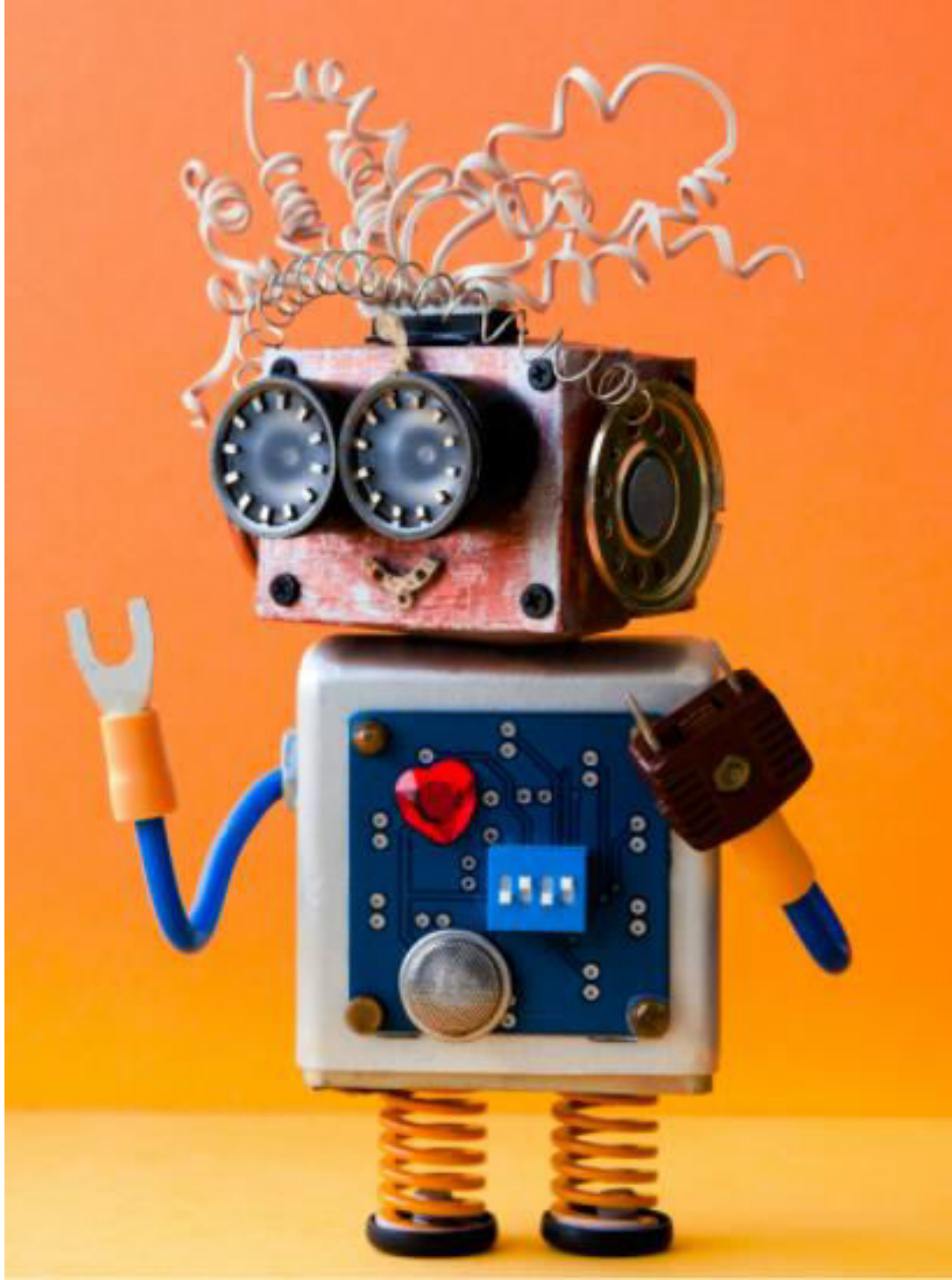
PROJECT OVERVIEW

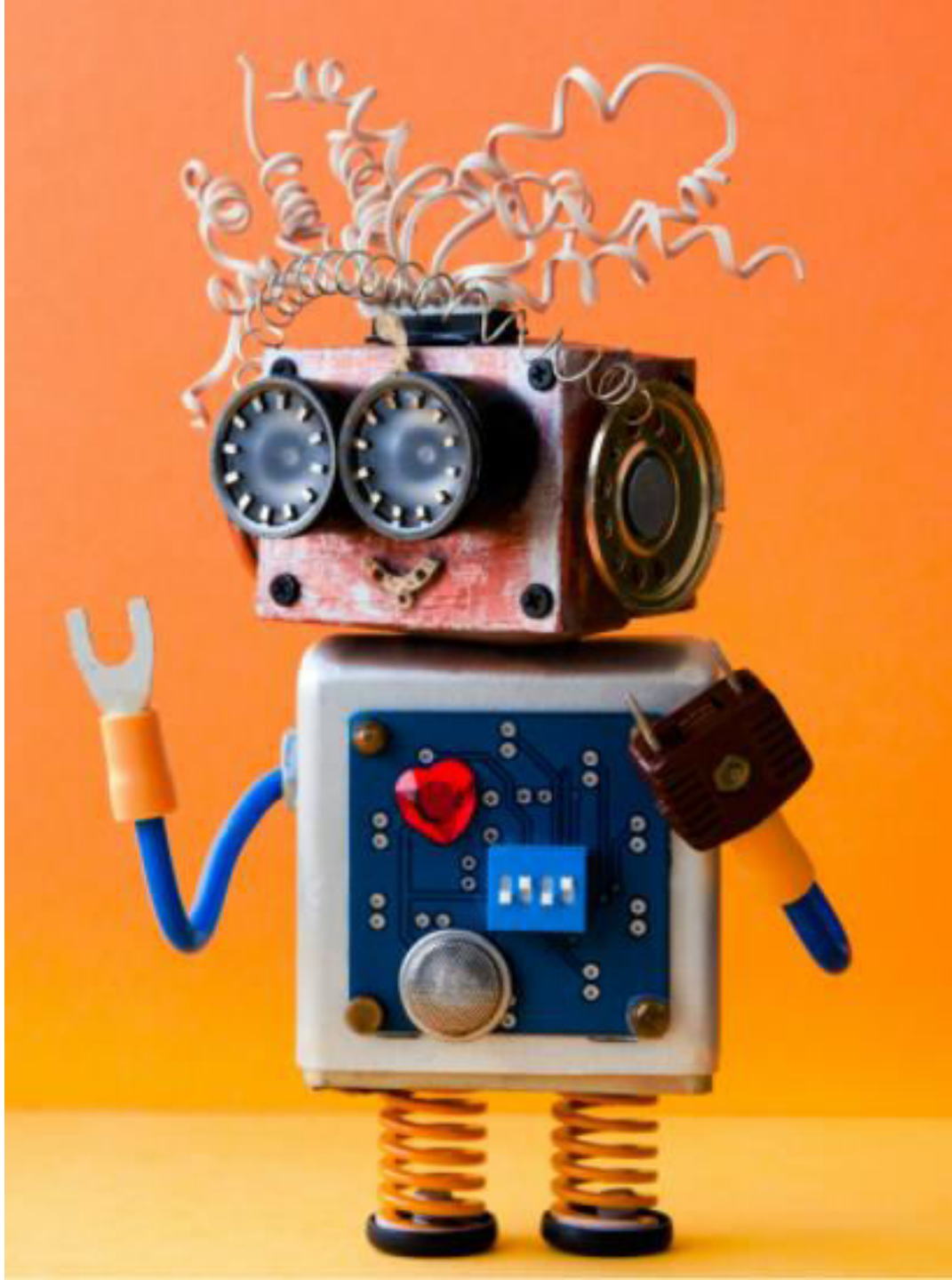
EASY ADVANCED



PROJECT OVERVIEW

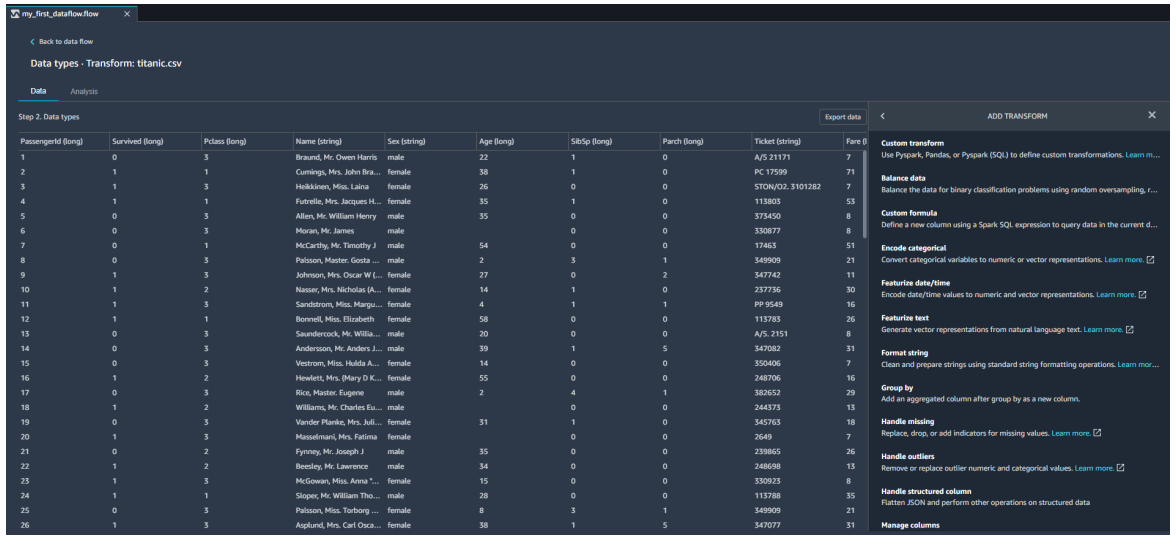
- In this project, we will leverage the power of data Wrangler service in AWS to prepare, clean and visualize the data.
- We will analyze the Titanic dataset which contains features related to Titanic passengers and cardiovascular disease datasets (final project).
- Here are the key learning outcomes:
  - Understand feature engineering strategies and tools.
  - Understand the fundamentals of Data Wrangler in AWS.
  - Perform one hot encoding and normalization.
  - Perform data visualization Using Data Wrangler.
  - Export a data wrangler workflow into Python Script.
  - Create a custom formula and apply it to a given column in the data.
  - Generate summary table tables in Data Wrangler.
  - Generate bias reports.

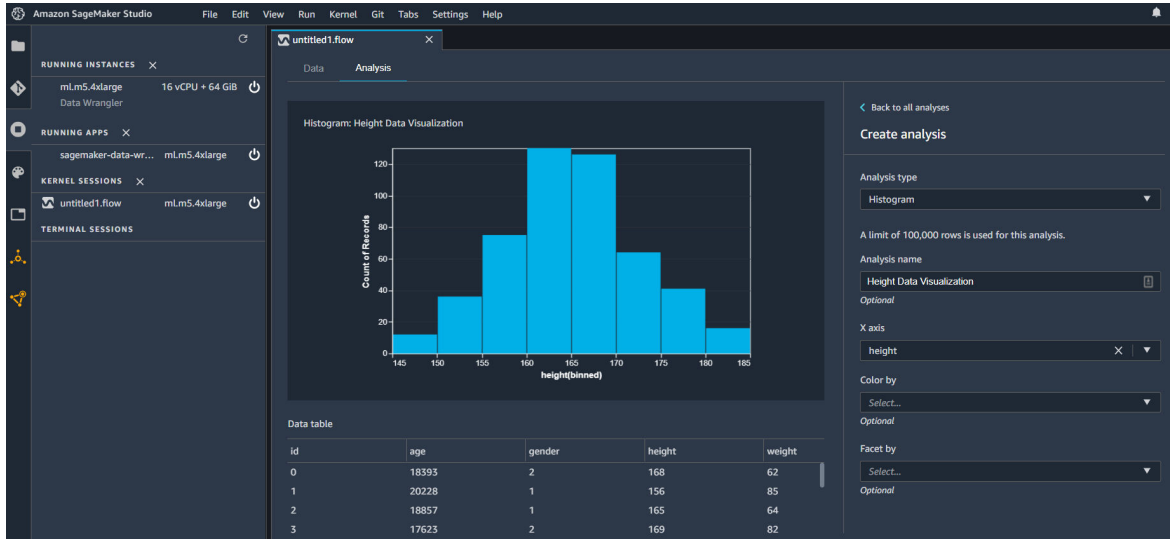
SAGEMAKER DATA WRANGLER 101



DATA WRANGLER 101

- Amazon SageMaker Data Wrangler accelerates the process of data preparation, exploration, cleaning, visualization and feature engineering. It makes creating Extract, Transform and Load (ETL) pipelines much easier.

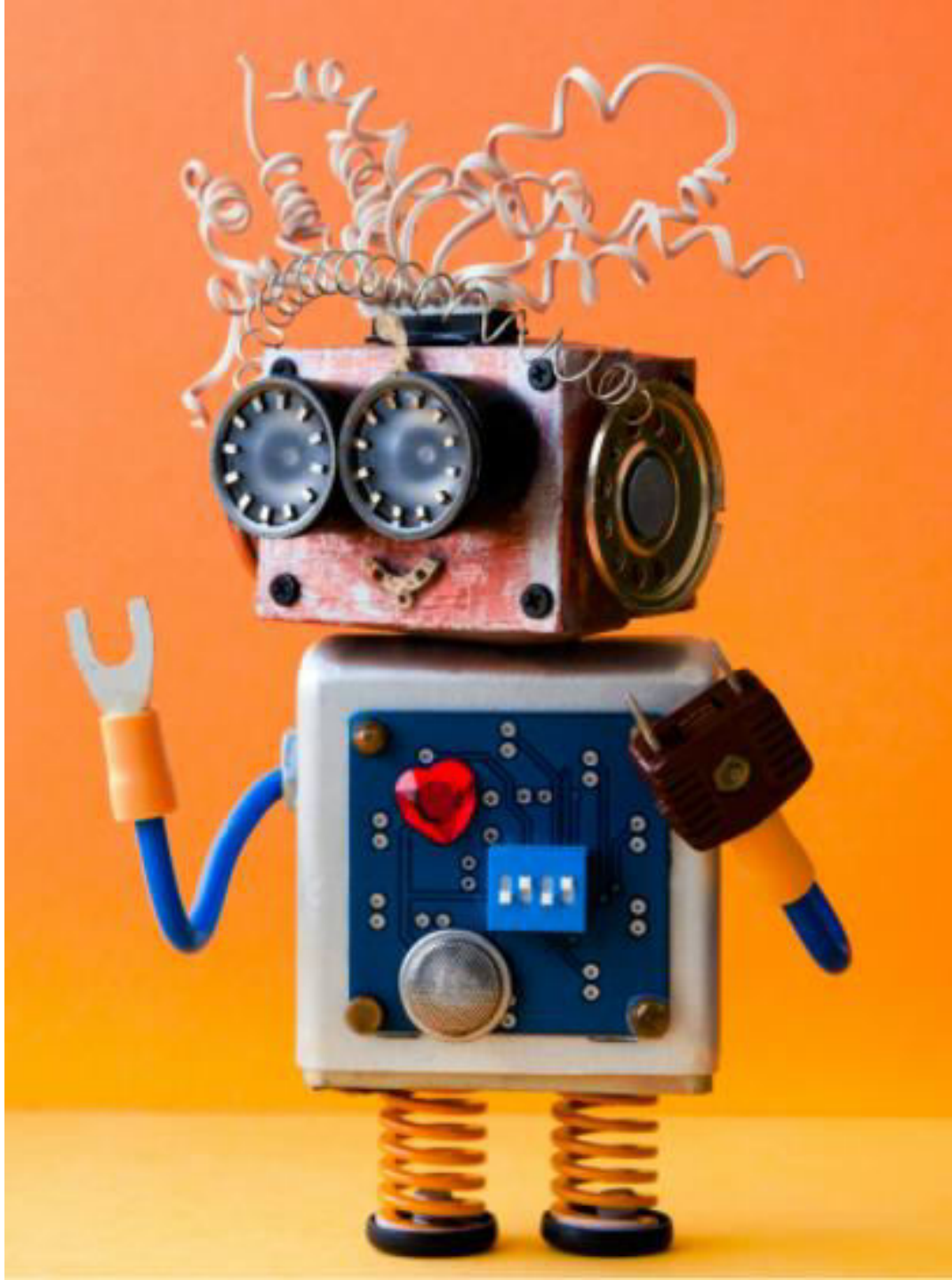




DATA WRANGLER 101

- SageMaker Data Wrangler is **cloud based** and **doesn't require any code**!
- Data can be imported into data Wrangler from more than one source such as **S3**, RedShift and SageMaker Feature Store.
- Data could be in **CSV, database tables and Parquet** formats.
- Data Wrangler includes over **300 data transformations** such as one hot encoding, normalization, imputation of missing data, ..etc.
- Several data visualization templates are available to generate **bar charts, line plots, histograms, scatterplots**...etc.
- Data Transformation **workflows can be exported** from data wrangler to a notebook or script so it can be automated with SageMaker Pipelines.
- Check out success stories from customers: https://aws.amazon.com/sagemaker/data-wrangler/

FEATURE ENGINEERING 101



WHAT IS FEATURE ENGINEERING?

- Machine Learning algorithms require training data to train.
- Feature engineering is a critical task that is performed by data scientists prior to training AI/ML models to ensure solid trained model performance.
- Feature engineering is an art of introducing new features that weren't existing before.
- Data scientists spend 80% of their time performing feature engineering.
- The remaining 20% is the easy part which includes training the model and performing hyperparameters optimization.
- As a data scientist, you may need to:
    1. Highlight important information in the data
    2. Remove/isolate unnecessary information (e.x.: outliers).
    3. Add your own expertise and domain knowledge to the alter the data.

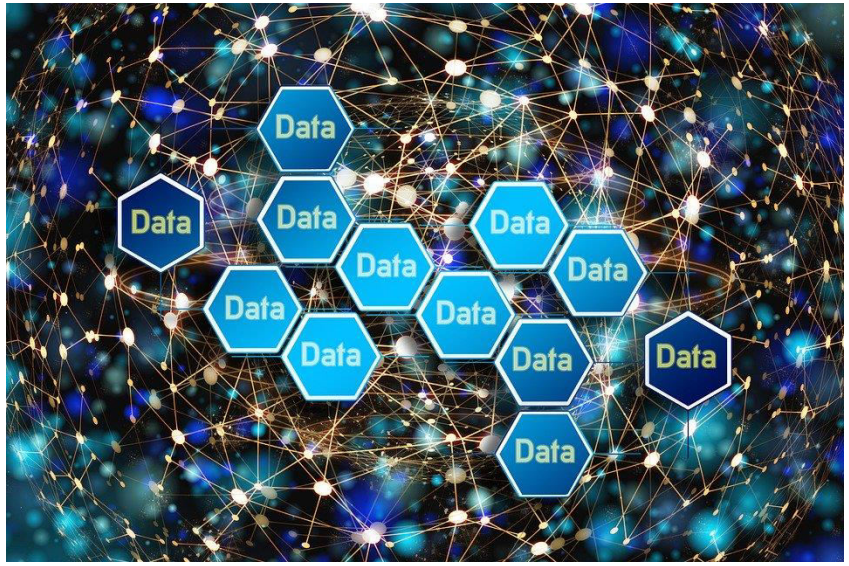


Photo Credit: https://pixabay.com/illustrations/network-data-memory-data-collection-4478146/

FEATURE ENGINEERING: PROPER QUESTIONS TO ASK?

- As a data scientist, you need to answer the following questions:

Which features should I select?
Can I add my domain knowledge to use less features?
Can I come up with new features from the data I have at hand?
What should I put in the missing data locations?
What are the capabilities of the ML model I have?

- It is important to choose features that are most relevant to the problem.
- Adding new features that are unnecessary will increase the computational requirements needed to train the model (curse of dimensionality).
- There are many techniques that could be used to reduce the number of features (compress/encode the data) such as Principal Component Analysis (PCA) – will be covered later.

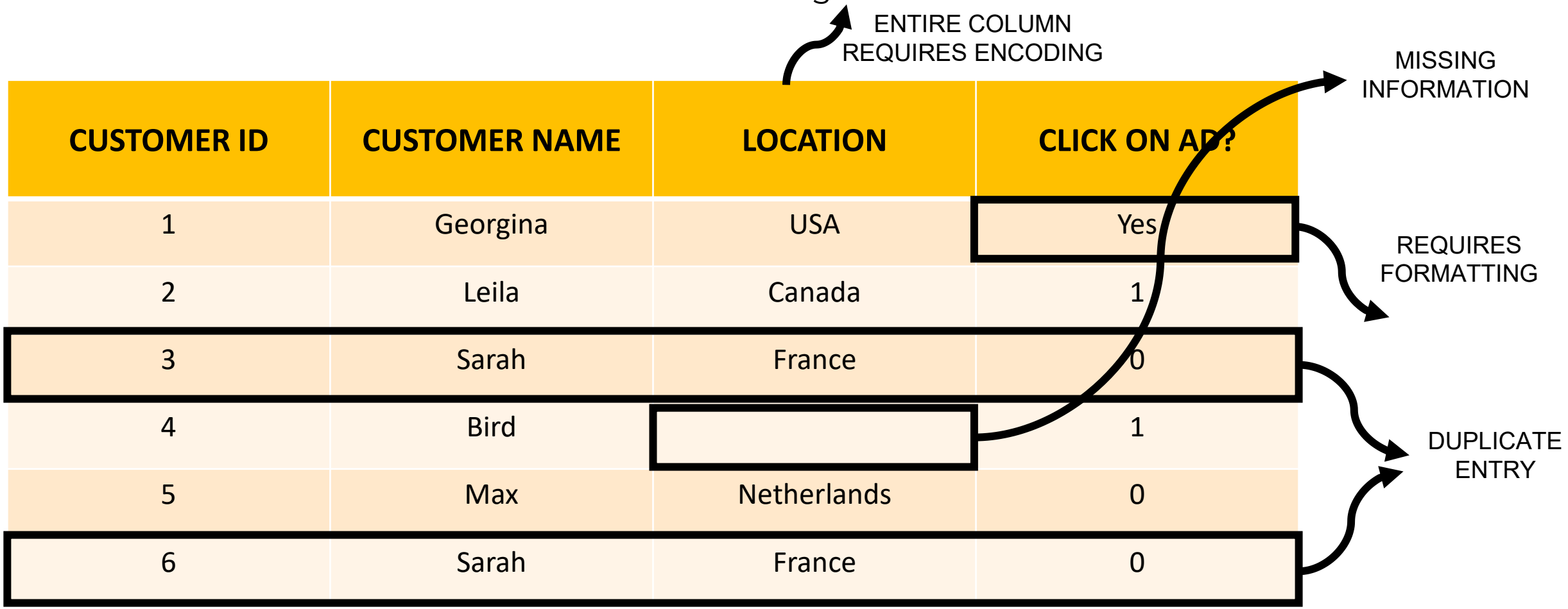
FEATURE ENGINEERING: QUIZ

- Let's take a look at this data and see what's wrong with it!

CUSTOMER ID	CUSTOMER NAME	LOCATION	CLICK ON AD?
1	Georgina	USA	Yes
2	Leila	Canada	1
3	Sarah	France	0
4	Bird		1
5	Max	Netherlands	0
6	Sarah	France	0

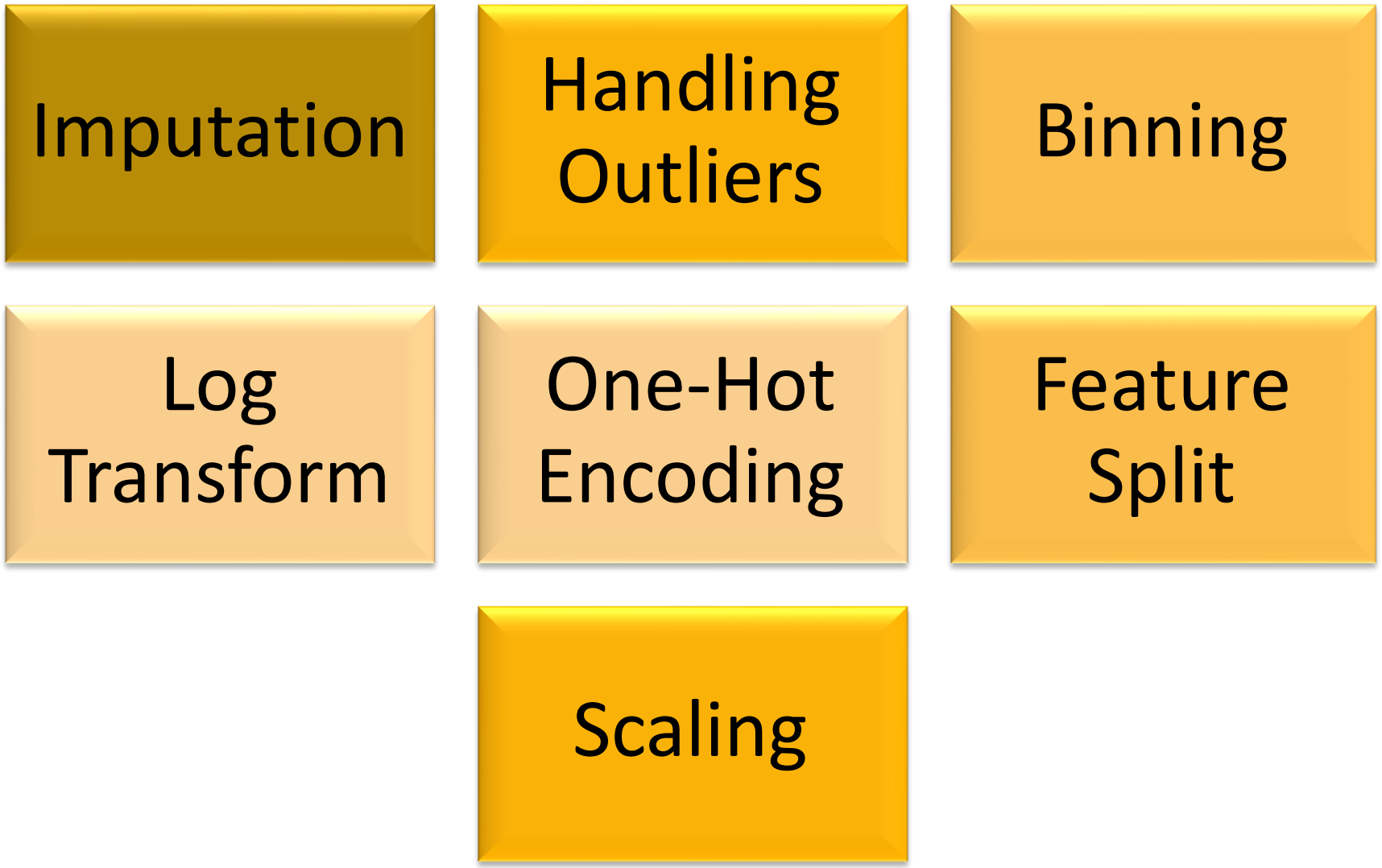
FEATURE ENGINEERING: SOLUTION

- Let's take a look at this data and see what's wrong with it!



CUSTOMER ID	CUSTOMER NAME	LOCATION	CLICK ON AD?
1	Georgina	USA	Yes
2	Leila	Canada	1
3	Sarah	France	0
4	Bird		1
5	Max	Netherlands	0
6	Sarah	France	0

FEATURE ENGINEERING TECHNIQUES



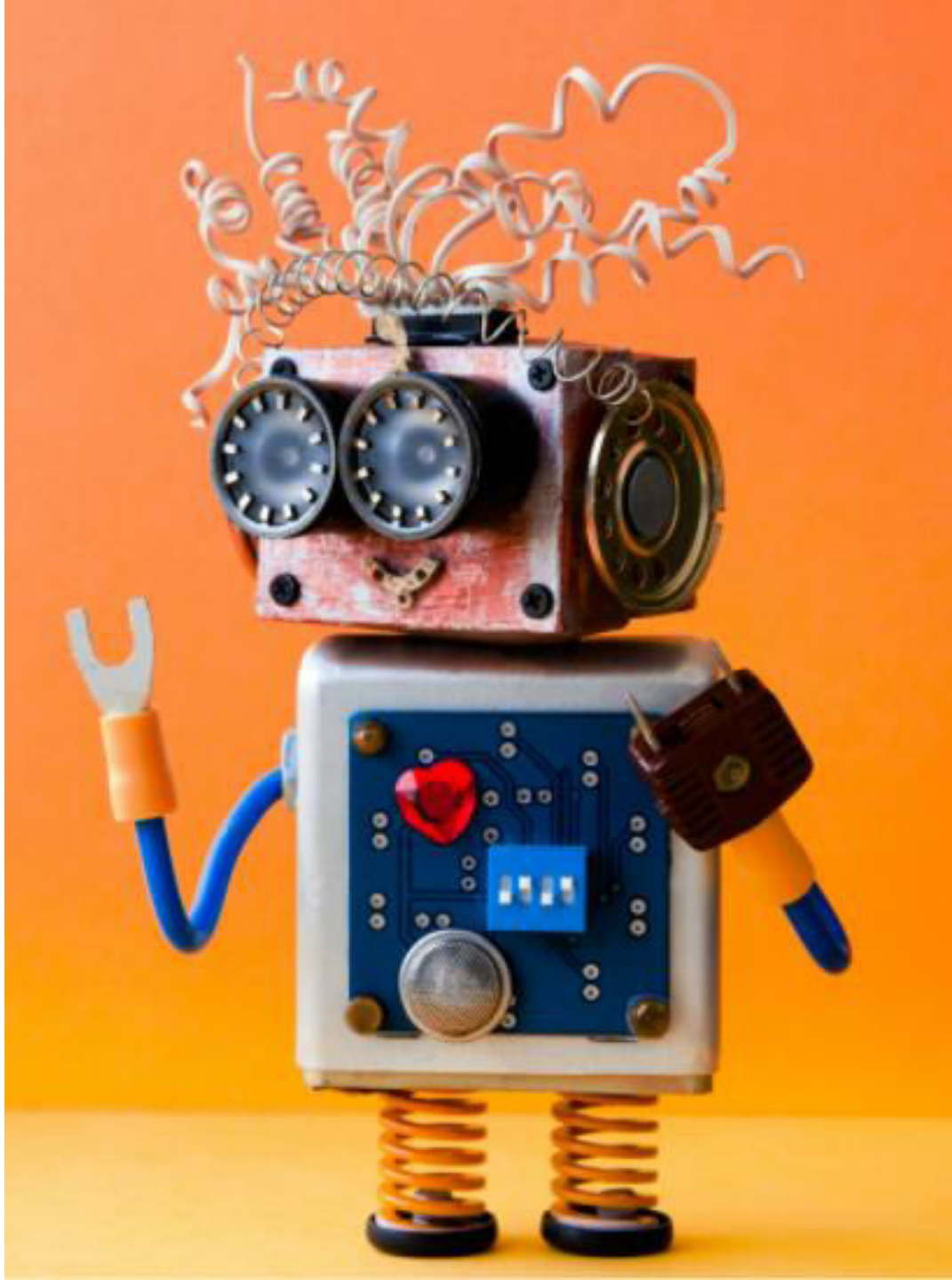
FEATURE ENGINEERING: TOOLS

JUPYTER NOTEBOOKS

AMAZON SAGEMAKER DATA WRANGLER

AWS GLUE

ONE-HOT ENCODING

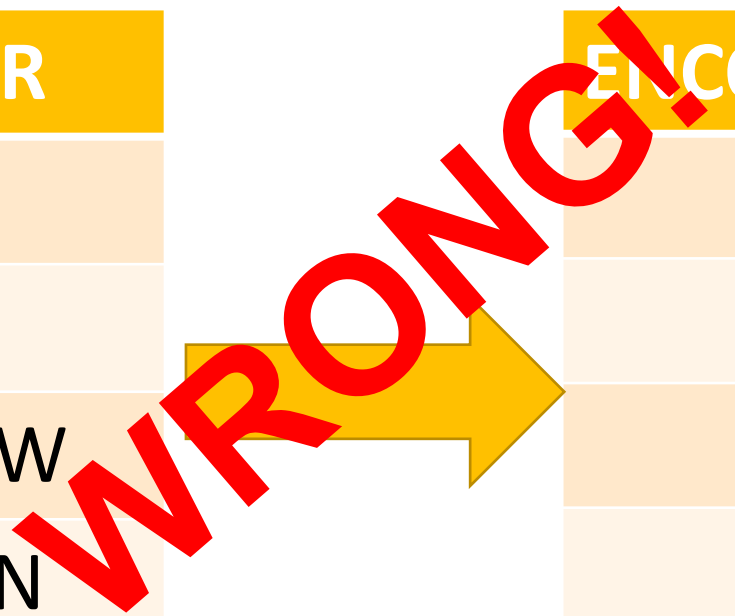


ONE-HOT ENCODING: WHY DO WE NEED IT?

- Can we simply replace colors with integer values?
- The machine learning model will assume that:

$$GREEN > YELLOW > RED$$

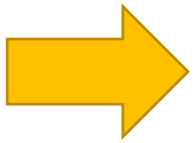
COLOR	ENCODED COLOR
RED	1
RED	1
YELLOW	2
GREEN	3
YELLOW	2



ONE-HOT ENCODING

- One hot encoding is widely used in machine learning.
- It works by converting values such as “color” into columns with 1’s and 0’s in them.
- Since machine learning models deal with numbers, we perform one hot encoding to convert from categorical data into numerical.
- If you have N categories, you will need N-1 binary columns to represent them.

COLOR
RED
RED
YELLOW
GREEN
YELLOW



RED	YELLOW	GREEN
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

ONE-HOT ENCODING: ORDINAL Vs. NOMINAL

- The difference between nominal and ordinal data is as follows:
  - In ordinal data, order is important.
  - In nominal data, order is not important.

