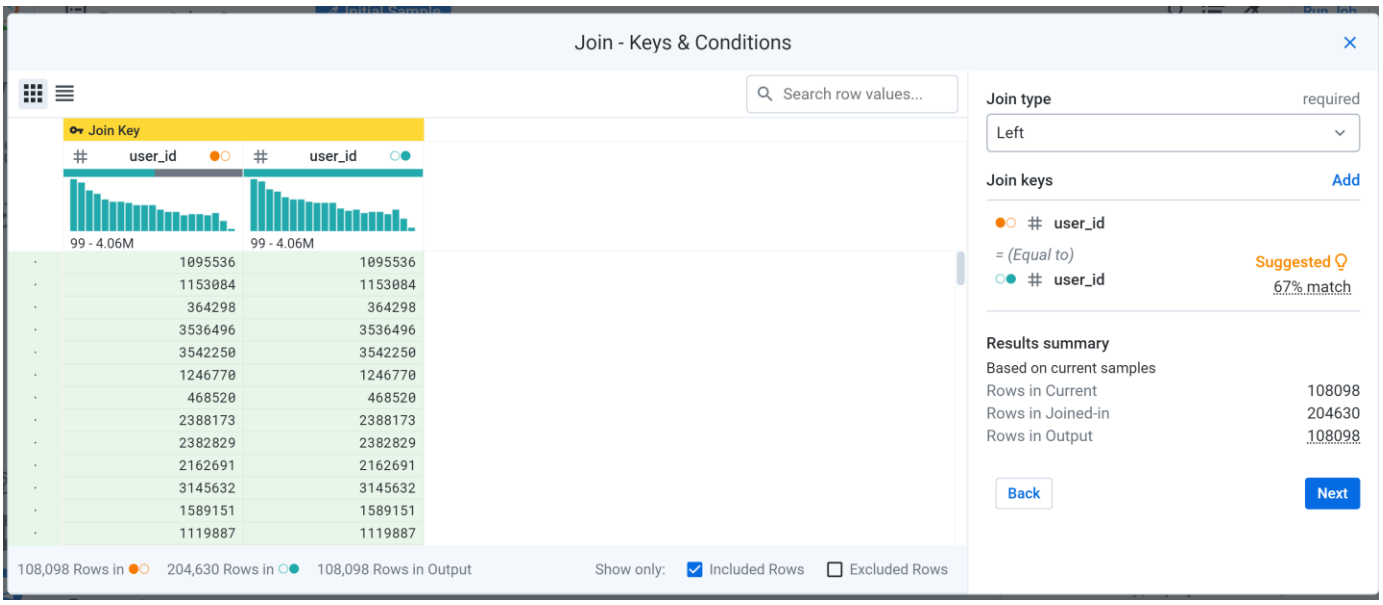
IF(DayofWeek == 1, 'Monday', IF(DayofWeek == 2, 'Tuesday', IF(DayofWeek == 3, 'Wednesday', IF(DayofWeek == 4, 'Thursday', IF(DayofWeek == 5, 'Friday', IF(DayofWeek == 6, 'Saturday', 'Sunday'))))))

