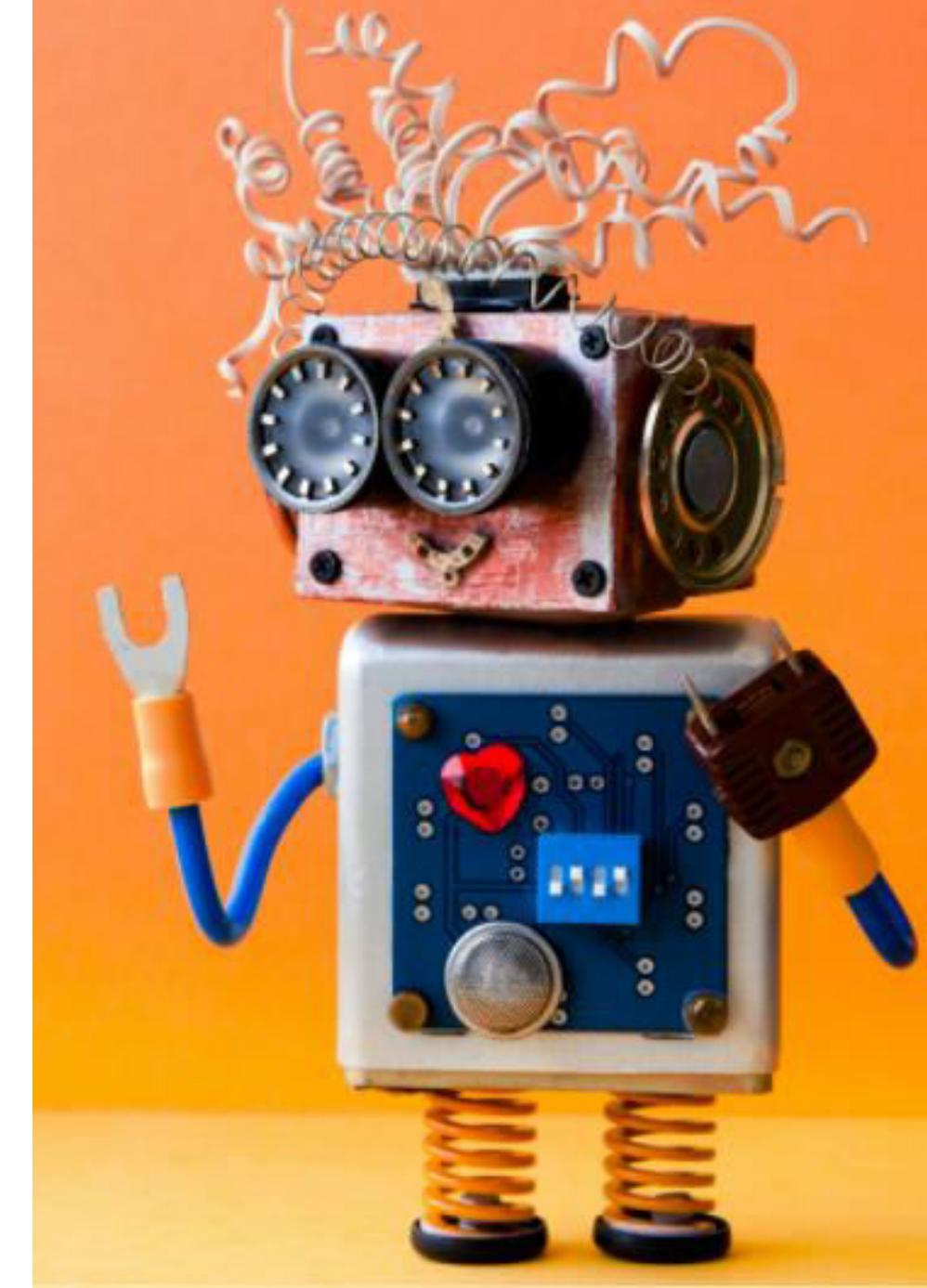


PROJECT OVERVIEW



EASY

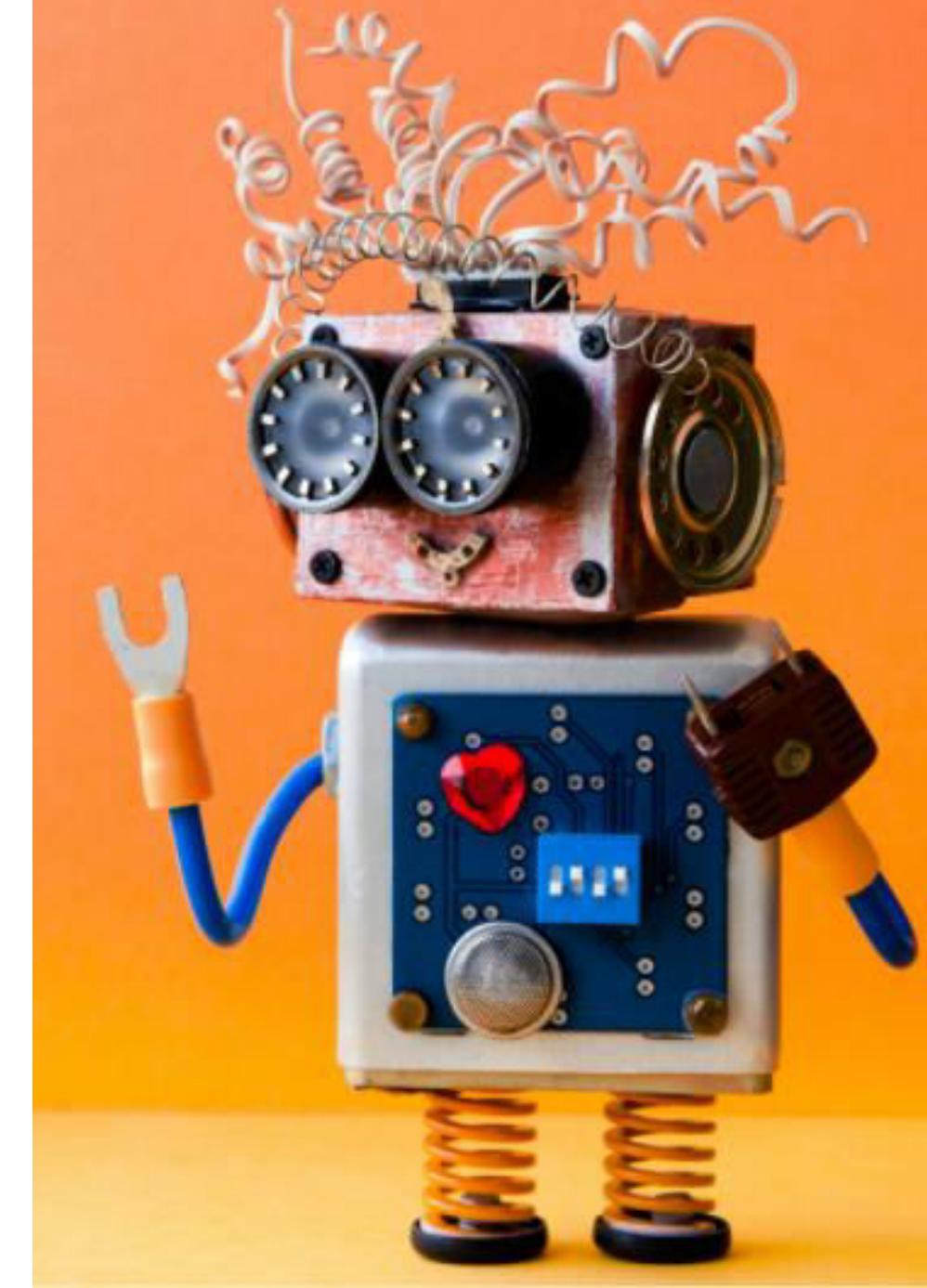
ADVANCED



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.