NOMINAL

Order of colors doesn't mean anything!

COLOR
RED
RED
YELLOW
GREEN
YELLOW

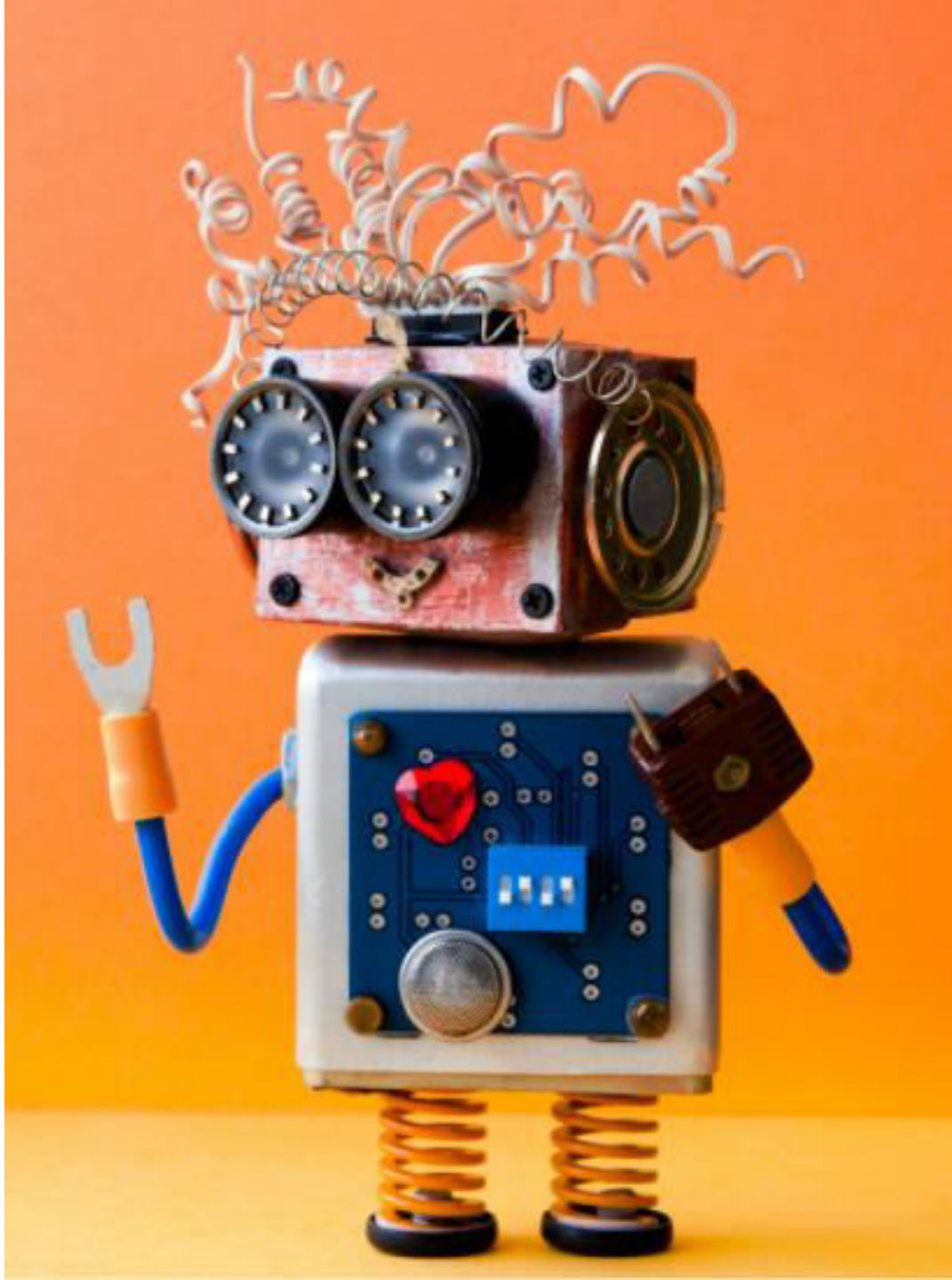
ORDINAL

Order is important!

- *1 star means poor quality course*
- *5 star means great quality course*

Photo Credit: https://pixabay.com/vectors/rating-stars-system-evaluation-153125/

SCALING: NORMALIZTION & STANDARDIZATION



FEATURE SCALING

- Feature Scaling is an important step to take prior to training of machine learning models to ensure that features are within the same scale.
- Example: interest rate and employment score are at a different scale. This will result in one feature dominating the other feature.
- Scikit Learn offers several tools to perform feature scaling.

RAW ORIGINAL DATASET

	Interest Rates	Employment	S&P 500 Price
0	1.943859	55.413571	2206.680582
1	2.258229	59.546305	2486.474488
2	2.215863	57.414687	2405.868337
3	1.977960	49.908353	2140.434475
4	2.437723	52.035492	2411.275663
5	2.143637	56.060598	2187.344909
6	2.148647	51.513208	2263.049249
7	2.176184	53.475909	2281.496374
8	2.125352	63.668422	2355.163011
9	2.225682	56.993396	2326.330337
10	1.814688	55.361780	2078.553895
11	2.281897	58.484752	2337.504507
12	2.426738	55.709328	2485.774097

QUICK STATS!

	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	2.20	56.25	2320.00
std	0.24	4.86	193.85
min	1.50	40.00	1800.00
25%	2.04	53.03	2190.45
50%	2.20	56.16	2312.44
75%	2.36	59.42	2455.76
max	3.00	70.00	3000.00

NORMALIZATION

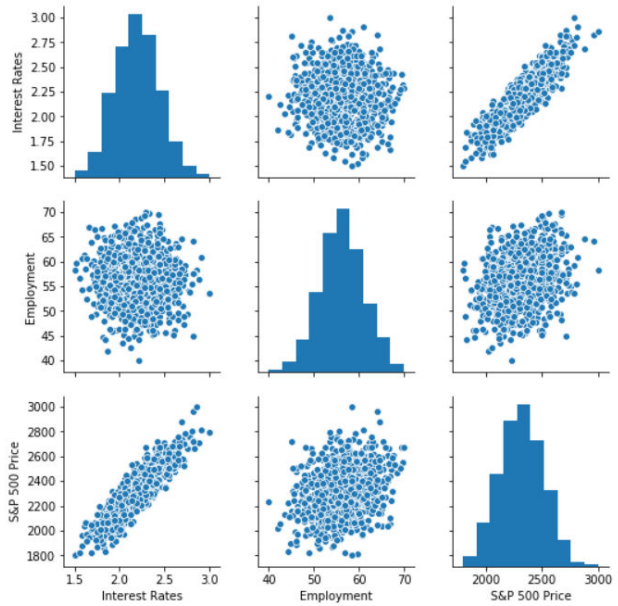
- Normalization is conducted to make feature values range from 0 to 1.

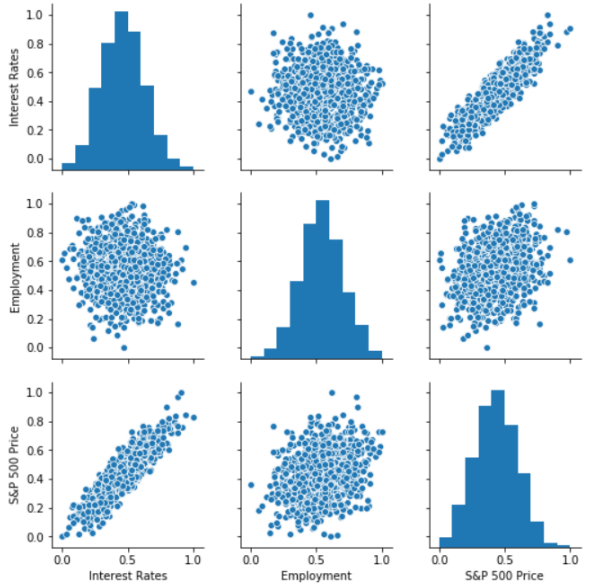
$$x' = \frac{x - \min(x)}{\max(x) - min(x)}$$

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
stock_df = scaler.fit_transform(stock_df)
```

NORMALIZATION

- Normalization is conducted to make feature values range from 0 to 1.





	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	2.20	56.25	2320.00
std	0.24	4.86	193.85
min	1.50	40.00	1800.00
25%	2.04	53.03	2190.45
50%	2.20	56.16	2312.44
75%	2.36	59.42	2455.76
max	3.00	70.00	3000.00

	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	0.46	0.54	0.43
std	0.16	0.16	0.16
min	0.00	0.00	0.00
25%	0.36	0.43	0.33
50%	0.47	0.54	0.43
75%	0.57	0.65	0.55
max	1.00	1.00	1.00

NORMALIZATION

$$x' = \frac{x - \min(x)}{\max(x) - min(x)} = \frac{2206 - 1800}{3000 - 1800} = 0.338$$

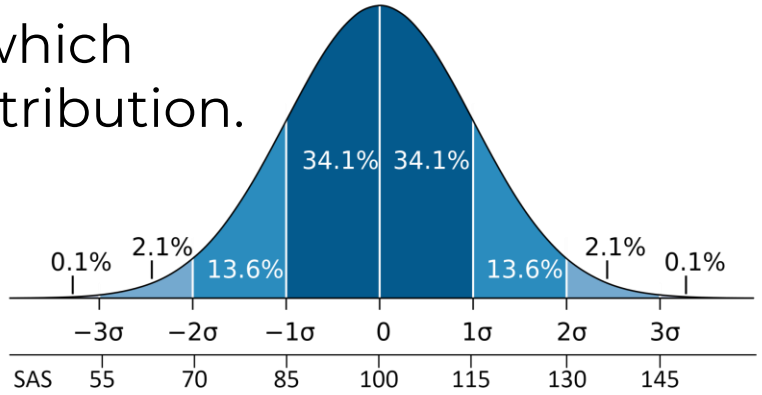
	S&P 500 PRICE (ORIGINAL)		S&P 500 PRICE (NORMALIZED)
Randomly selected Value	2206	NORMALIZATION/ SCALING →	0.338
Average	2319		0.432
Maximum	3000		1
Minimum	1800		0

- S&P 500 PRICE (Max) = 3000
- S&P 500 PRICE (Min) = 1800

NOTE THAT NUMBERS NOW RANGE FROM 0 TO 1 AFTER NORMALIZATION

STANDARDIZATION

- Standardization is conducted to transform the data to have a mean of zero and standard deviation of 1.
- Standardization is also known as Z-score normalization in which properties will have the behaviour of a standard normal distribution.



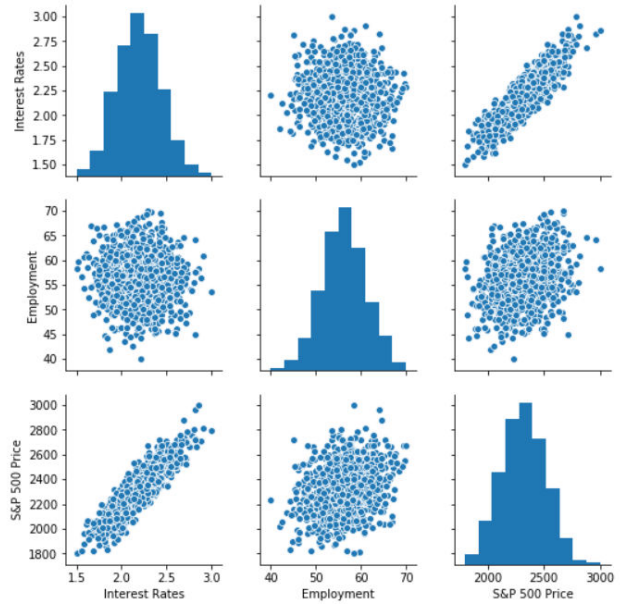
$$z = \frac{x - \overline{x}}{\sigma}$$

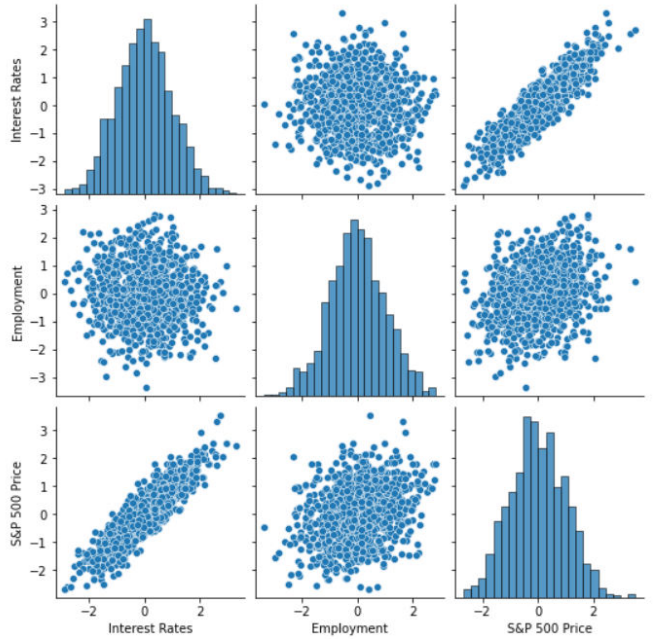
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
stock_df = scaler.fit_transform(stock_df)
```

Image Source: https://commons.wikimedia.org/wiki/File:Wechsler.svg

STANDARDIZATION

- Standardization transforms data to have a mean of zero and standard deviation of 1.





	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	2.20	56.25	2320.00
std	0.24	4.86	193.85
min	1.50	40.00	1800.00
25%	2.04	53.03	2190.45
50%	2.20	56.16	2312.44
75%	2.36	59.42	2455.76
max	3.00	70.00	3000.00

	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	0.00	0.00	-0.00
std	1.00	1.00	1.00
min	-2.88	-3.34	-2.68
25%	-0.66	-0.66	-0.67
50%	0.01	-0.02	-0.04
75%	0.68	0.65	0.70
max	3.33	2.83	3.51

STANDARDIZATION

$$z = \frac{x - \bar{x}}{\sigma} = \frac{2206 - 2319}{193.8} = -0.583$$

	S&P 500 PRICE (ORIGINAL)
Randomly selected Value	2206
Average	2319
Maximum	3000
Minimum	1800

STANDARIZATION

S&P 500 PRICE (STANDARIZED)
-0.583
0
3.513
-2.67

- S&P 500 PRICE (Mean) = 2319
- S&P 500 PRICE (Std) = 193.8

NOTE THAT AFTER STANDARDIZATION THE AVERAGE IS SET TO ZERO

ALWAYS REMEMBER!

“A normalized dataset will always range from 0 to 1”

“A standardized dataset will always have a mean of 0 and standard deviation of 1, but can have any upper and lower values”

WHEN SHOULD I PERFORM STANDARDIZATION VS. NORMALIZATION?

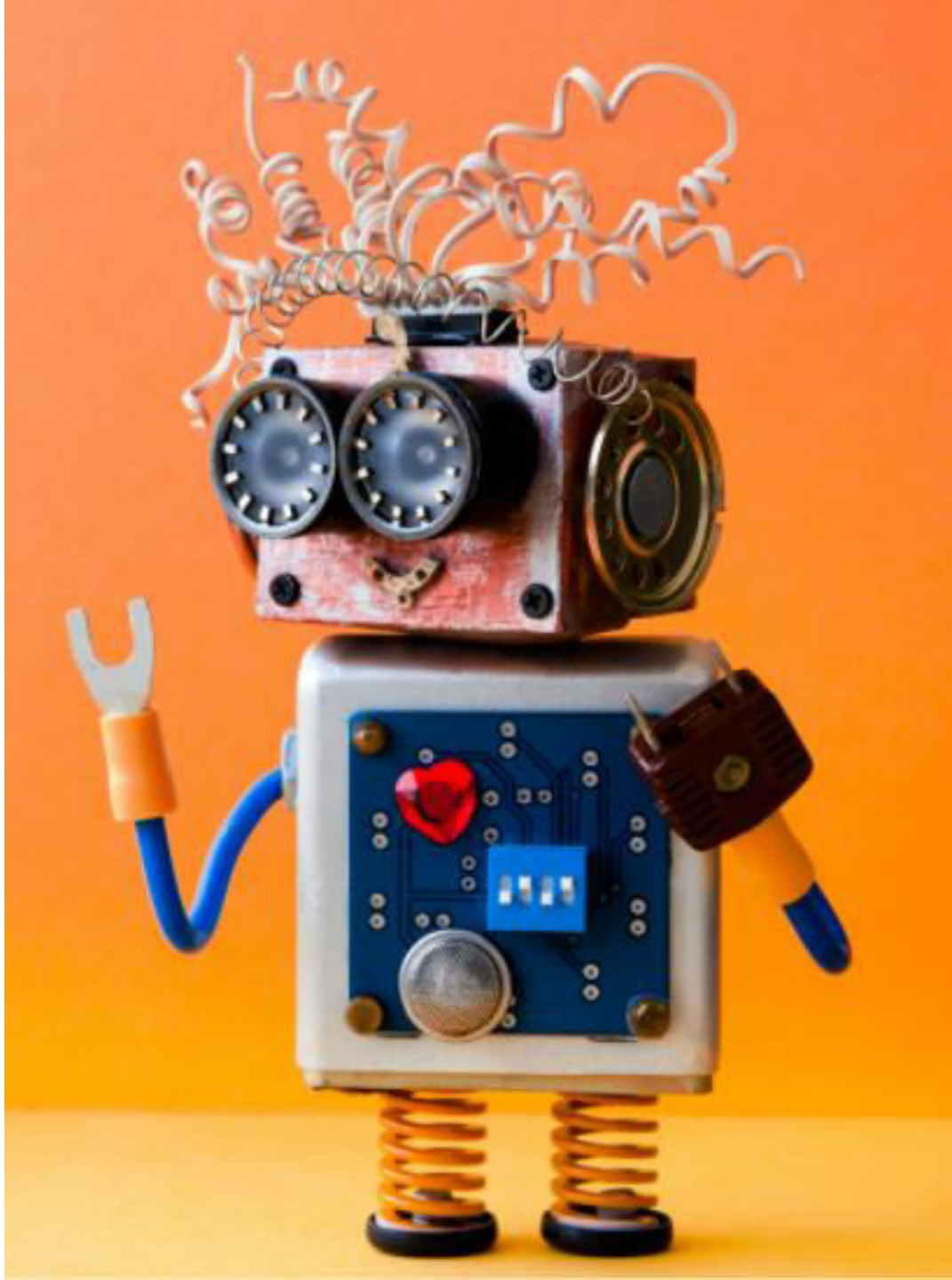
- Scaling (standardization or normalization) is required when we use any machine learning algorithm that require **gradient calculation.**
- Examples of machine learning algorithms that require gradient calculations are: linear/logistic regression and artificial neural networks
- Having different scales for each feature will result in a different step size which in turn jeopardizes the process of reaching a minimum point.
- Scaling is not required for distance-based and tree-based algorithms such as K-Means Clustering, Support Vector Machines and K Nearest Neighbors, decision trees, random forest, and XG-Boost.

STANDARDIZATION Vs. NORMALIZATION?

- Generally speaking, there is no right or wrong answer!
- In case of neural networks, normalization is preferred since we don’t assume any data distribution.
- Standardization is preferred when data follows gaussian distribution
- Standardization is preferred over normalization when there are a lot of outliers.

DATA WRANGLER DEMO #1: LAUNCHING & UPLOADING DATA TO S3

EASY ADVANCED

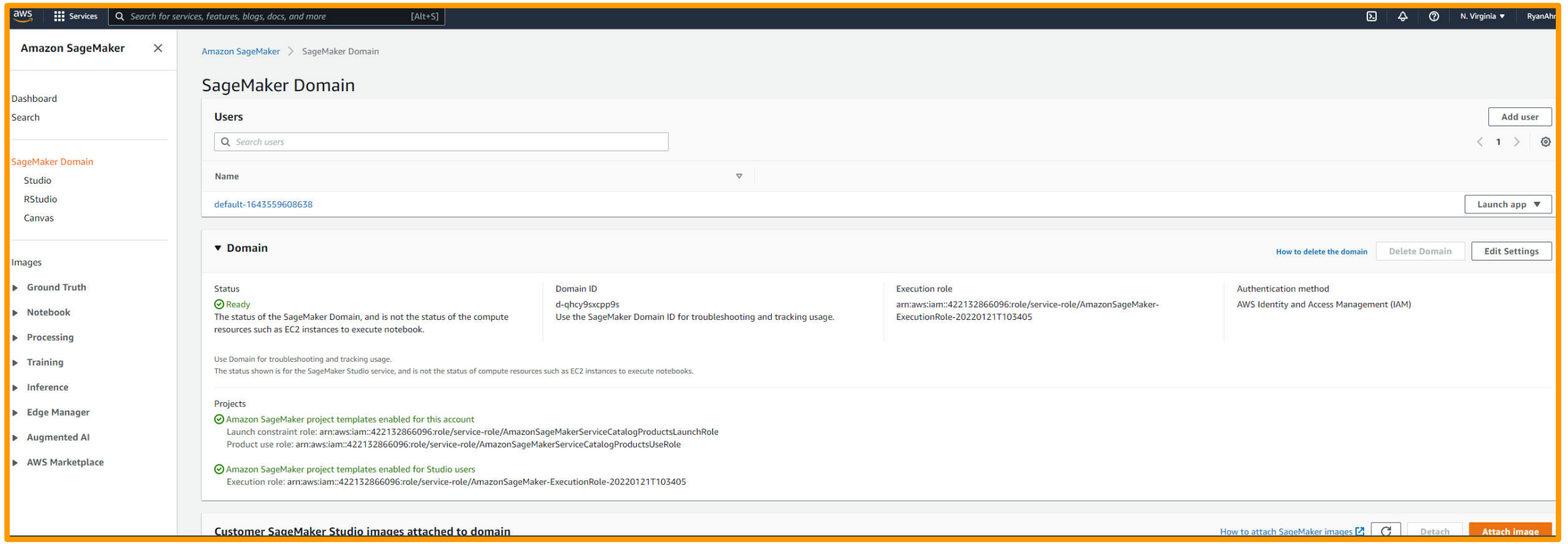


DATA WRANGLER DEMO

GO TO SAGEMAKER HOMEPAGE

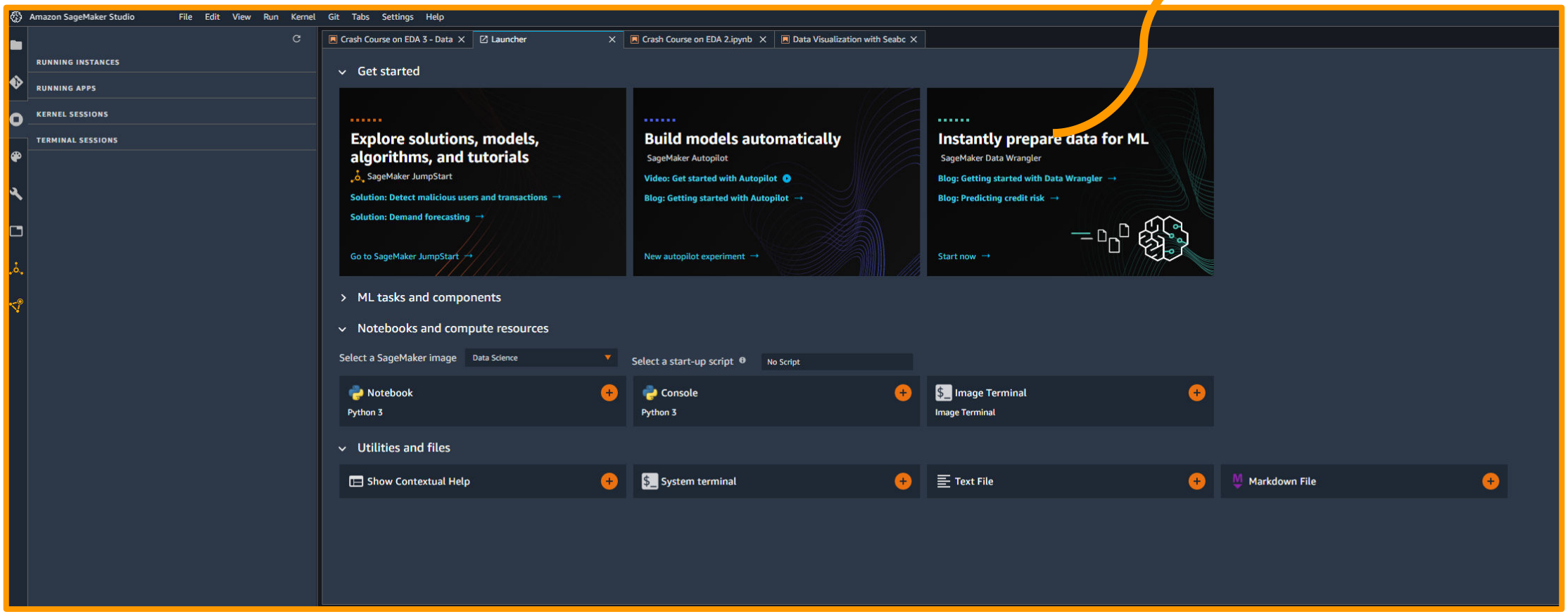
DATA WRANGLER DEMO

CLICK ON LAUNCH APP



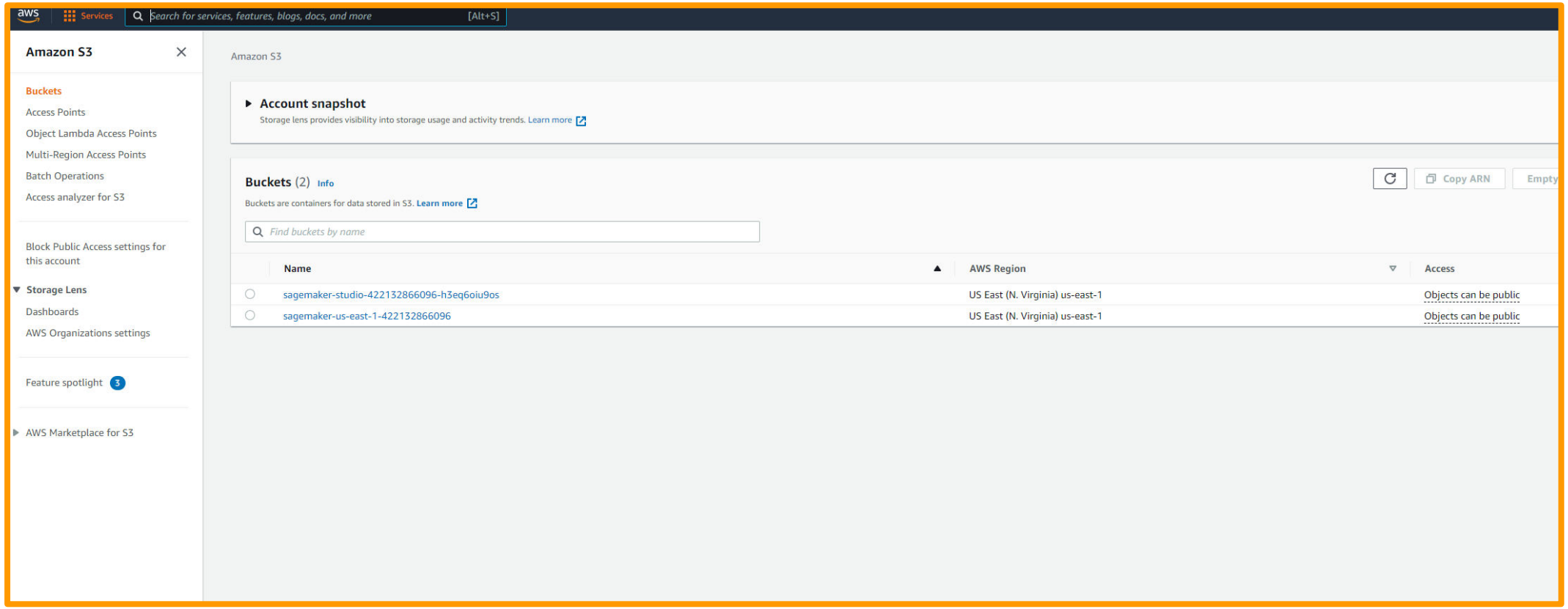
DATA WRANGLER DEMO

CLICK ON INSTANTLY PREPARE DATA FOR ML WITH DATA WRANGLER



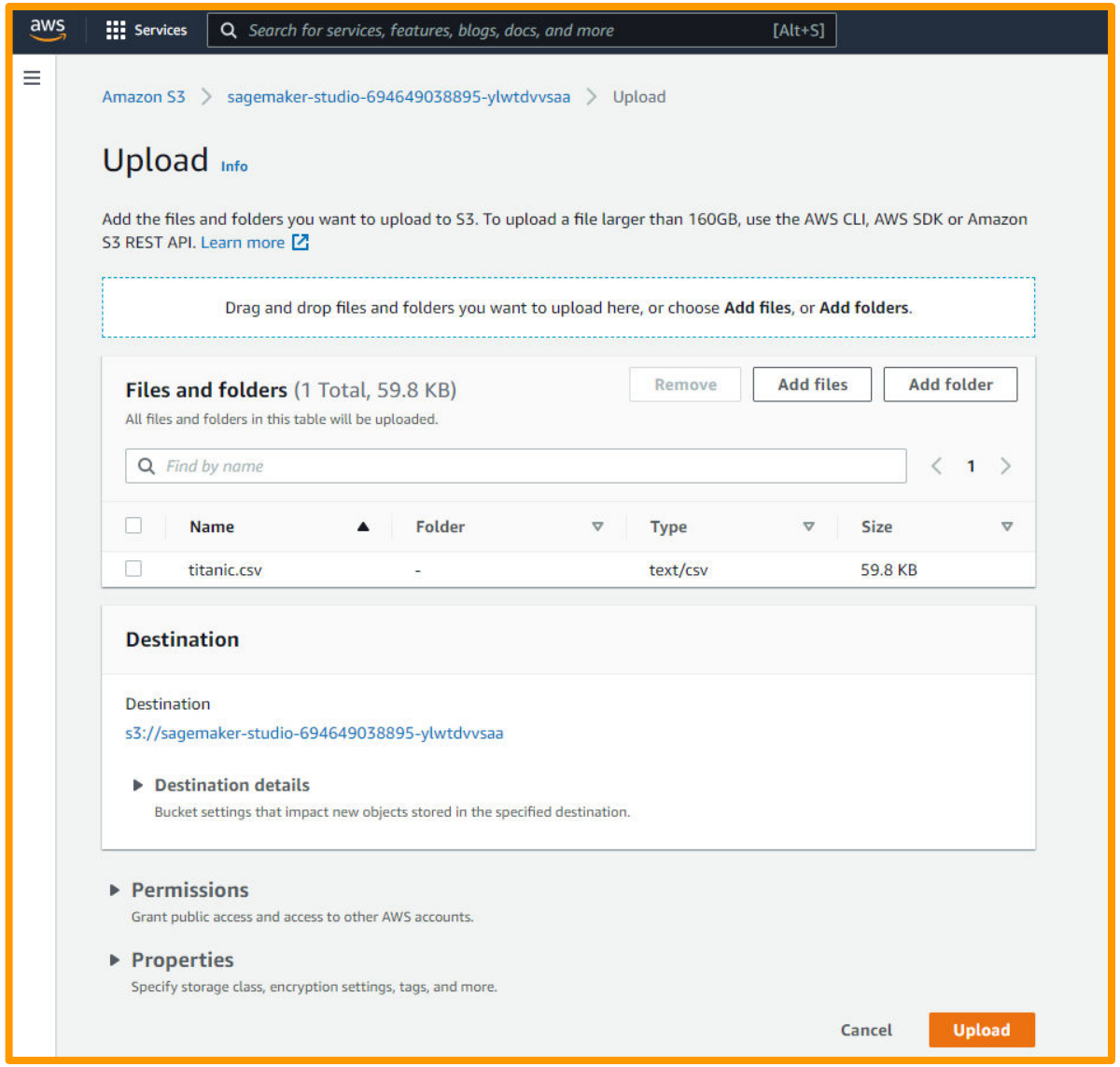
DATA WRANGLER DEMO

LET'S UPLOAD THE DATA TO S3.
GO TO S3 AND NAVIGATE TO THE SAGEMAKER STUDIO BUCKET.