Now lets join this dataset with the other file – user_data

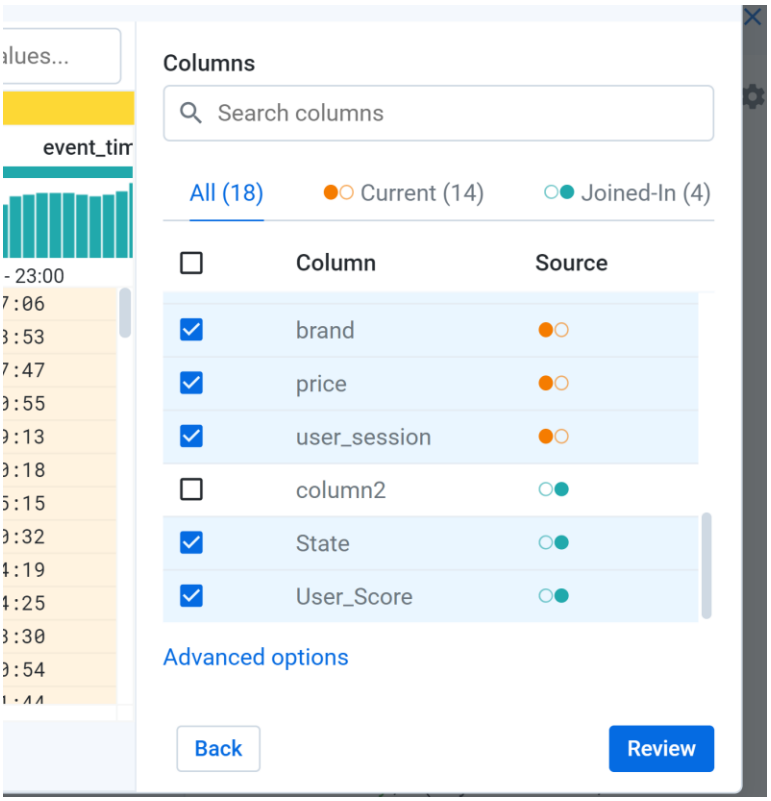
- Search Join Datasets



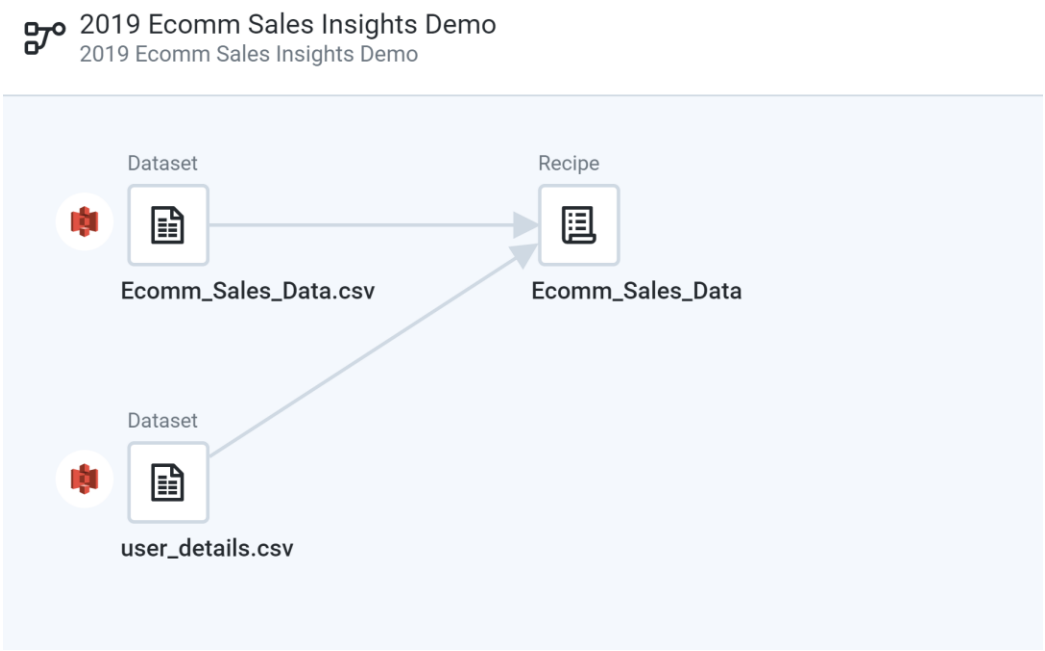
We can select different types of joins as well. Here we will select Left join to get all the data from our main table