The screenshot displays two main windows from the Amazon SageMaker Studio interface:

- Data Wrangler Window:** Titled "my_first_dataflow.flow". It shows a preview of the "titanic.csv" dataset with 26 rows and 12 columns. The columns include PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, and Fare. A sidebar on the right lists various data transformation options like "Custom transform", "Balance data", and "Encode categorical".
- Amazon SageMaker Studio Home Window:** Shows the "RUNNING INSTANCES" section with one instance named "ml.m5.4xlarge" (16 vCPU + 64 GB). The "RUNNING APPS" section shows "sagemaker-data-wrangler" and "untitled1.flow" both running on "ml.m5.4xlarge". The "KERNEL SESSIONS" section shows "untitled1.flow" running on "ml.m5.4xlarge".

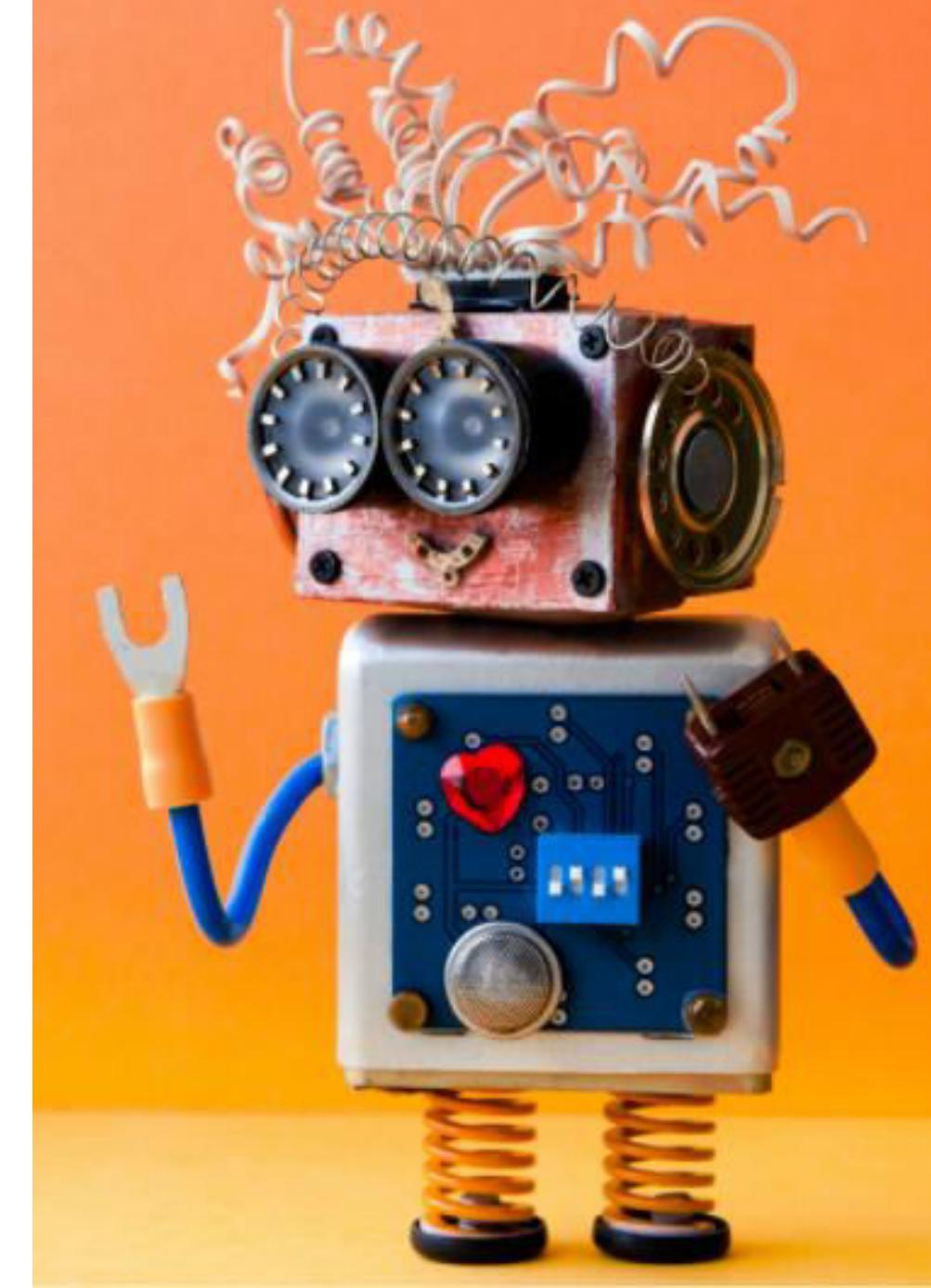
In the bottom right of the main window, there is a histogram titled "Histogram: Height Data Visualization" showing the distribution of height. The x-axis is labeled "height(binned)" and the y-axis is labeled "Count of Records". The histogram has several bars, with the highest frequency occurring between 160 and 165 units. Below the histogram, a "Data table" is shown with columns: id, age, gender, height, and weight. The first few rows of the table are:

id	age	gender	height	weight
0	18393	2	168	62
1	20228	1	156	85
2	18857	1	165	64
3	17623	2	169	82

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

ENTIRE COLUMN REQUIRES ENCODING

MISSING INFORMATION

REQUIRES FORMATTING

DUPLICATE ENTRY

The diagram illustrates several issues with the dataset:

- Entire Column Requires Encoding:** An annotation points to the "CLICK ON AD?" column, indicating that the entire column needs encoding.
- Missing Information:** An annotation points to the empty cell in the "LOCATION" column for customer ID 4.
- Requires Formatting:** An annotation points to the "Yes" entry in the "CLICK ON AD?" column for customer ID 1, suggesting the need for binary conversion.
- Duplicate Entry:** An annotation points to the row for customer ID 6, which has the same name and location as customer ID 3.