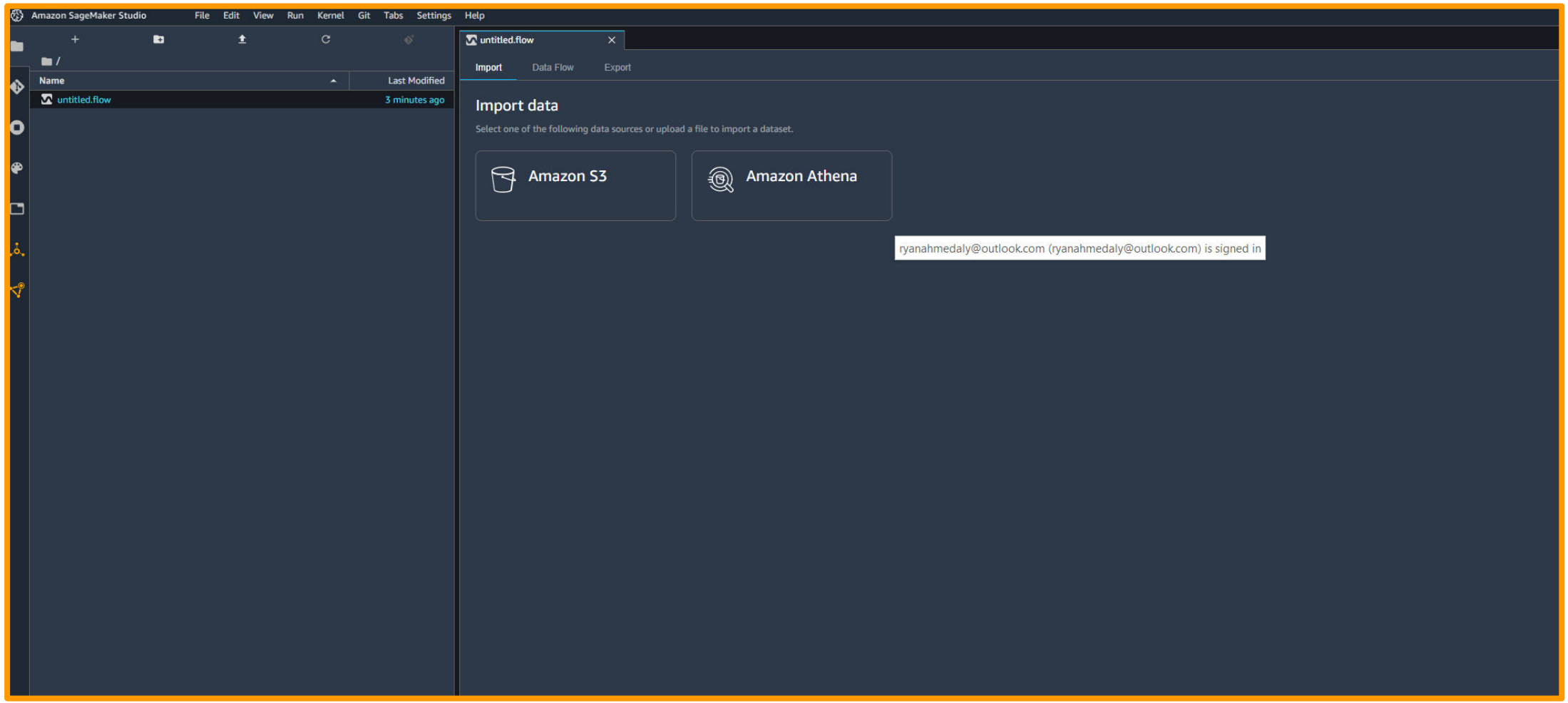
DATA WRANGLER DEMO

DRAG AND DROP TITANIC.CSV to S3



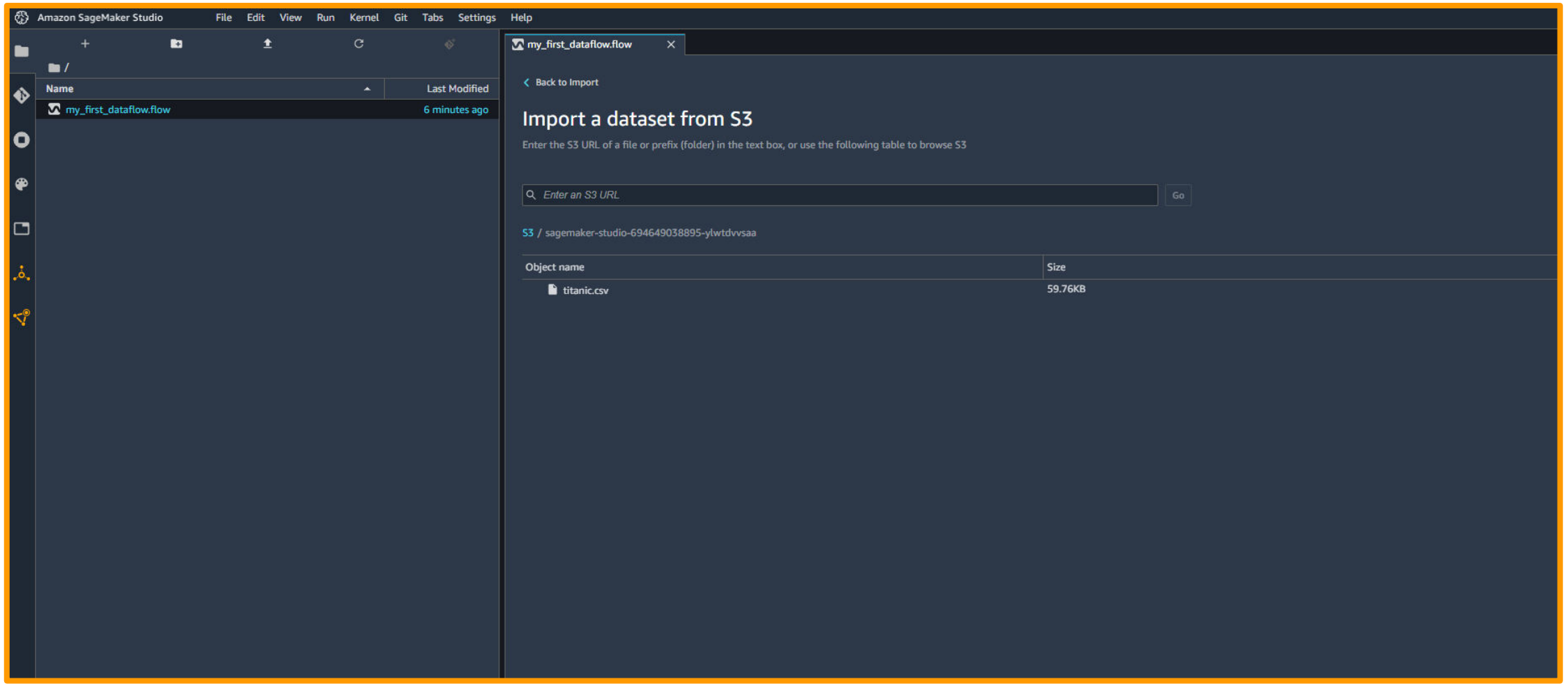
DATA WRANGLER DEMO

SELECT DATA FROM S3

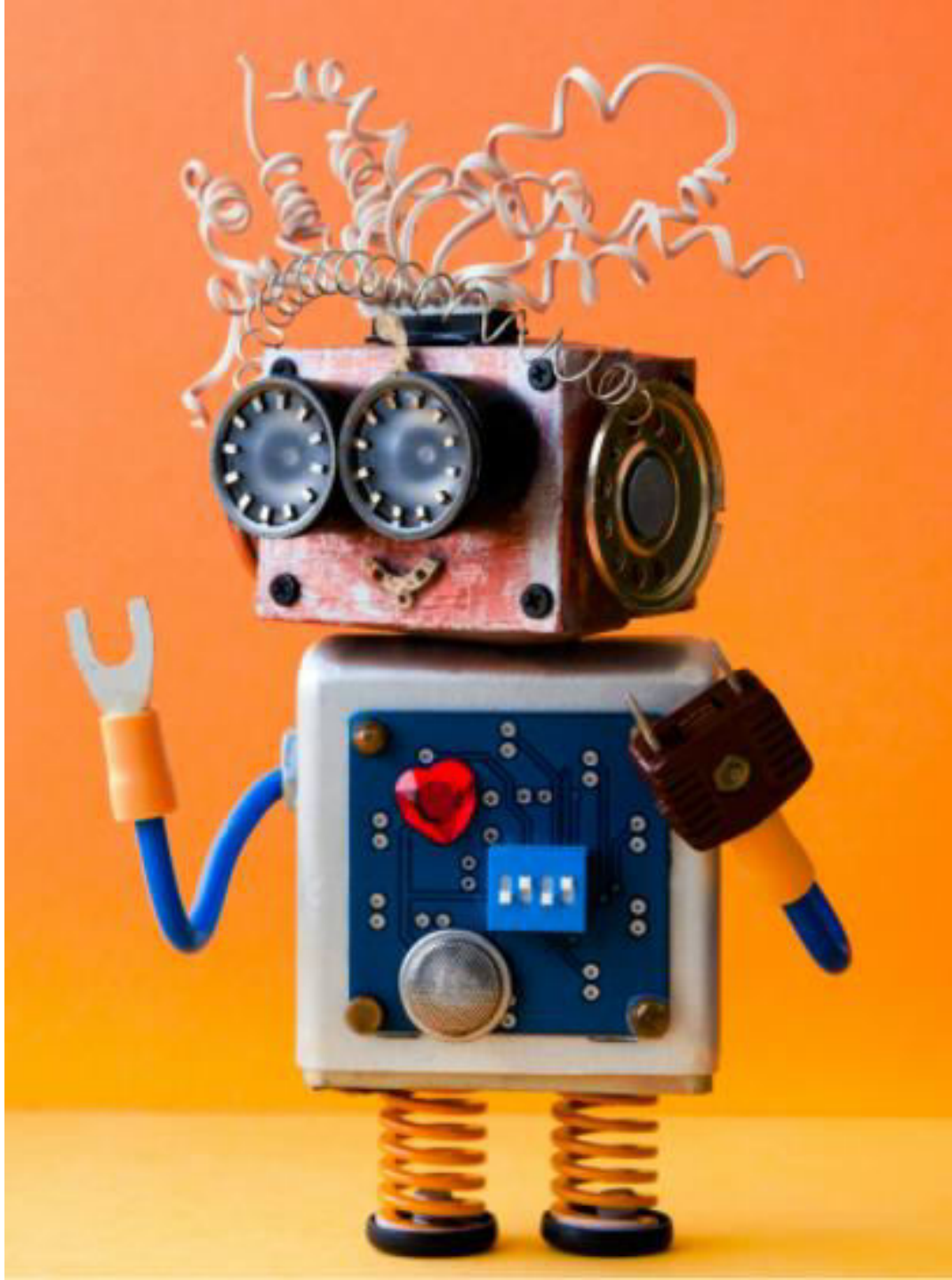


DATA WRANGLER DEMO

CLICK ON S3 AND SELECT THE DATASET YOU RECENTLY UPLOADED TO S3

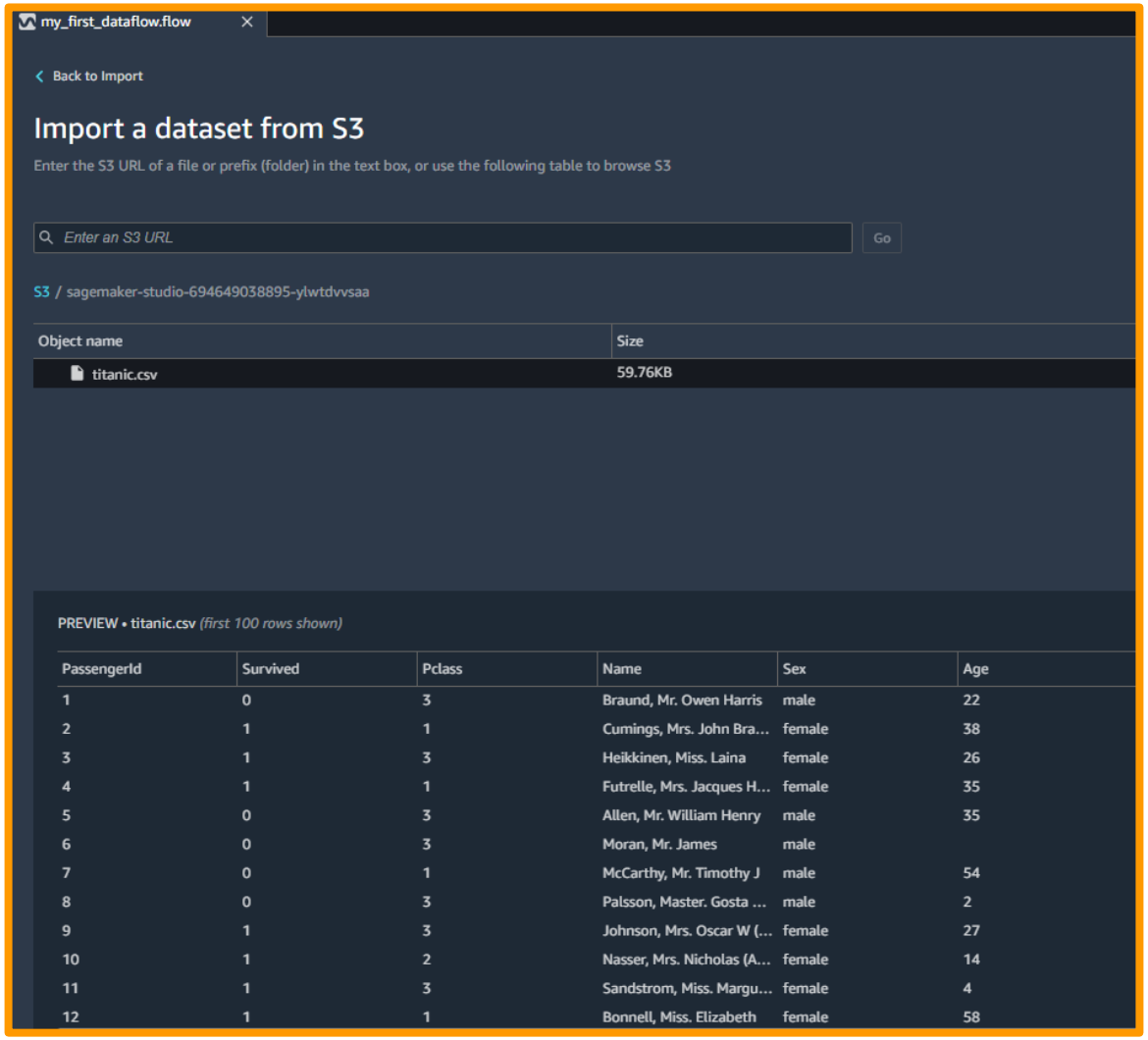


DATA WRANGLER DEMO #2: PERFORM EDA (CHANGE DATA TYPE & SUMMARY)



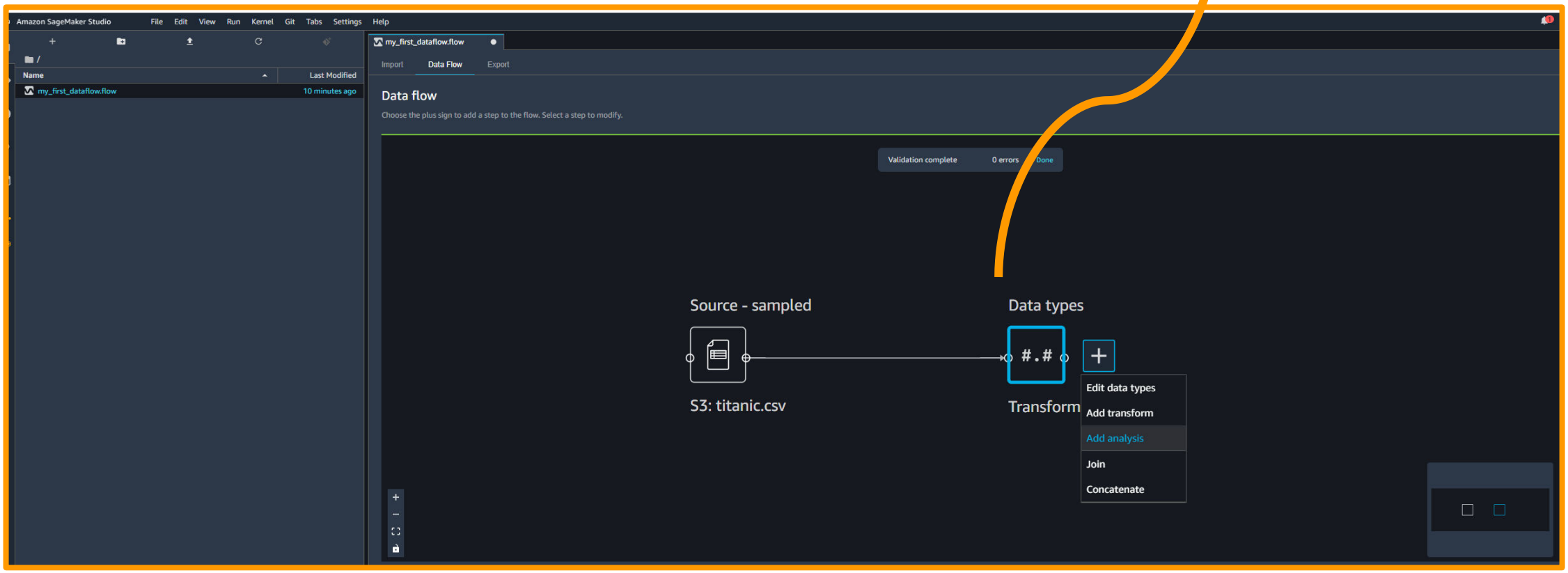
DATA WRANGLER DEMO

YOU SHOULD SEE THE PREVIEW OF THE DATASET BELOW, CLICK IMPORT!



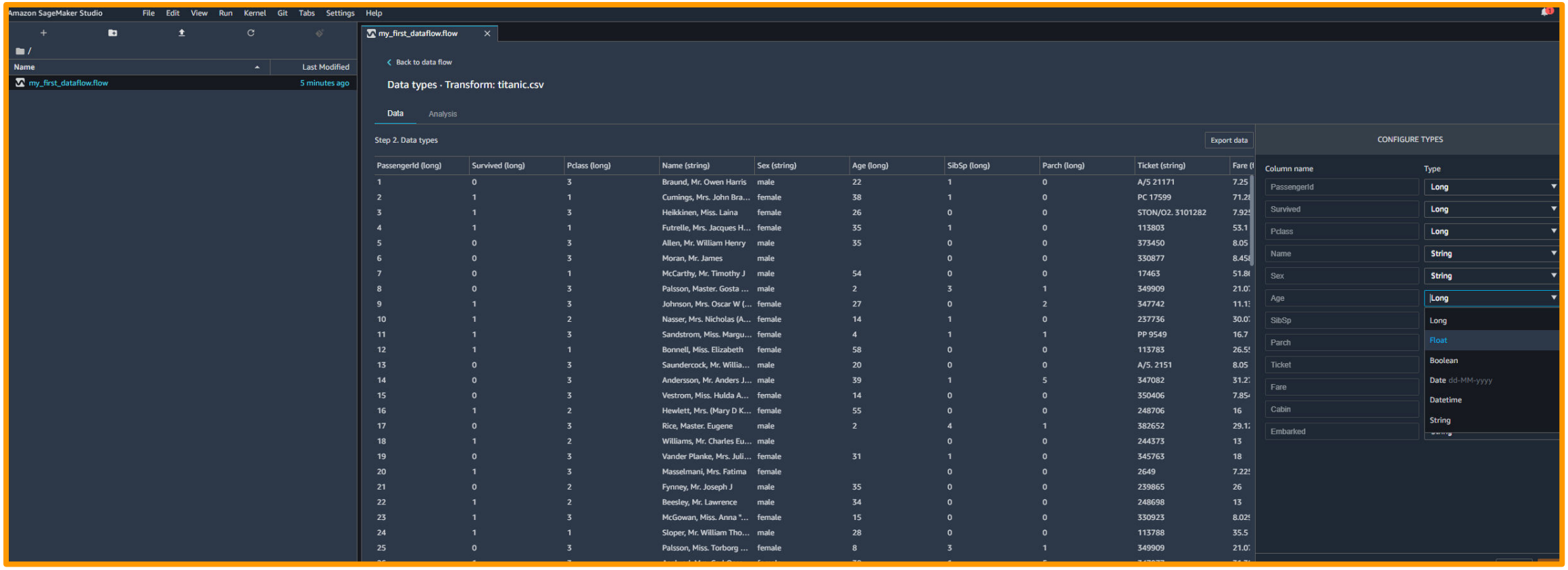
DATA WRANGLER DEMO

YOU SHOULD SEE THE DATA FLOW,
CLICK ON THE “+” NEXT TO TRANSFORM



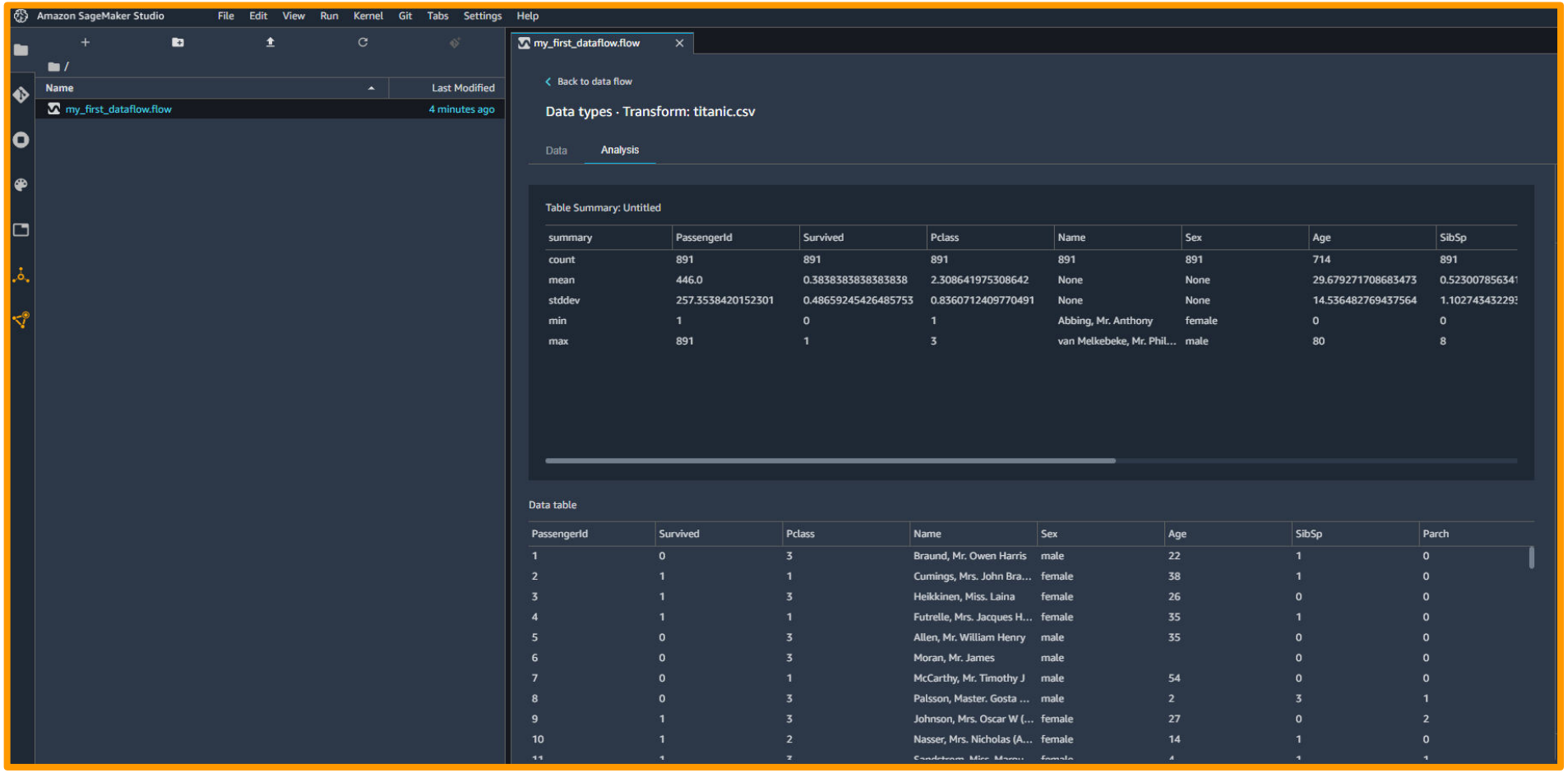
DATA WRANGLER DEMO

YOU CAN EASILY EDIT DATATYPES IN DATA WRANGLER. LET'S CHANGE THE FARE FROM "FLOAT" TO "LONG". NOTE THAT LONG IS A LARGER DATA TYPE THAN INTEGER. INT IS 32 BITS IN WIDTH WHILE LONG IS 64 BITS. YOU CAN PREVIEW THE CHANGE AND ACCEPT IT



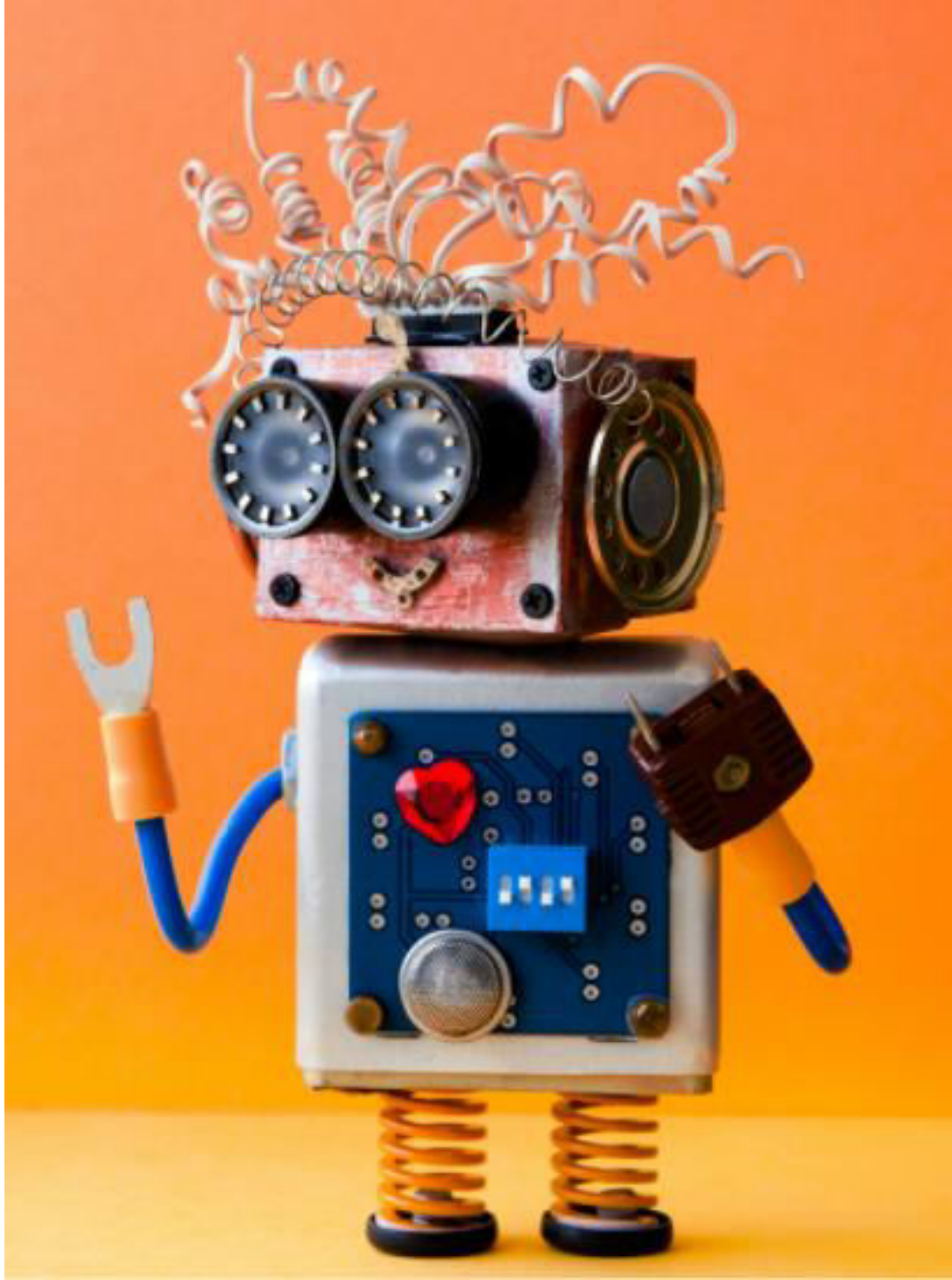
DATA WRANGLER DEMO

LET'S SEE IF WE HAVE MISSING DATA AND PRINT OUT A STATISCAL SUMMARY SIMILAR TO THE *.describe()* method. CLICK ON ANALYSIS>TABLE SUMMARY
NOTE THAT WE HAVE MISSING DATA IN "AGE" AND "CABIN"



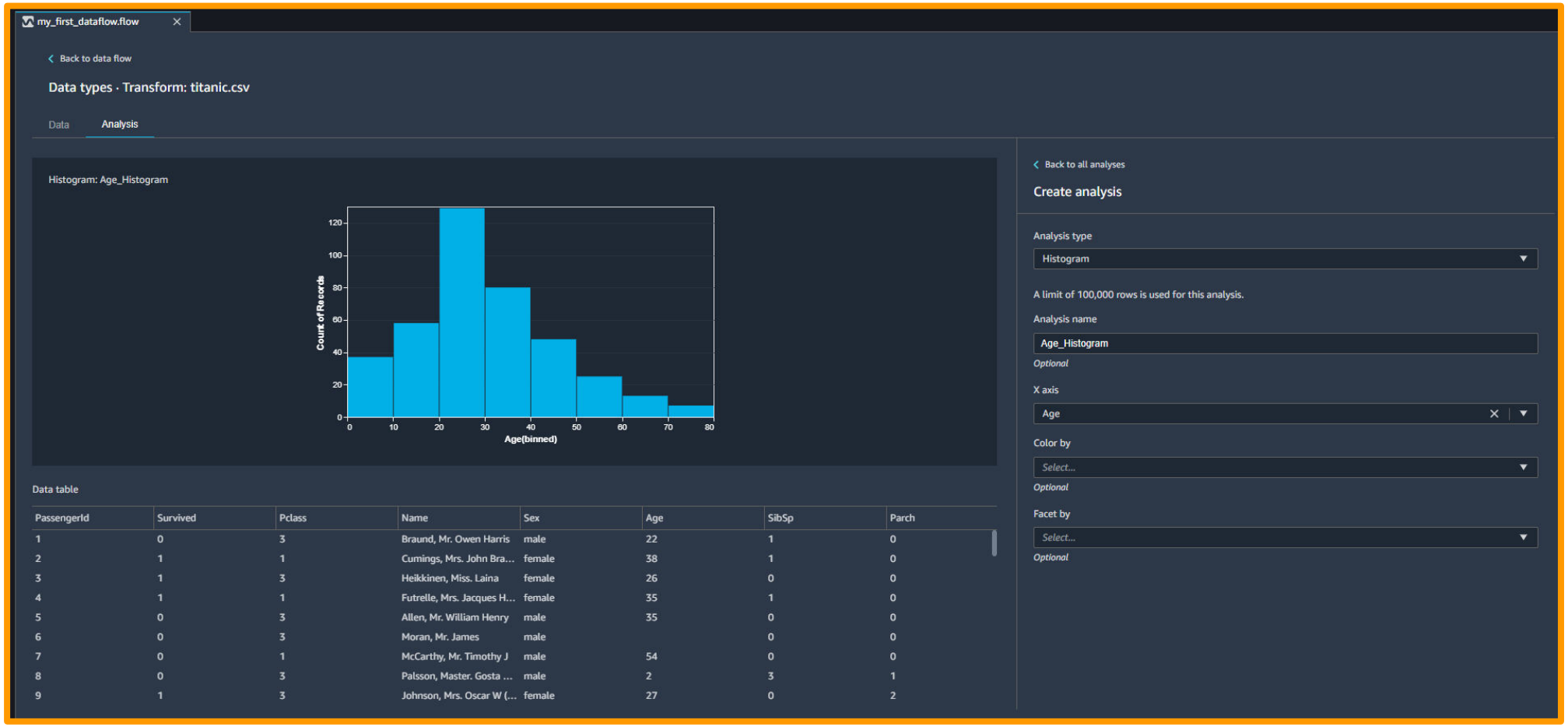
DATA WRANGLER DEMO #3: PERFORM DATA VISUALIZATION

EASY ADVANCED



DATA WRANGLER DEMO

LET'S PLOT THE HISTOGRAM OF THE AGE COLUMN



DATA WRANGLER DEMO

LET'S PLOT CORRELATIONS

my_first_dataflow.flow

Back to data flow

Data types · Transform: titanic.csv

Data | Analysis

Feature Correlation: Feature_Correlations

Linear feature correlation is based on Pearson's correlation. Numeric to categorical correlation is calculated by encoding the categorical features as the floating point numbers that best predict the numeric feature before calculating Pearson's correlation. Linear categorical to categorical correlation is not supported.

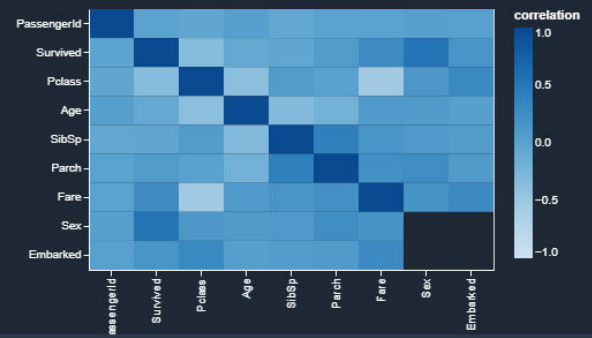
Numeric to numeric correlation is in the range [-1, 1] where 0 implies no correlation, 1 implies perfect correlation and -1 implies perfect inverse correlation. Numeric to categorical and categorical to categrical correlations are in the range [0, 1] where 0 implies no correlation and 1 implies perfect correlation

Features that are not either numeric or categorical are ignored.

The table below lists for each feature what is the most correlated feature to it.

	Most correlated feature	Correlation
Pclass (numeric)	Fare (numeric)	-0.550553
Fare (numeric)	Pclass (numeric)	-0.550553
Survived (numeric)	Sex (categorical)	0.543351
Sex (categorical)	Survived (numeric)	0.543351
SibSp (numeric)	Parch (numeric)	0.414838
Parch (numeric)	SibSp (numeric)	0.414838
Age (numeric)	Pclass (numeric)	-0.36945
Embarked (categorical)	Pclass (numeric)	0.308249
PassengerId (numeric)	SibSp (numeric)	-0.0575268

Correlation matrix:



Back to all analyses

Create analysis

Analysis type

Feature Correlation

A limit of 100,000 rows is used for this analysis.