Select the columns you want to add from other table



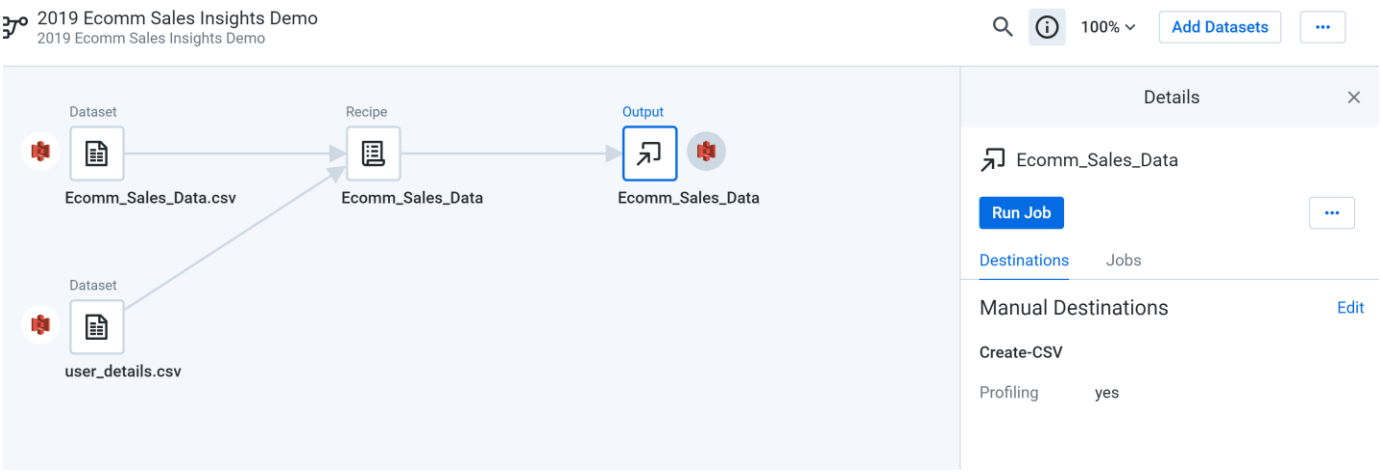
If you check the flow now it will look something like this



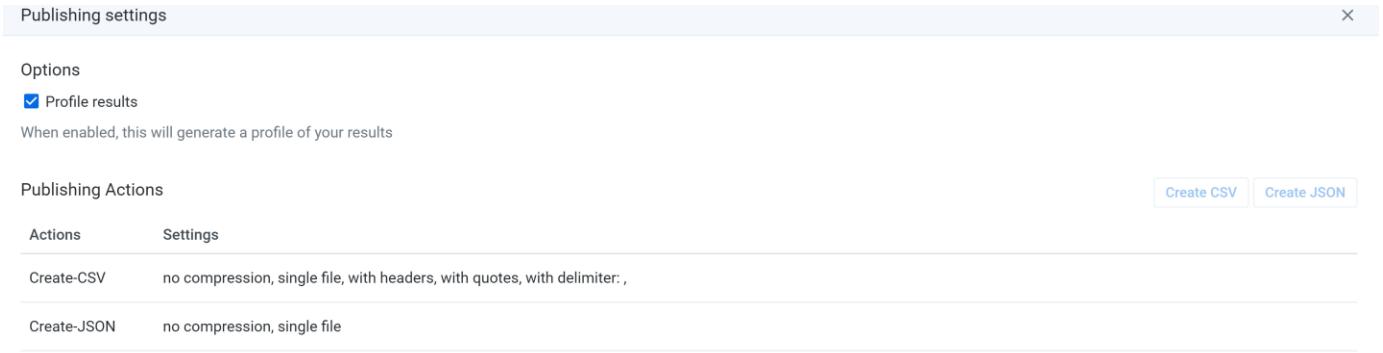
Download the transformed data -

After making any additional modifications to the dataset (find the full range of transformations here), you can generate results for the entire dataset by clicking Generate Results.

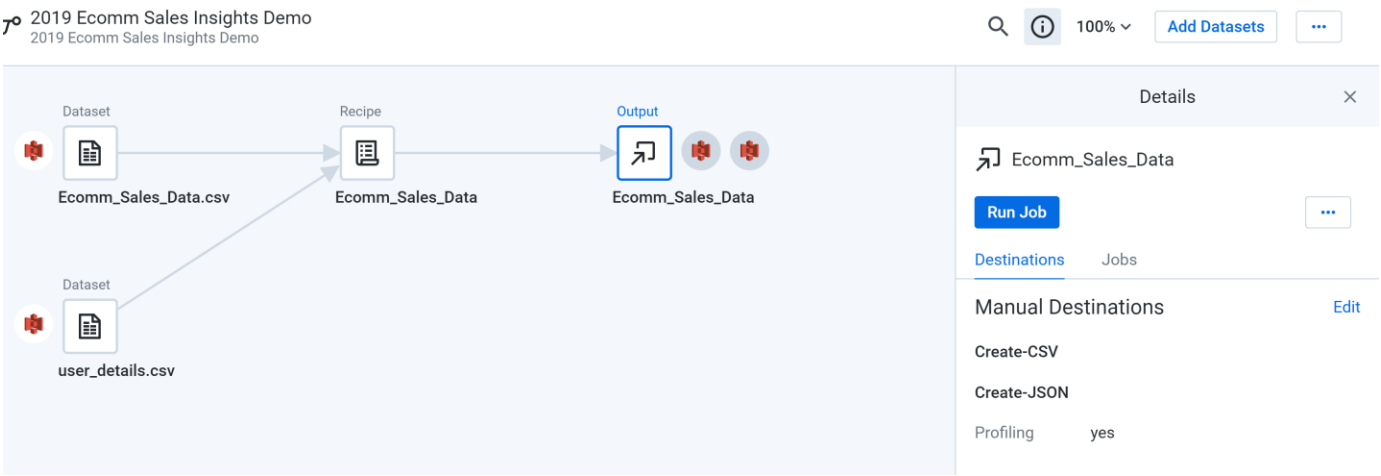
- You can choose the format to download your data, an optional method of compression, and view a summary of the dataset.