FEATURE ENGINEERING TECHNIQUES

Imputation

Handling
Outliers

Binning

Log
Transform

One-Hot
Encoding

Feature
Split

Scaling

FEATURE ENGINEERING: TOOLS



JUPYTER
NOTEBOOKS

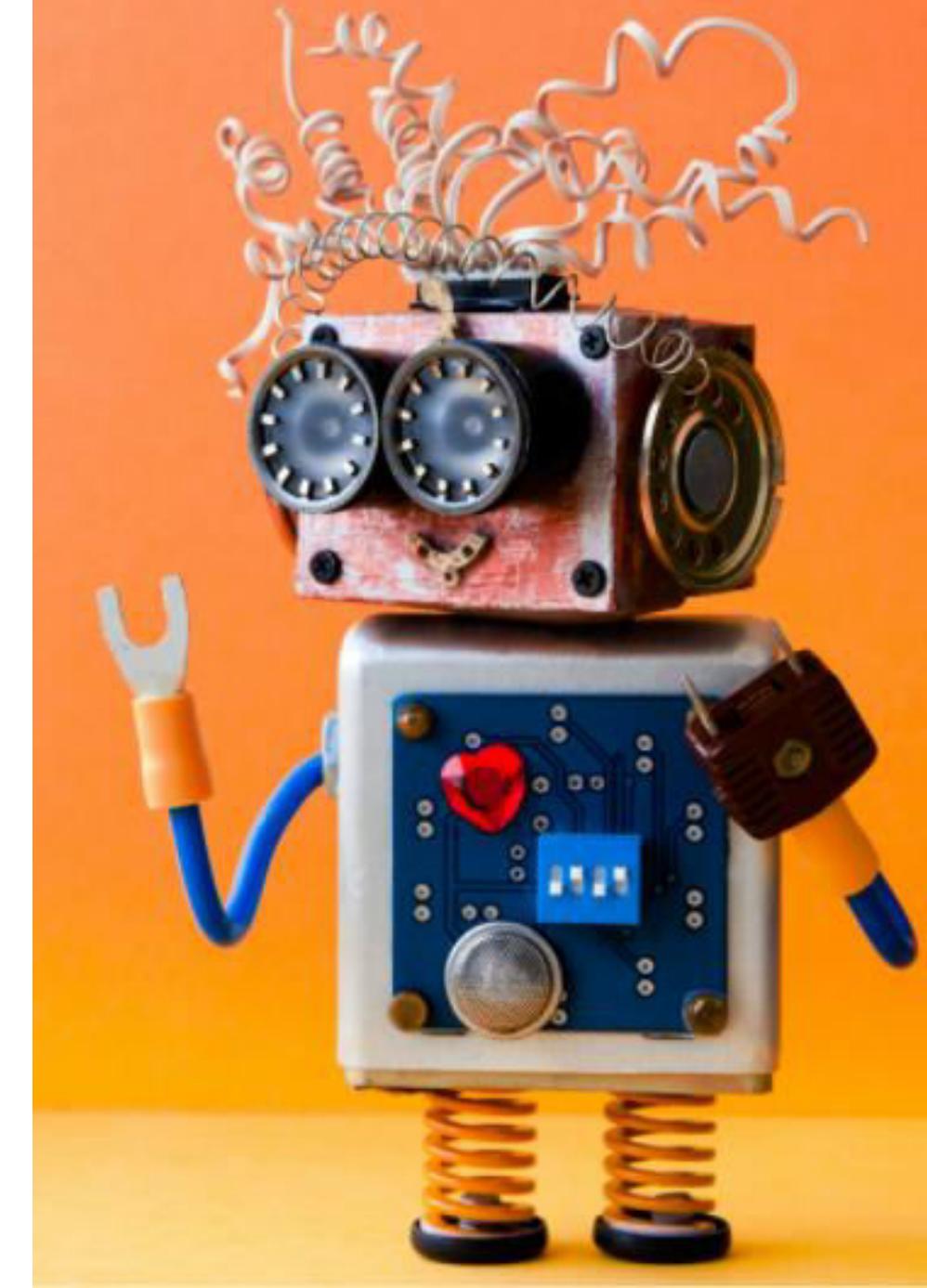


AMAZON
SAGEMAKER DATA
WRANGLER



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

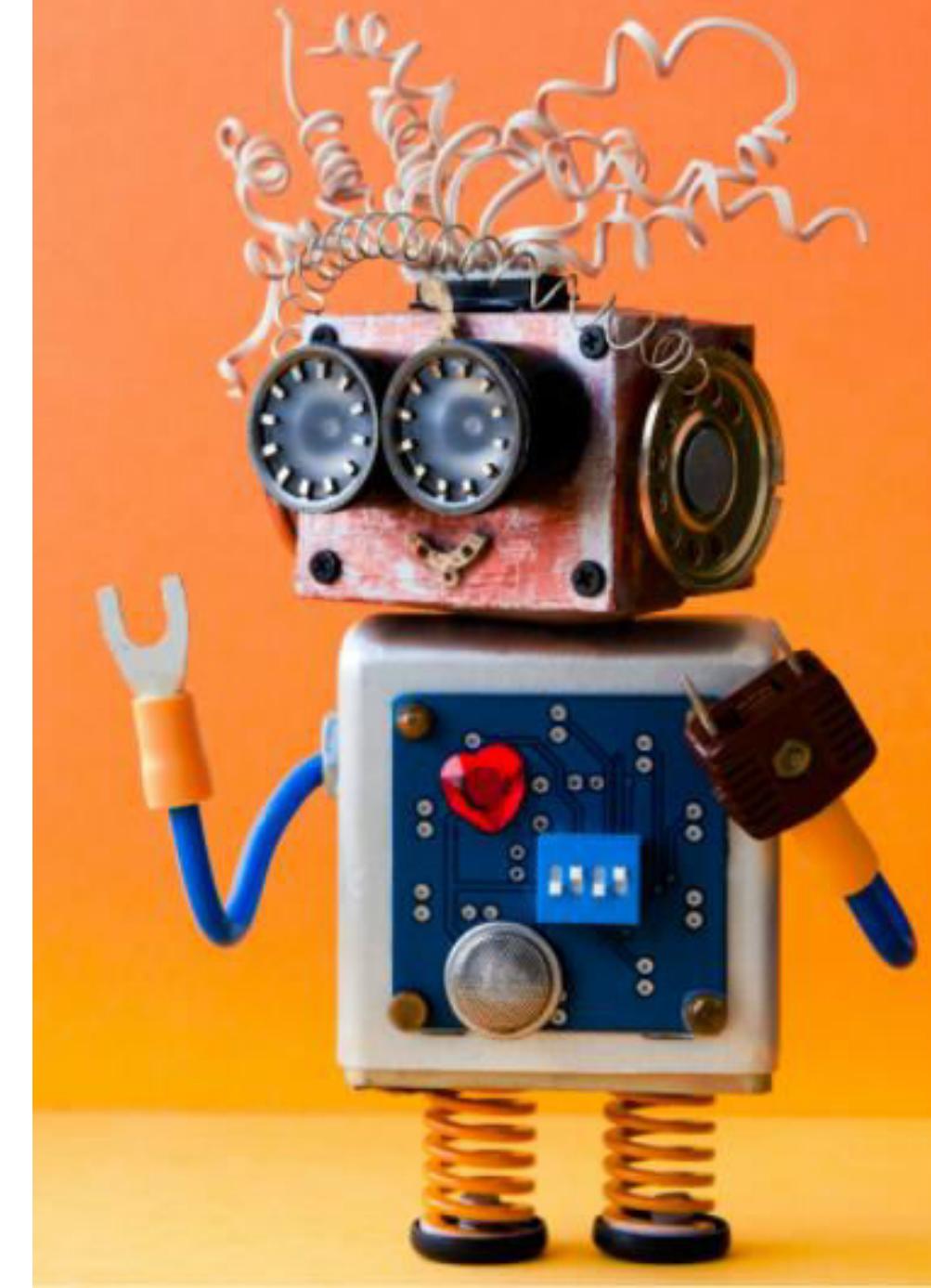
Order is important!