Analysis name

Feature_Correlations

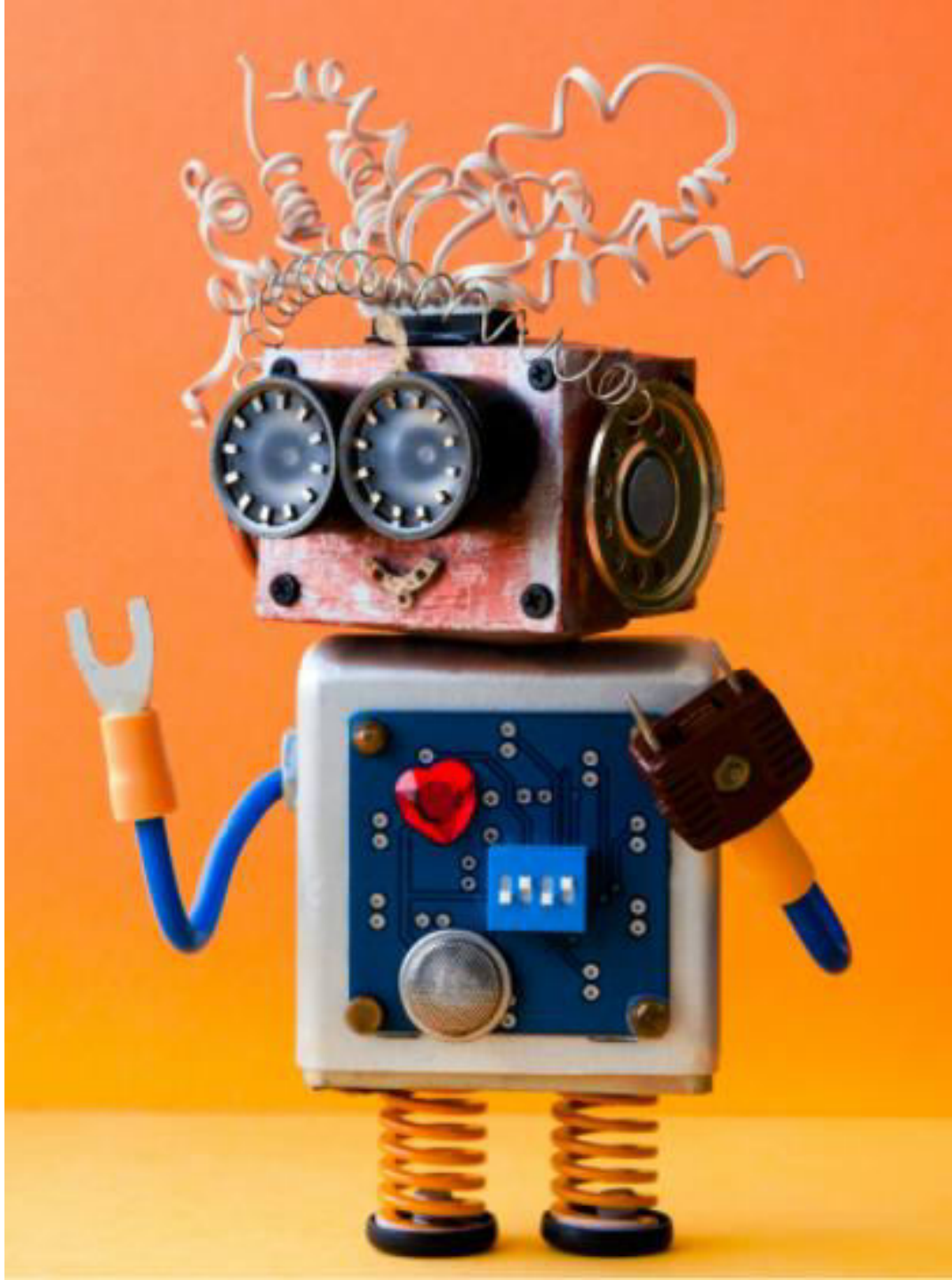
Optional

Correlation type

linear

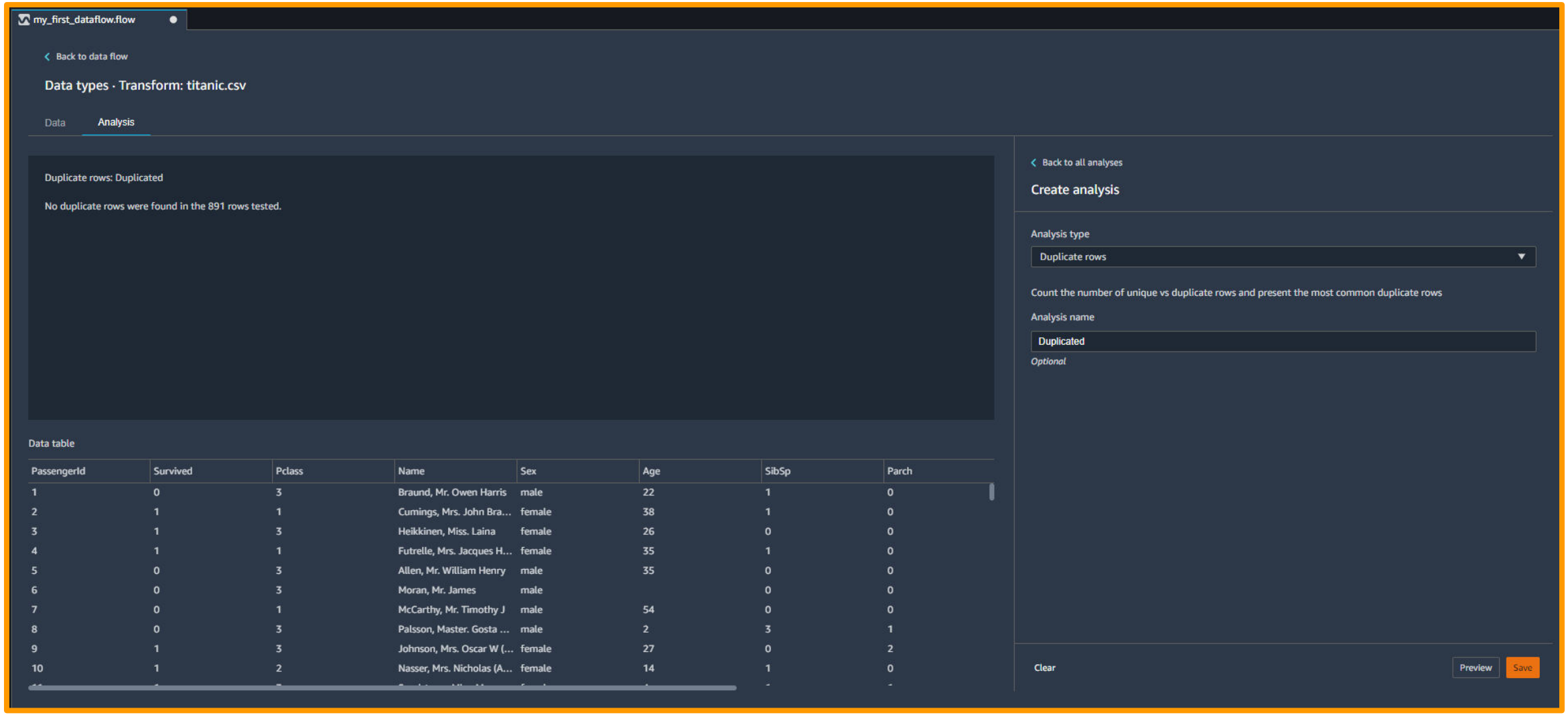
DATA WRANGLER DEMO #4: DUPLICATES, BIAS, AND FEATURE IMPORTANCE

EASY ADVANCED



DATA WRANGLER DEMO

LET'S SEE IF THERE ARE ANY DUPLICATED ROWS



DATA WRANGLER DEMO

LET'S GET A QUICK MODEL AND PLOT FEATURE IMPORTANCE

my_first_dataflow.flow

Back to data flow

Data types · Transform: titanic.csv

Data | Analysis

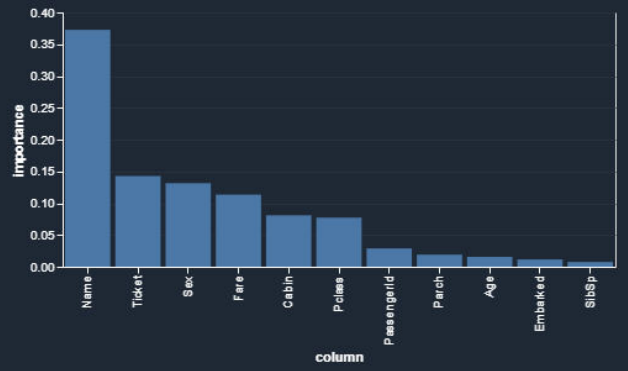
Quick Model: quick_model

We train a random forest with 10 trees on 499 observations and measure prediction quality on the remaining 215 observations. For classification, we use stratified sampling for both the training dataset and the test dataset. For stratified sampling, we divide your data into groups based on the labels in your dataset. For both your training and test datasets, we choose a random sample that is proportional to the dataset that you provide. For example, if you have a dataset about cars with 25% of the cars being minivans, 50% being SUVs, and 25% being sedans, the training and test datasets will have the same proportion of minivans, SUVs, and sedans.

The Random Forest is trained with the default hyper parameters. There is minimal preprocessing of the features before the model is trained.

The model achieved an f1 of 0.796 on the test set.

We use Gini importance scores as feature importance scores. For more information, see documentation.



Data table

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
1	0	3	Braund, Mr. Owen Harris	male	22	1	0
2	1	1	Cumings, Mrs. John Bra...	female	38	1	0
3	1	3	Heikkinen, Miss. Laina	female	26	0	0

Back to all analyses

Create analysis

Analysis type

Quick Model

A limit of 100,000 rows is used for this analysis. You can use the Quick Model feature to provide a rough estimate of the expected predicted quality and the predictive power of the features in your dataset. We don't recommend using a quick model to fine tune the data preprocessing pipeline or to optimize feature selection.

Analysis name

quick_model

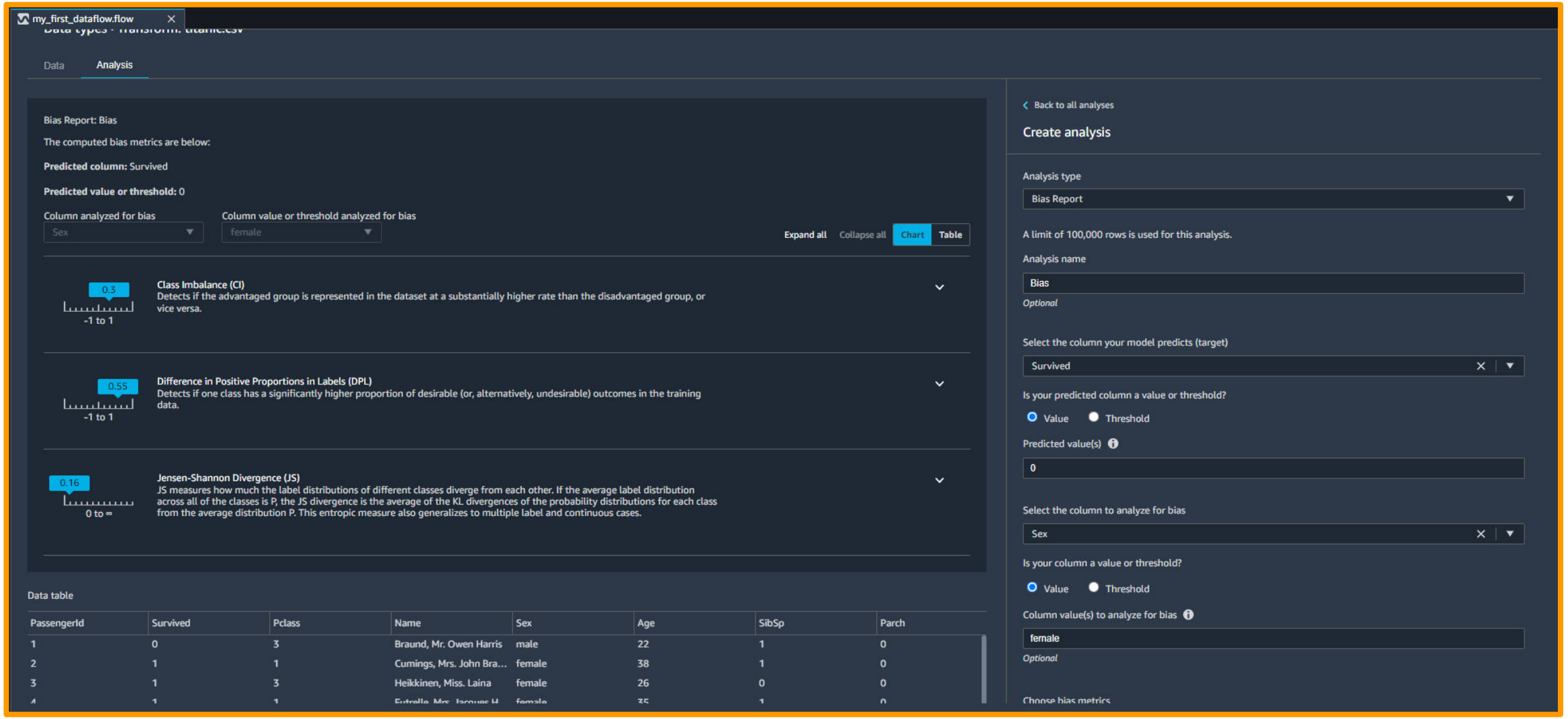
Optional

Label

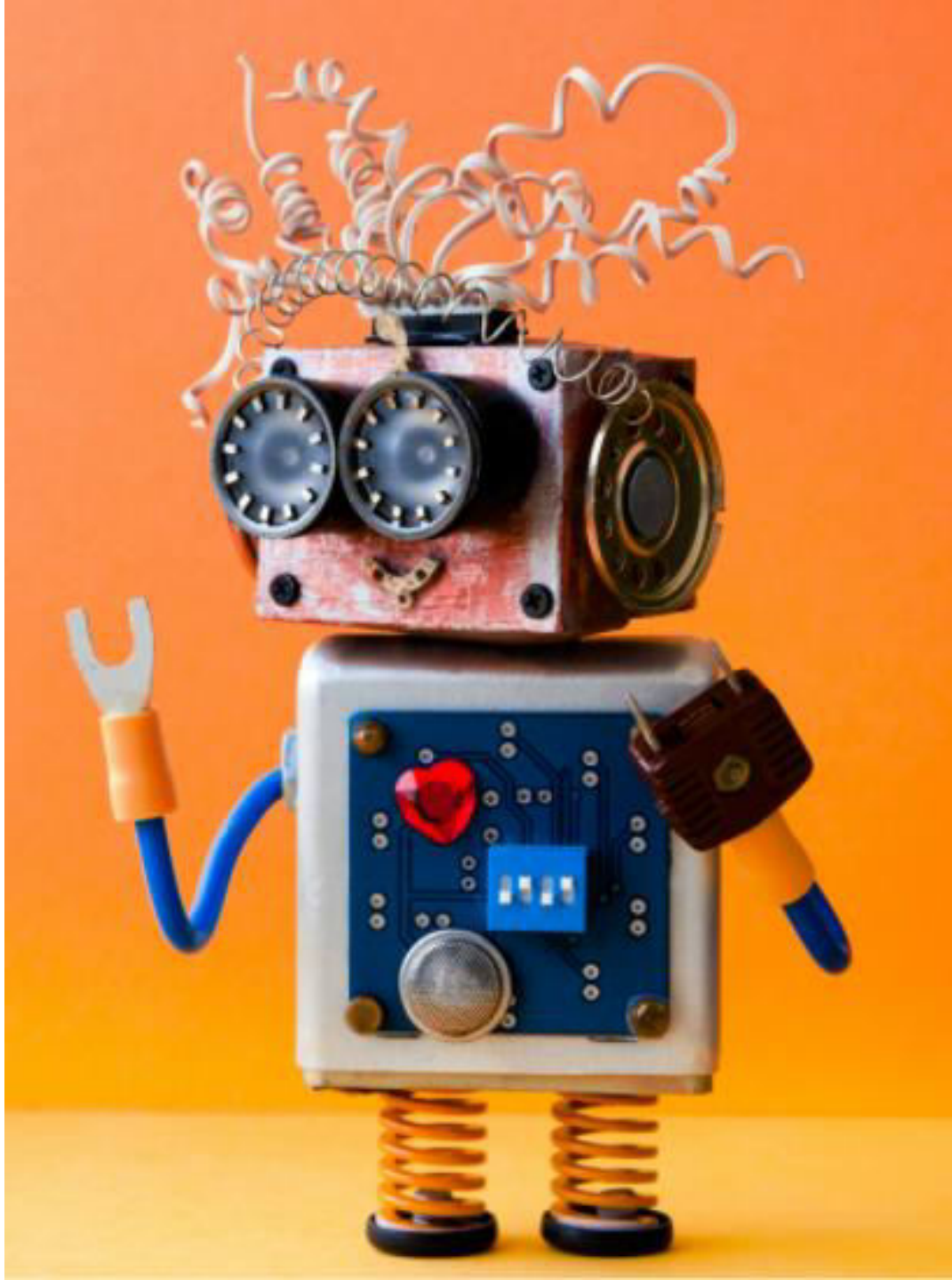
Survived

DATA WRANGLER DEMO

CHECK FOR DATA BIAS – UNBALANCED DATASETS

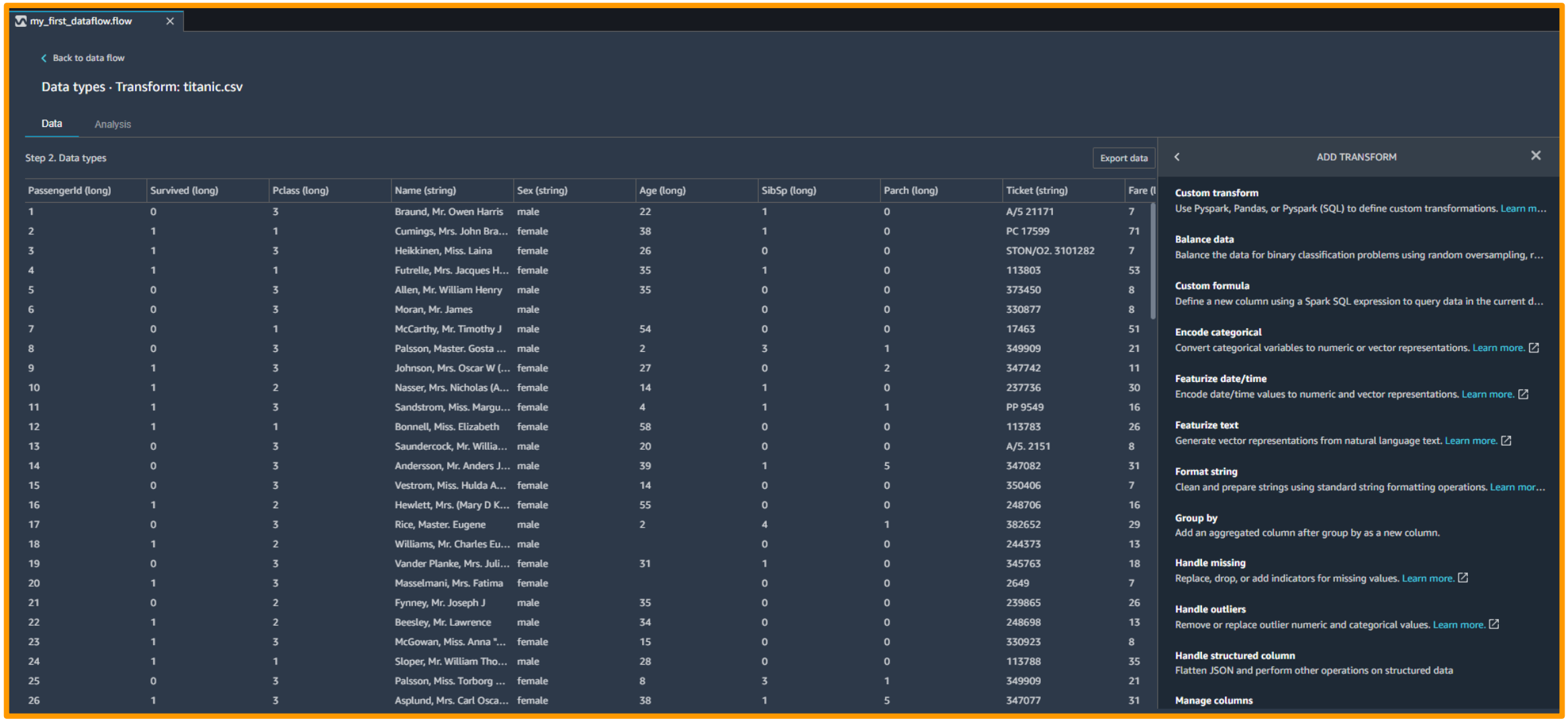


DATA WRANGLER DEMO #5: TRANSFORMATIONS (IMPUTE, DROP, & ENCODING)



DATA WRANGLER DEMO

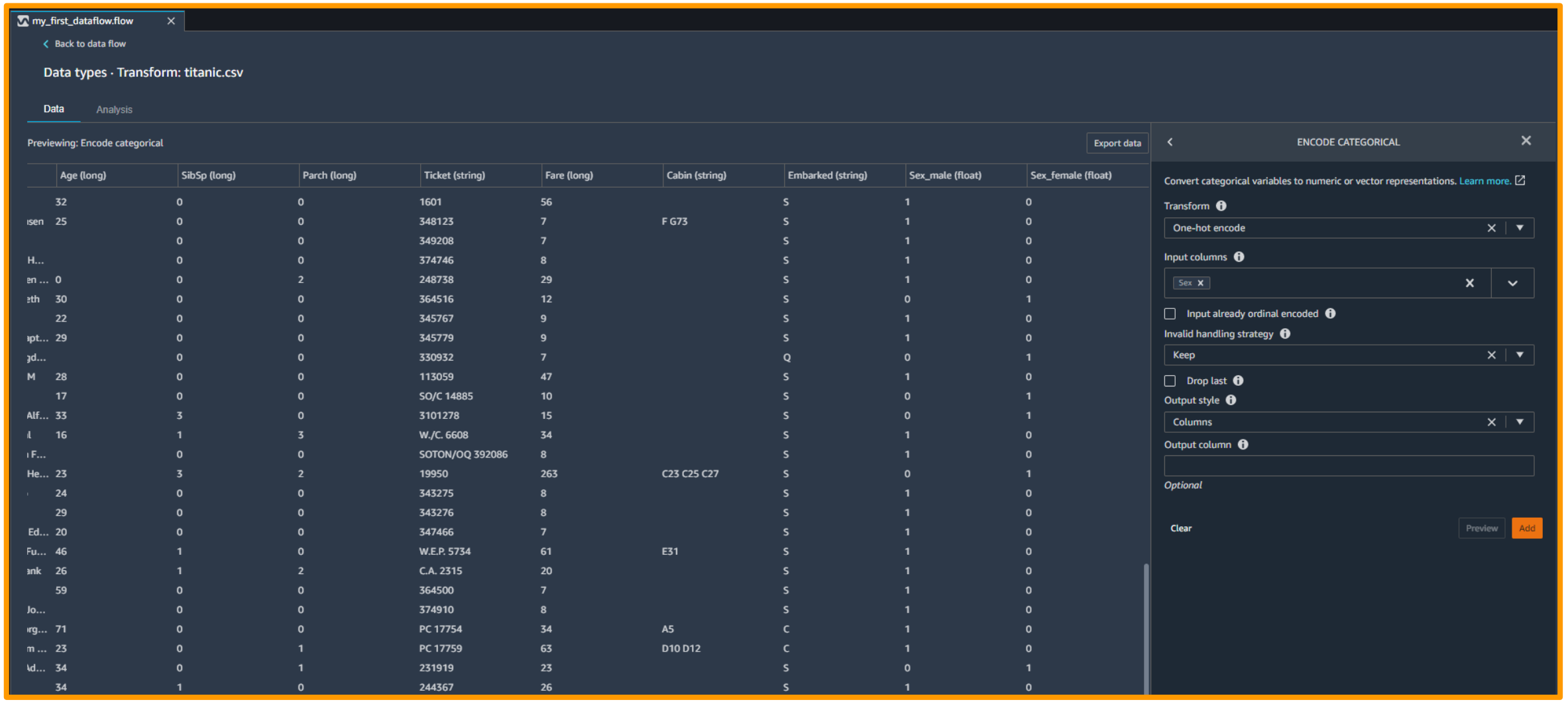
CLICK ON ADD TRANSFORM AND SELECT FROM 300+ TRANSFORMATIONS!!



PassengerId (long)	Survived (long)	Pclass (long)	Name (string)	Sex (string)	Age (long)	SibSp (long)	Parch (long)	Ticket (string)	Fare (l
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7
2	1	1	Cumings, Mrs. John Bra...	female	38	1	0	PC 17599	71
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7
4	1	1	Futrelle, Mrs. Jacques H...	female	35	1	0	113803	53
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8
6	0	3	Moran, Mr. James	male		0	0	330877	8
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51
8	0	3	Palsson, Master. Gosta ...	male	2	3	1	349909	21
9	1	3	Johnson, Mrs. Oscar W (...	female	27	0	2	347742	11
10	1	2	Nasser, Mrs. Nicholas (A...	female	14	1	0	237736	30
11	1	3	Sandstrom, Miss. Margu...	female	4	1	1	PP 9549	16
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26
13	0	3	Saundercock, Mr. Willia...	male	20	0	0	A/5. 2151	8
14	0	3	Andersson, Mr. Anders J...	male	39	1	5	347082	31
15	0	3	Vestrom, Miss. Hulda A...	female	14	0	0	350406	7
16	1	2	Hewlett, Mrs. (Mary D K...	female	55	0	0	248706	16
17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29
18	1	2	Williams, Mr. Charles Eu...	male		0	0	244373	13
19	0	3	Vander Planke, Mrs. Juli...	female	31	1	0	345763	18
20	1	3	Masselmani, Mrs. Fatima	female		0	0	2649	7
21	0	2	Fynney, Mr. Joseph J	male	35	0	0	239865	26
22	1	2	Beesley, Mr. Lawrence	male	34	0	0	248698	13
23	1	3	McGowan, Miss. Anna "...	female	15	0	0	330923	8
24	1	1	Sloper, Mr. William Tho...	male	28	0	0	113788	35
25	0	3	Palsson, Miss. Torborg ...	female	8	3	1	349909	21
26	1	3	Asplund, Mrs. Carl Osca...	female	38	1	5	347077	31

DATA WRANGLER DEMO

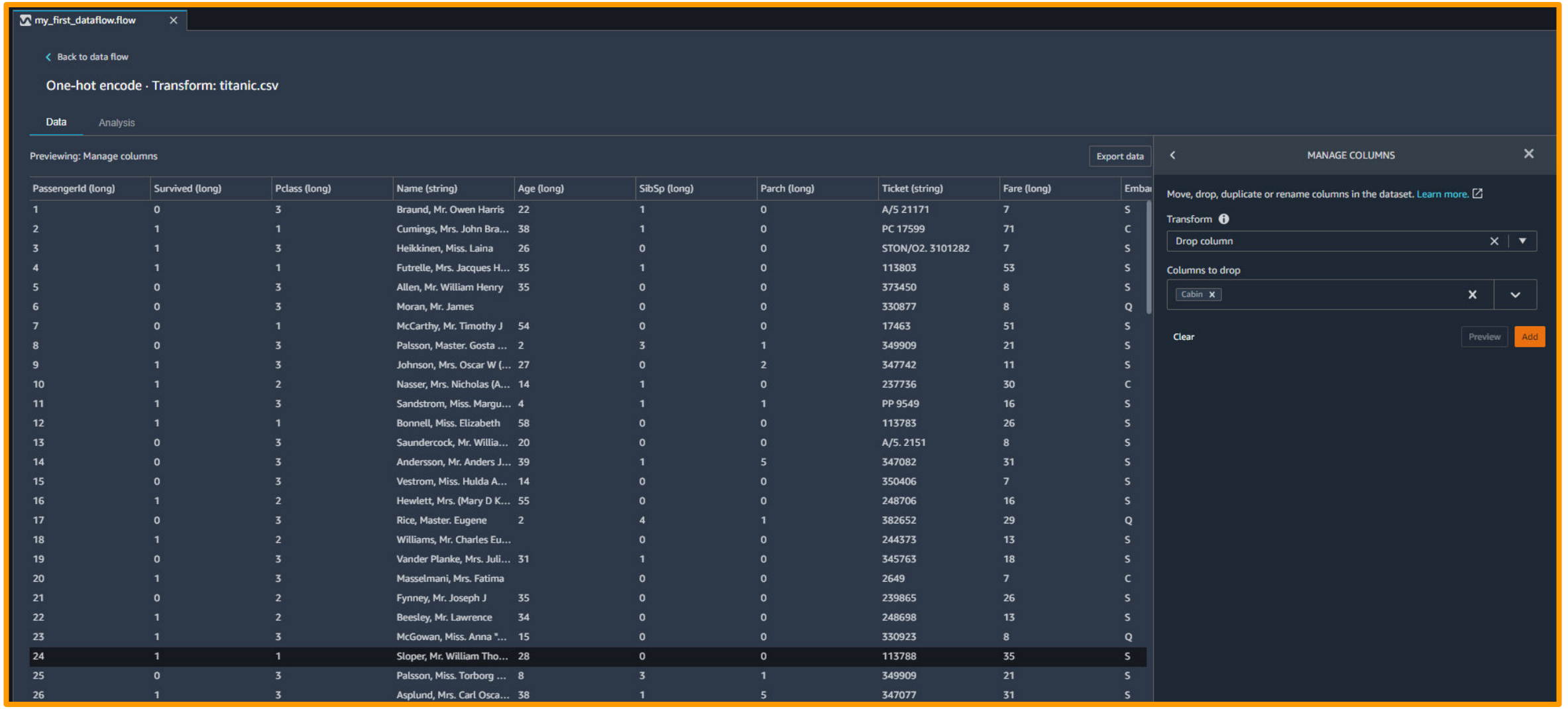
SELECT ONE HOT ENCODING FOR THE SEX COLUMN



	Age (long)	SibSp (long)	Parch (long)	Ticket (string)	Fare (long)	Cabin (string)	Embarked (string)	Sex_male (float)	Sex_female (float)
	32	0	0	1601	56		S	1	0
ısen	25	0	0	348123	7	F G73	S	1	0
		0	0	349208	7		S	1	0
H...		0	0	374746	8		S	1	0
en ...	0	0	2	248738	29		S	1	0
eth	30	0	0	364516	12		S	0	1
	22	0	0	345767	9		S	1	0
apt...	29	0	0	345779	9		S	1	0
gd...		0	0	330932	7		Q	0	1
M	28	0	0	113059	47		S	1	0
	17	0	0	SO/C 14885	10		S	0	1
Alf...	33	3	0	3101278	15		S	0	1
l	16	1	3	W./C. 6608	34		S	1	0
l F...		0	0	SOTON/OQ 392086	8		S	1	0
He...	23	3	2	19950	263	C23 C25 C27	S	0	1
	24	0	0	343275	8		S	1	0
	29	0	0	343276	8		S	1	0
Ed...	20	0	0	347466	7		S	1	0
Fu...	46	1	0	W.E.P. 5734	61	E31	S	1	0
ank	26	1	2	C.A. 2315	20		S	1	0
	59	0	0	364500	7		S	1	0
Jo...		0	0	374910	8		S	1	0
rg...	71	0	0	PC 17754	34	A5	C	1	0
m ...	23	0	1	PC 17759	63	D10 D12	C	1	0
ld...	34	0	1	231919	23		S	0	1
	34	1	0	244367	26		S	1	0

