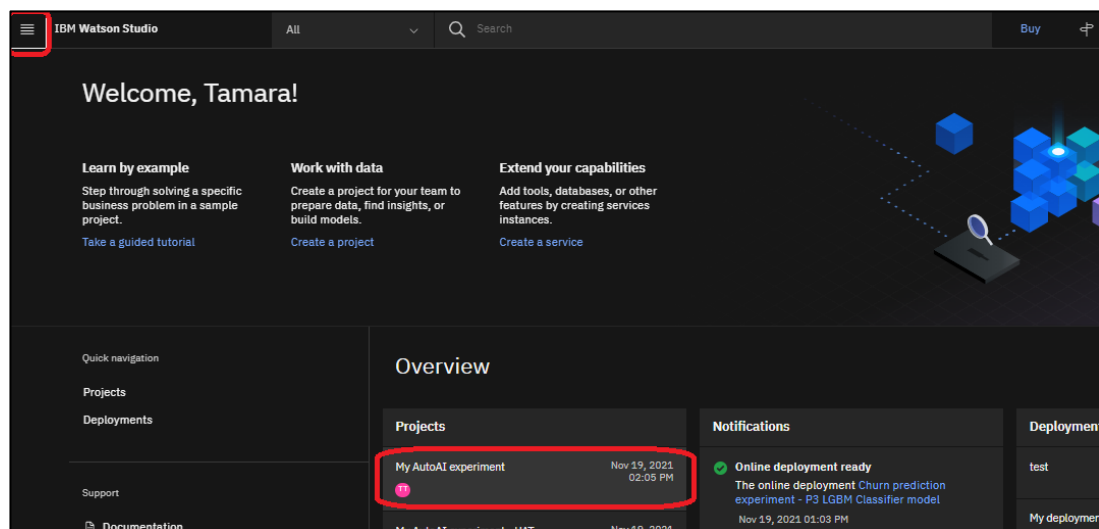


# Exploring and preparing data with Data Refinery

Log in to your CPDaaS account: <https://eu-de.dataplatform.cloud.ibm.com/>

Navigate to your My AutoAI Experiment project you created as part of the pre-work – you can do that from the home page tile, or, alternatively – by going through the main menu (top left of the screen).

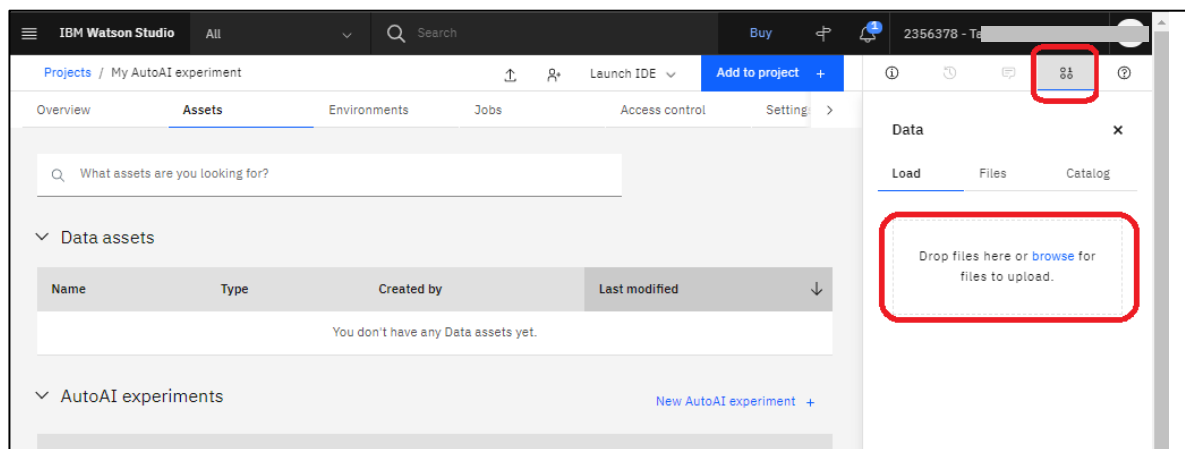
A **Project** is a collaborative workspace where you work with data and other assets to achieve a particular goal. Your project resources can include data, collaborators, tools, and operational assets that run code, like notebooks and models. Projects allow you to work with different analytical tools and IDEs built into the platform, and will spin up and run various runtimes, environments and jobs as/where needed.



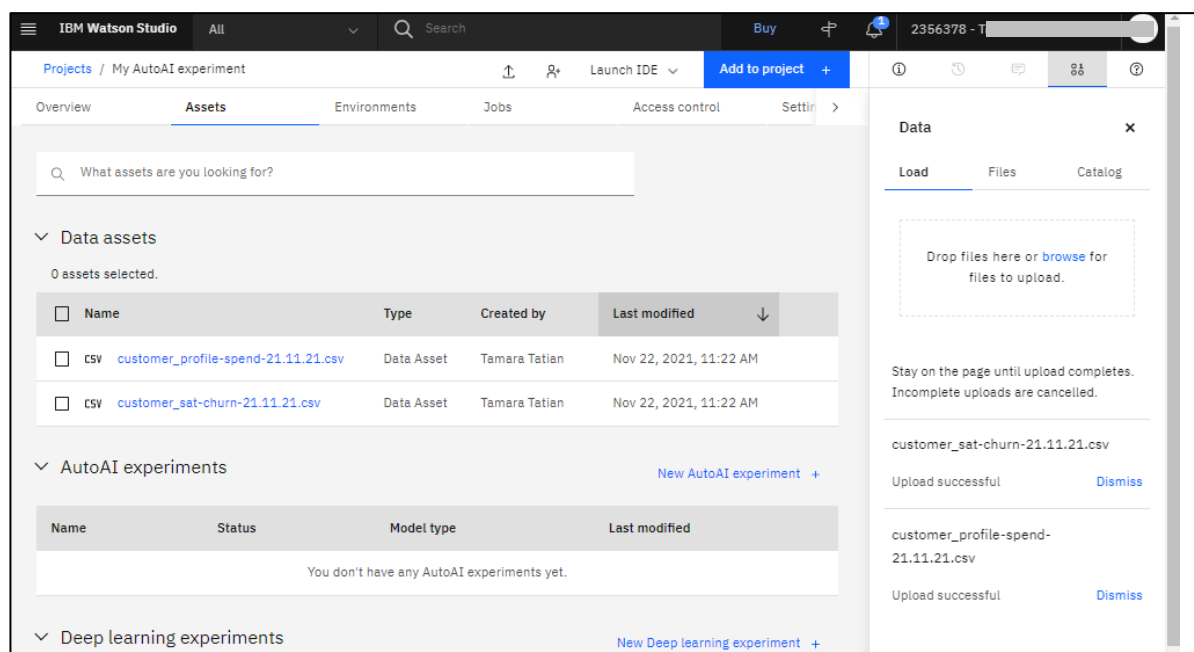
Download the .zip file from the following folder: <https://github.ibm.com/Hendrik-Loeffel/End-to-End-Data-Science-for-Business-Users/tree/main/DataRefinery>

And extract it on your local machine. You should see two .csv files, namely customer\_sat-churn-21.11.21.csv and customer\_profile-spend-21.11.21.csv.