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

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

SCALING: NORMALIZATION & 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

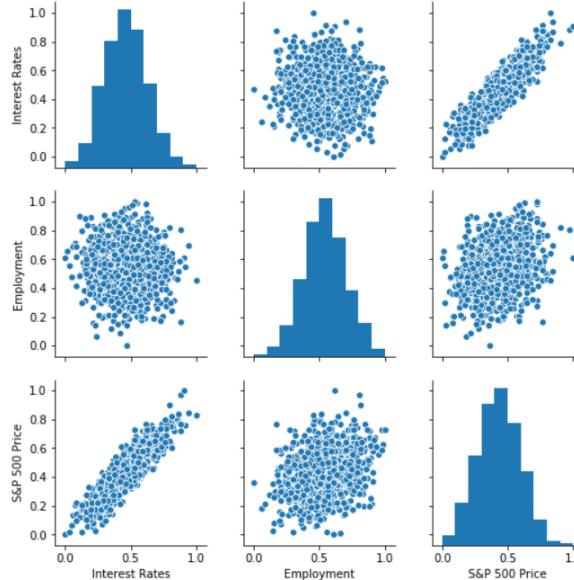
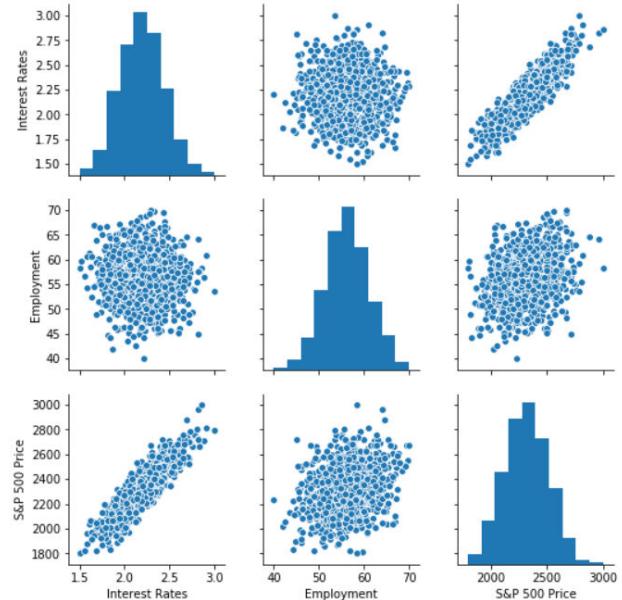
- Normalization is conducted to make feature values range from 0 to 1.