DATA WRANGLER DEMO

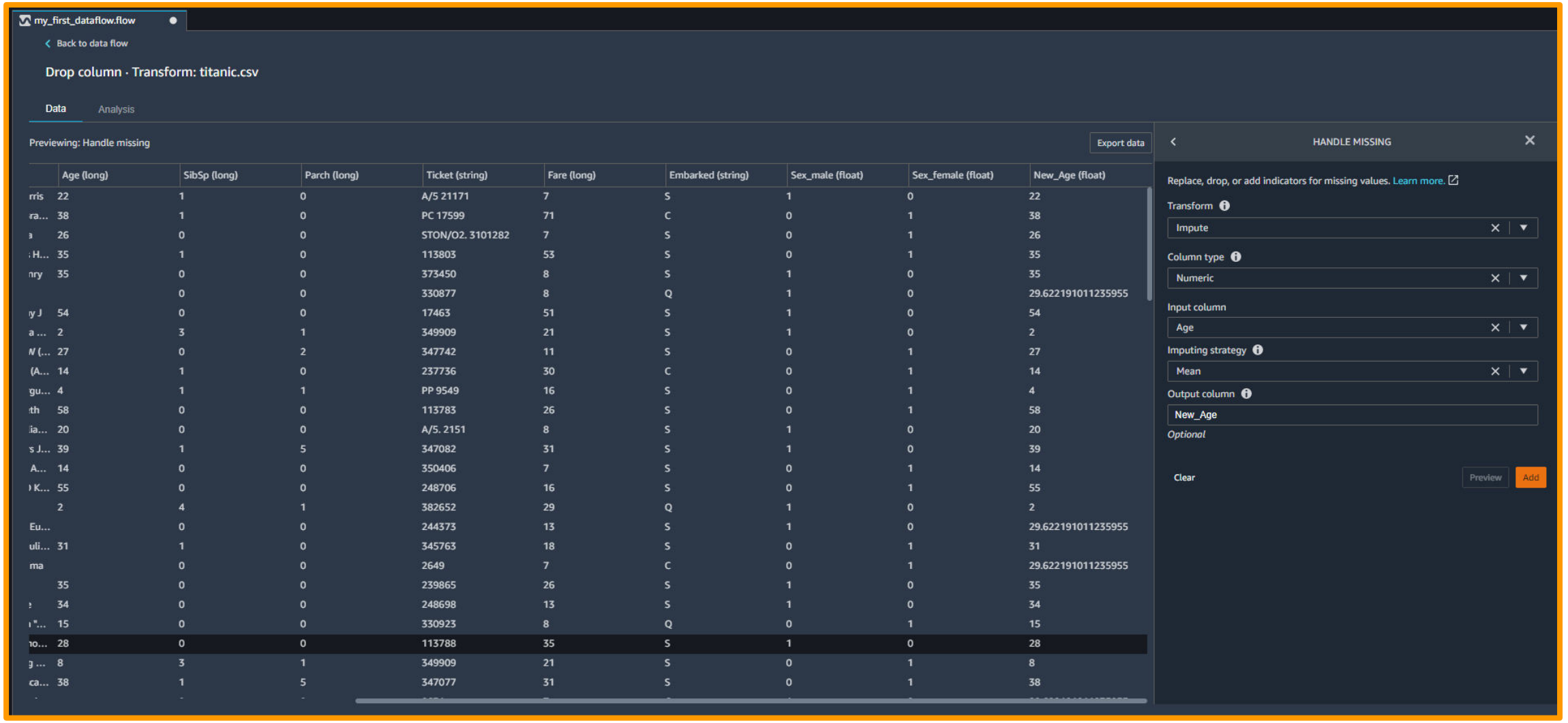
LET'S DROP THE CABIN COLUMN



PassengerId (long)	Survived (long)	Pclass (long)	Name (string)	Age (long)	SibSp (long)	Parch (long)	Ticket (string)	Fare (long)	Embar
1	0	3	Braund, Mr. Owen Harris	22	1	0	A/5 21171	7	S
2	1	1	Cumings, Mrs. John Bra...	38	1	0	PC 17599	71	C
3	1	3	Heikkinen, Miss. Laina	26	0	0	STON/O2. 3101282	7	S
4	1	1	Futrelle, Mrs. Jacques H...	35	1	0	113803	53	S
5	0	3	Allen, Mr. William Henry	35	0	0	373450	8	S
6	0	3	Moran, Mr. James		0	0	330877	8	Q
7	0	1	McCarthy, Mr. Timothy J	54	0	0	17463	51	S
8	0	3	Palsson, Master. Gosta ...	2	3	1	349909	21	S
9	1	3	Johnson, Mrs. Oscar W (...	27	0	2	347742	11	S
10	1	2	Nasser, Mrs. Nicholas (A...	14	1	0	237736	30	C
11	1	3	Sandstrom, Miss. Margu...	4	1	1	PP 9549	16	S
12	1	1	Bonnell, Miss. Elizabeth	58	0	0	113783	26	S
13	0	3	Saundercock, Mr. Willia...	20	0	0	A/5. 2151	8	S
14	0	3	Andersson, Mr. Anders J...	39	1	5	347082	31	S
15	0	3	Vestrom, Miss. Hulda A...	14	0	0	350406	7	S
16	1	2	Hewlett, Mrs. (Mary D K...	55	0	0	248706	16	S
17	0	3	Rice, Master. Eugene	2	4	1	382652	29	Q
18	1	2	Williams, Mr. Charles Eu...		0	0	244373	13	S
19	0	3	Vander Planke, Mrs. Juli...	31	1	0	345763	18	S
20	1	3	Masselmani, Mrs. Fatima		0	0	2649	7	C
21	0	2	Fynney, Mr. Joseph J	35	0	0	239865	26	S
22	1	2	Beesley, Mr. Lawrence	34	0	0	248698	13	S
23	1	3	McGowan, Miss. Anna "...	15	0	0	330923	8	Q
24	1	1	Sloper, Mr. William Tho...	28	0	0	113788	35	S
25	0	3	Palsson, Miss. Torborg ...	8	3	1	349909	21	S
26	1	3	Asplund, Mrs. Carl Osca...	38	1	5	347077	31	S

DATA WRANGLER DEMO

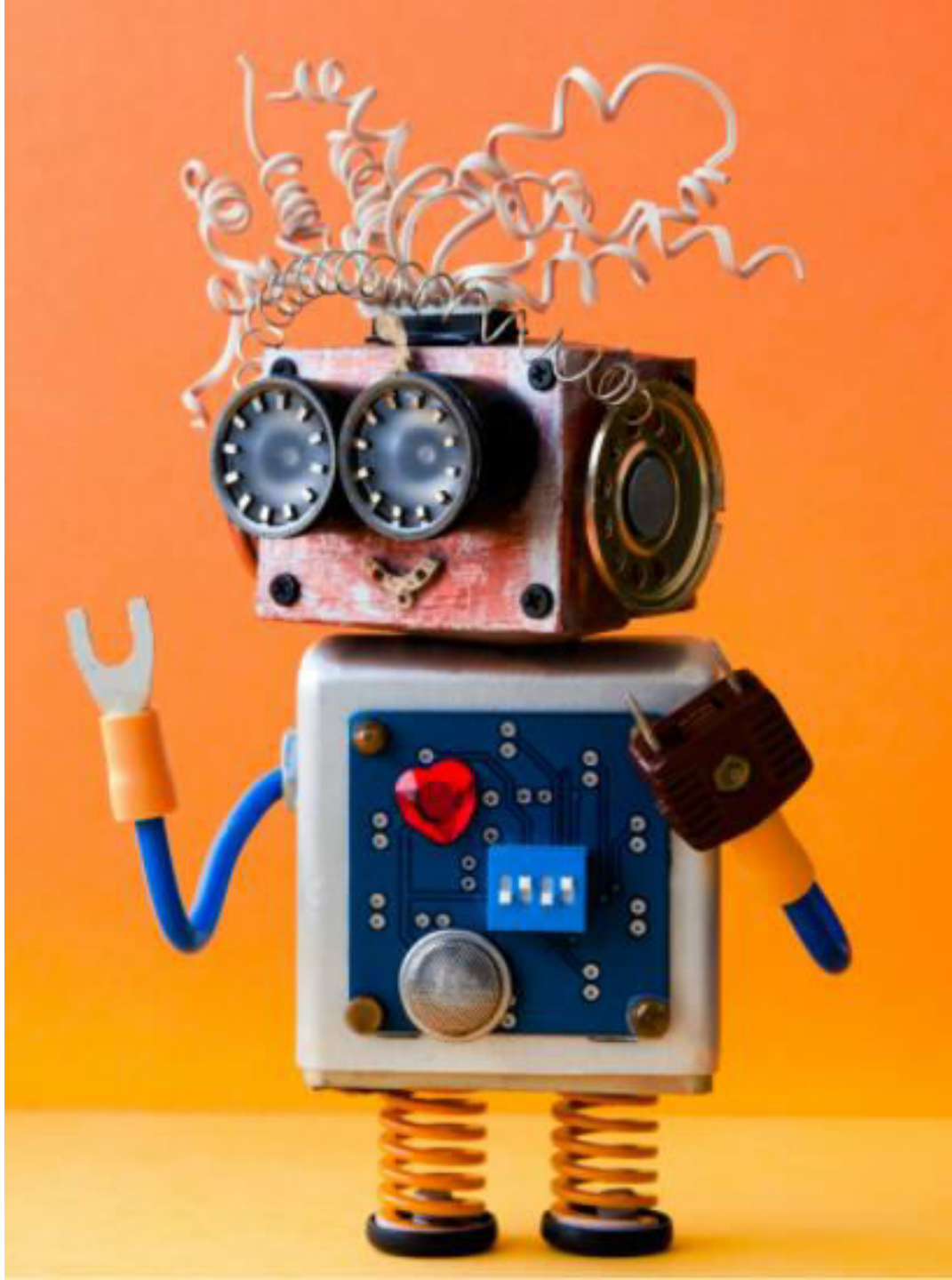
FILL OUT MISSING AGE VALUES



	Age (long)	SibSp (long)	Parch (long)	Ticket (string)	Fare (long)	Embarked (string)	Sex_male (float)	Sex_female (float)	New_Age (float)
rris	22	1	0	A/5 21171	7	S	1	0	22
ra...	38	1	0	PC 17599	71	C	0	1	38
a	26	0	0	STON/O2. 3101282	7	S	0	1	26
: H...	35	1	0	113803	53	S	0	1	35
nry	35	0	0	373450	8	S	1	0	35
		0	0	330877	8	Q	1	0	29.622191011235955
y J	54	0	0	17463	51	S	1	0	54
a ...	2	3	1	349909	21	S	1	0	2
N (...	27	0	2	347742	11	S	0	1	27
(A...	14	1	0	237736	30	C	0	1	14
gu...	4	1	1	PP 9549	16	S	0	1	4
th	58	0	0	113783	26	S	0	1	58
ia...	20	0	0	A/5. 2151	8	S	1	0	20
s J...	39	1	5	347082	31	S	1	0	39
A...	14	0	0	350406	7	S	0	1	14
) K...	55	0	0	248706	16	S	0	1	55
	2	4	1	382652	29	Q	1	0	2
Eu...		0	0	244373	13	S	1	0	29.622191011235955
uli...	31	1	0	345763	18	S	0	1	31
ma		0	0	2649	7	C	0	1	29.622191011235955
	35	0	0	239865	26	S	1	0	35
	34	0	0	248698	13	S	1	0	34
ı "...	15	0	0	330923	8	Q	0	1	15
no...	28	0	0	113788	35	S	1	0	28
g ...	8	3	1	349909	21	S	0	1	8
ca...	38	1	5	347077	31	S	0	1	38

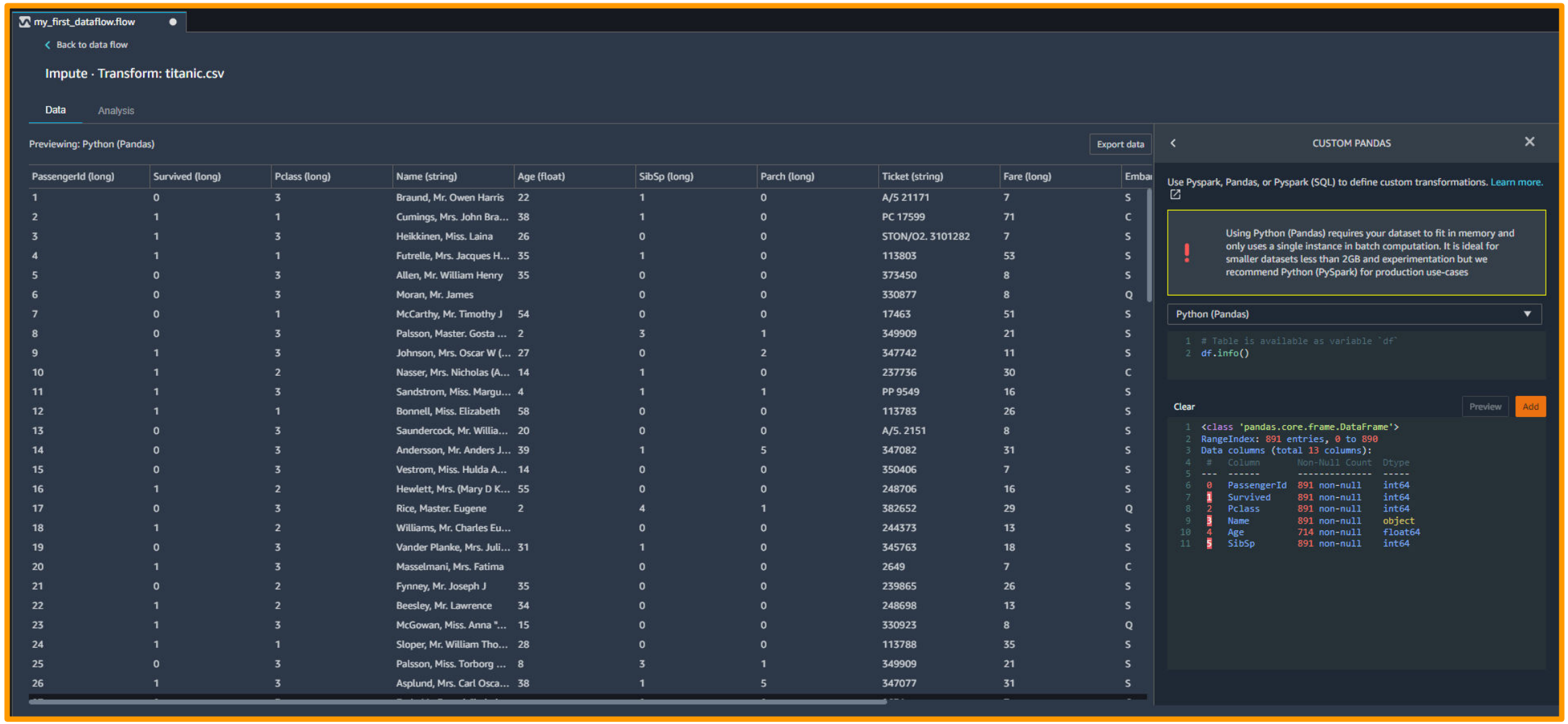
DATA WRANGLER DEMO #6: TRANSFORMATIONS (CUSTOM TRANSFORM & SCALING)

EASY ADVANCED



DATA WRANGLER DEMO

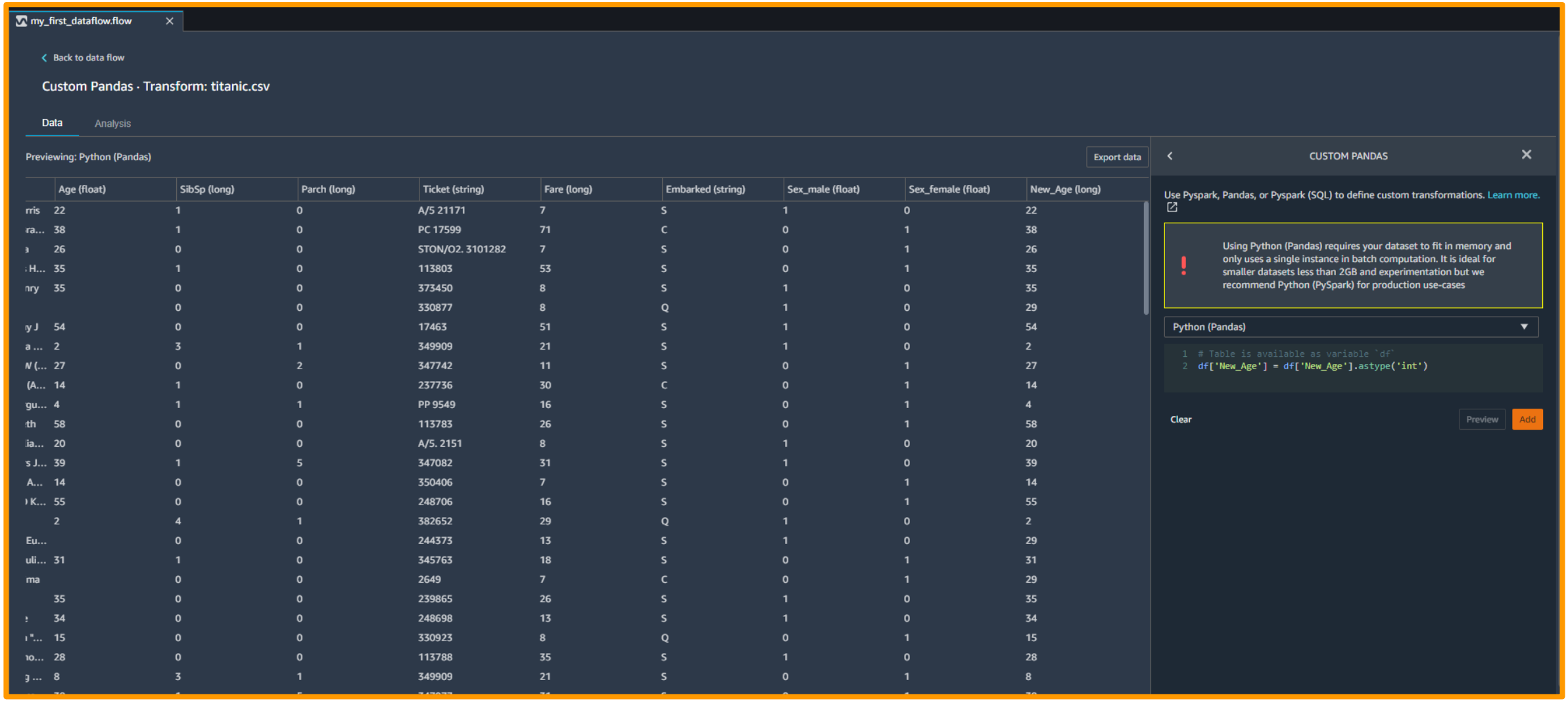
INTEGRATE A CUSTOM-BASED info() METHOD INTO THE WORKFLOW



PassengerId (long)	Survived (long)	Pclass (long)	Name (string)	Age (float)	SibSp (long)	Parch (long)	Ticket (string)	Fare (long)	Emba
1	0	3	Braund, Mr. Owen Harris	22	1	0	A/5 21171	7	S
2	1	1	Cumings, Mrs. John Bra...	38	1	0	PC 17599	71	C
3	1	3	Heikkinen, Miss. Laina	26	0	0	STON/O2. 3101282	7	S
4	1	1	Futrelle, Mrs. Jacques H...	35	1	0	113803	53	S
5	0	3	Allen, Mr. William Henry	35	0	0	373450	8	S
6	0	3	Moran, Mr. James		0	0	330877	8	Q
7	0	1	McCarthy, Mr. Timothy J	54	0	0	17463	51	S
8	0	3	Palsson, Master. Gosta ...	2	3	1	349909	21	S
9	1	3	Johnson, Mrs. Oscar W (...	27	0	2	347742	11	S
10	1	2	Nasser, Mrs. Nicholas (A...	14	1	0	237736	30	C
11	1	3	Sandstrom, Miss. Margu...	4	1	1	PP 9549	16	S
12	1	1	Bonnell, Miss. Elizabeth	58	0	0	113783	26	S
13	0	3	Saundercock, Mr. Willia...	20	0	0	A/5. 2151	8	S
14	0	3	Andersson, Mr. Anders J...	39	1	5	347082	31	S
15	0	3	Vestrom, Miss. Hulda A...	14	0	0	350406	7	S
16	1	2	Hewlett, Mrs. (Mary D K...	55	0	0	248706	16	S
17	0	3	Rice, Master. Eugene	2	4	1	382652	29	Q
18	1	2	Williams, Mr. Charles Eu...		0	0	244373	13	S
19	0	3	Vander Planke, Mrs. Juli...	31	1	0	345763	18	S
20	1	3	Masselmani, Mrs. Fatima		0	0	2649	7	C
21	0	2	Fynney, Mr. Joseph J	35	0	0	239865	26	S
22	1	2	Beesley, Mr. Lawrence	34	0	0	248698	13	S
23	1	3	McGowan, Miss. Anna "...	15	0	0	330923	8	Q
24	1	1	Sloper, Mr. William Tho...	28	0	0	113788	35	S
25	0	3	Palsson, Miss. Torborg ...	8	3	1	349909	21	S
26	1	3	Asplund, Mrs. Carl Osca...	38	1	5	347077	31	S

DATA WRANGLER DEMO

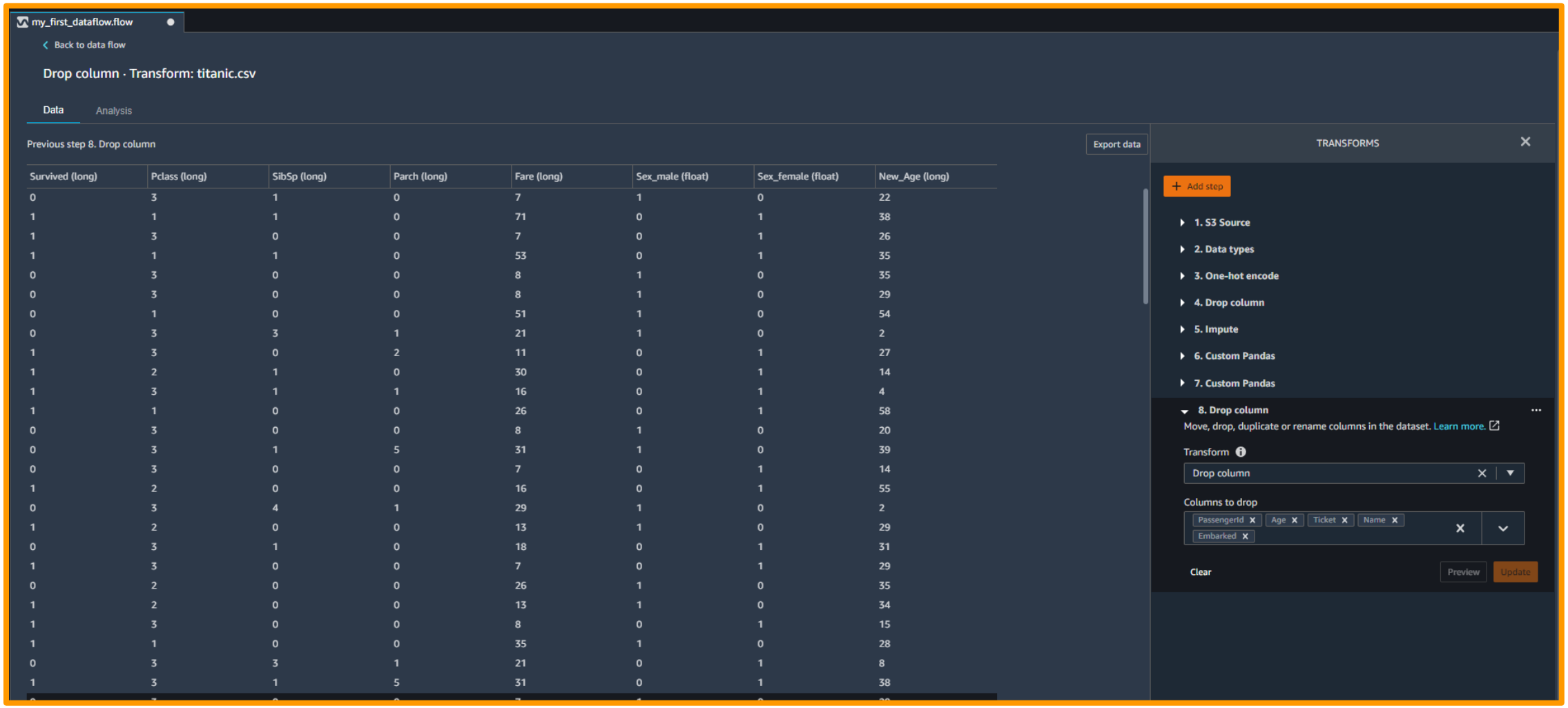
LET'S CHANGE THE DATATYPE USING CODE!



	Age (float)	SibSp (long)	Parch (long)	Ticket (string)	Fare (long)	Embarked (string)	Sex_male (float)	Sex_female (float)	New_Age (long)
rris	22	1	0	A/5 21171	7	S	1	0	22
ra...	38	1	0	PC 17599	71	C	0	1	38
a	26	0	0	STON/O2. 3101282	7	S	0	1	26
H...	35	1	0	113803	53	S	0	1	35
nry	35	0	0	373450	8	S	1	0	35
		0	0	330877	8	Q	1	0	29
y J	54	0	0	17463	51	S	1	0	54
a ...	2	3	1	349909	21	S	1	0	2
N (...	27	0	2	347742	11	S	0	1	27
(A...	14	1	0	237736	30	C	0	1	14
gu...	4	1	1	PP 9549	16	S	0	1	4
th	58	0	0	113783	26	S	0	1	58
ia...	20	0	0	A/5. 2151	8	S	1	0	20
s J...	39	1	5	347082	31	S	1	0	39
A...	14	0	0	350406	7	S	0	1	14
) K...	55	0	0	248706	16	S	0	1	55
	2	4	1	382652	29	Q	1	0	2
Eu...		0	0	244373	13	S	1	0	29
uli...	31	1	0	345763	18	S	0	1	31
ma		0	0	2649	7	C	0	1	29
	35	0	0	239865	26	S	1	0	35
	34	0	0	248698	13	S	1	0	34
...	15	0	0	330923	8	Q	0	1	15
no...	28	0	0	113788	35	S	1	0	28
g ...	8	3	1	349909	21	S	0	1	8

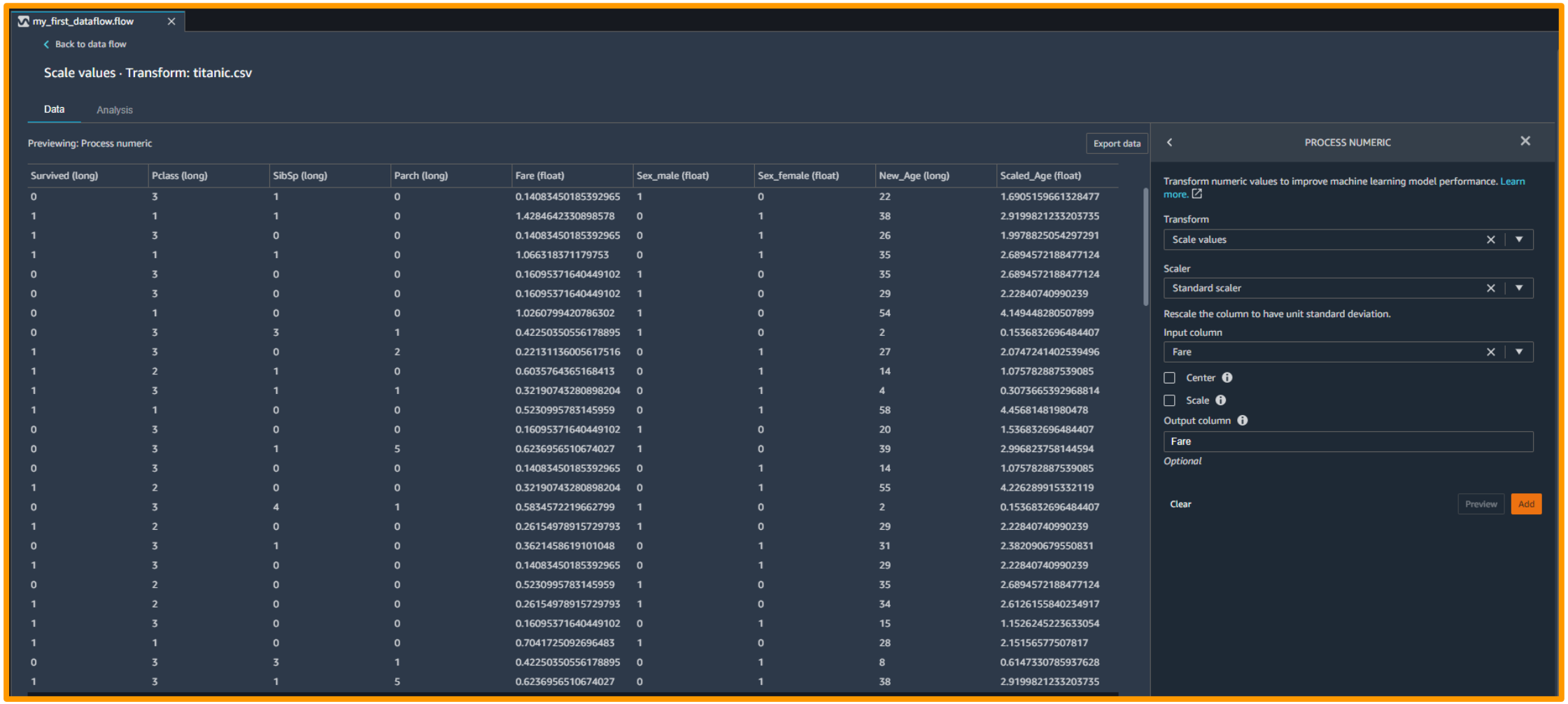
DATA WRANGLER DEMO

LET'S DROP THE PASSENGER ID, EMBARKED, TICKET NUMBER, NAME AND THE AGE (NOTE THAT WE HAVE A NEW_AGE COLUMN AFTER WE FILLED OUT MISSING ROWS!



DATA WRANGLER DEMO

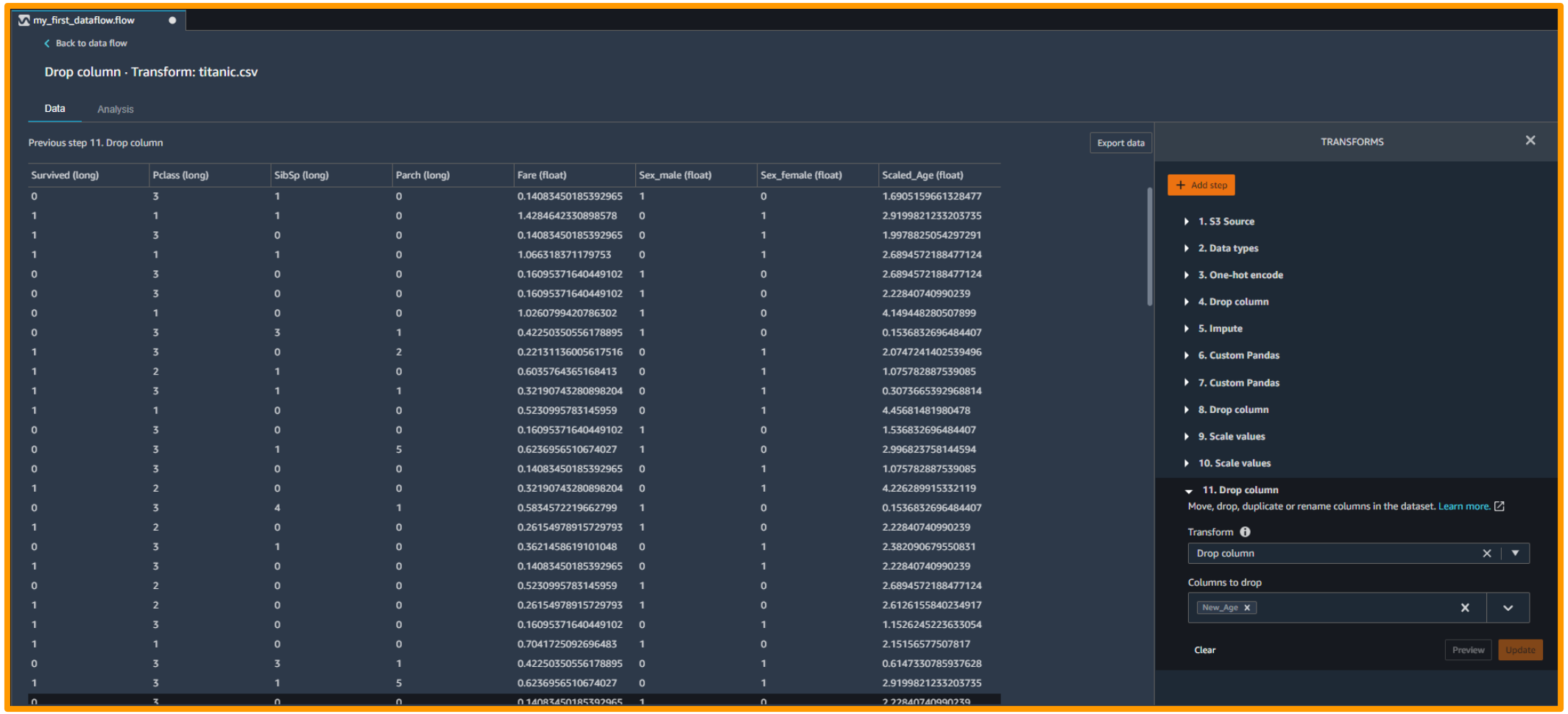
LET'S PERFORM STANDARD SCALER WHICH WILL TRANSFORM NUMERIC DATA INTO ZERO MEAN. NOTE THAT YOU CAN OVERRIDE THE ORIGINAL COLUMN NAME



Survived (long)	Pclass (long)	SibSp (long)	Parch (long)	Fare (float)	Sex_male (float)	Sex_female (float)	New_Age (long)	Scaled_Age (float)
0	3	1	0	0.14083450185392965	1	0	22	1.6905159661328477
1	1	1	0	1.4284642330898578	0	1	38	2.9199821233203735
1	3	0	0	0.14083450185392965	0	1	26	1.9978825054297291
1	1	1	0	1.066318371179753	0	1	35	2.6894572188477124
0	3	0	0	0.16095371640449102	1	0	35	2.6894572188477124
0	3	0	0	0.16095371640449102	1	0	29	2.22840740990239
0	1	0	0	1.0260799420786302	1	0	54	4.149448280507899
0	3	3	1	0.42250350556178895	1	0	2	0.1536832696484407
1	3	0	2	0.22131136005617516	0	1	27	2.0747241402539496
1	2	1	0	0.6035764365168413	0	1	14	1.075782887539085
1	3	1	1	0.32190743280898204	0	1	4	0.3073665392968814
1	1	0	0	0.5230995783145959	0	1	58	4.45681481980478
0	3	0	0	0.16095371640449102	1	0	20	1.536832696484407
0	3	1	5	0.6236956510674027	1	0	39	2.996823758144594
0	3	0	0	0.14083450185392965	0	1	14	1.075782887539085
1	2	0	0	0.32190743280898204	0	1	55	4.226289915332119
0	3	4	1	0.5834572219662799	1	0	2	0.1536832696484407
1	2	0	0	0.26154978915729793	1	0	29	2.22840740990239
0	3	1	0	0.3621458619101048	0	1	31	2.382090679550831
1	3	0	0	0.14083450185392965	0	1	29	2.22840740990239
0	2	0	0	0.5230995783145959	1	0	35	2.6894572188477124
1	2	0	0	0.26154978915729793	1	0	34	2.6126155840234917
1	3	0	0	0.16095371640449102	0	1	15	1.1526245223633054
1	1	0	0	0.7041725092696483	1	0	28	2.15156577507817
0	3	3	1	0.42250350556178895	0	1	8	0.6147330785937628
1	3	1	5	0.6236956510674027	0	1	38	2.9199821233203735