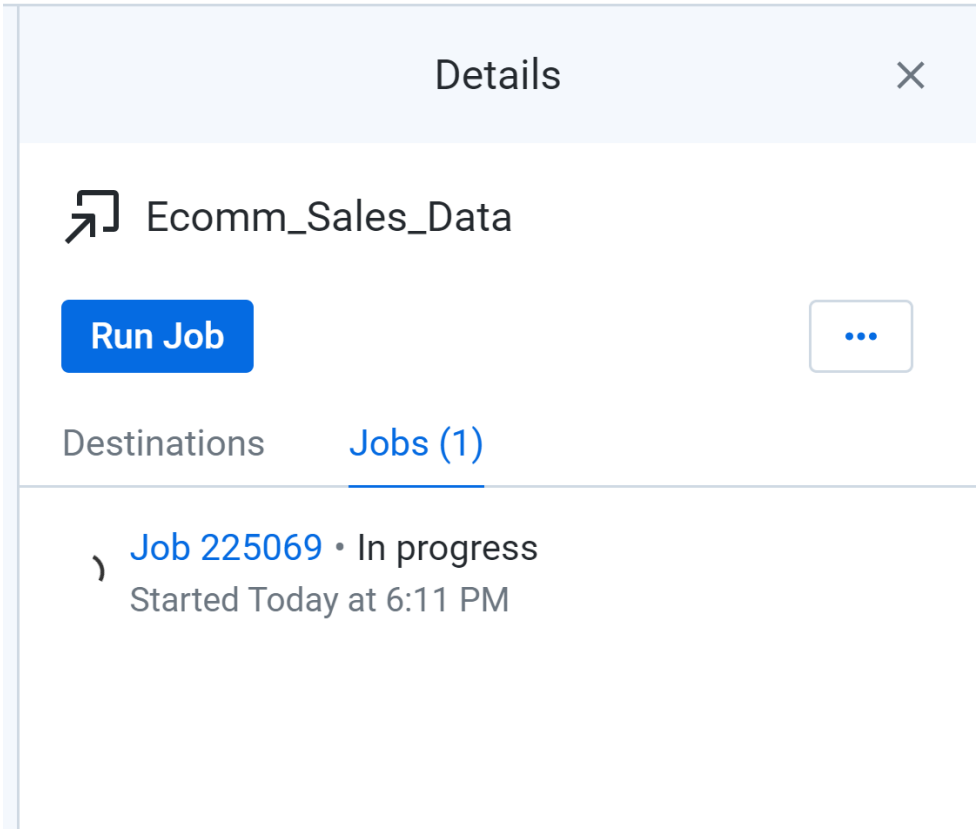


You can generate json or csv file as well for your output

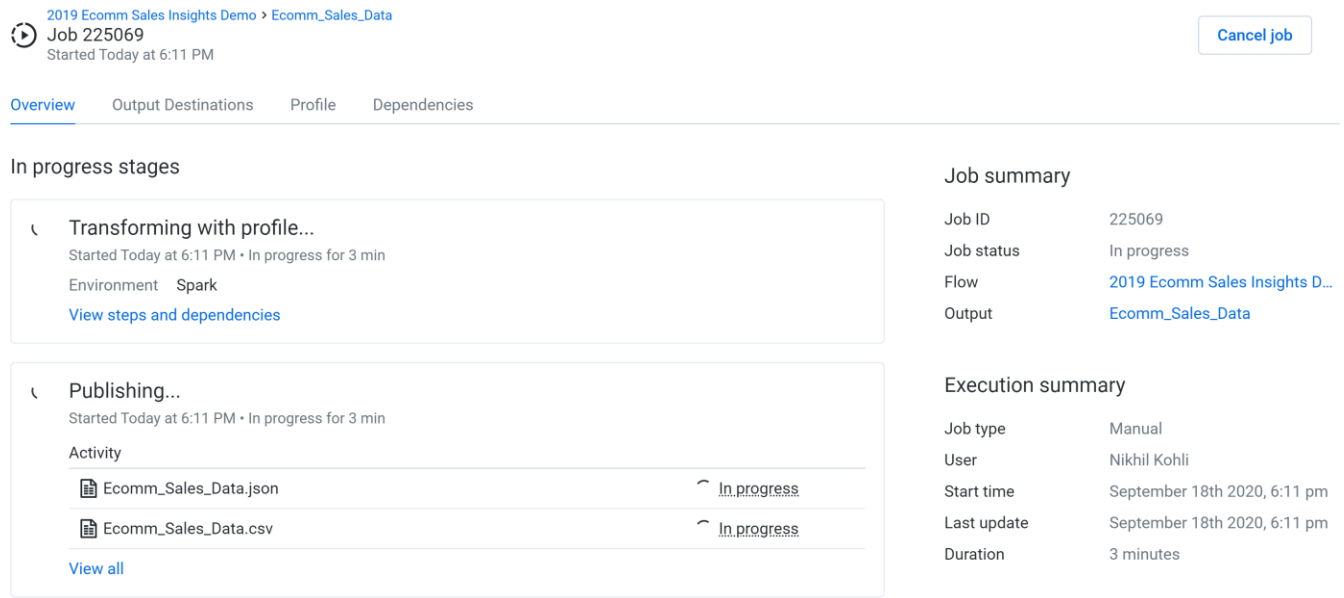


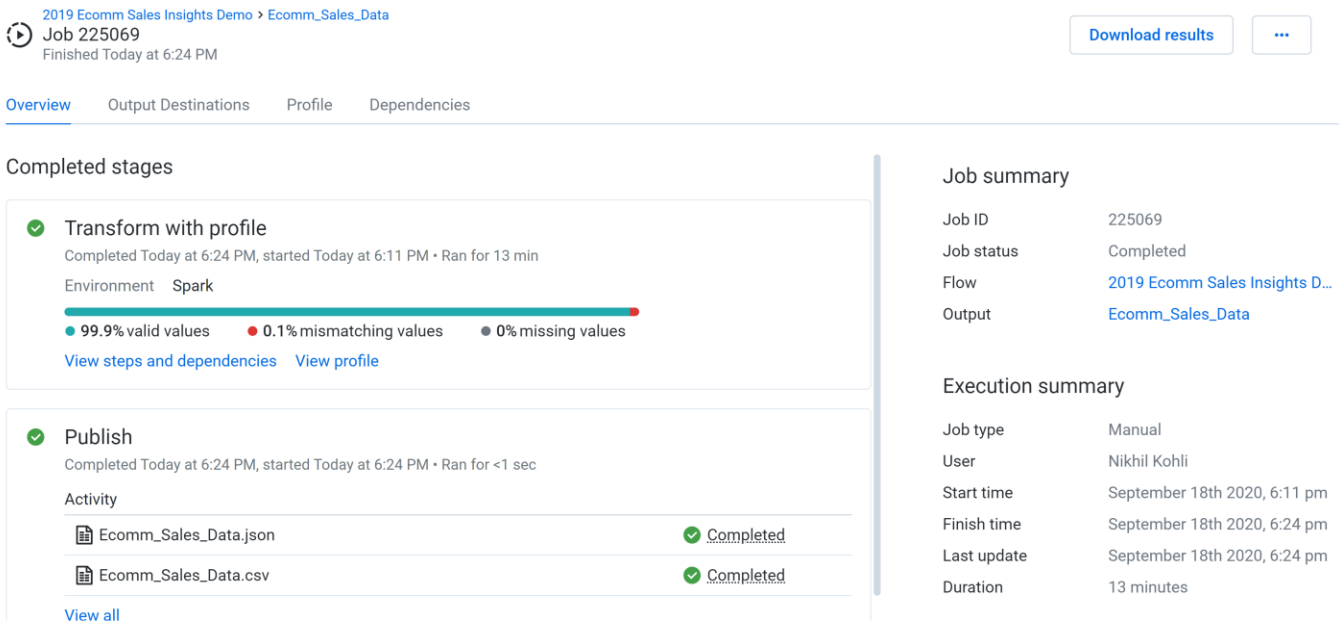
Click on run job





Click on that job id





Click on Download results and it will download the json and csv for output

Other Transformations that you can apply

1. One hot encoding:

Source	Preview		
# IsPenalty	# 0	# 1	# IsPenaltyAccepte_New
0 - 1	0 - 1	0 - 1	0 - 1
0	1	0	null
0	1	0	null
0	1	0	null
0	1	0	null
0	1	0	null
0	1	0	null
0	1	0	null
0	1	0	null
0	1	0	null
0	1	0	null
0	1	0	null
0	1	0	null
0	1	0	null