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

```
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.



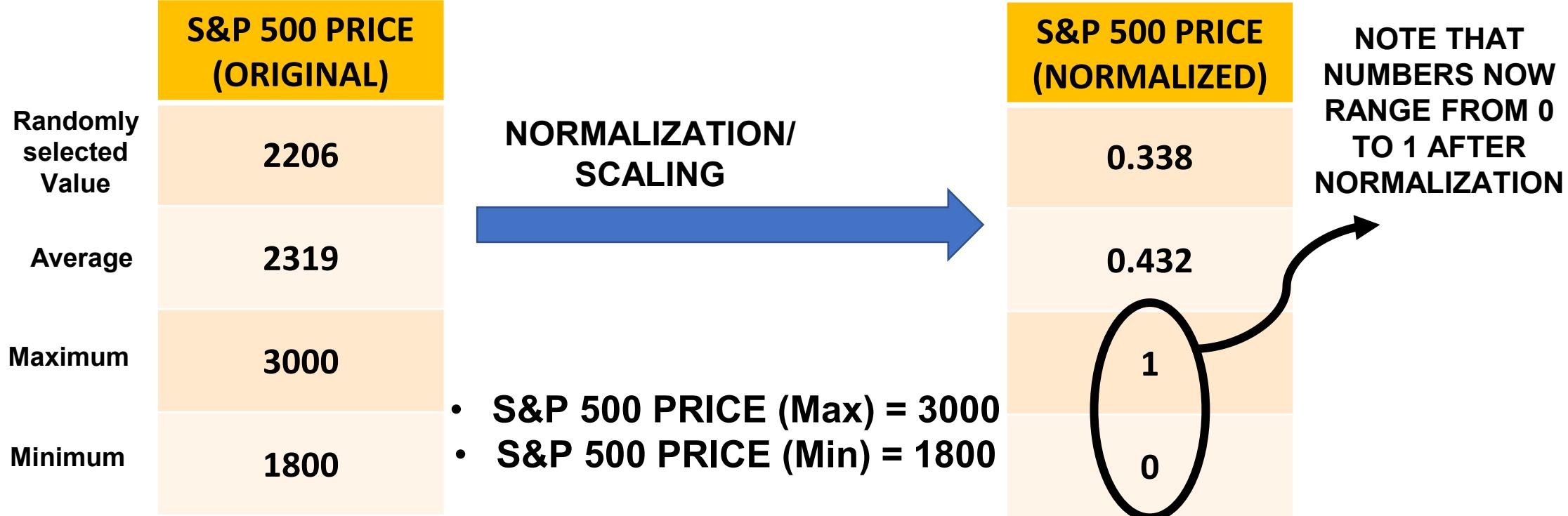
	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	2.20	56.25	2320.00
std	0.24	4.86	193.85
min	1.50	40.00	1800.00
25%	2.04	53.03	2190.45
50%	2.20	56.16	2312.44
75%	2.36	59.42	2455.76
max	3.00	70.00	3000.00



	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

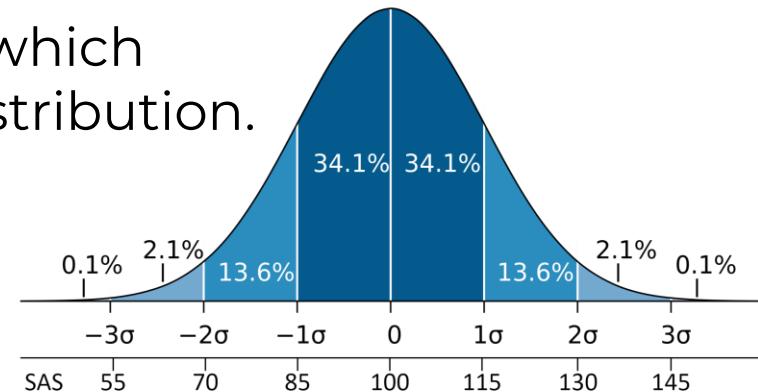
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} = \frac{2206 - 1800}{3000 - 1800} = 0.338$$



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.

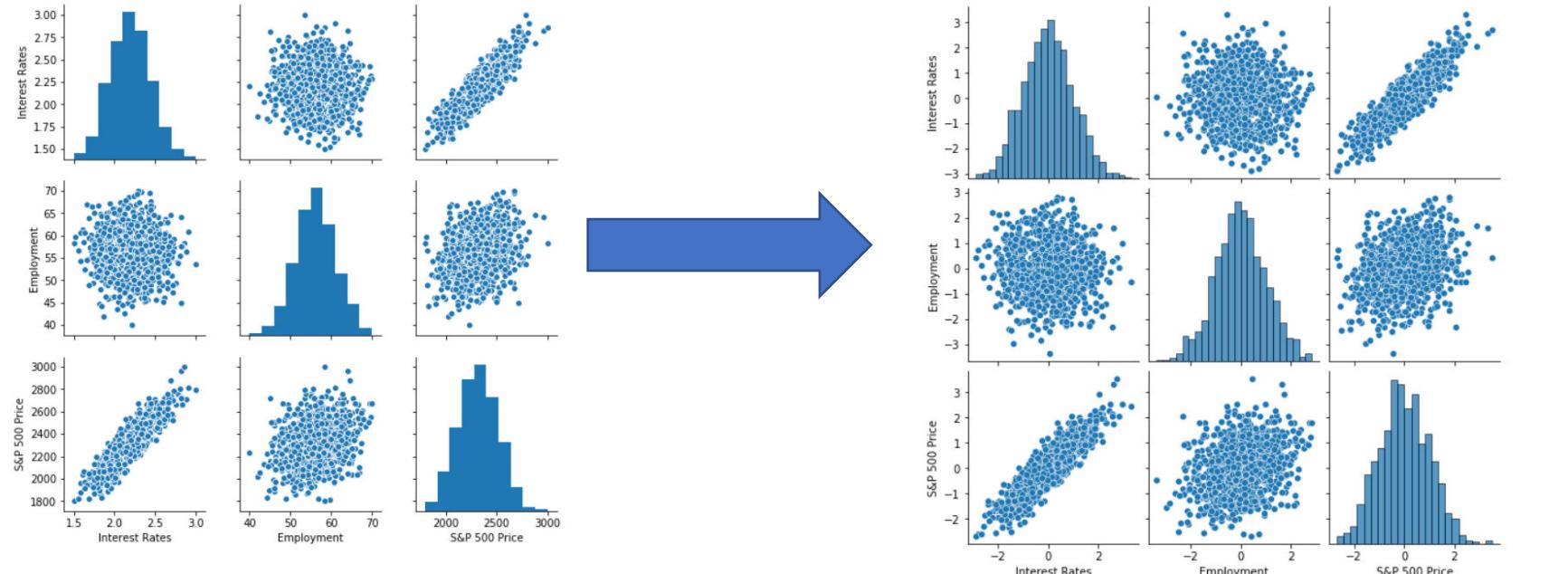
$$z = \frac{x - \bar{x}}{\sigma}$$



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

STANDARDIZATION

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



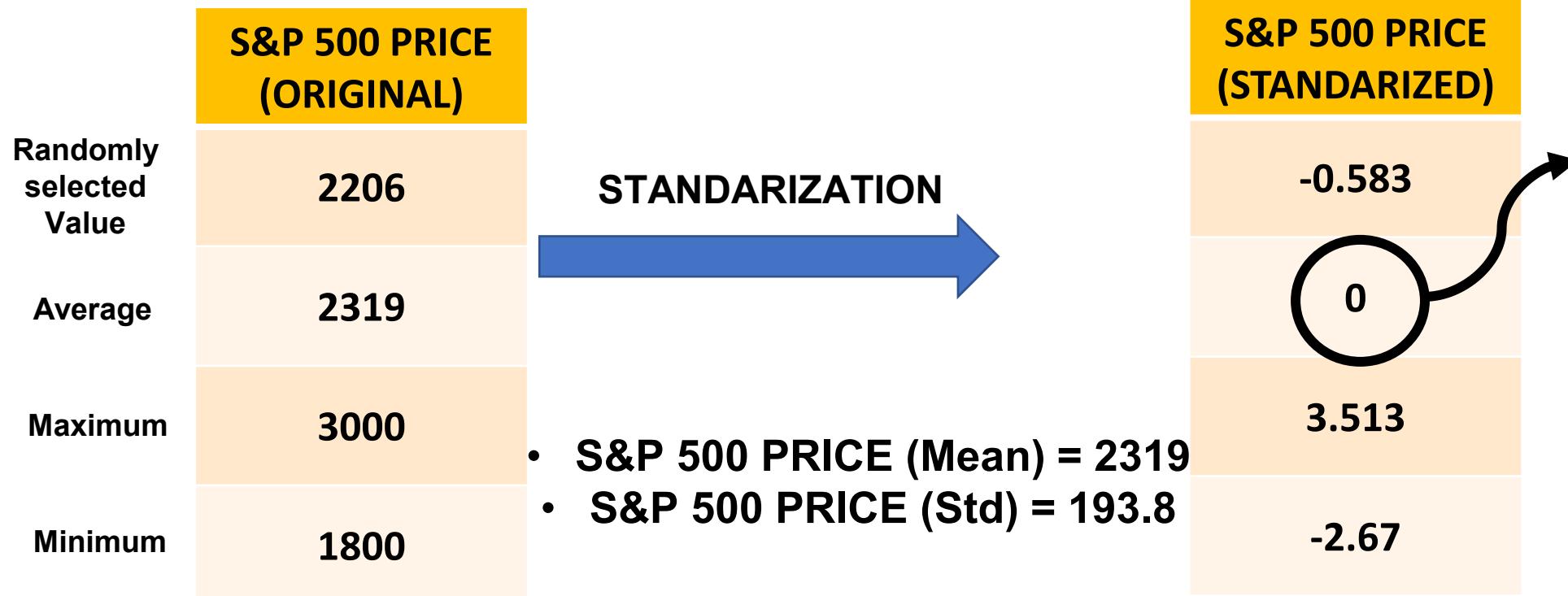
	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	2.20	56.25	2320.00
std	0.24	4.86	193.85
min	1.50	40.00	1800.00
25%	2.04	53.03	2190.45
50%	2.20	56.16	2312.44
75%	2.36	59.42	2455.76
max	3.00	70.00	3000.00

The figure displays a 3x3 grid of plots illustrating the transformation of data through standardization. The variables are Interest Rates, Employment, and S&P 500 Price. The first row shows histograms for each variable. The second row shows scatter plots between pairs of variables. The third row shows histograms for each variable again. Red ovals highlight the mean and std values in the summary statistics table below.