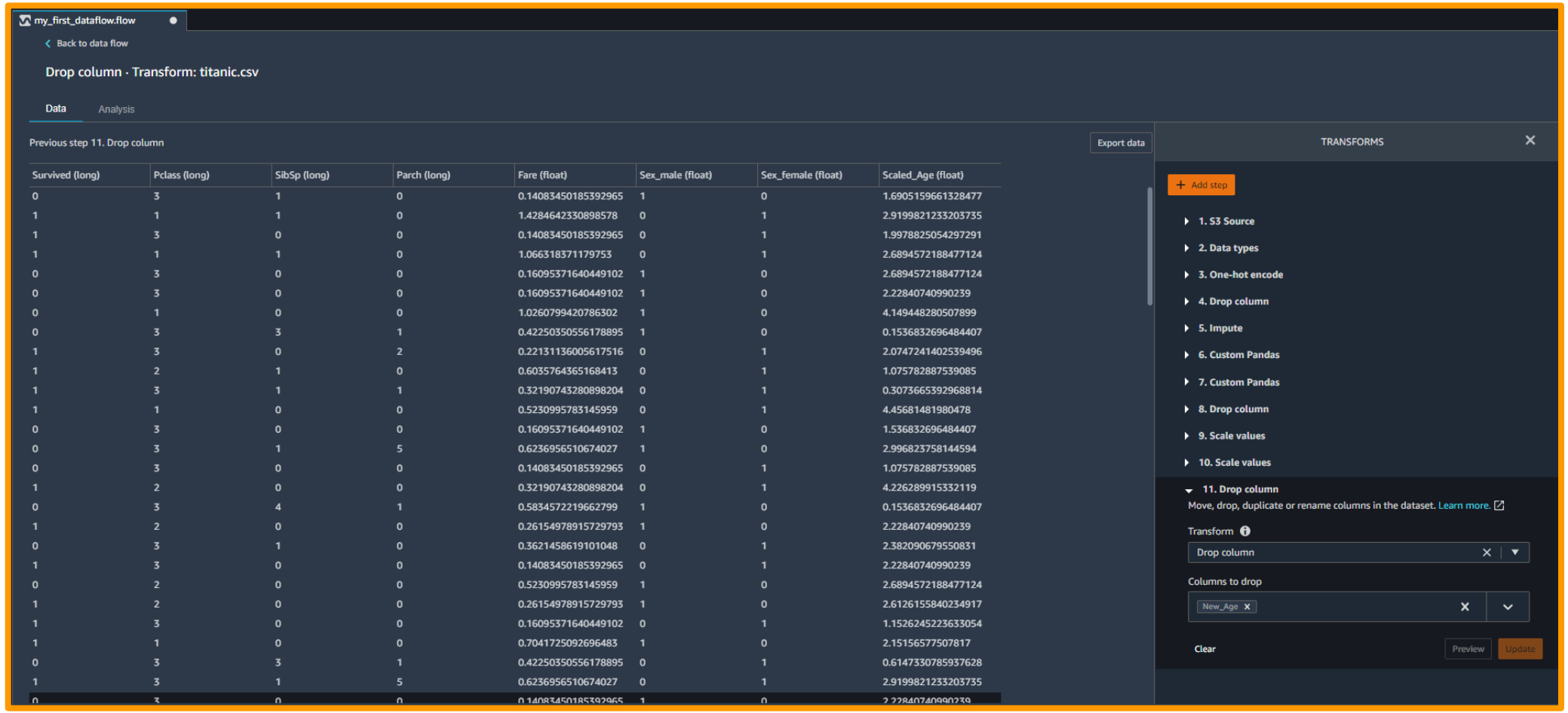
DATA WRANGLER DEMO

LET'S DROP THE AGE, NOTE THAT WE COULD HAVE REMOVED THIS STEP IF WE PUT THE OUTPUT COLUMN NAME SIMILAR TO THE INPUT



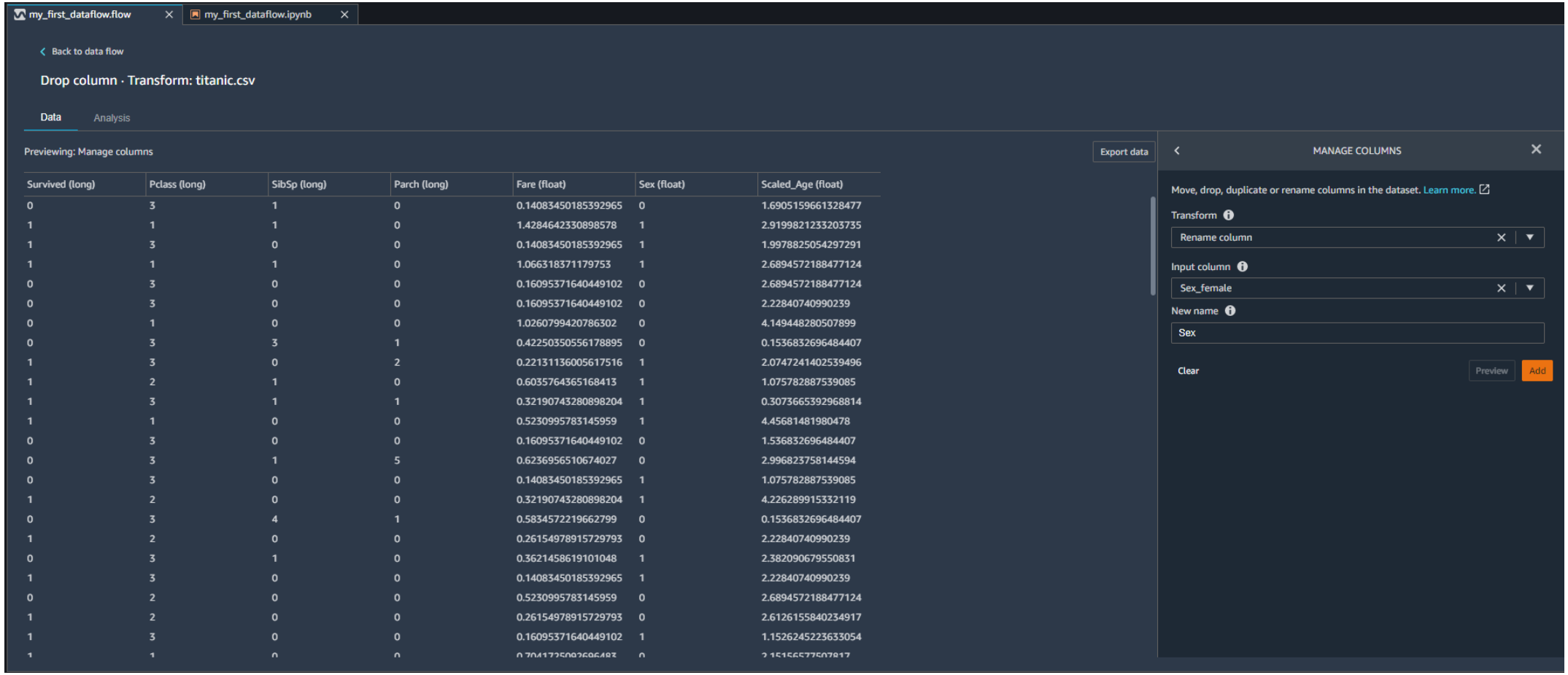
DATA WRANGLER DEMO

ONE LAST STEP! NOTE THAT SEX_FEMALE AND SEX_MALE ARE NEGATIVELY CORRELATED. DROP ONE OF THEM BEFORE YOU TRAIN YOUR MACHINE LEARNING MODEL



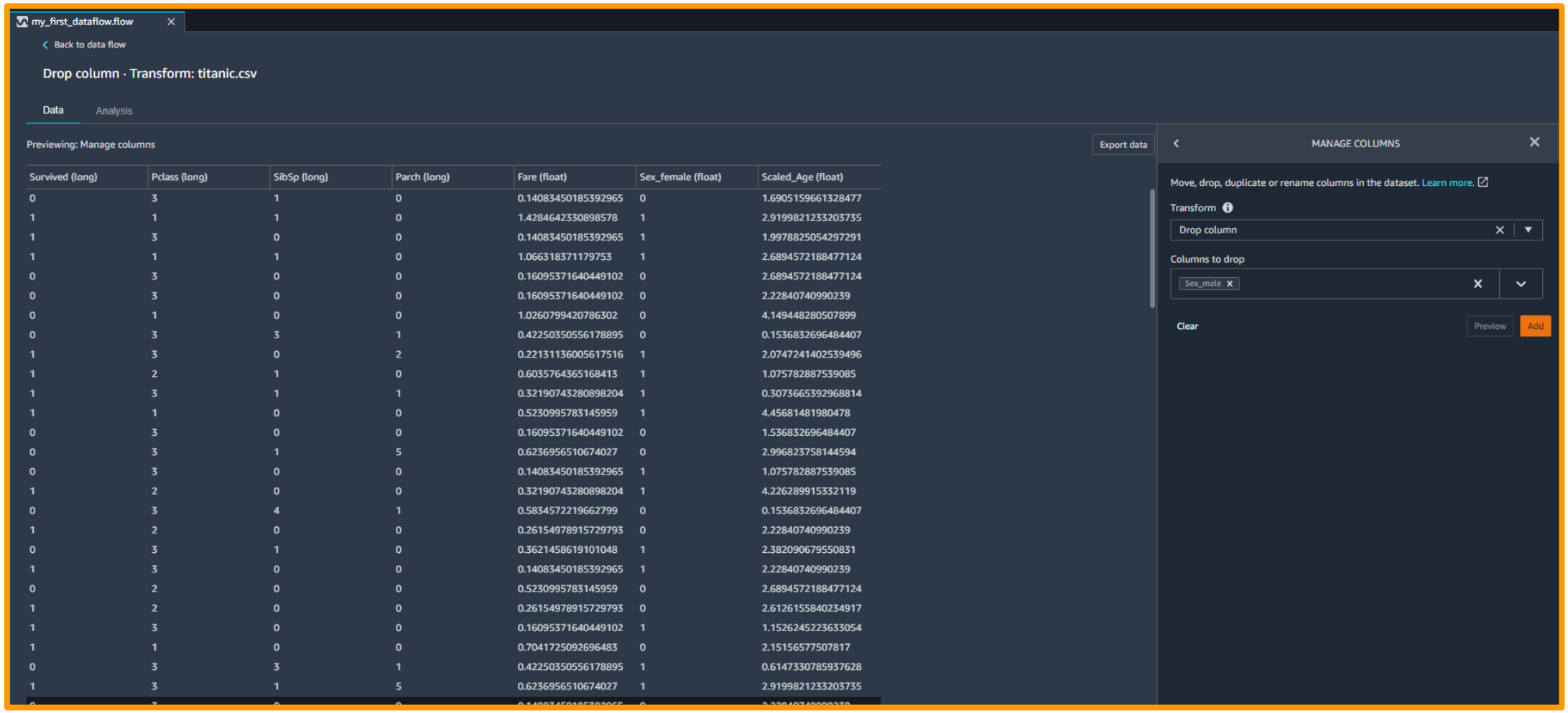
DATA WRANGLER DEMO

RENAME COLUMN



DATA WRANGLER DEMO

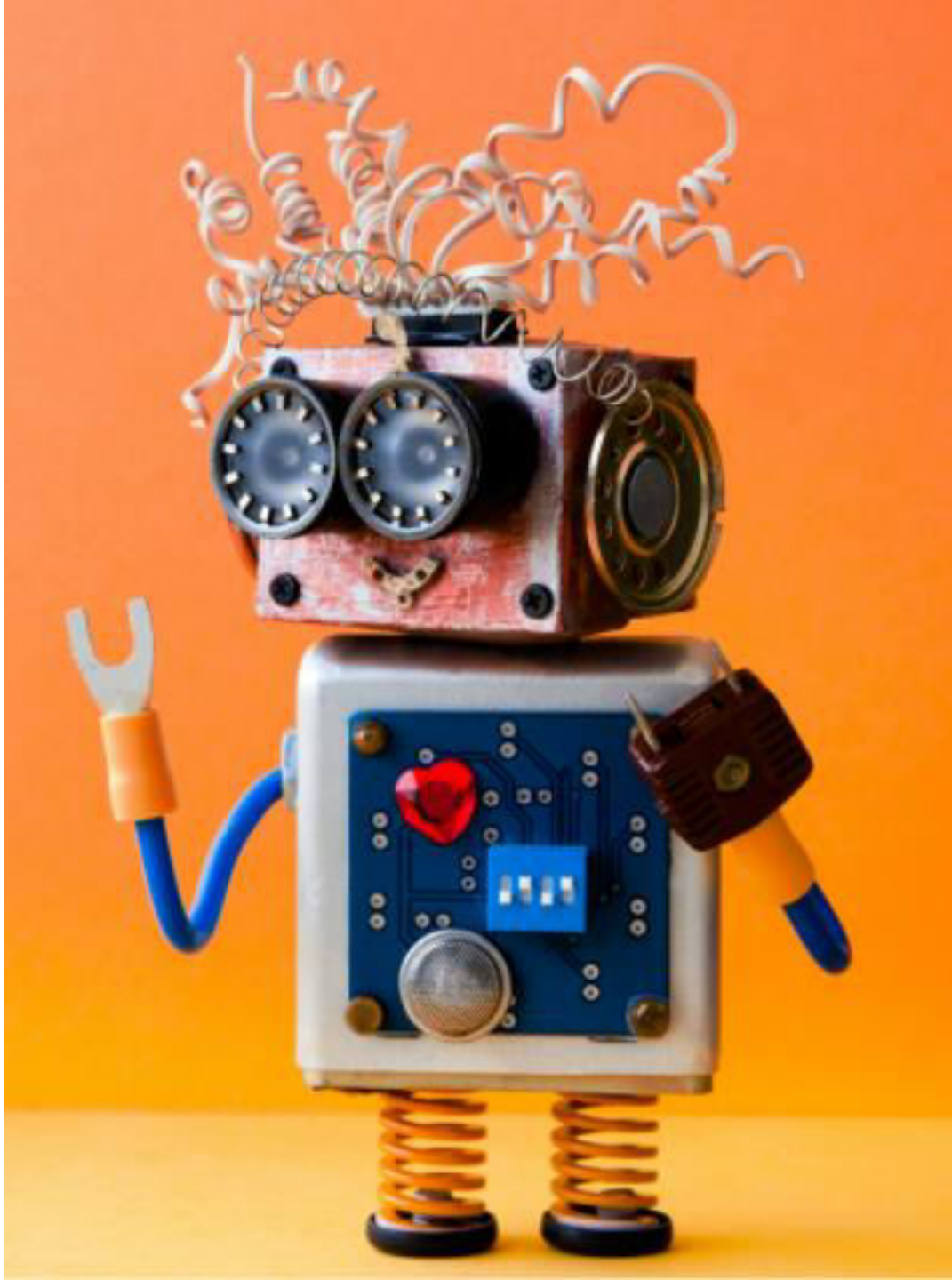
NOW YOU'RE GOOD TO GO!



Survived (long)	Pclass (long)	SibSp (long)	Parch (long)	Fare (float)	Sex_female (float)	Scaled_Age (float)
0	3	1	0	0.14083450185392965	0	1.6905159661328477
1	1	1	0	1.4284642330898578	1	2.9199821233203735
1	3	0	0	0.14083450185392965	1	1.9978825054297291
1	1	1	0	1.066318371179753	1	2.6894572188477124
0	3	0	0	0.16095371640449102	0	2.6894572188477124
0	3	0	0	0.16095371640449102	0	2.22840740990239
0	1	0	0	1.0260799420786302	0	4.149448280507899
0	3	3	1	0.42250350556178895	0	0.1536832696484407
1	3	0	2	0.22131136005617516	1	2.0747241402539496
1	2	1	0	0.6035764365168413	1	1.075782887539085
1	3	1	1	0.32190743280898204	1	0.3073665392968814
1	1	0	0	0.5230995783145959	1	4.45681481980478
0	3	0	0	0.16095371640449102	0	1.536832696484407
0	3	1	5	0.6236956510674027	0	2.996823758144594
0	3	0	0	0.14083450185392965	1	1.075782887539085
1	2	0	0	0.32190743280898204	1	4.226289915332119
0	3	4	1	0.5834572219662799	0	0.1536832696484407
1	2	0	0	0.26154978915729793	0	2.22840740990239
0	3	1	0	0.3621458619101048	1	2.382090679550831
1	3	0	0	0.14083450185392965	1	2.22840740990239
0	2	0	0	0.5230995783145959	0	2.6894572188477124
1	2	0	0	0.26154978915729793	0	2.6126155840234917
1	3	0	0	0.16095371640449102	1	1.1526245223633054
1	1	0	0	0.7041725092696483	0	2.15156577507817
0	3	3	1	0.42250350556178895	1	0.6147330785937628
1	3	1	5	0.6236956510674027	1	2.9199821233203735

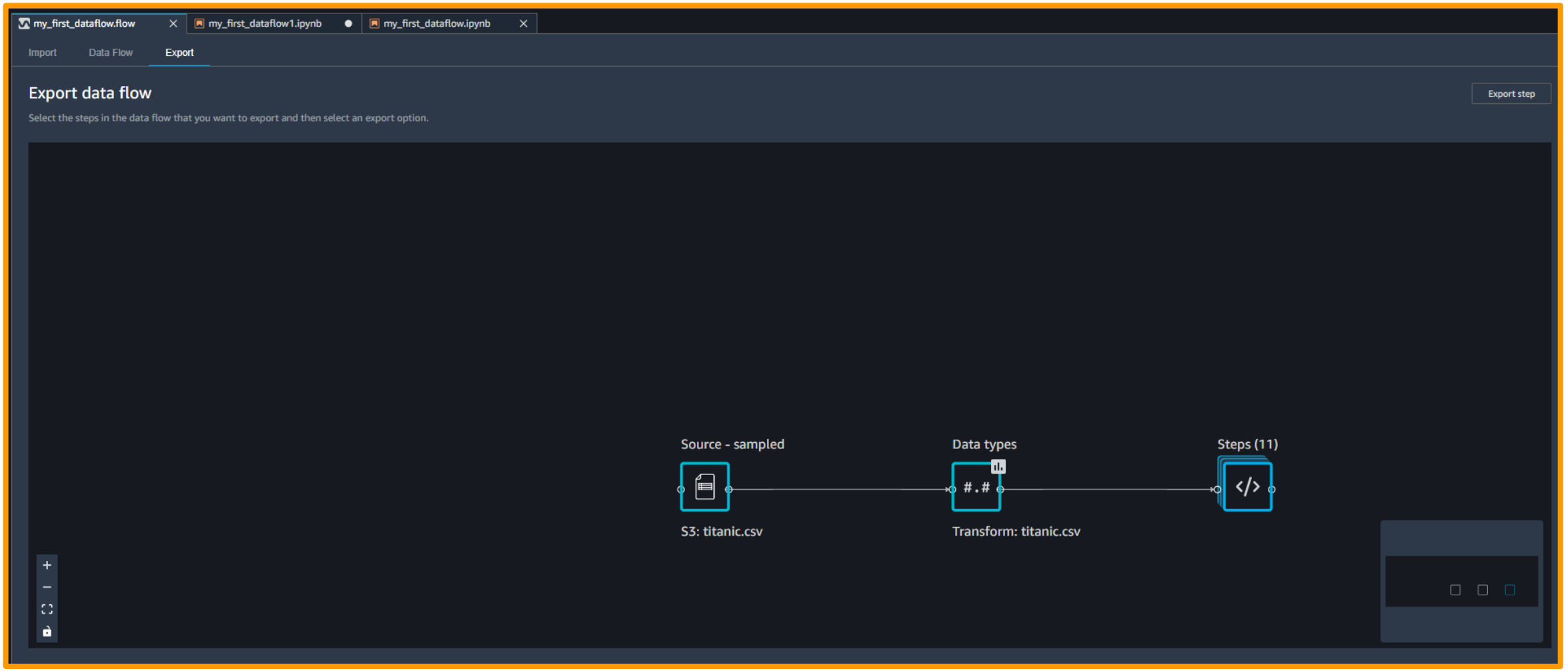
DATA WRANGLER DEMO #7: EXPORT WORKFLOW

EASY ADVANCED



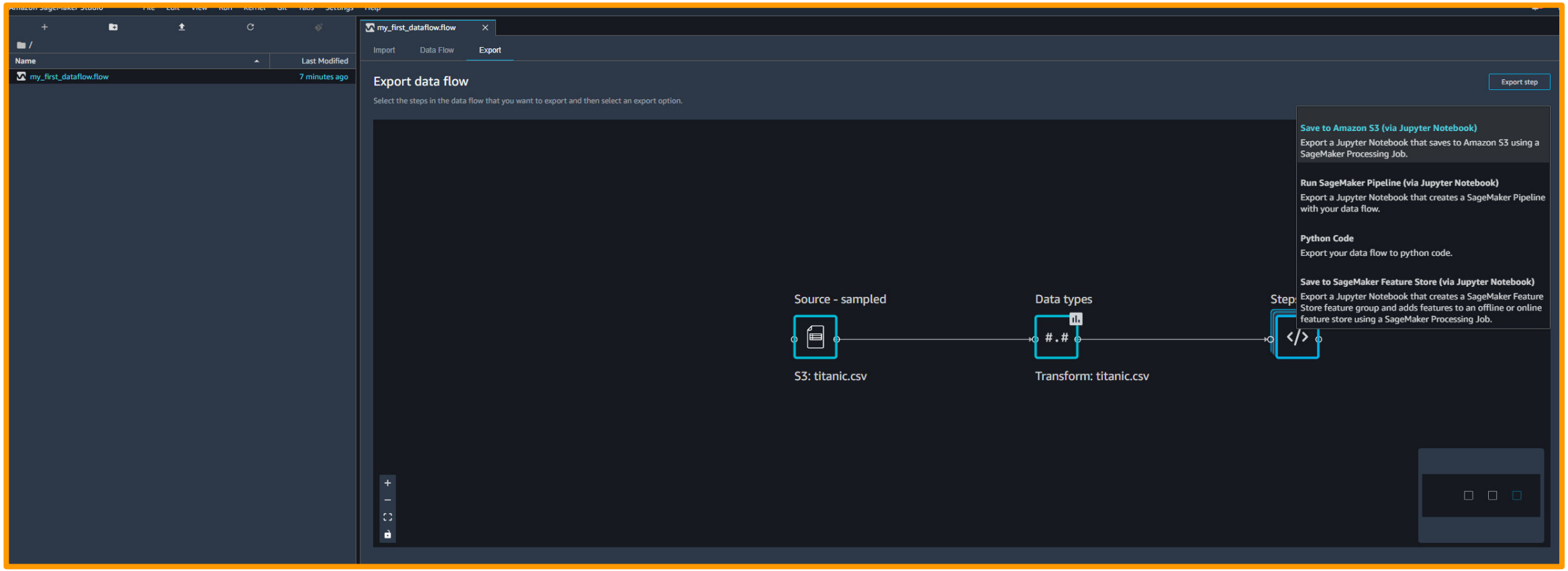
DATA WRANGLER DEMO

GO THE EXPORT TAB AND CLICK ON THE LAST STEP TO INCLUDE ALL THE STEPS



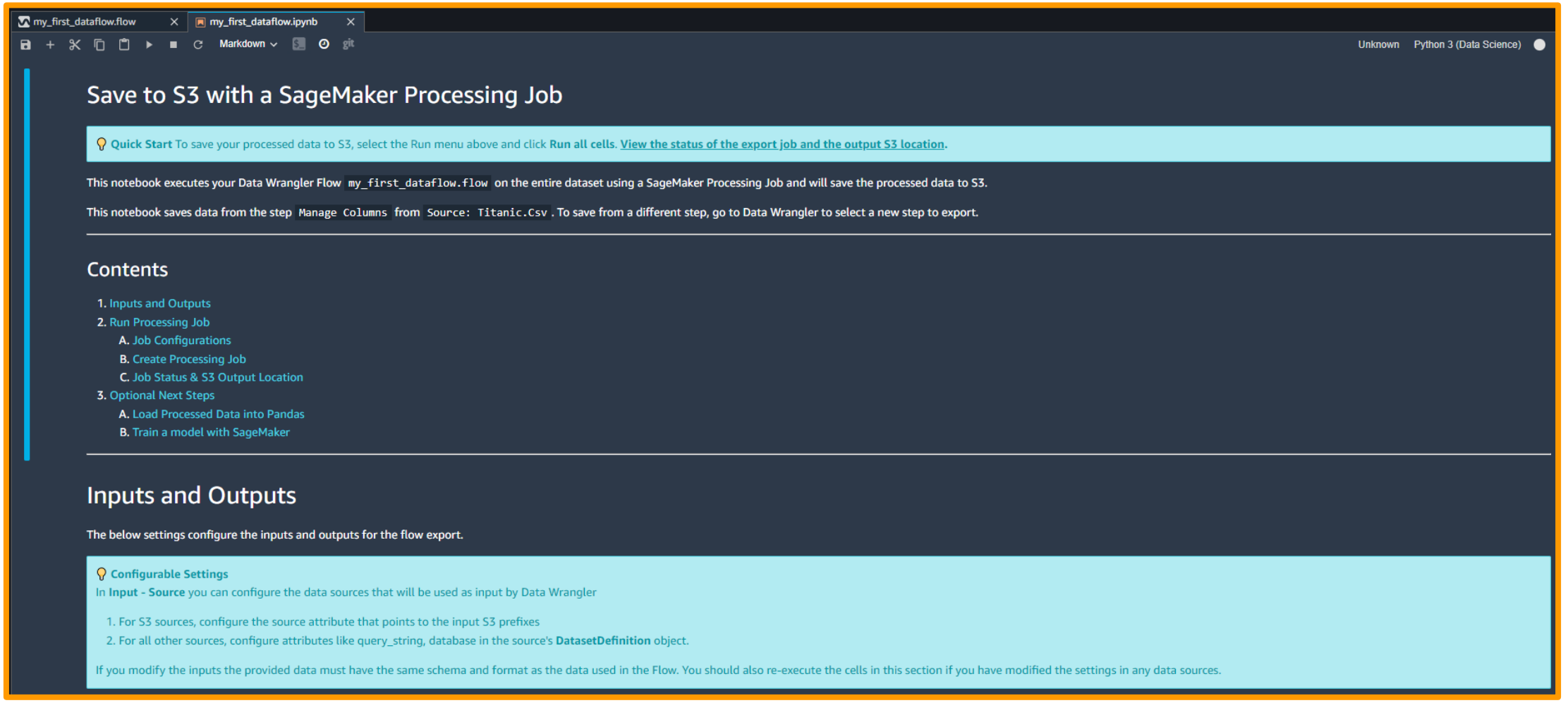
DATA WRANGLER DEMO

EXPLORE THE MANY OPTIONS TO EXPORT THE WORKFOW



DATA WRANGLER DEMO

SELECT THE S3 OPTION AND RUN THE GENERATED NOTEBOOK



DATA WRANGLER DEMO

RUN THE CELLS. NOTE THAT THE RESULTS ARE ALSO SAVED IN S3

my_first_dataflow.flow | my_first_dataflow.ipynb

Markdown | 2 vCPU + 4 GiB | Cluster | Python 3 (Data Science)

[9]:
```
from sagemaker.processing import Processor
from sagemaker.network import NetworkConfig

processor = Processor(
    role=iam_role,
    image_uri=container_uri,
    instance_count=instance_count,
    instance_type=instance_type,
    volume_size_in_gb=volume_size_in_gb,
    network_config=NetworkConfig(enable_network_isolation=enable_network_isolation),
    sagemaker_session=sess,
    output_kms_key=kms_key,
    tags=user_tags
)

# Start Job
processor.run(
    inputs=[flow_input] + data_sources,
    outputs=[processing_job_output],
    arguments=[f"--output-config '{json.dumps(output_config)}'"],
    wait=False,
    logs=False,
    job_name=processing_job_name
)
```

```
Job Name:  data-wrangler-flow-processing-02-10-19-38-966039ef
Inputs:  [{'InputName': 'flow', 'AppManaged': False, 'S3Input': {'S3Uri': 's3://sagemaker-us-east-1-694649038895/data_wrangler_flows/flow-02-10-19-38-966039ef.flow', 'LocalPath': '/opt/ml/processing/flow', 'S3DataType': 'S3Prefix', 'S3InputMode': 'Fil
e', 'S3DataDistributionType': 'FullyReplicated', 'S3CompressionType': 'None'}}, {'InputName': 'titanic.csv', 'AppManaged': False, 'S3Input': {'S3Uri': 's3://sagemaker-studio-694649038895-ylwtdvvsaa/titanic.csv', 'LocalPath': '/opt/ml/processing/titani
c.csv', 'S3DataType': 'S3Prefix', 'S3InputMode': 'File', 'S3DataDistributionType': 'FullyReplicated', 'S3CompressionType': 'None'}}]
Outputs:  [{'OutputName': '67e10d8a-7319-4add-bf94-2fe248ea255d.default', 'AppManaged': False, 'S3Output': {'S3Uri': 's3://sagemaker-us-east-1-694649038895/export-flow-02-10-19-38-966039ef/output', 'LocalPath': '/opt/ml/processing/output', 'S3UploadMo
e': 'EndOfJob'}}]
```

Job Status & S3 Output Location

Below you wait for processing job to finish. If it finishes successfully, the raw parameters used by the Processing Job will be printed

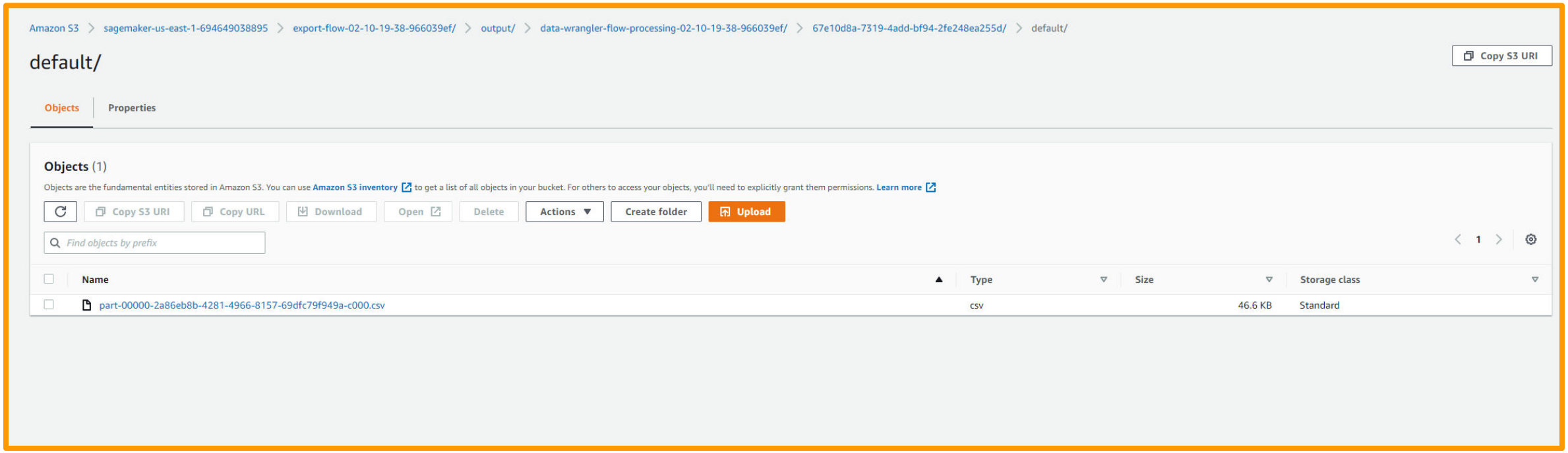
[*]:
```
s3_job_results_path = f"s3://{bucket}/{s3_output_prefix}/{processing_job_name}"
print(f"Job results are saved to S3 path: {s3_job_results_path}")

job_result = sess.wait_for_processing_job(processing_job_name)
job_result
```

```
Job results are saved to S3 path: s3://sagemaker-us-east-1-694649038895/export-flow-02-10-19-38-966039ef/output/data-wrangler-flow-processing-02-10-19-38-966039ef
..
```