Drag and drop both csv files into the “Drop files here” box under Data – Load

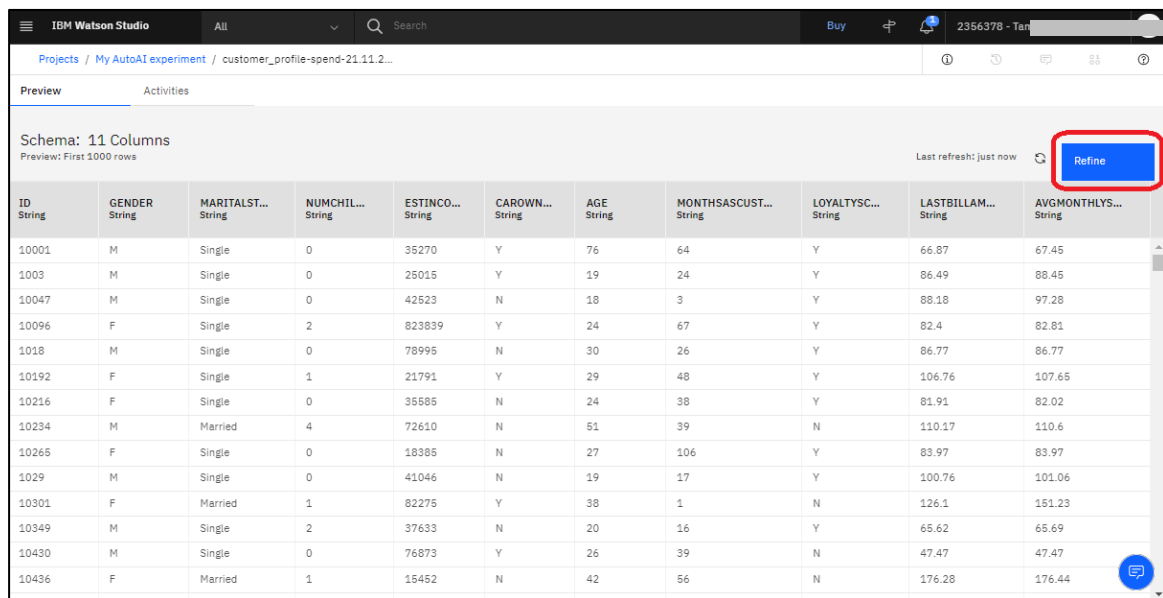


Once the upload finishes, navigate to the customer\_profile-spend-21.11.21.csv by clicking the corresponding link in the Data Assets section



Cloud Pak for Data allows you to preview the data in your data asset. This applies to both files that you physically load to projects and catalogues, and to “Connected assets” – files and tables residing in remote data sources (that you can connect to Cloud Pak for Data using a wide range of standard connectors through “Connections”). In this lab, we will be working with uploaded project files only.

Let's explore the data further and do some data wrangling. Click the Refine button



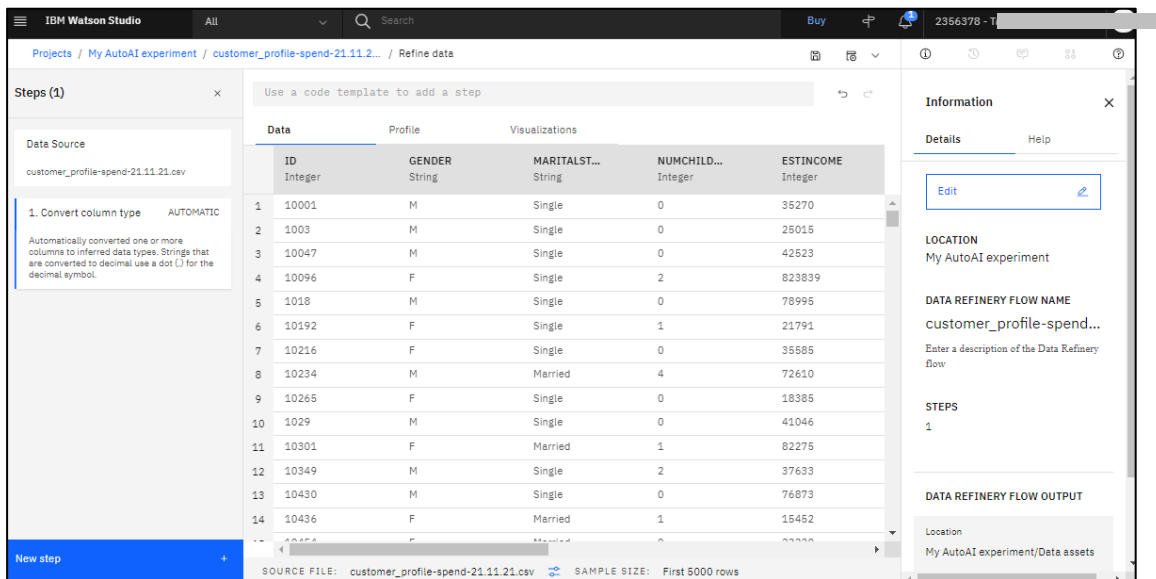
The screenshot shows the IBM Watson Studio interface. At the top, there's a navigation bar with 'IBM Watson Studio', a search bar, and a 'Buy' button. Below the navigation bar, the breadcrumb path is 'Projects / My AutoAI experiment / customer\_profile-spend-21.11.2...'. The main area is divided into 'Preview' and 'Activities' tabs. The 'Preview' tab is active, showing a table with 11 columns and 15 rows of data. The columns are: ID (String), GENDER (String), MARITALST... (String), NUMCHIL... (String), ESTINCO... (String), CAROWN... (String), AGE (String), MONTHSASCUST... (String), LOYALTYS... (String), LASTBILLAM... (String), and AVGMONTHLYS... (String). The 'Refine' button is highlighted in a red box in the top right corner of the table area.

ID String	GENDER String	MARITALST... String	NUMCHIL... String	ESTINCO... String	CAROWN... String	AGE String	MONTHSASCUST... String	LOYALTYS... String	LASTBILLAM... String	AVGMONTHLYS... String
10001	M	Single	0	35270	Y	76	64	Y	66.87	67.45
1003	M	Single	0	25015	Y	19	24	Y	86.49	88.45
10047	M	Single	0	42523	N	18	3	Y	88.18	97.28
10096	F	Single	2	823839	Y	24	67	Y	82.4	82.81
1018	M	Single	0	78995	N	30	26	Y	86.77	86.77
10192	F	Single	1	21791	Y	29	48	Y	106.76	107.65
10216	F	Single	0	35585	N	24	38	Y	81.91	82.02
10234	M	Married	4	72610	N	51	39	N	110.17	110.6
10265	F	Single	0	18385	N	27	106	Y	83.97	83.97
1029	M	Single	0	41046	N	19	17	Y	100.76	101.06
10301	F	Married	1	82275	Y	38	1	N	126.1	151.23
10349	M	Single	2	37633	N	20	16	Y	65.62	65.69
10430	M	Single	0	76873	Y	26	39	N	47.47	47.47
10436	F	Married	1	15452	N	42	56	N	176.28	176.44

The platform will fire up a Data Refinery instance (sandbox) for your file.

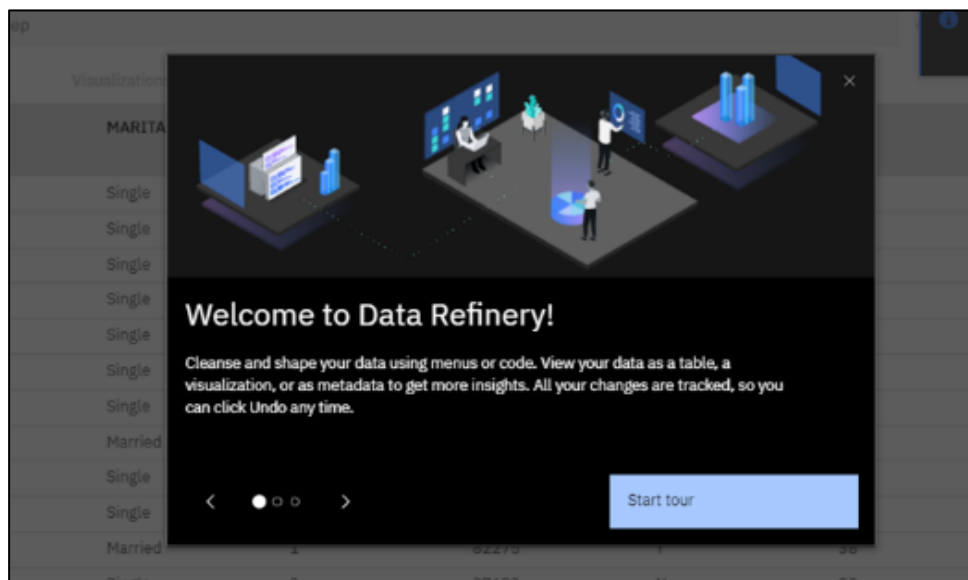
[Data Refinery](#) is a built-in data wrangling and data preparation tool available in Watson Studio. It helps reduce the amount of time it takes to prepare data for analysis and data science and allows you to cleanse and shape tabular data with a graphical flow editor. You can also use interactive templates to code operations, functions, and logical operators (R code is used). With Data Refinery, you can:

- Interactively discover, cleanse, and transform your data with over 100 built-in operations. No coding is required.
- Understand the quality and distribution of your data using dozens of built-in charts, graphs, and statistics.
- Automatically detect data types and business classifications.
- Schedule data flow executions for repeatable outcomes.