One-hot encode values to columns

Column required

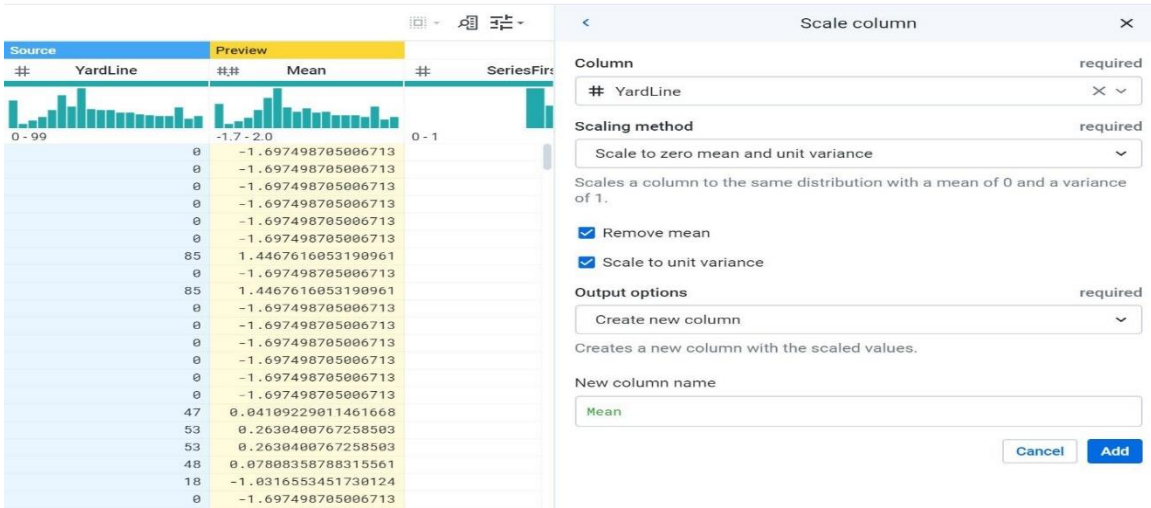
IsPenalty

Max number of columns to create required

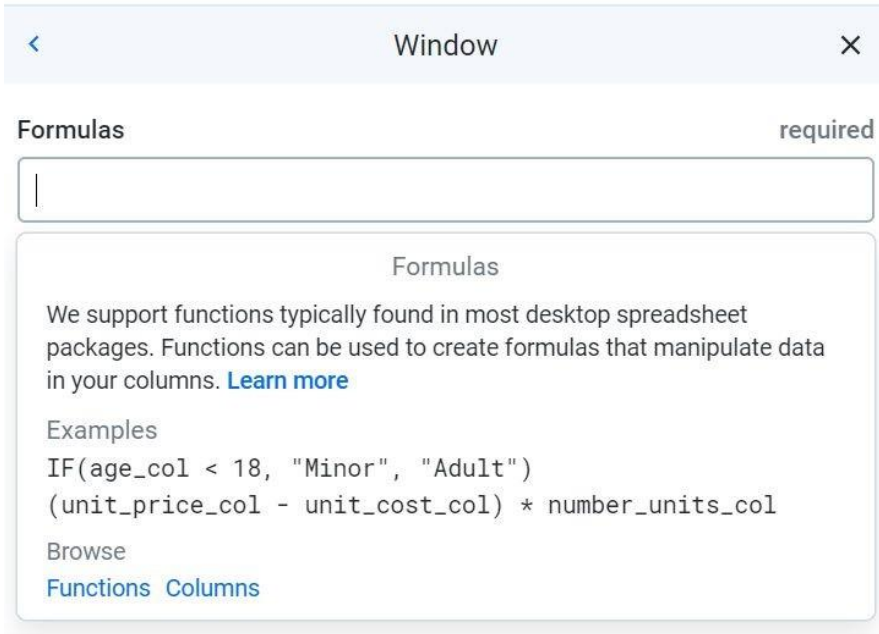
2

Cancel Add

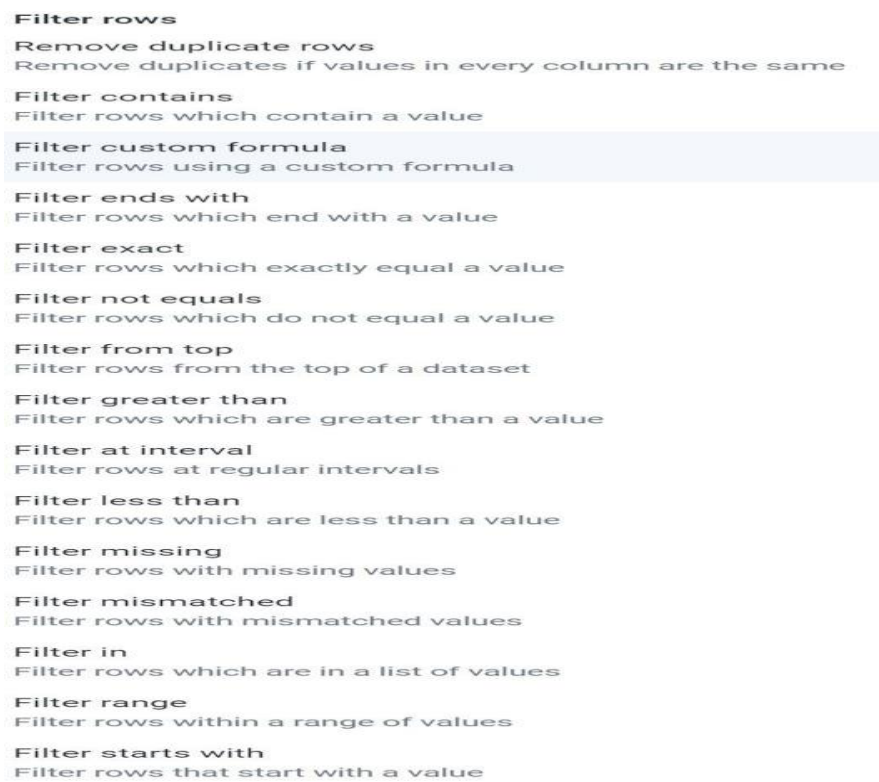
2. Scale Column:



3. Window:



4. Filtering Rows:



5. Functions:

MODE
(Function) Returns the most frequent value for each group. If multiple values ...

NULL
(Function) Returns a null value

NUMFORMAT
(Function) Converts the value into a custom number format

PAD
(Function) Pads a value to a specified length

PARSEDATE
(Function) Parses a string into a datetime object

PI
(Function) Returns the value of PI to fifteen decimal places

PROPER
(Function) Converts a string to Propercase by capitilizing the first letter of eac...

RADIANS
(Function) Converts degrees into radians

RAND
(Function) Returns a random number between 0 and 1

RANDBETWEEN
(Function) Returns a random integer between two specified integers, inclusive

RANGE
(Function) Returns an array of integers sequenced between two values by a st...

REMOVESYMBOLS
(Function) Removes all characters that are not alphanumeric or whitespace fr...

REMOVEWHITESPACE
(Function) Removes all whitespace characters from a string

REPEAT
(Function) Repeats a string a specified number of times

RIGHT
(Function) Returns a sub-string from the end of a string

RIGHTFIND
(Function) Returns the position at which a string or a pattern is last found with...

ROUND
(Function) Rounds the value to the specified decimal place

SIN
(Function) Returns the sine of an angle provided in radians

SINH
(Function) Returns the hyperbolic sine of the value provided

References

https://docs.trifacta.com/display/SS/Documentation