	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

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



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

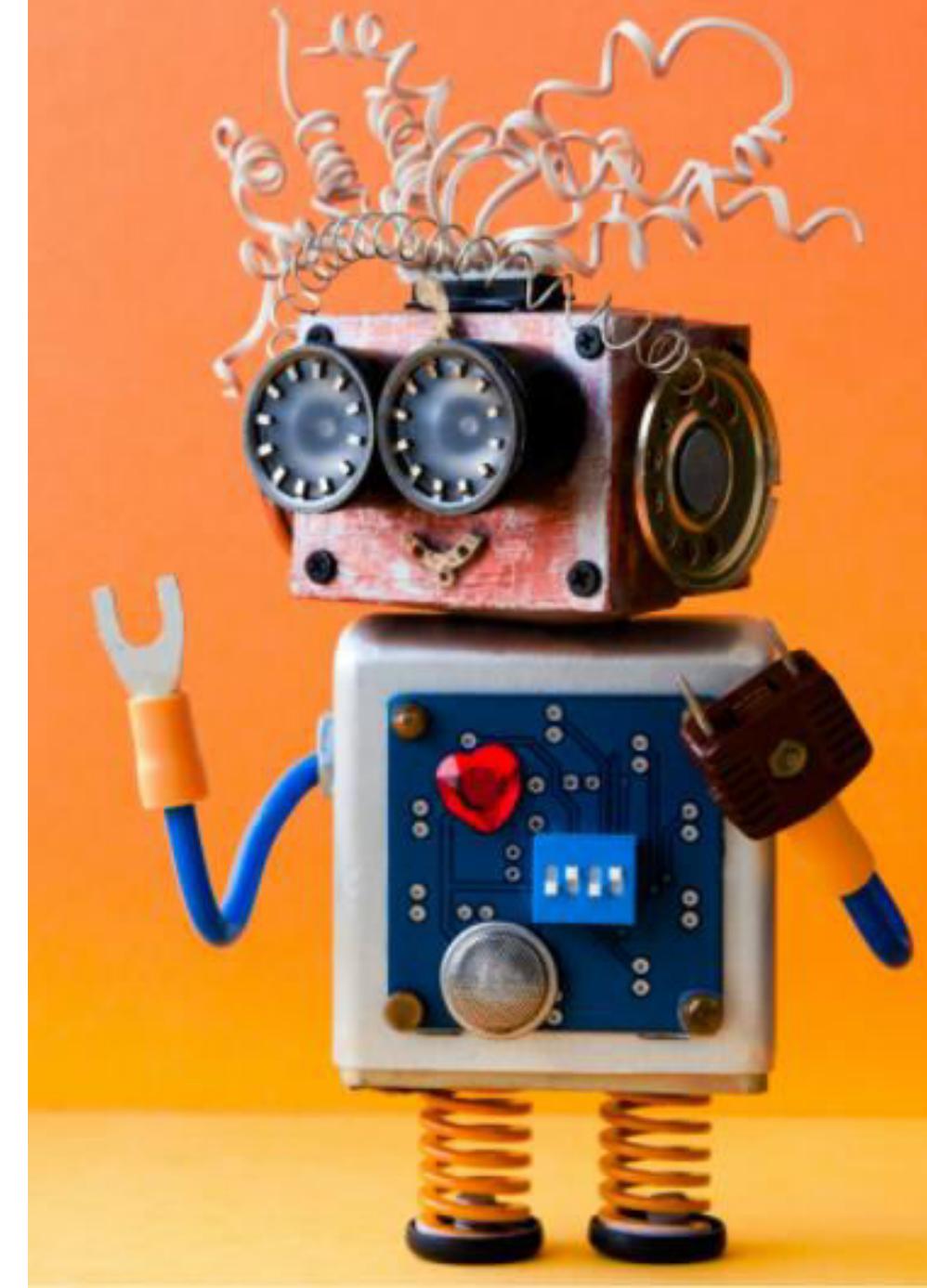
DEMO #1: LAUNCHING & UPLOADING DATA TO S3



EASY



ADVANCED



DATA WRANGLER DEMO

GO TO SAGEMAKER HOMEPAGE



The screenshot shows the Amazon SageMaker homepage within the AWS Management Console. The left sidebar contains navigation links for Dashboard, Search, SageMaker Domain (Studio, RStudio, Canvas), Images (Ground Truth, Notebook, Processing, Training, Inference, Edge Manager, Augmented AI, AWS Marketplace), and a large 'How it works' section. The main content area features the title 'Amazon SageMaker' and the subtitle 'Build, train, and deploy machine learning models at scale'. Below this is a sub-subtitle 'The quickest and easiest way to get ML models from idea to production.' To the right, there are sections for 'Get started' (with a 'SageMaker Studio' button), 'Pricing (US)', and 'Related services' (AWS Glue, Amazon EC2, Amazon Elastic Block Store (EBS)). The 'How it works' section illustrates the workflow with three stages: 'Label' (cloud icon with a person and network), 'Build' (cloud icon with documents and a gear), and 'Train' (cloud icon with a neural network and a gear). A yellow border surrounds the entire screenshot.

DATA WRANGLER DEMO