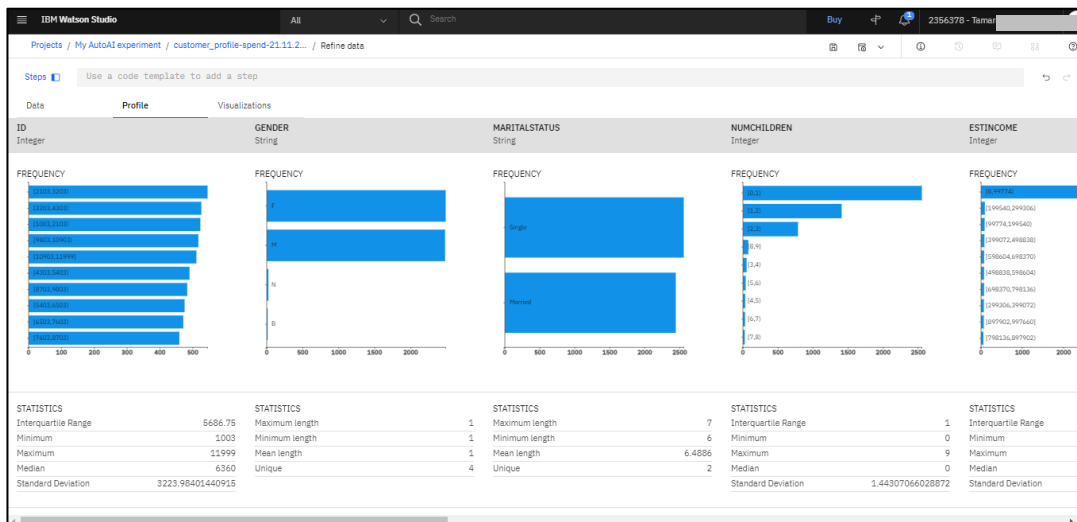
In your Data Refinery sandbox you can explore your data, design data wrangling and cleansing ‘recipes’ (flows) that you can further save and execute as jobs. While you are building your data wrangling recipes, all the transformations and changes are effectively performed in-memory and actual data is not touched at that point. It is only once you choose to execute your flow by running a Data Refinery job that the actual data will be transformed and changed.

On first load, the system may offer you to take a Tour – please feel free to explore or skip it.



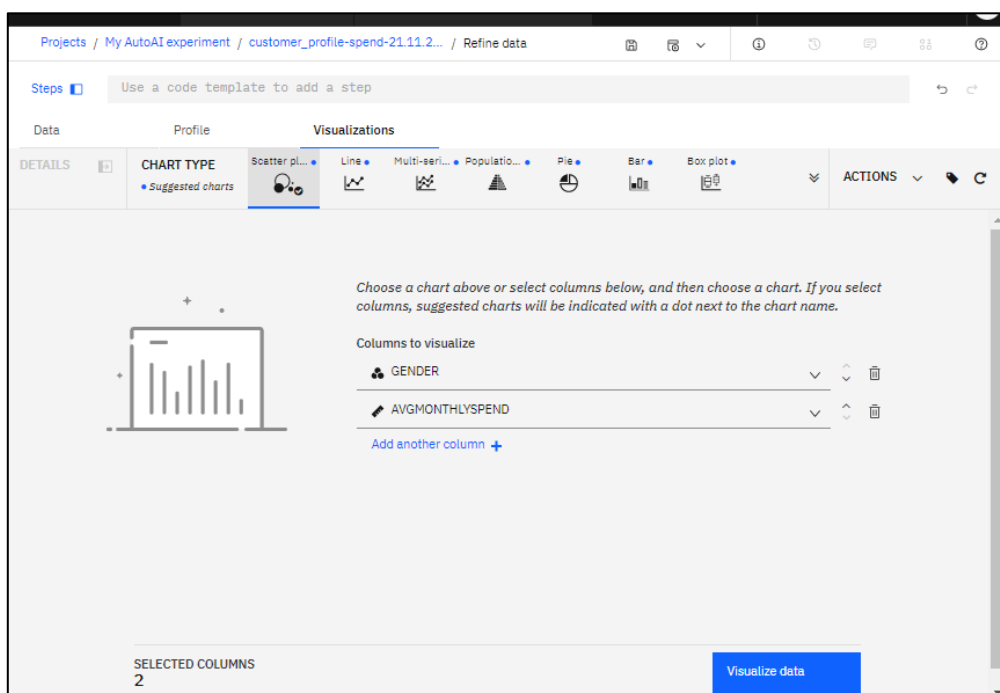
First, let’s explore our data. Minimise the Information and Steps panes on the screen (click X in those sections). Click on the Profile tab above the Gender column title.

The Profile tab shows you statistics and frequency analysis for each of the columns in your data set.

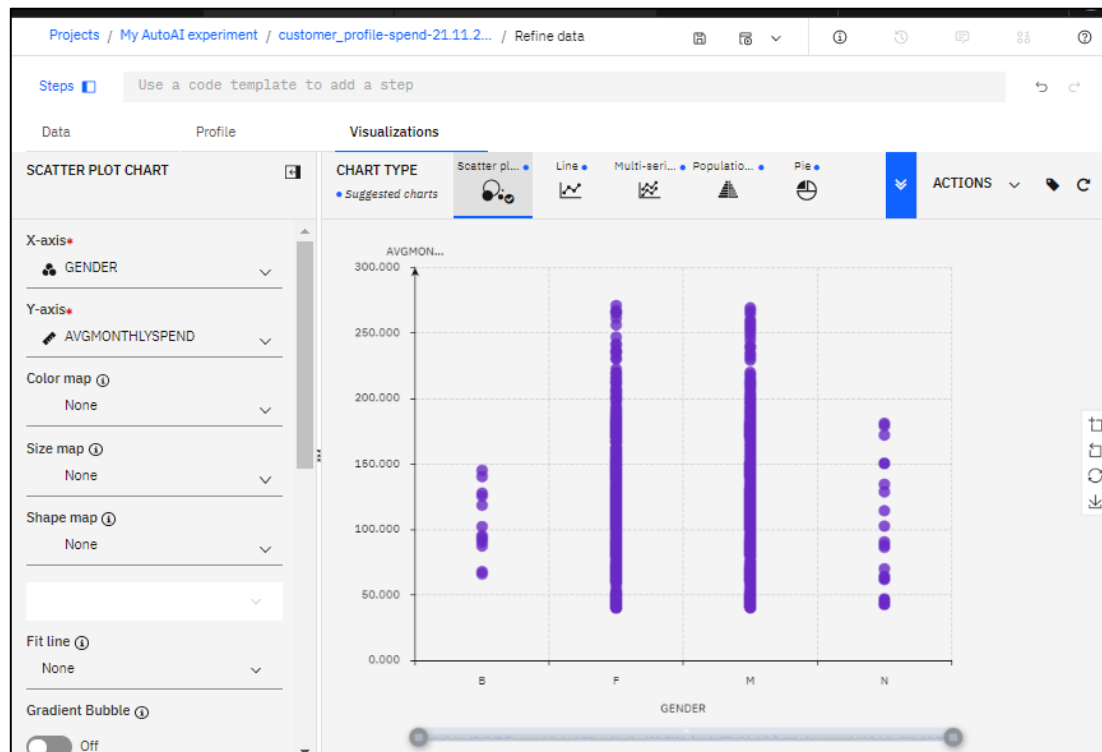


Next, let's visualize the data to try to explore and understand it a bit more. Navigate to the Visualizations tab above the Gender column title. Select GENDER as your first column, and add AVGMONTHLYSPEND as your second one, then click the Visualize Data button.

Refinery will automatically suggest the best fit type of visualization based on the data and number of columns that you choose.