(Optional)Next Steps

DATA WRANGLER DEMO

GO TO S3 AND DOWNLOAD THE PROCESSED DATASETS

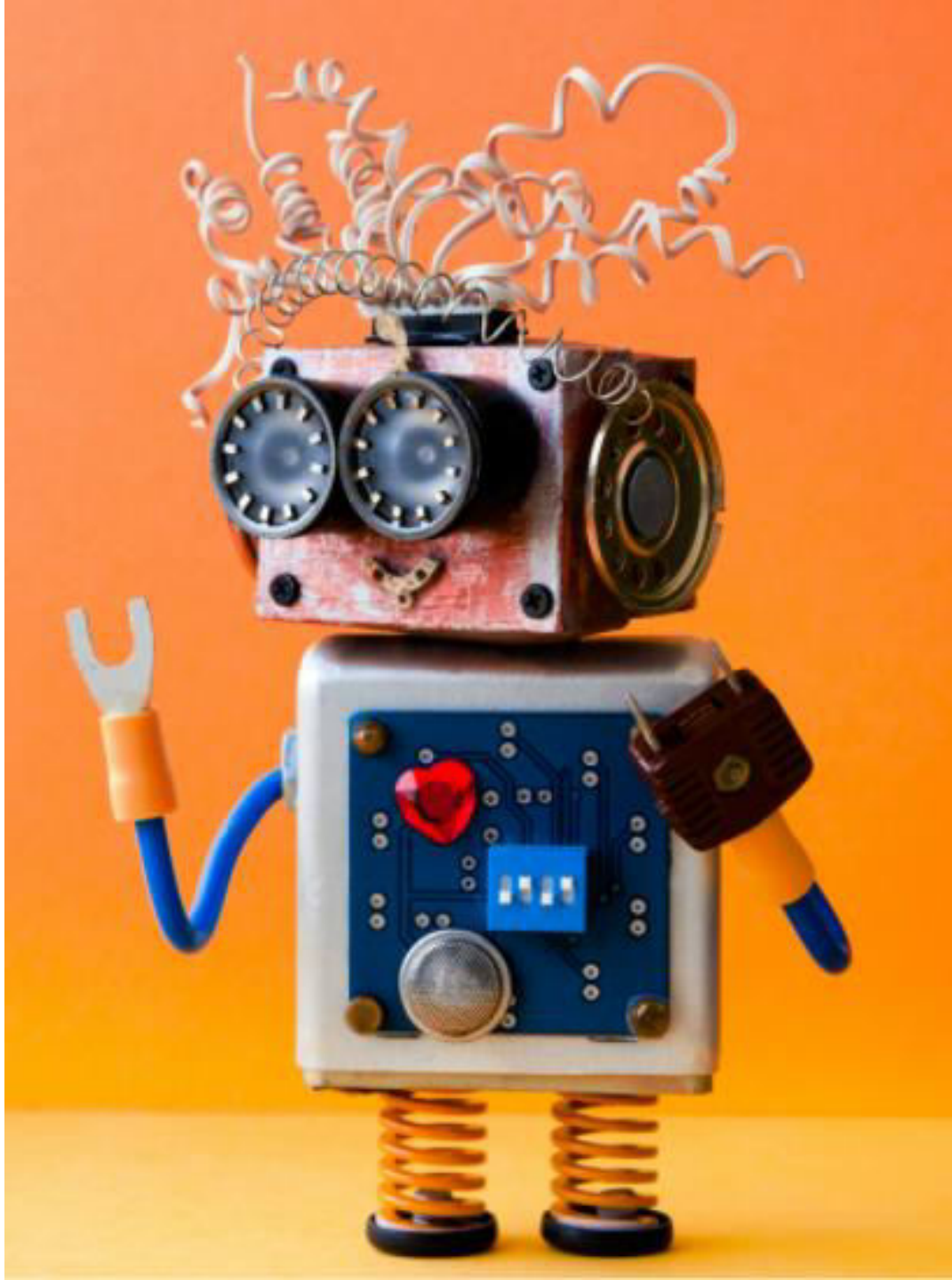


DATA WRANGLER DEMO

RUN THE CELLS. NOTE THAT THE RESULTS ARE ALSO SAVED IN S3

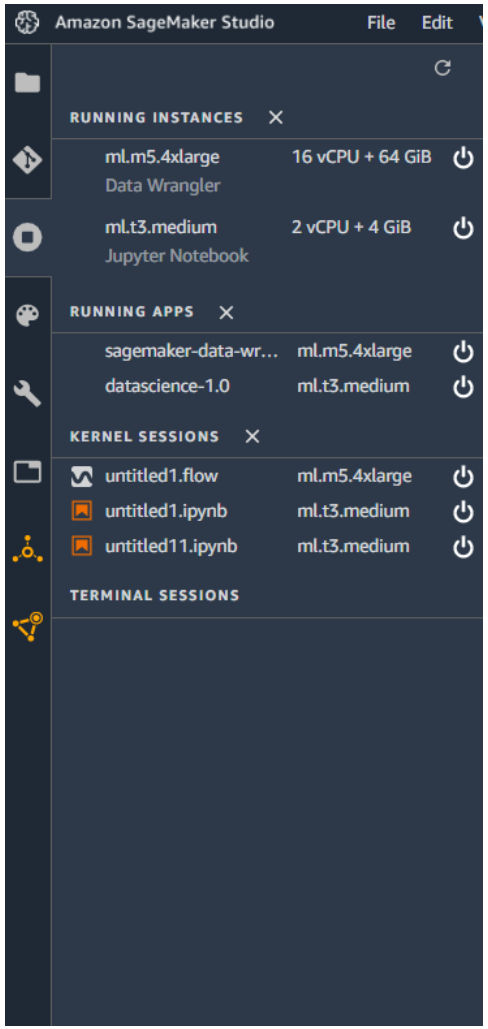
Survived	Pclass	SibSp	Parch	Fare	Sex	Scaled_Age
0	3	1	0	0.14083	0	1.69051597
1	1	1	0	1.42846	1	2.91998212
1	3	0	0	0.14083	1	1.99788251
1	1	1	0	1.06632	1	2.68945722
0	3	0	0	0.16095	0	2.68945722
0	3	0	0	0.16095	0	2.22840741
0	1	0	0	1.02608	0	4.14944828
0	3	3	1	0.4225	0	0.15368327
1	3	0	2	0.22131	1	2.07472414
1	2	1	0	0.60358	1	1.07578289
1	3	1	1	0.32191	1	0.30736654
1	1	0	0	0.5231	1	4.45681482
0	3	0	0	0.16095	0	1.5368327
0	3	1	5	0.6237	0	2.99682376
0	3	0	0	0.14083	1	1.07578289
1	2	0	0	0.32191	1	4.22628992
0	3	4	1	0.58346	0	0.15368327
1	2	0	0	0.26155	0	2.22840741
0	3	1	0	0.36215	1	2.38209068
1	3	0	0	0.14083	1	2.22840741
0	2	0	0	0.5231	0	2.68945722
1	2	0	0	0.26155	0	2.61261558
1	3	0	0	0.16095	1	1.15262452
1	1	0	0	0.70417	0	2.15156578

DATA WRANGLER DEMO #8: SHUTDOWN RESOURCES [IMPORTANT]



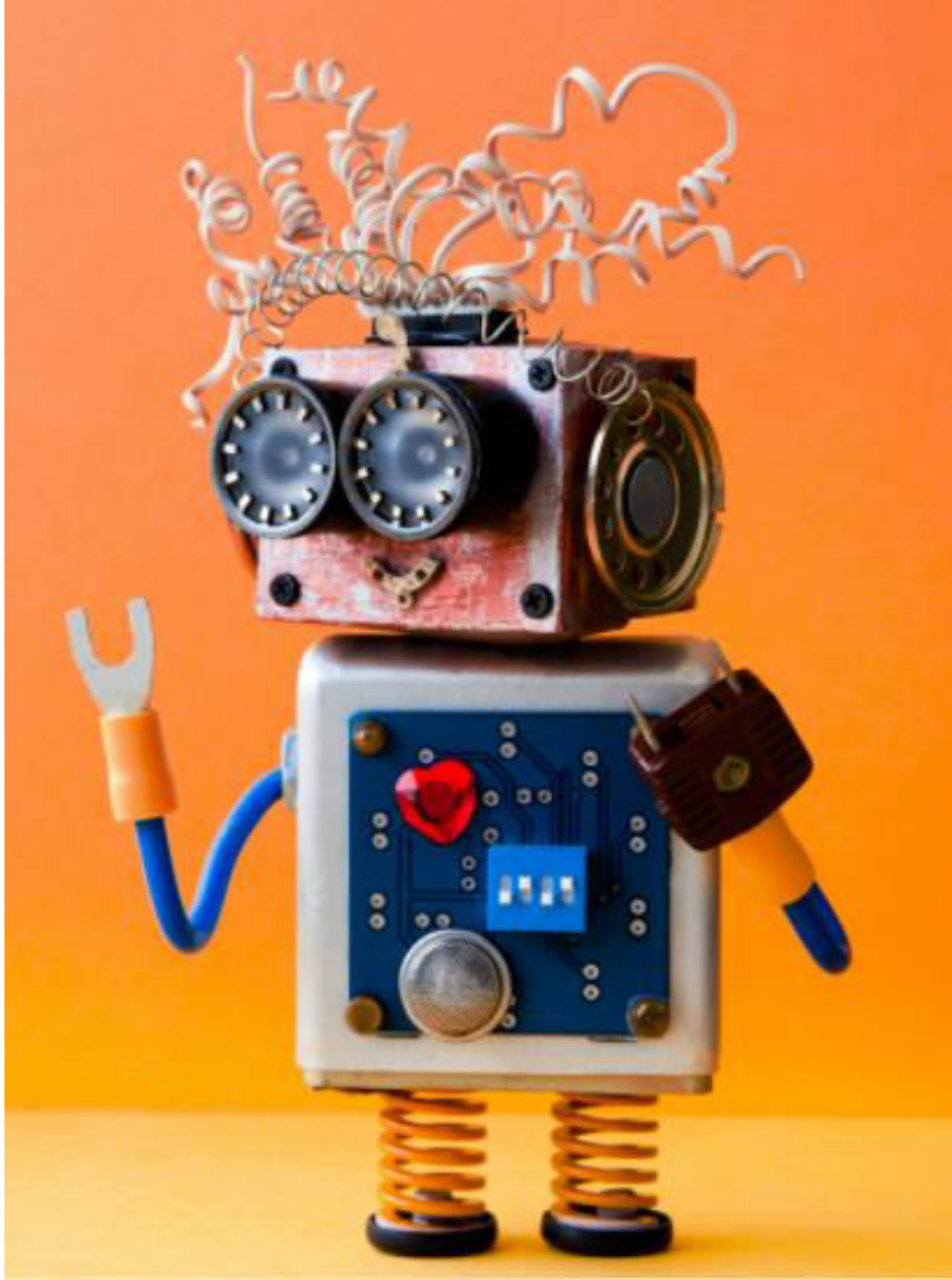
DATA WRANGLER DEMO [IMPORTANT]

CLICK ON POWER BUTTON TO SHUTDOWN ALL INSTANCES. THIS IS CRITICAL TO AVOID INCURRING ANY ADDITIONAL CHARGES.



FINAL END-OF-DAY CAPSTONE PROJECT

EASY ADVANCED

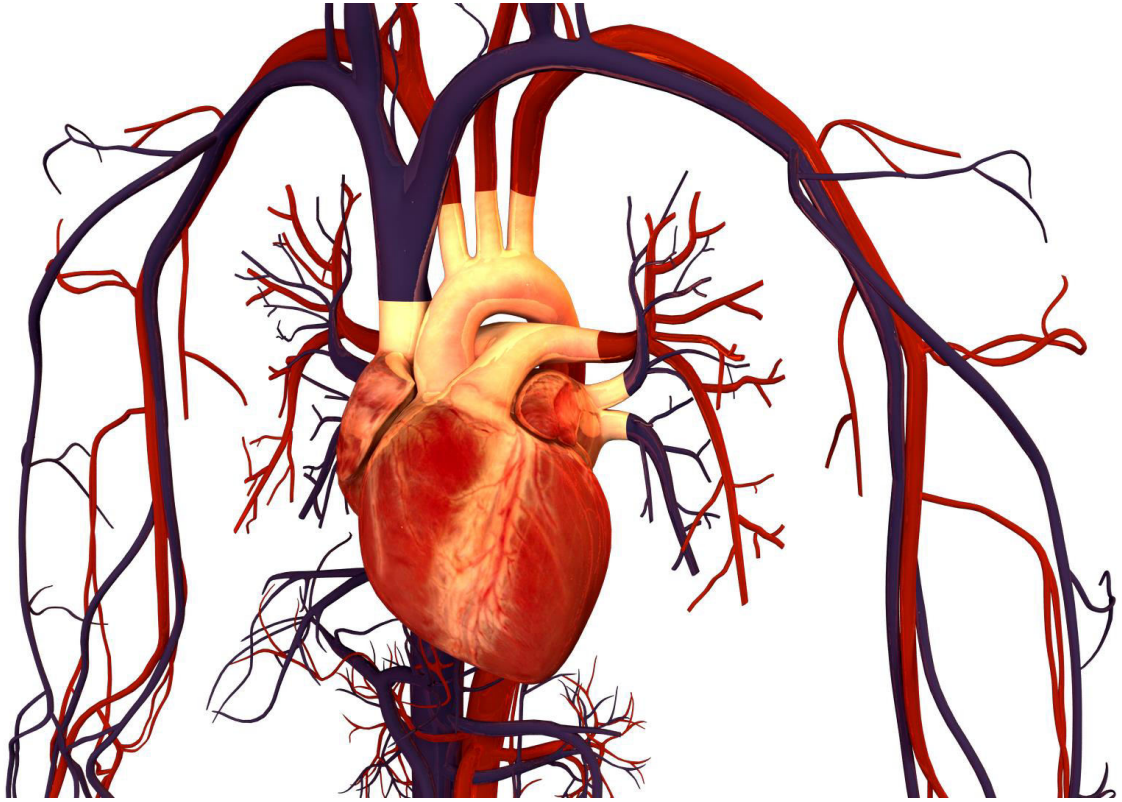


FINAL PROJECT OVERVIEW

- Using the “cardo.csv” dataset, use AWS data wrangler to perform the following tasks:
- 1. Upload the data to S3
- 2. Plot the histogram for the height column
- 3. Plot the scatterplot between the height and weight
- 4. Plot the correlation matrix between features
- 5. Drop the ID column
- 6. Create a custom formula to convert age column from days to years. Round the age to the nearest integer
- 7. Plot the histogram for the newly created age column
- 8. Generate a summary table and list the average age value
- 9. Sort the dataframe according the age column in an ascending order
- 10. Generate a bias report for the cholesterol column
- 11. Scale the weight, height, ap_lo and ap_hi using min max scaler
- 12. Export the workflow
- 13. Execute the code and explore the generated csv file

FINAL PROJECT OVERVIEW: DATA

- Aim of the problem is to detect the presence or absence of cardiovascular disease in person based on the given features.
- Features available are:
  - Age
  - Height
  - Weight
  - Gender
  - Smoking
  - Alcohol intake
  - Physical activity
  - Systolic blood pressure
  - Diastolic blood pressure
  - Cholesterol
  - Glucose

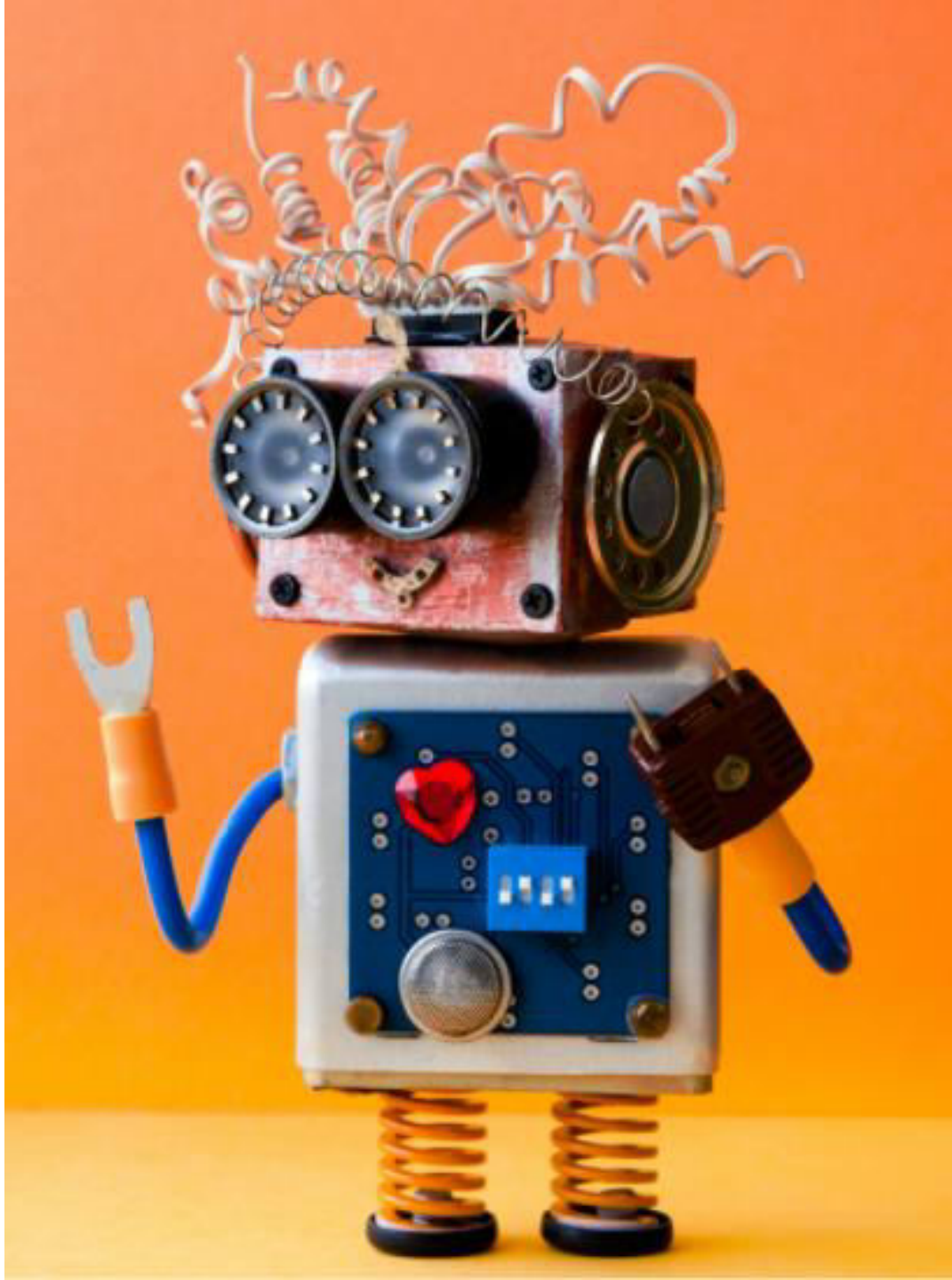


- **Data Source:** https://www.kaggle.com/sulianova/cardiovascular-disease-dataset
- **Image Source:** https://commons.wikimedia.org/wiki/File:Human_Heart_and_Circulatory_System.png