CLICK ON LAUNCH APP

The screenshot shows the Amazon SageMaker console interface. The left sidebar has a 'SageMaker Domain' section selected. The main content area is titled 'SageMaker Domain' and shows details for a domain named 'default-1643559608638'. The domain status is 'Ready'. It includes sections for 'Users', 'Domain', 'Projects', and 'Customer SageMaker Studio images attached to domain'. A 'Launch app' button is visible in the 'Domain' section. The entire screenshot is highlighted with a thick orange border.

SageMaker Domain

Users

Name

default-1643559608638

Domain

Status: Ready

Domain ID: d-qhcy9sxcpp9s

Execution role: arn:aws:iam::422132866096:role/service-role/AmazonSageMaker-ExecutionRole-20220121T103405

Authentication method: AWS Identity and Access Management (IAM)

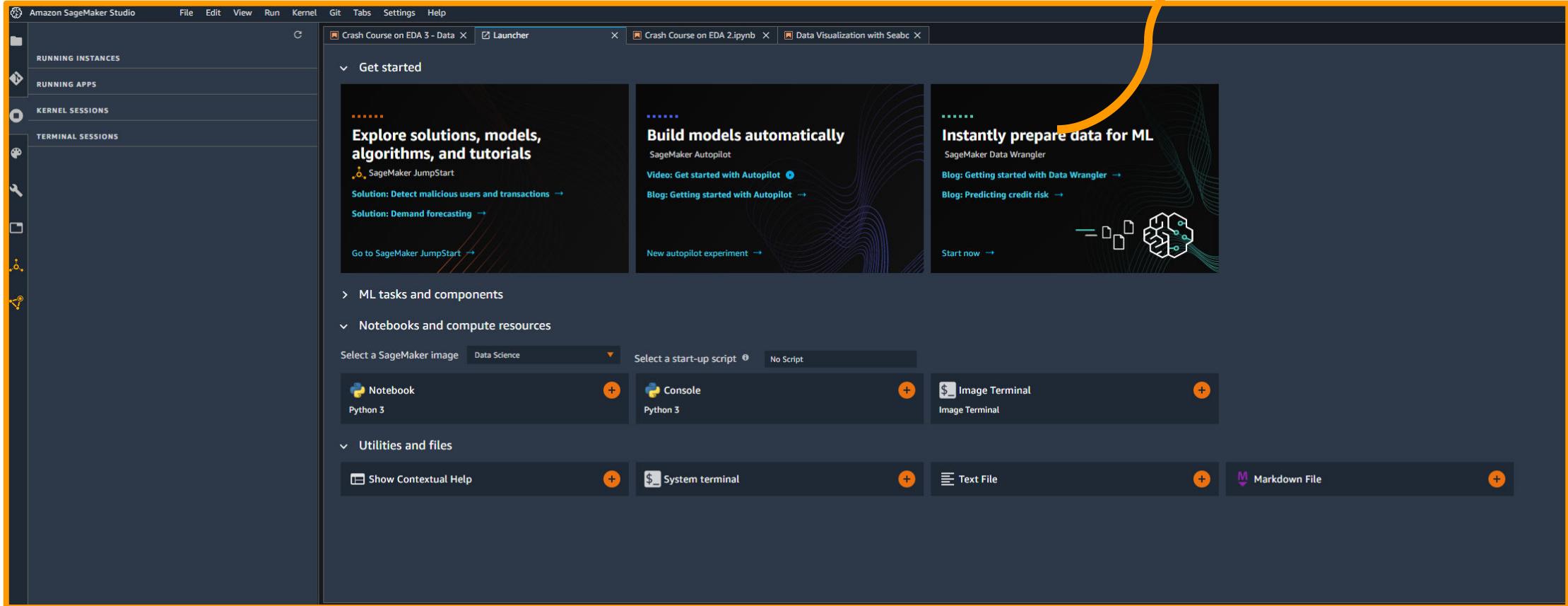
Projects:

- Amazon SageMaker project templates enabled for this account
- Amazon SageMaker project templates enabled for Studio users

Customer SageMaker Studio images attached to domain

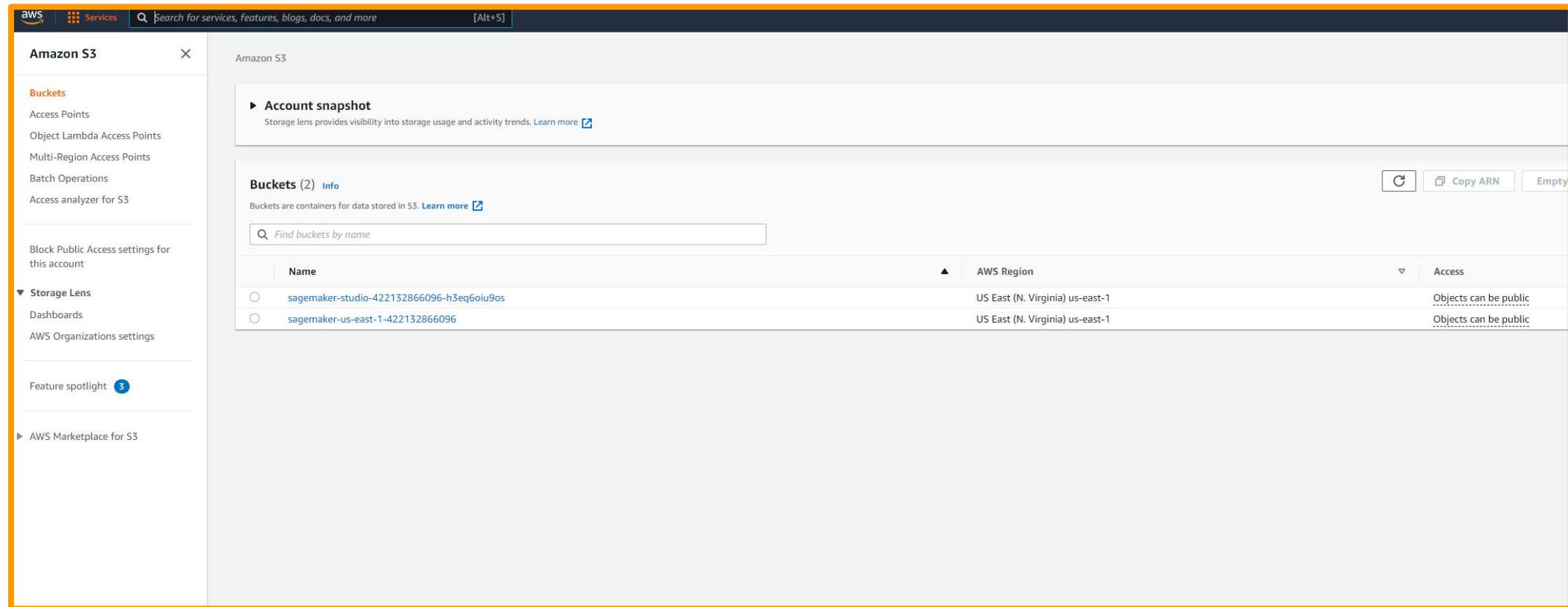
DATA WRANGLER DEMO

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.



The screenshot shows the AWS S3 console interface. On the left, there is a navigation sidebar with the following sections:

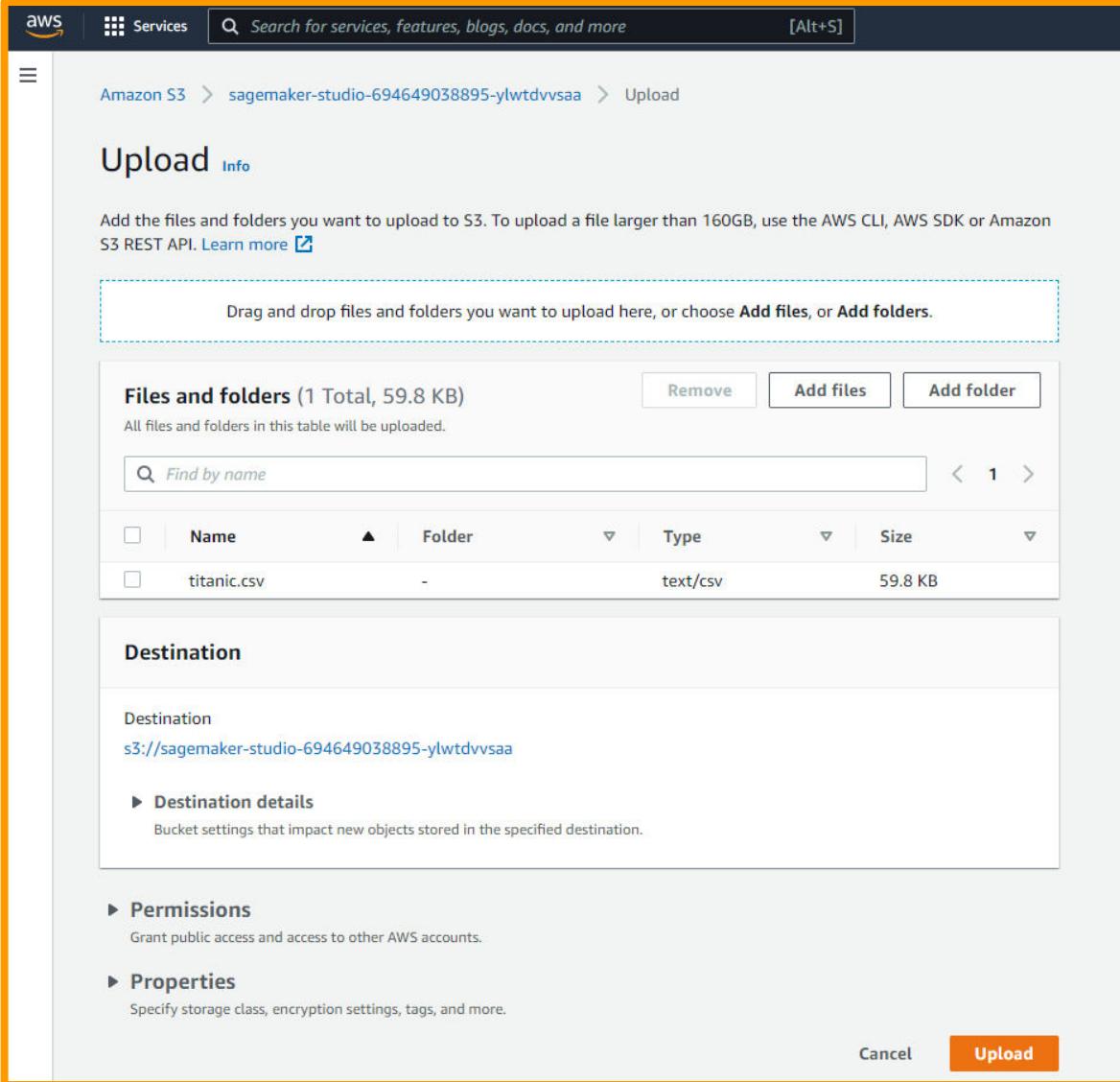
- Buckets**: Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, Access analyzer for S3.
- Storage Lens**: Dashboards, AWS Organizations settings.