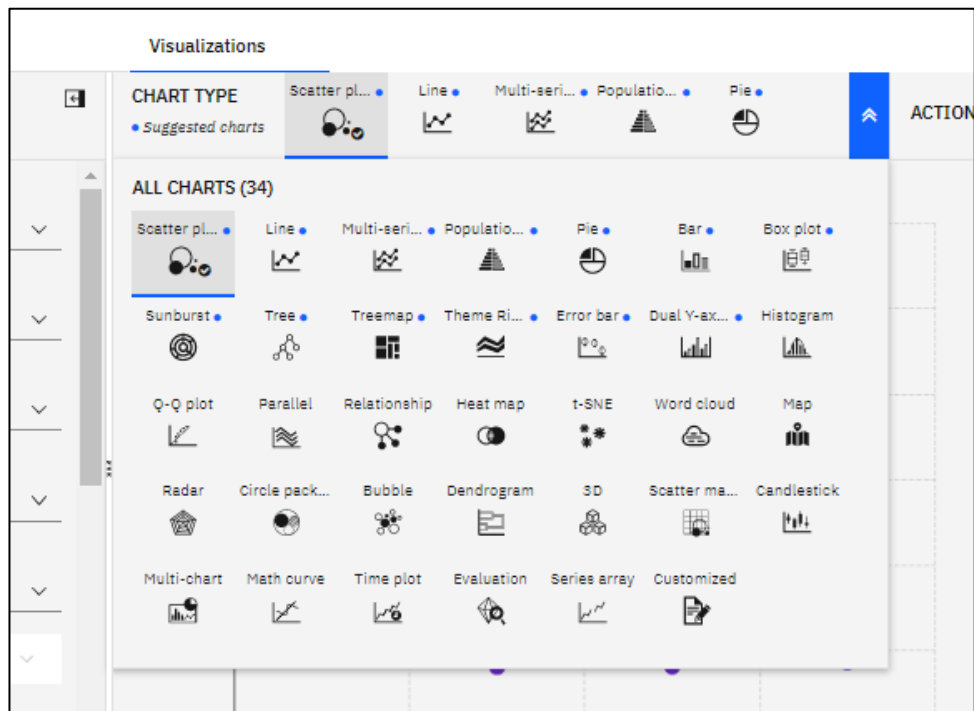


In our case, it picked Scatter plot.

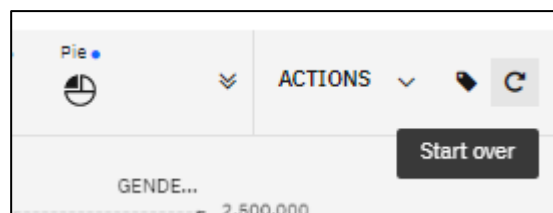


Note that the visualization highlighted to us that in our data set, our customers' gender data contains not only the more typical F and M values, but also B and N. This may warrant further investigation – there may be issue with data quality, or those could be legitimately valid values, depending on our company's data governance and capture policies and rules.

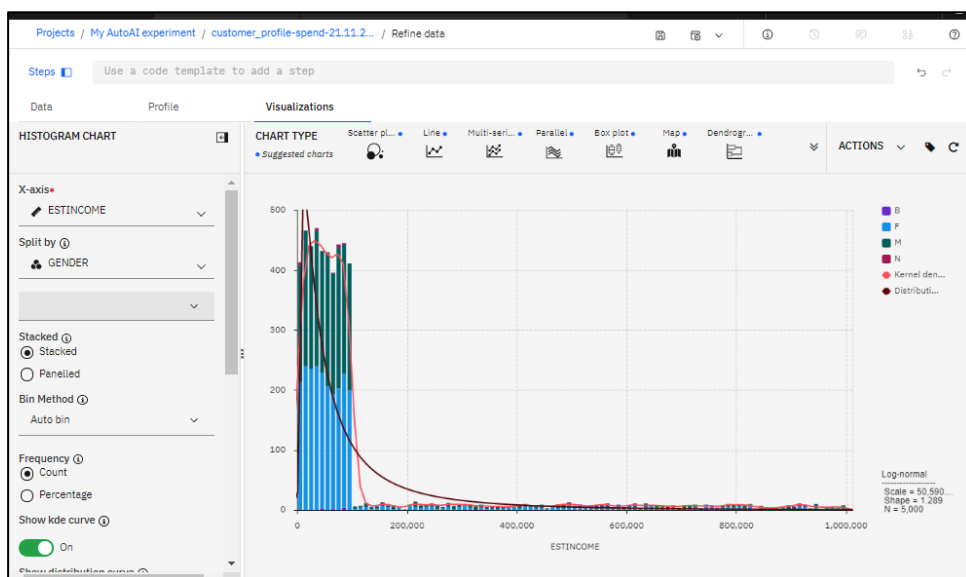
Expand the Chart Type menu by clicking on the chevron button. Note that the most suitable chart types are marked with a blue dot next to them. Feel free to switch between them and explore.

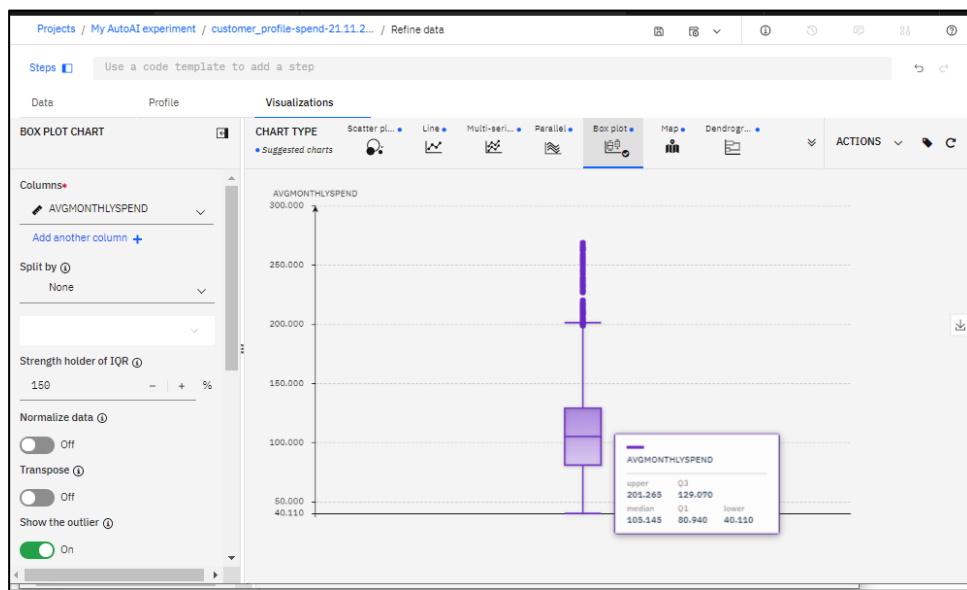


If you wanted to reset the visualization and start over, reset the chart by clicking the Start Over button. Note that you can also save your visualizations as an image.



Some more examples of visualizations built using our data set:





Switch back to the Data tab. We are going to address the GENDER data quality issue, data type/quality issues with some of the columns, create a new feature in our data called TOTALSPEND, and join our data set to the other csv from our project.

Expand the Steps pane.

Projects / My AutoAI experiment / customer\_profile-spend-21.11.2... / Refine data

Steps [ ] Use a code template to add a step

Data Profile Visualizations

	ID Integer	GENDER String	MARITALST... String	NUMCHILD... Integer	ESTINCOME Integer	CAROWNER String	AGE Integer	MONTHSAS... Integer
1	10001	M	Single	0	35270	Y	76	64
2	1003	M	Single	0	25015	Y	19	24
3	10047	M	Single	0	42523	N	18	3
4	10096	F	Single	2	823839	Y	24	67
5	1018	M	Single	0	78995	N	30	26
6	10192	F	Single	1	21791	Y	29	48
7	10216	F	Single	0	35585	N	24	38
8	10234	M	Married	4	72610	N	51	39
9	10265	F	Single	0	18385	N	27	106
10	1029	M	Single	0	41046	N	19	17
11	10301	F	Married	1	82275	Y	38	1
12	10349	M	Single	2	37633	N	20	16
13	10430	M	Single	0	76873	Y	26	39
14	10436	F	Married	1	15452	N	42	56
15	10454	F	Married	0	23220	Y	40	47

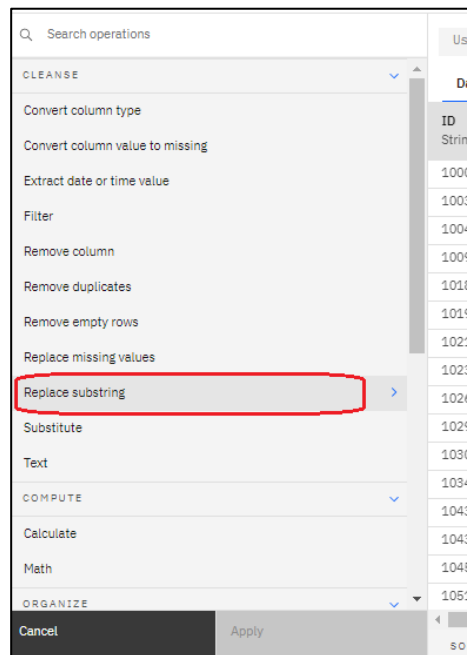


First, let's get rid of the automated transformation step the tool did for us – Data Refinery can autoconvert column types to the best fit/most suitable ones based on the data they contain. It can prove useful, but because in our case we wanted to join the data set to another one by the ID feature later on, we need to make sure the data types of the ID column match in both of our CSVs – so the original type String would work best for us. Click on the Bin icon next to step – this will remove it. Please note that you can remove any steps we build later on the same way – e.g. if you make a mistake or decide you no longer need them.