PROJECT SOLUTION

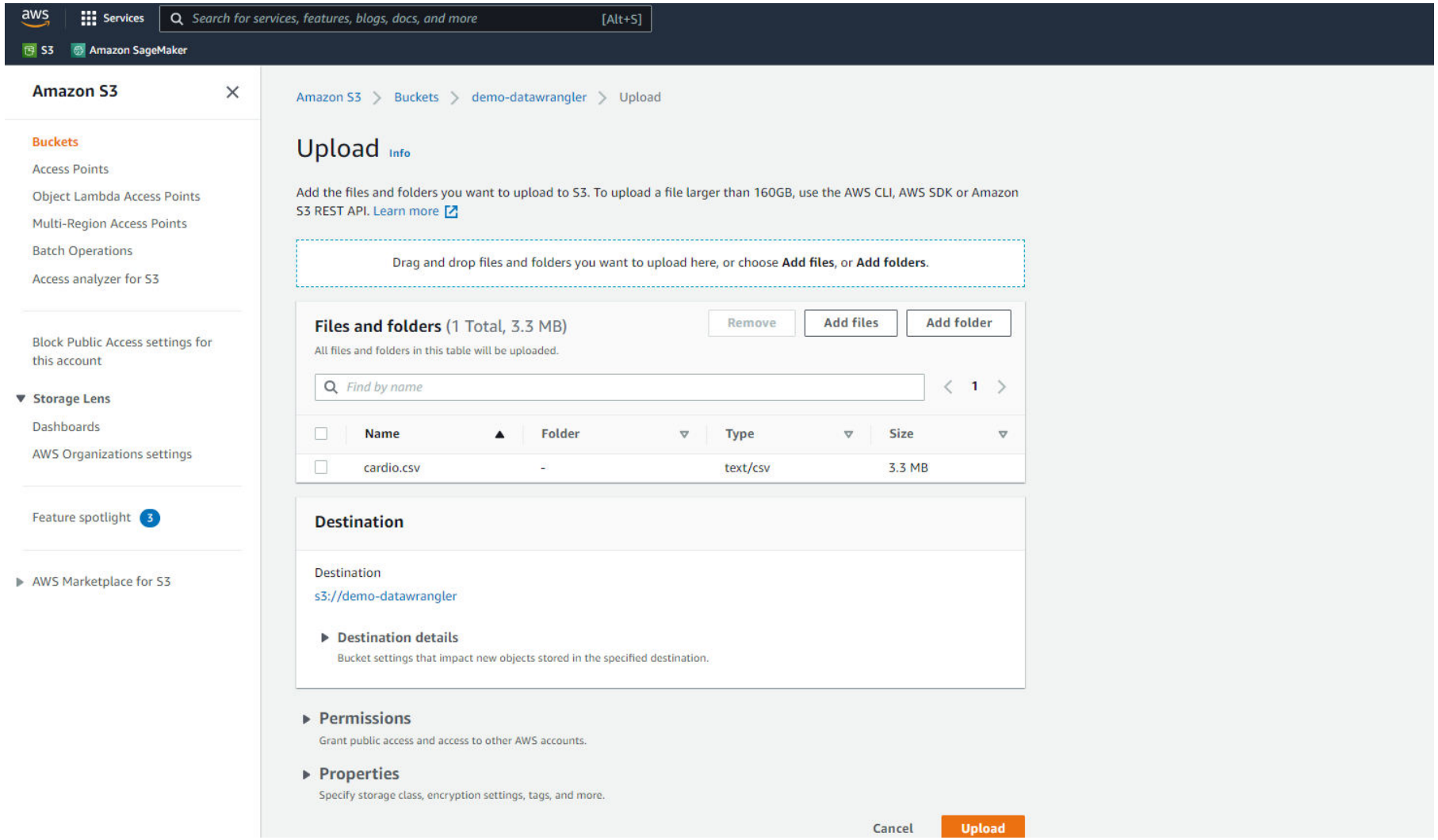
EASY

ADVANCED



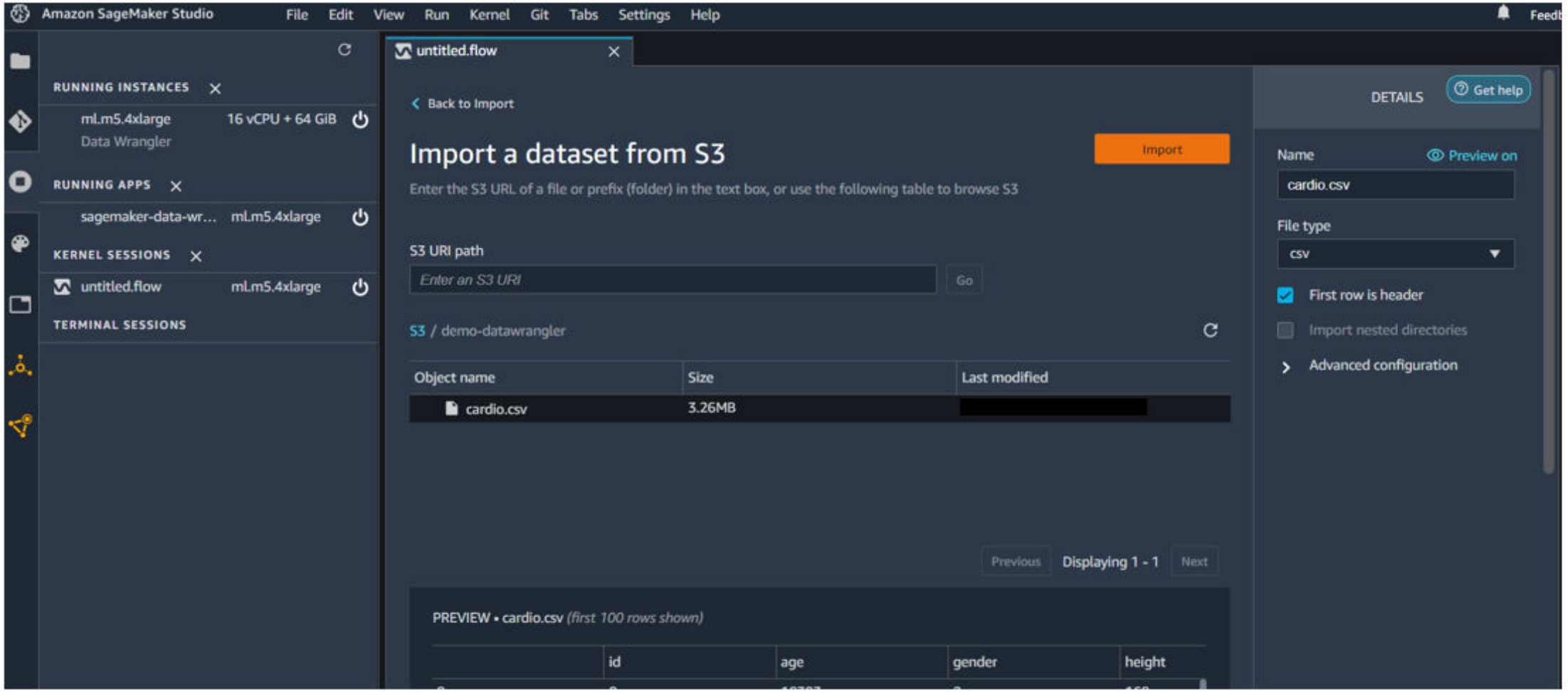
PROJECT

UPLOAD THE DATA TO S3



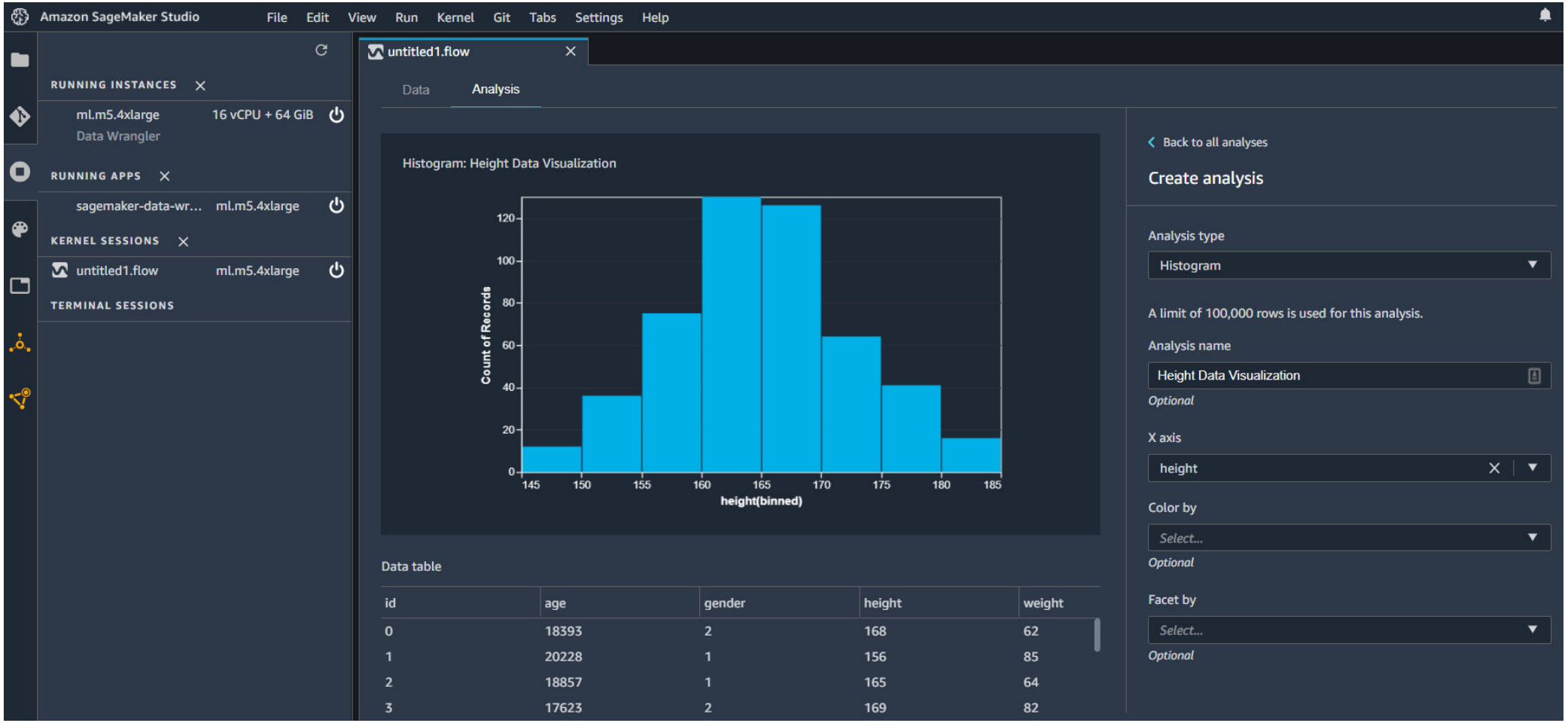
PROJECT

UPLOAD THE DATA TO S3



PROJECT

PLOT THE HEIGHT HISTOGRAM

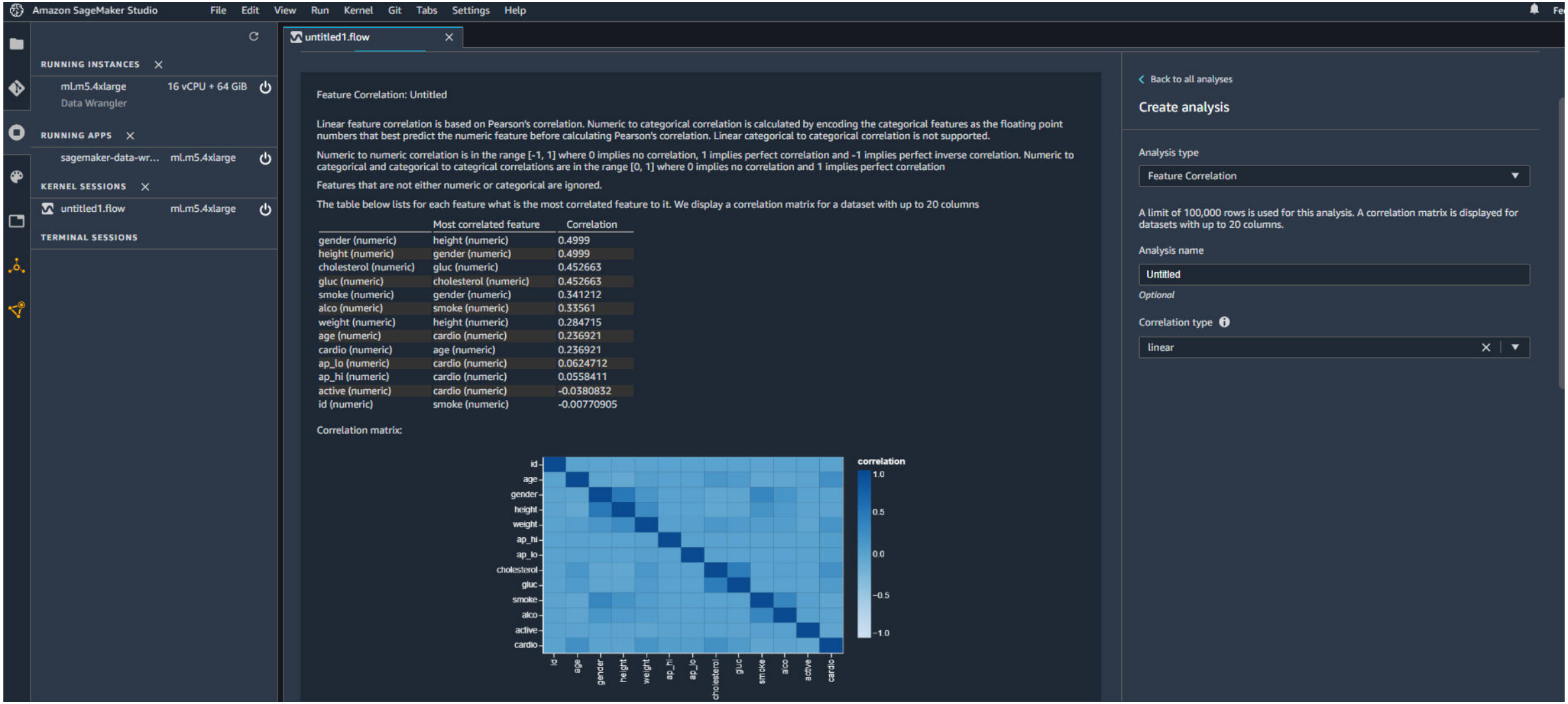


PROJECT

PLOT SCATTERPLOT BETWEEN HEIGHT AND WEIGHT

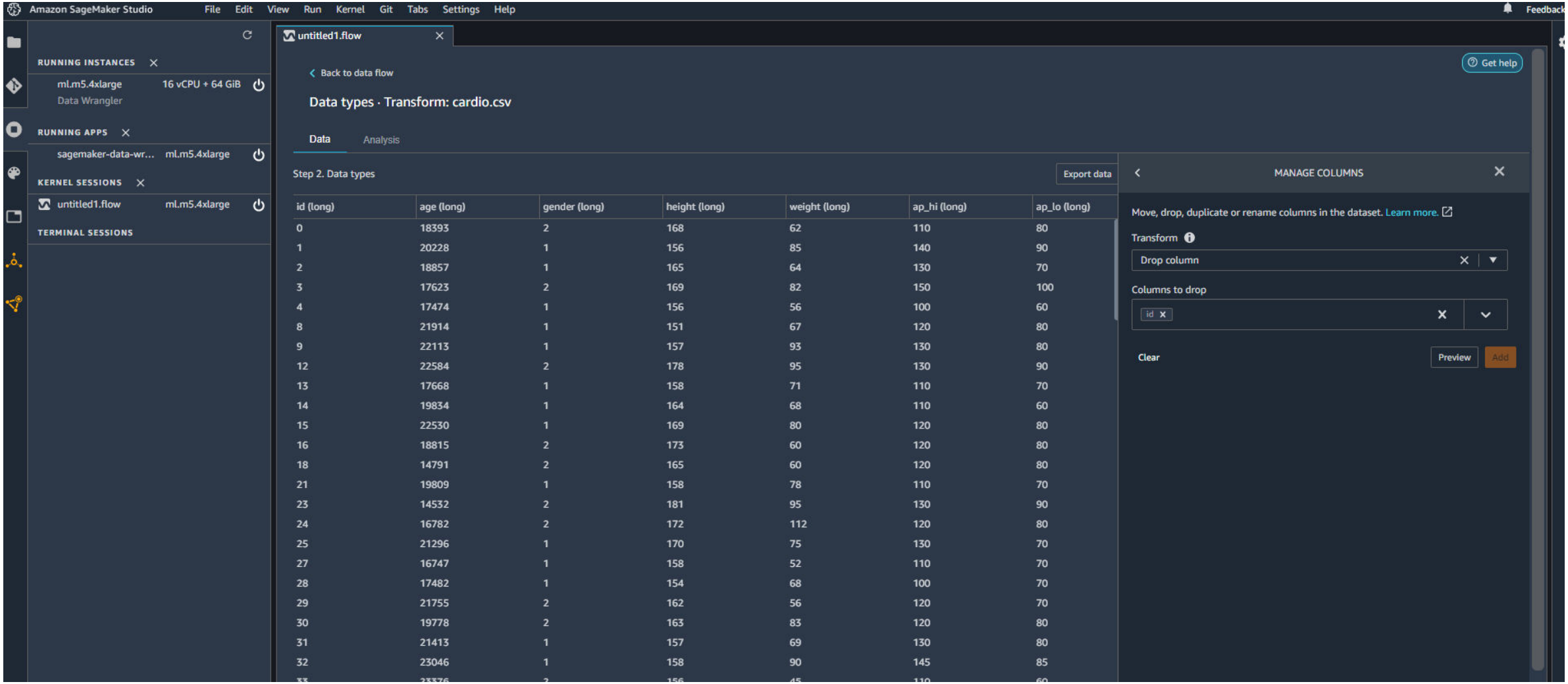
PROJECT

PLOT CORRELATION BETWEEN FEATURES



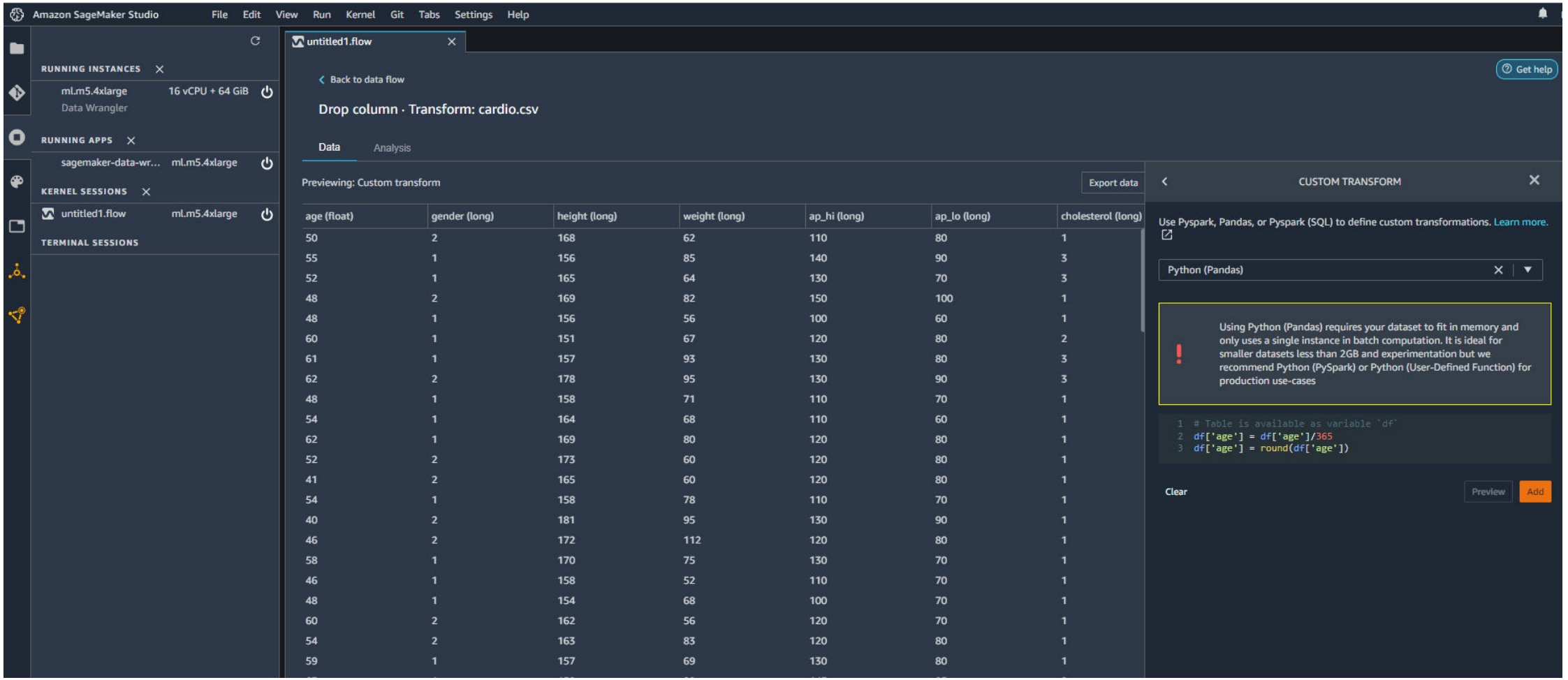
PROJECT

DROP THE ID COLUMN



PROJECT

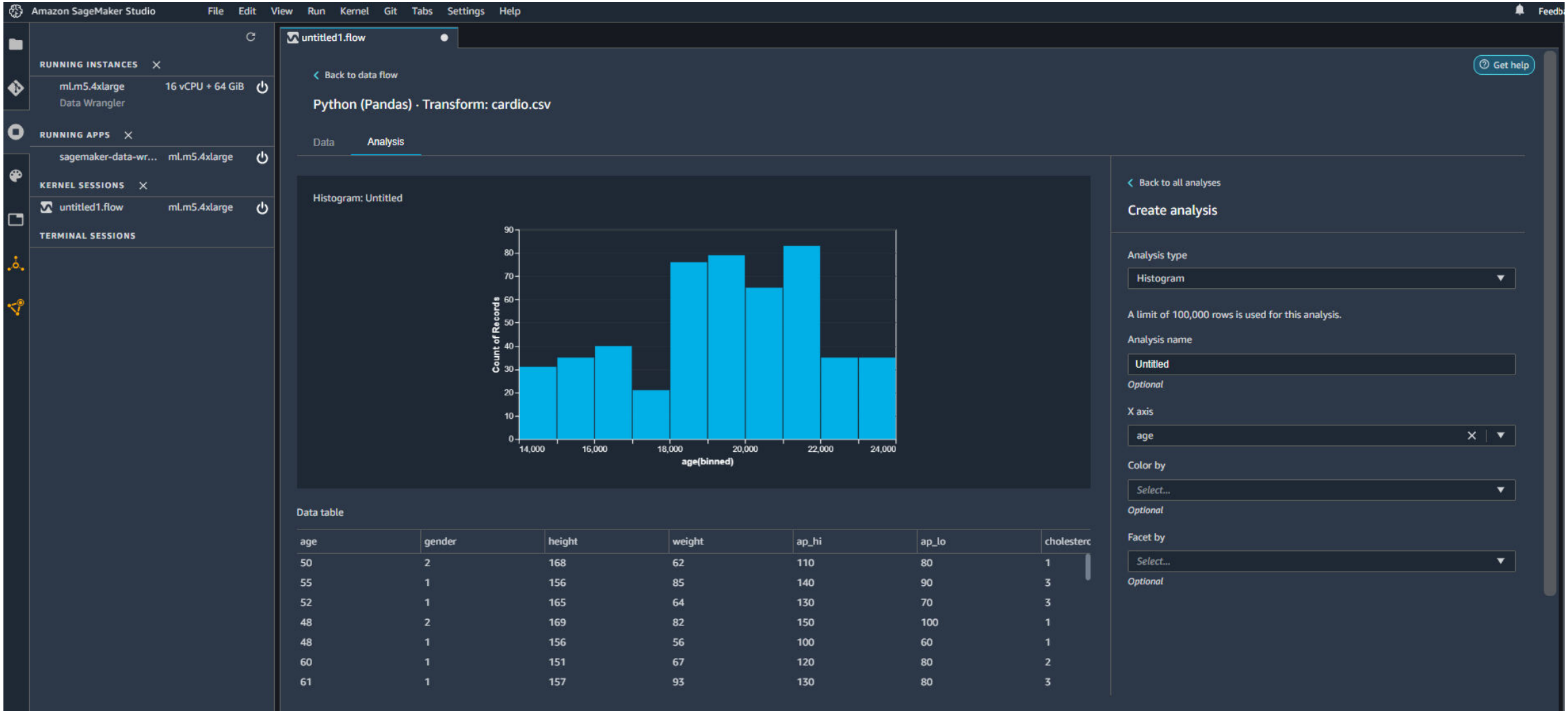
CREATE A CUSTOM FORMULA TO CONVERT DAYS TO YEARS. ROUND THE VALUE TO THE NEAREST INTEGER.



age (float)	gender (long)	height (long)	weight (long)	ap_hi (long)	ap_lo (long)	cholesterol (long)
50	2	168	62	110	80	1
55	1	156	85	140	90	3
52	1	165	64	130	70	3
48	2	169	82	150	100	1
48	1	156	56	100	60	1
60	1	151	67	120	80	2
61	1	157	93	130	80	3
62	2	178	95	130	90	3
48	1	158	71	110	70	1
54	1	164	68	110	60	1
62	1	169	80	120	80	1
52	2	173	60	120	80	1
41	2	165	60	120	80	1
54	1	158	78	110	70	1
40	2	181	95	130	90	1
46	2	172	112	120	80	1
58	1	170	75	130	70	1
46	1	158	52	110	70	1
48	1	154	68	100	70	1
60	2	162	56	120	70	1
54	2	163	83	120	80	1
59	1	157	69	130	80	1

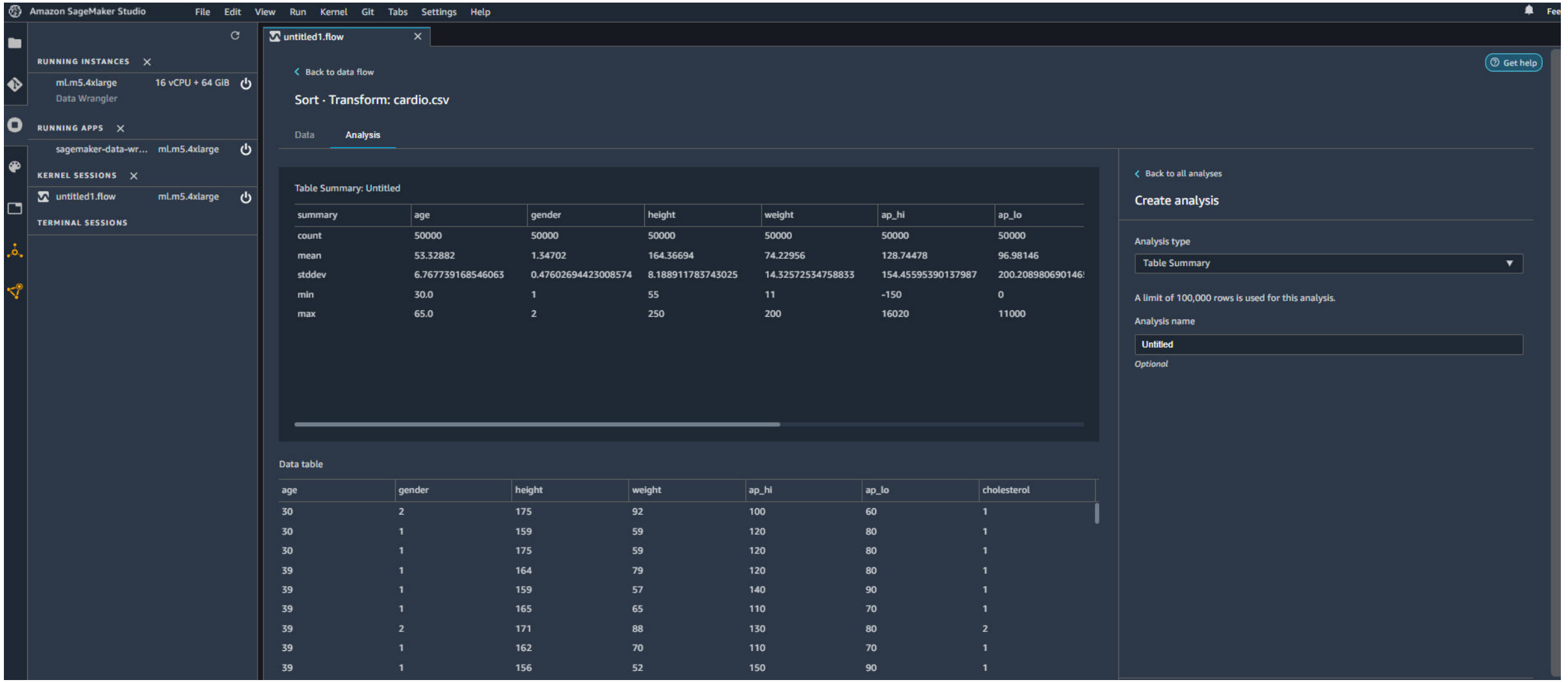
PROJECT

PLOT THE HISTOGRAM FOR THE AGE COLUMN



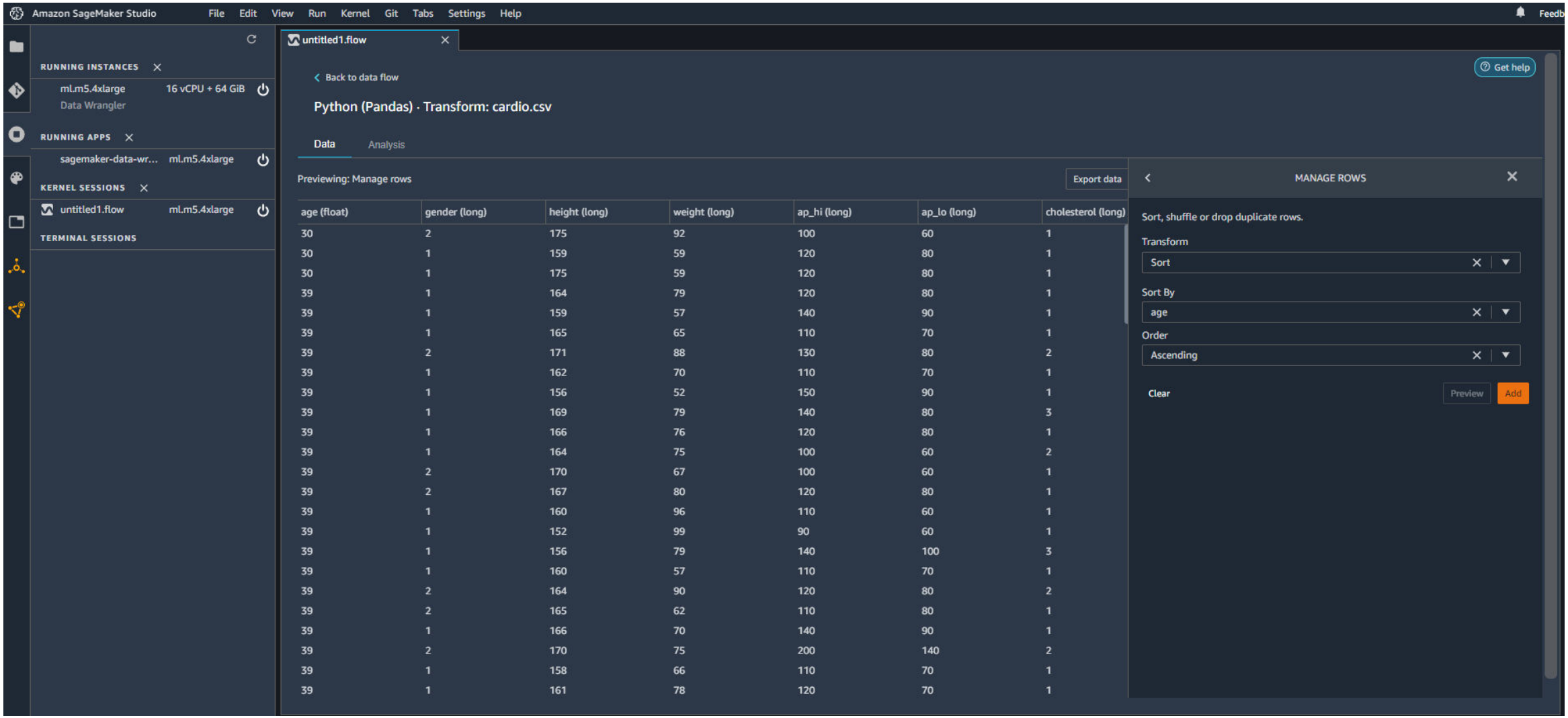
PROJECT

GENERATE A SUMMARY TABLE.



PROJECT

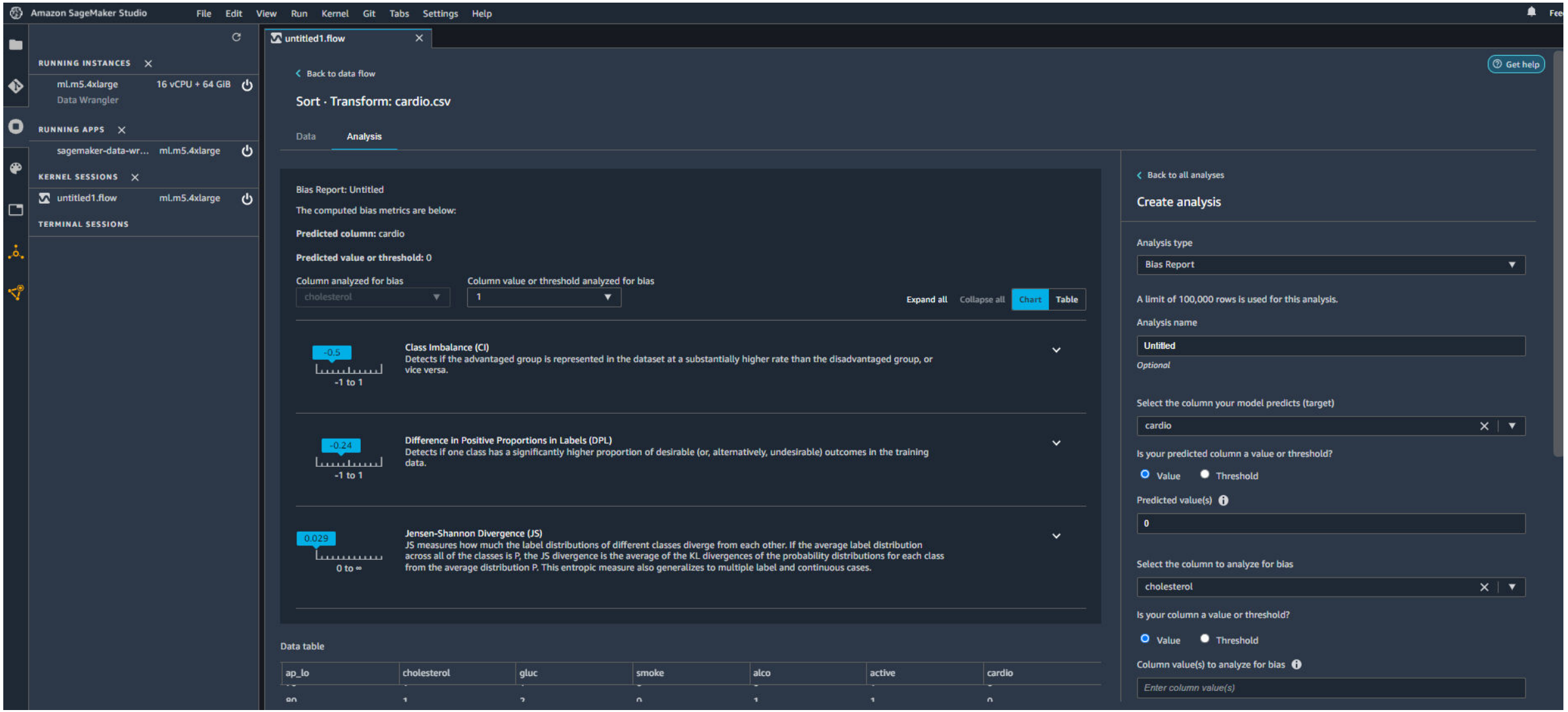
SORT DATAFRAME IN AN ASCENDING ORDER USING AGE COLUMN



age (float)	gender (long)	height (long)	weight (long)	ap_hi (long)	ap_lo (long)	cholesterol (long)
30	2	175	92	100	60	1
30	1	159	59	120	80	1
30	1	175	59	120	80	1
39	1	164	79	120	80	1
39	1	159	57	140	90	1
39	1	165	65	110	70	1
39	2	171	88	130	80	2
39	1	162	70	110	70	1
39	1	156	52	150	90	1
39	1	169	79	140	80	3
39	1	166	76	120	80	1
39	1	164	75	100	60	2
39	2	170	67	100	60	1
39	2	167	80	120	80	1
39	1	160	96	110	60	1
39	1	152	99	90	60	1
39	1	156	79	140	100	3
39	1	160	57	110	70	1
39	2	164	90	120	80	2
39	2	165	62	110	80	1
39	1	166	70	140	90	1
39	2	170	75	200	140	2
39	1	158	66	110	70	1
39	1	161	78	120	70	1

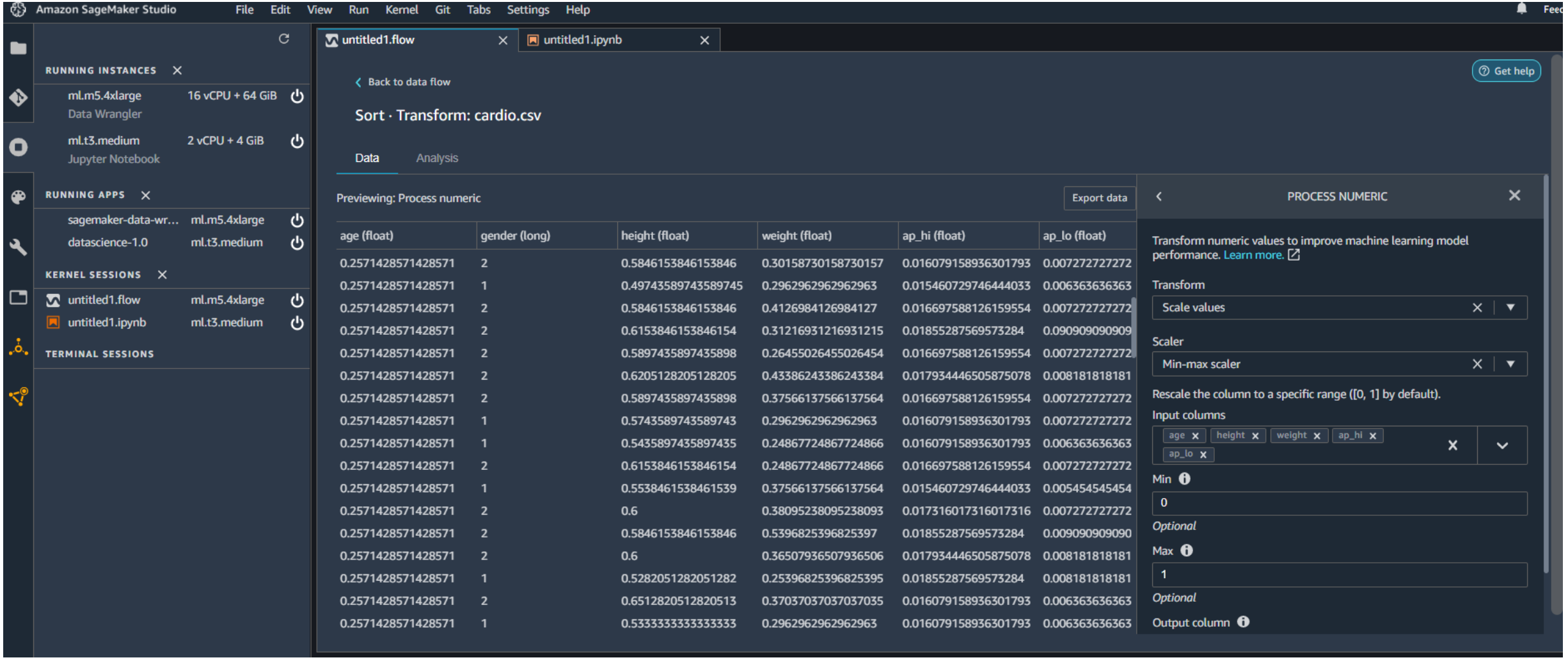
PROJECT

GENERATE A BIAS REPORT USING CHOLESTROL COLUMN



PROJECT

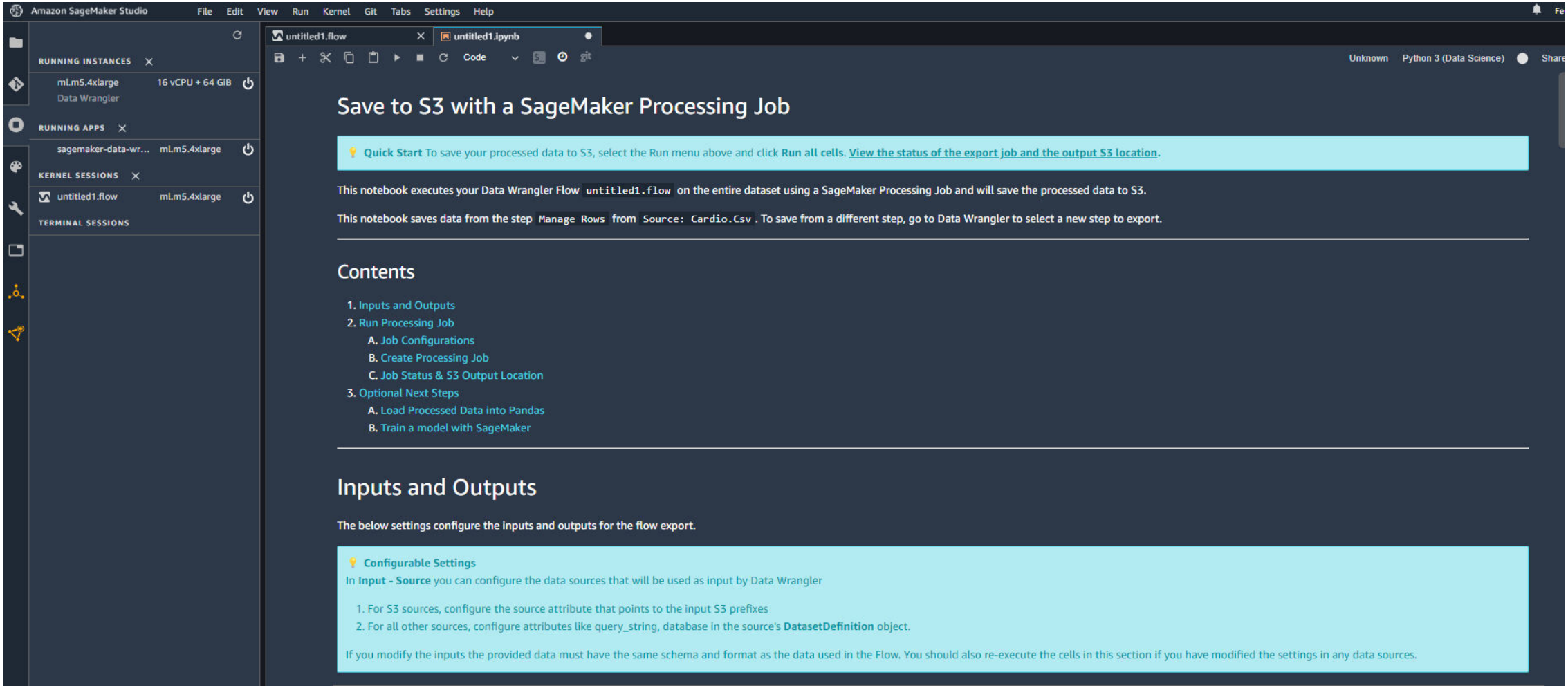
SCALE AGE, HEIGHT, WEIGHT, AP_HI AND AP_LO



age (float)	gender (long)	height (float)	weight (float)	ap_hi (float)	ap_lo (float)
0.2571428571428571	2	0.5846153846153846	0.30158730158730157	0.016079158936301793	0.007272727272
0.2571428571428571	1	0.49743589743589745	0.2962962962962963	0.015460729746444033	0.006363636363
0.2571428571428571	2	0.5846153846153846	0.4126984126984127	0.016697588126159554	0.007272727272
0.2571428571428571	2	0.6153846153846154	0.31216931216931215	0.01855287569573284	0.090909090909
0.2571428571428571	2	0.5897435897435898	0.26455026455026454	0.016697588126159554	0.007272727272
0.2571428571428571	2	0.6205128205128205	0.43386243386243384	0.017934446505875078	0.008181818181
0.2571428571428571	2	0.5897435897435898	0.37566137566137564	0.016697588126159554	0.007272727272
0.2571428571428571	1	0.5743589743589743	0.2962962962962963	0.016079158936301793	0.007272727272
0.2571428571428571	1	0.5435897435897435	0.24867724867724866	0.016079158936301793	0.006363636363
0.2571428571428571	2	0.6153846153846154	0.24867724867724866	0.016697588126159554	0.007272727272
0.2571428571428571	1	0.5538461538461539	0.37566137566137564	0.015460729746444033	0.005454545454
0.2571428571428571	2	0.6	0.38095238095238093	0.017316017316017316	0.007272727272
0.2571428571428571	2	0.5846153846153846	0.5396825396825397	0.01855287569573284	0.009090909090
0.2571428571428571	2	0.6	0.36507936507936506	0.017934446505875078	0.008181818181
0.2571428571428571	1	0.5282051282051282	0.25396825396825395	0.01855287569573284	0.008181818181
0.2571428571428571	2	0.6512820512820513	0.37037037037037035	0.016079158936301793	0.006363636363
0.2571428571428571	1	0.5333333333333333	0.2962962962962963	0.016079158936301793	0.006363636363

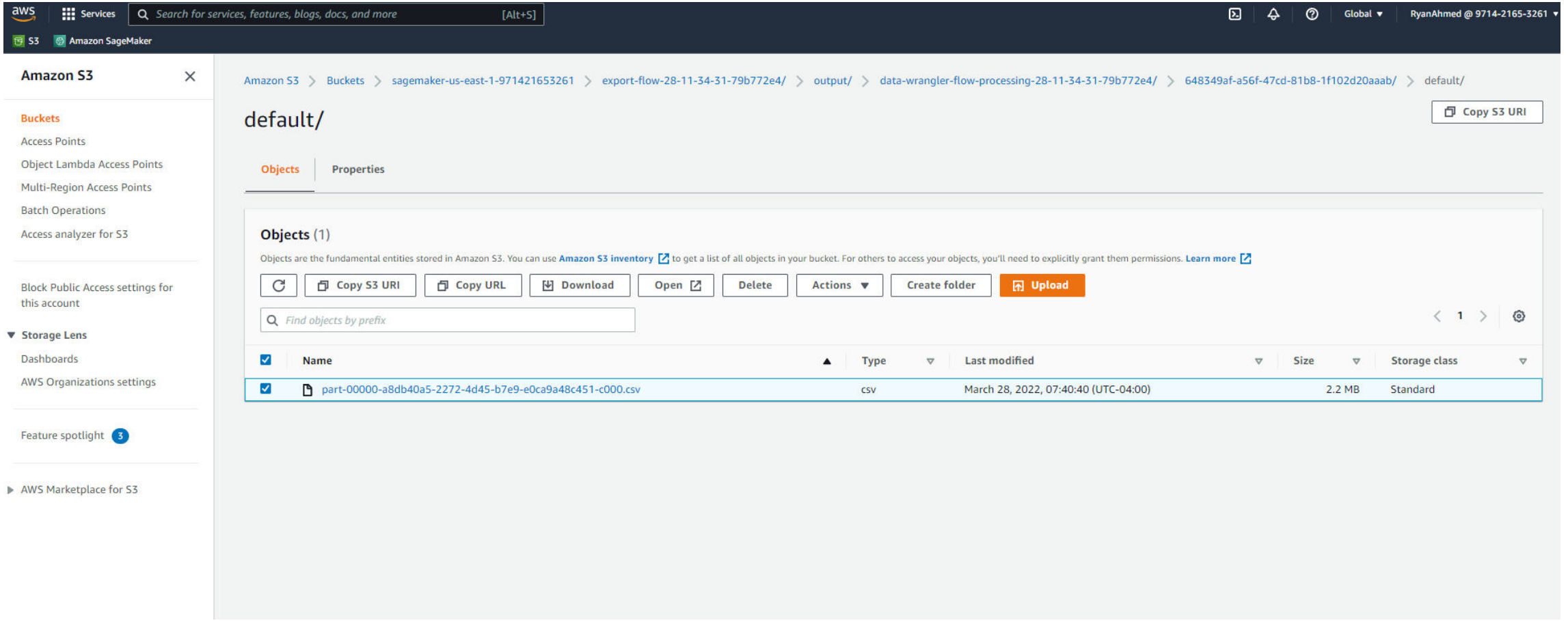
PROJECT

EXPORT TO S3



PROJECT

NAVIGATE TO THE PATH SHOWN BELOW AND DOWNLOAD THE FILE



DATA WRANGLER DEMO [IMPORTANT]

CLICK ON POWER BUTTON TO SHUTDOWN ALL INSTANCES. THIS IS CRITICAL TO AVOID INCURRING ANY ADDITIONAL CHARGES.