The screenshot shows the Data Refinery interface. On the left, under 'Steps (1)', there is a step named 'Convert column type' which is highlighted with a red box. Next to this step is a bin icon. The main area displays a data table with columns: ID (Integer), GENDER (String), MARITALST... (String), NUMCHILD... (Integer), ESTINCOME (Integer), and CAROWNER (String). The table contains 16 rows of data. At the bottom, it shows 'SOURCE FILE: customer\_profile-spend-21.11.21.csv' and 'SAMPLE SIZE: First 5000 rows'.

	ID Integer	GENDER String	MARITALST... String	NUMCHILD... Integer	ESTINCOME Integer	CAROWNER String
1	10001	M	Single	0	35270	Y
2	1003	M	Single	0	25015	Y
3	10047	M	Single	0	42523	N
4	10096	F	Single	2	823839	Y
5	1018	M	Single	0	78995	N
6	10192	F	Single	1	21791	Y
7	10216	F	Single	0	35585	N
8	10234	M	Married	4	72610	N
9	10265	F	Single	0	18385	N
10	1029	M	Single	0	41046	N
11	10301	F	Married	1	82275	Y
12	10349	M	Single	2	37633	N
13	10430	M	Single	0	76873	Y
14	10436	F	Married	1	15452	N
15	10454	F	Married	0	23220	Y
16	10518	M	Single	1	70863	Y

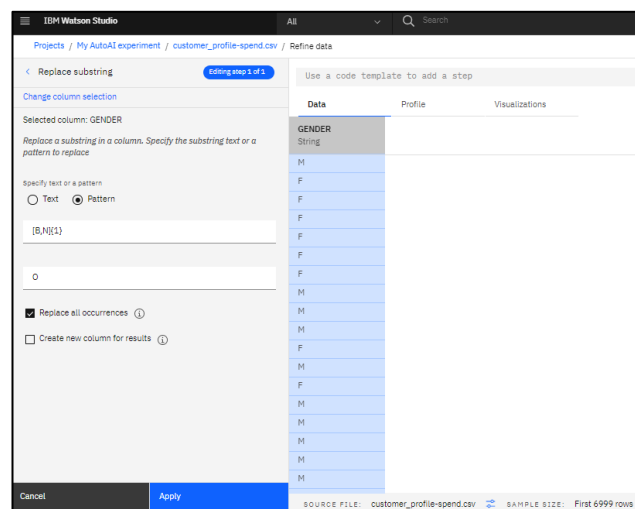
Select Replace Substring from the menu – we are going to replace our B and N entries in the GENDER column with a single new gender type of O, as we happen to know that our company's data capture rules allow for "Other/Prefer Not to Say" option in addition to F (female) and M (male).



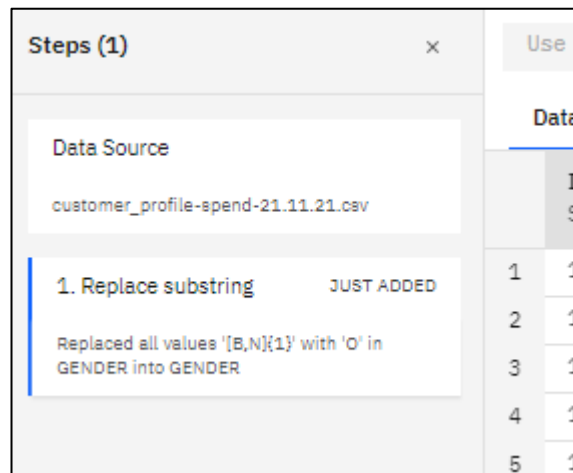
Select GENDER, click Next, switch to Pattern recognition on the next screen and enter the following values:

**Regular expression: [B,N]{1}**

**Replacement string: O**



Click Apply – you now have a new step in your data preparation flow.



Next, we are going to convert several columns with numerical data to more appropriate data formats. Click on the three dots icon next to NUMCHILDREN column's title, select Convert Column Type > Integer.

Use a code template to add a step

	Data	Profile	Visualizations				
	ID String	GENDER String	MARITALST... String	NUMCHILD... String	ESTINCOME ...	CAROWNER String	AGE Str...
1	10001	M	Single	0	Remove column	Y	76
2	1003	M	Single	0	Remove duplicates	Y	19
3	10047	M	Single	0	Remove empty rows	N	18
4	10096	F	Single	2	Sort ascending	Y	24
5	1018	M	Single	0	Sort descending	N	30
6	10192	F	Single	1	Substitute	Y	29
7	10216	F	Single	0	CONVERT CO... >	Boolean	14
8	10234	M	Married	4	TEXT >	Date	11
9	10265	F	Single	0	View All	Decimal	17
10	1029	M	Single	0		Integer	9
11	10301	F	Married	1	82275	String Integer	18
12	10349	M	Single	2	97633	Timestamp	10
13	10430	M	Single	0	76873		16
14	10436	F	Married	1	15452	N	42
15	10464	F	Married	0	23220	Y	40
16	10618	M	Single	1	70863	Y	29
17	10635	F	Single	1	9570	Y	21

Add more columns on the next screen - click Select Column and add the following conversions:

ESTINCOME – Integer

AGE – Integer

MONTHSASCUSTOMER – Integer

LASTBILLAMOUNT – Decimal

AVGMONTHLYSPEND - Decimal



Enter the following on the next screen:

Calculation type: Multiplication

By: Column – MONTHSASCUSTOMER

Create new column for results checkbox ticked (yes)

Column name: TOTALSPEND

Click Apply.

Projects / My AutoAI experiment / customer\_profile-spend-21.11.2... / Refine data

< Calculate

Use a code template

Change column selection

Selected column: AVGMONTHLYSPEND

Apply a calculation with another column or a value. Overwrite the existing column or create a new column for the results.

Multiplication

Specify value or a column

☐ Value ☒ Column

MONTHSASCUSTOMER

Showing columns with supported data types.

☒ Create new column for results ⓘ

TOTALSPEND

Cancel Apply

SOURCE FILE: custo

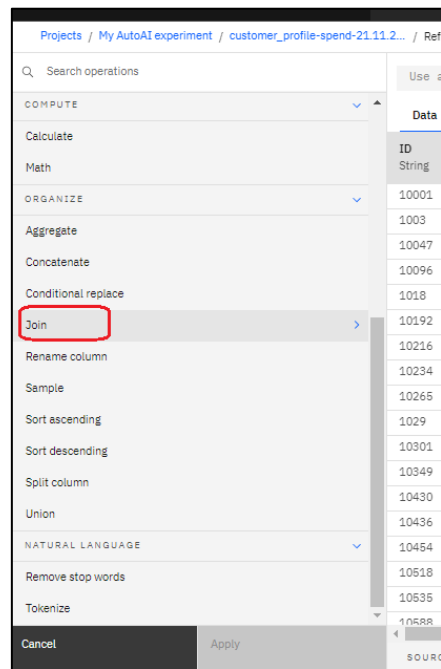
AVGMONTH...
67.45
88.45
97.28
82.81
86.77
107.65
82.02
110.6
83.97
101.06
151.23
65.69
47.47
176.44
87.42
71.08
46.99
88.64

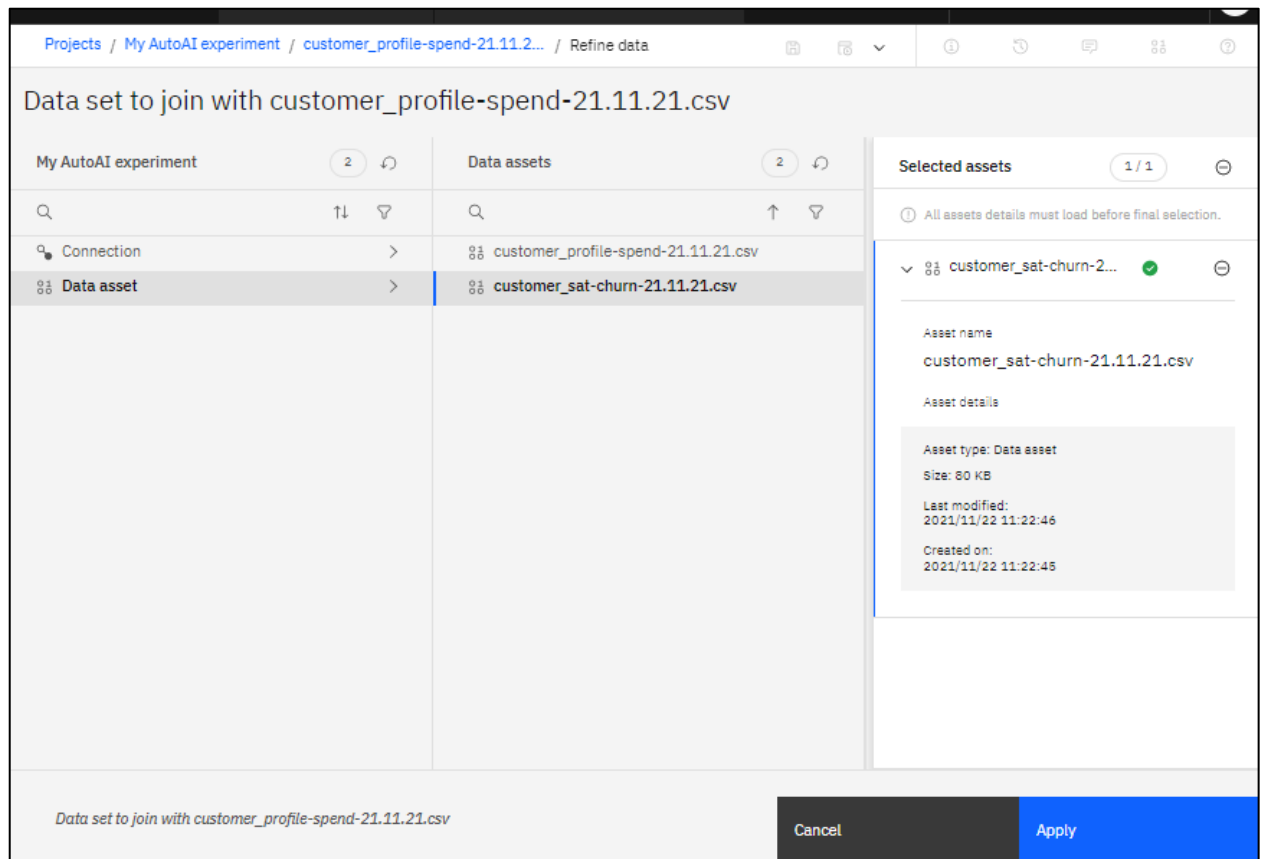
Preview now shows:

AVGMONTH...	TOTALSP...
Decimal	Decimal
67.45	4316.8
88.45	2122.8
97.28	291.84
82.81	5548.27
86.77	2256.02
107.65	5167.2
82.02	3116.76
110.6	4313.4
83.97	8900.82
101.06	1718.02
151.23	151.23
65.69	1051.04
47.47	1851.33
176.44	9880.64
87.42	4108.74
71.08	1563.76
46.99	93.98
88.64	5229.76

Finally, let's join our two csvs. We are going to be predicting customer CHURN and building a model for it later on – at the moment, our current data set does not include CHURN data. However, the other csv file we loaded does have it – so let's join them together.

Click Next Step – select Join from the menu.





Select ID as join keys for both the files (note that for the join to work both the columns you are using as join keys need to be of the same type – e.g. String and String, or Integer and Integer etc. – the tool will only let you pick columns of the same type once you specify your join key for the first data set). Click Next. On the next screen exclude CAROWNER and SUPPORTCALLSMONTH columns, then click Apply.

Projects / My AutoAI experiment / customer\_profile-spend-21.11.2... / Refine data

< Join

Returns all rows in the original data set and returns only matching rows in the joining data set. Returns one row in the original data set for each matching row in the joining data set.

The default suffix for each data set will be used to differentiate any duplicate column names in the resulting data set.

Source: customer\_profile-spend-21...  
\*Suffix: \_X

Data set to join: customer\_sat-churn-21...  
\*Suffix: \_Y

Enter unique suffixes to differentiate duplicate column names in the resulting data set

JOIN KEYS <sup>1</sup>

customer_profile-spend-21... (1/12)	customer_sat-churn-21.11.2... (1/6)
ID	ID

Add join key +

Cancel Next

Use a code template to add a step

Data	Profile	Visualizations	
ID	GENDER	MARITALST...	NUMCHILD...
String	String	String	Integer
10001	M	Single	0
1003	M	Single	0
10047	M	Single	0
10096	F	Single	2
1018	M	Single	0
10192	F	Single	1
10216	F	Single	0
10234	M	Married	4
10265	F	Single	0
1029	M	Single	0
10301	F	Married	1
10349	M	Single	2
10430	M	Single	0
10436	F	Married	1
10454	F	Married	0
10518	M	Single	1

SOURCE FILE: customer\_profile-spend-21.11.21.csv SAMPLE SIZE: First 5000 rows

< Join

☒ MARITALSTATUS

☒ NUMCHILDREN

☒ ESTINCOME

☐ CAROWNER

☒ AGE

☒ MONTHSASCUSTOMER

☒ LOYALTYScheme

☒ LASTBILLAMOUNT

☒ AVGMONTHLYSPEND

☒ TOTALSPEND

☒ SUPPORTCALLSYEAR

☐ SUPPORTCALLSMONTH

☒ COMPLAINTSYEAR

☒ COMPLAINTSMONTH

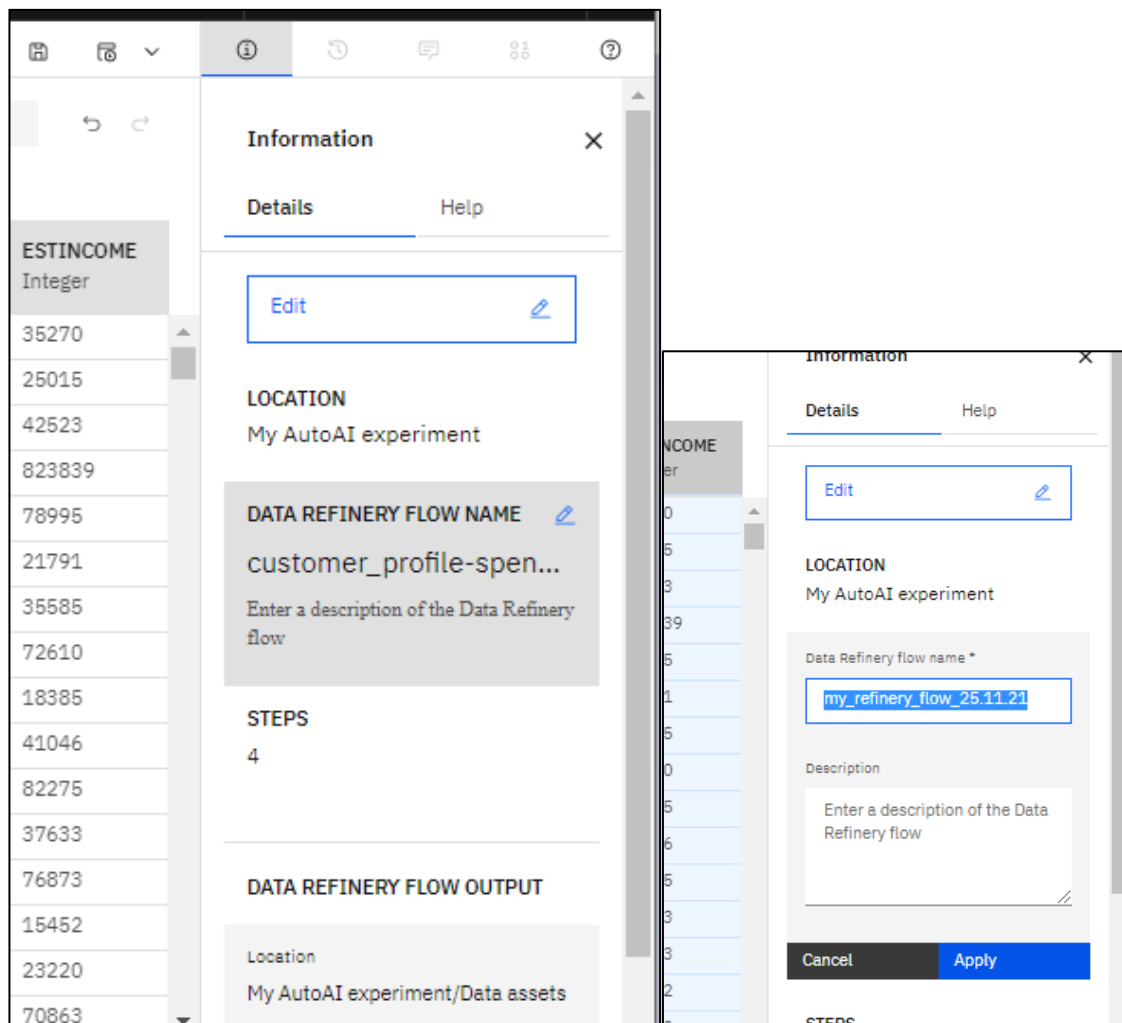
☒ CHURN

Back Apply

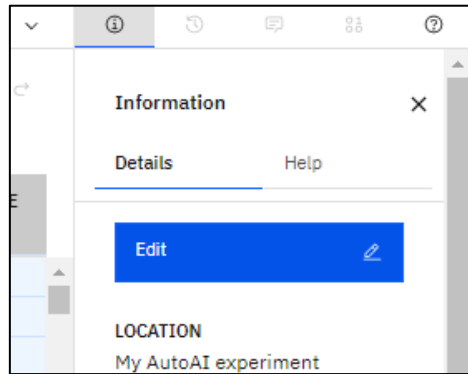


We finished building the data preparation and cleansing flow – let's now save it and run our shaping job.

First, let's give it a name and decide how and where we would want to save the output of our shaping flow. Click the **i** icon to expand the Information pane, then click on the pencil icon next to the data refinery flow name to edit it. Name your flow `my_refinery_flow_25.11.21`, then click Apply.



Next, let's check where and how Refinery is going to output the results of our data cleansing and shaping flow. Click the Edit button, then Edit Output on the next screen



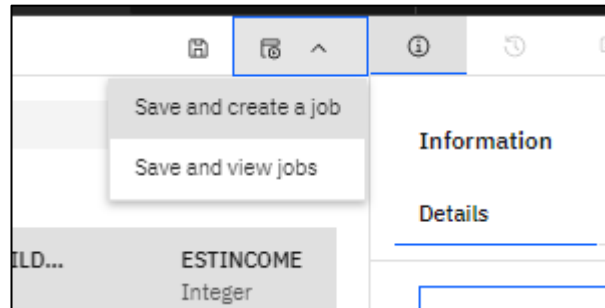
Name your data set customer-profile-churn-joined.csv

By default, Refinery will create a new csv file within your project (a new local data asset in the project). Please note that you can choose other file and format types if/where relevant, and choose to write the results to a connection (remote data source) as well – if you have any defined in the project. We are going to go with defaults - click the tickbox icon once you finish renaming the file, then click the Done button (bottom right of the screen)

Projects / My AutoAI experiment / customer\_profile-spend-21.11.2... / Refine data

DATA REFINERY FLOW DETAILS	DATA REFINERY FLOW OUTPUT
<p><b>LOCATION</b> My AutoAI experiment</p> <p><b>DATA REFINERY FLOW NAME</b> my_refinery_flow_25.11.21 <small>Enter a description of the Data Refinery flow</small></p> <p><b>STEPS</b> 4</p>	<div><div>Edit output × ✓</div><p>Location *</p><p>My AutoAI experiment/Data assets</p><p>Data set name *</p><p>customer-profile-churn-joined.csv</p><p>Description</p><p>Enter a description of the resulting data set.</p><p>File format</p><p>CSV</p><p>Encoding</p><p>UTF-8</p></div>

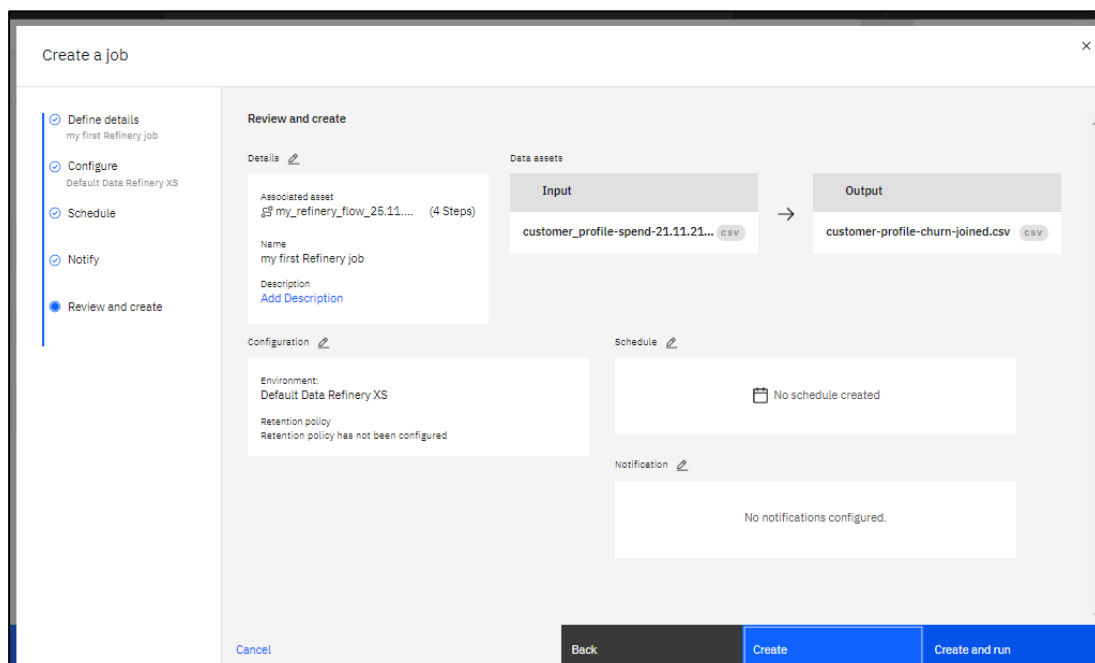
Now, let's create and run a job that will execute our flow. From the menu on the top right hand side, select Save and create a job.



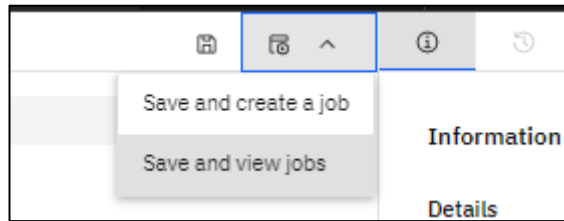
Give your job a name – e.g. “my first Refinery job”, click Next. Review and go with the defaults on the next screen. Note that if you have the Analytics Engine for Spark service deployed as part of your CPDaaS, you can select different environment / runtime configurations and sizes for your Refinery jobs (e.g. create and use larger specs for more complex and data/compute intensive jobs).

Next screen allows you to specify a schedule for your job. We are going to run it as a one-off, but feel free to explore the scheduling options. Disable the schedule before you move to the next screen.

On the Notifications screen click next, and then Create and Run on the last screen.



Select save and view jobs – then navigate to your newly created job



You can monitor its progress, as well as see the logs and execution details.

Projects / My AutoAI experiment / my first Refinery job

## Job Details

Overview

1 Runs Completed

0 Runs Failed

No schedule created

Edit Configuration

Find a job run

Last updated: 22/11/2021, 13:36

Start time	Status	Duration	Asset type
Nov 22, 2021 1:34:41 PM Started by Tamara Tatian	Completed	00:00:28	Data Refinery Flow

Items per page: 10 1-1 of 1 item

Projects / My AutoAI experiment / my first Refinery job / Job run details

## Nov 22, 2021 1:34:41 PM

### About this run

Completed

Run details

Duration (seconds): 28

Started by: Tamara Tatian

Associated job: my first Refinery job

Associated asset

my\_refinery\_flow\_25.11.21

Data assets

Input

Output

customer\_prof... CSV

customer-profi... CSV

Rows read

5000(168 KB)

Rows written

5000(278 KB)

Resource consumed: 0.04

Configuration

### Log

Total 98 lines

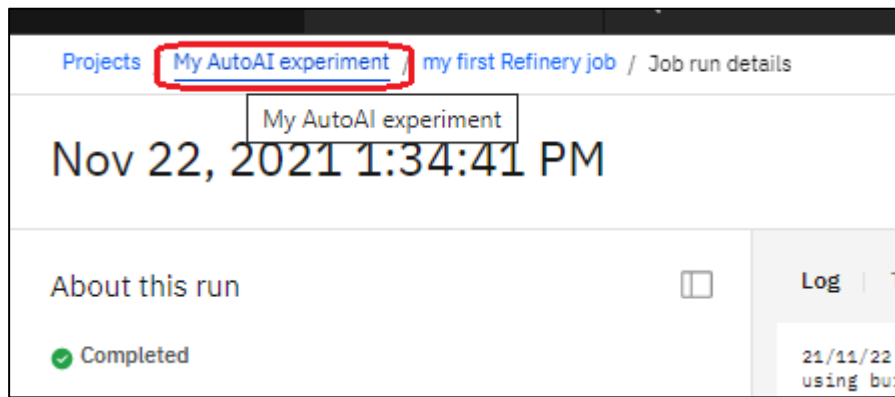
Download log

```

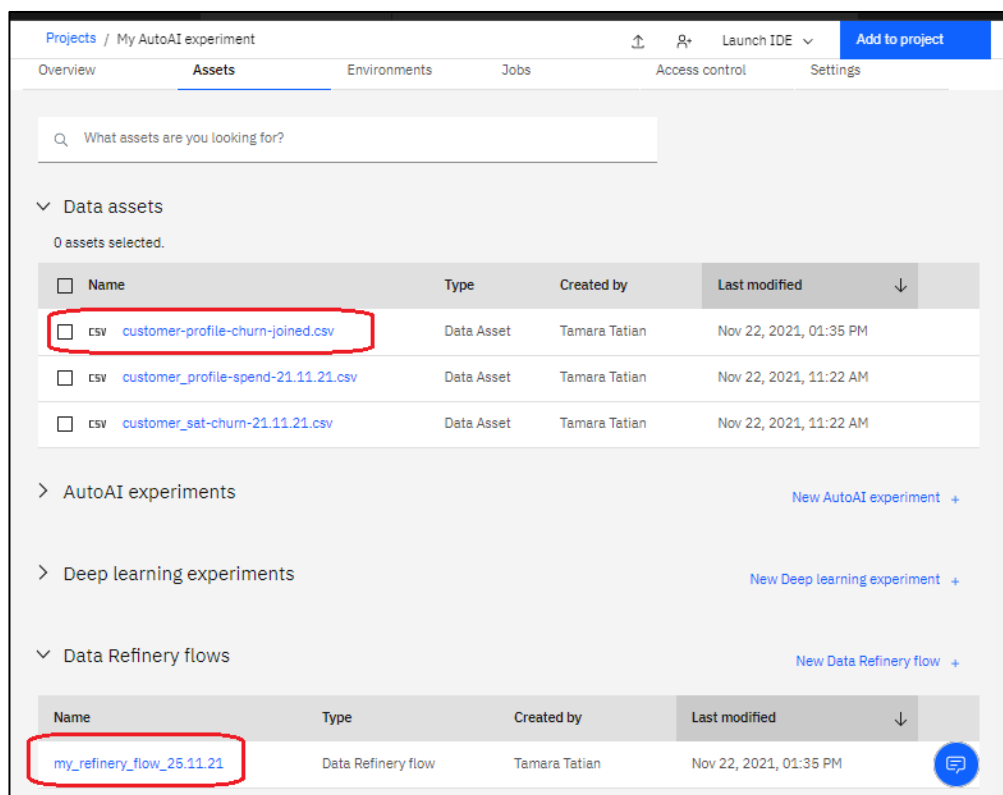
21/11/22 13:55:04 WARN NativeCodeLoader - Unable to load native-hadoop library for your platform...
using builtin-java classes where applicable
21/11/22 13:55:04 INFO SecurityManager - Changing view acls to: spark
21/11/22 13:55:04 INFO SecurityManager - Changing modify acls to: spark
21/11/22 13:55:04 INFO SecurityManager - Changing view acls groups to:
21/11/22 13:55:04 INFO SecurityManager - Changing modify acls groups to:
21/11/22 13:55:04 INFO SecurityManager - SecurityManager: authentication enabled; ui acls disabled;
users with view permissions: Set(spark); groups with view permissions: Set(); users with modify
permissions: Set(spark); groups with modify permissions: Set()
2021-11-22 13:55:05.150+0000 INFO: Job run ID: 27f69aa0-e80d-439a-92c0-97098c9b5f91
2021-11-22 13:55:05.173+0000 INFO: Loading libraries.
2021-11-22 13:55:05.200+0000 INFO: Running jsonlite v 1.7.2
2021-11-22 13:55:05.201+0000 INFO: Running dplyr v 0.8.2
2021-11-22 13:55:07.349+0000 INFO: Runtime environment: Spark
2021-11-22 13:55:07.356+0000 INFO: Initializing parameters for Data Refinery flow.
Spark package found in SPARK_HOME: /opt/lib/spark
21/11/22 13:55:08 INFO SparkContext - Running Spark version 3.0.2
21/11/22 13:55:08 INFO ResourceUtils - =====
21/11/22 13:55:08 INFO ResourceUtils - Resources for spark.driver:
21/11/22 13:55:08 INFO ResourceUtils - =====
21/11/22 13:55:08 INFO SparkContext - Submitted application: SparkR
21/11/22 13:55:08 INFO SecurityManager - Changing view acls to: spark
21/11/22 13:55:08 INFO SecurityManager - Changing modify acls to: spark
21/11/22 13:55:08 INFO SecurityManager - Changing view acls groups to:
21/11/22 13:55:08 INFO SecurityManager - Changing modify acls groups to:
21/11/22 13:55:08 INFO SecurityManager - SecurityManager: authentication enabled; ui acls disabled;
users with view permissions: Set(spark); groups with view permissions: Set(); users with modify
permissions: Set(spark); groups with modify permissions: Set()
21/11/22 13:55:08 INFO Utils - Successfully started service 'sparkDriver' on port 43947.
21/11/22 13:55:08 INFO SparkEnv - Registering MapOutputTracker
21/11/22 13:55:08 INFO SparkEnv - Registering BlockManagerMaster
21/11/22 13:55:08 INFO BlockManagerMasterEndpoint - Using
org.apache.spark.storage.DefaultTopologyMapper for getting topology information
21/11/22 13:55:08 INFO BlockManagerMasterEndpoint - BlockManagerMasterEndpoint up
21/11/22 13:55:08 INFO SparkEnv - Registering BlockManagerMasterHeartbeat
21/11/22 13:55:08 INFO DiskBlockManager - Created local directory at /tmp/spark/scratch/blockmgr-
27f69aa0-e80d-439a-92c0-97098c9b5f91

```

Navigate back to your project Assets view by clicking on the project name



You new data set and flow are now available – please feel free to preview the joined data set.



Projects / My AutoAI experiment

Launch IDE

Add to project

0 assets selected.

<input type="checkbox"/>	Name	Type	Created by	Last modified	
<input type="checkbox"/>	csv <a href="#">customer-profile-churn-joined.csv</a>	Data Asset	Tamara Tatian	Nov 22, 2021, 01:35 PM	<div>RefineDownloadPromoteRemove</div>
<input type="checkbox"/>	csv <a href="#">customer_profile-spend-21.11.21.csv</a>	Data Asset	Tamara Tatian	Nov 22, 2021, 01:35 PM	
<input type="checkbox"/>	csv <a href="#">customer_sat-churn-21.11.21.csv</a>	Data Asset	Tamara Tatian	Nov 22, 2021, 01:35 PM	

> AutoAI experiments

> Deep learning experiments

> Data Refinery flows

my\_refinery\_flow\_25.11.21

Data Refinery flow

Tamara Tatian

Nov 22, 2021, 01:35 PM

CloneCreate jobPromoteView jobRemove