- Feature spotlight (3)
- AWS Marketplace for S3

The main content area is titled "Amazon S3" and "Amazon S3". It features an "Account snapshot" section with a link to "Storage lens provides visibility into storage usage and activity trends. Learn more". Below this is a "Buckets (2) Info" section with a link to "Buckets are containers for data stored in S3. Learn more". A search bar labeled "Find buckets by name" is present. The bucket list table has columns for Name, AWS Region, and Access. The data is as follows:

Name	AWS Region	Access
sagemaker-studio-422132866096-h3eq6oiu9os	US East (N. Virginia) us-east-1	Objects can be public
sagemaker-us-east-1-422132866096	US East (N. Virginia) us-east-1	Objects can be public

DATA WRANGLER DEMO

DRAG AND DROP TITANIC.CSV to S3

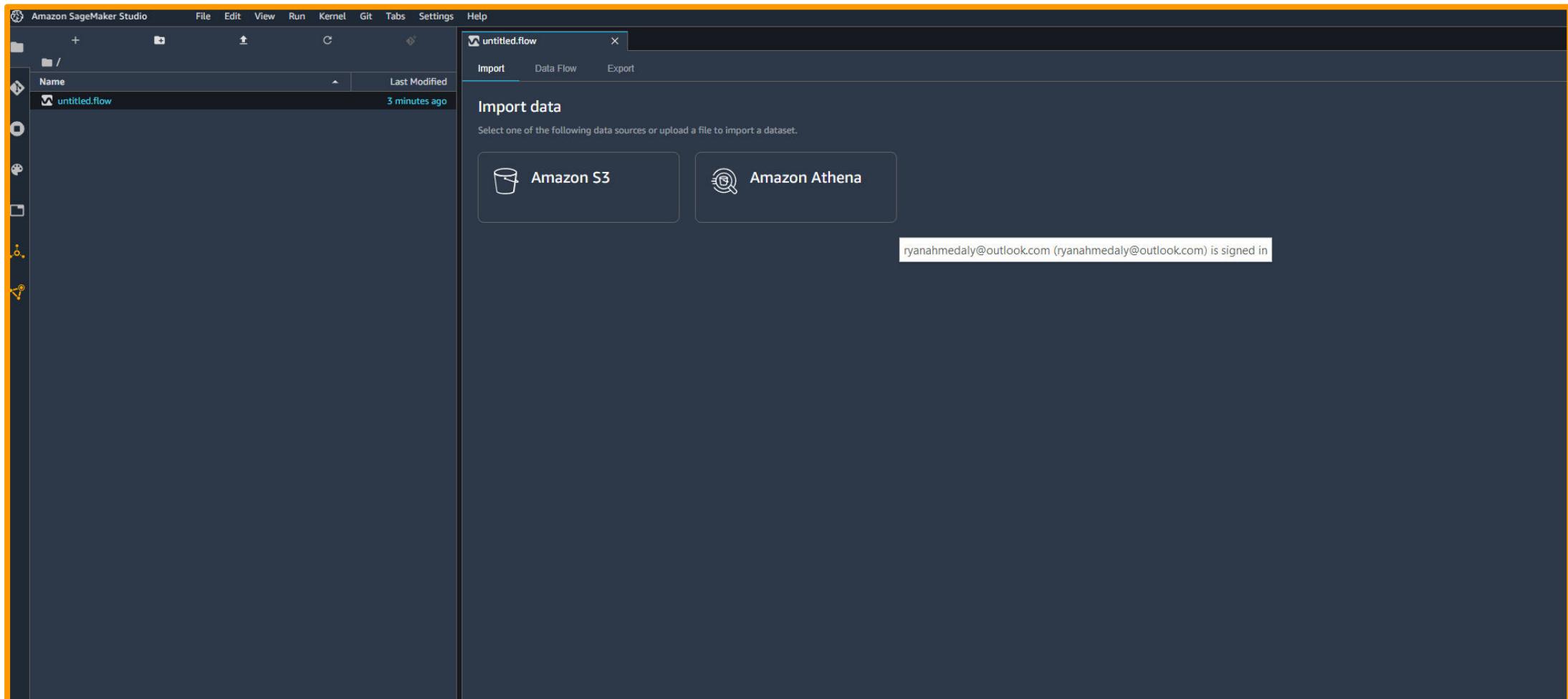


The screenshot shows the AWS S3 'Upload' interface. At the top, the navigation bar includes the AWS logo, 'Services' dropdown, search bar ('Search for services, features, blogs, docs, and more'), and keyboard shortcut '[Alt+S]'. Below the navigation, the path 'Amazon S3 > sagemaker-studio-694649038895-ylwtdvvsaa > Upload' is displayed. The main section is titled 'Upload' with an 'Info' link. A descriptive text explains how to upload files and folders, mentioning the AWS CLI, AWS SDK, or Amazon S3 REST API, with a 'Learn more' link. A large dashed blue box provides a placeholder for dragging and dropping files. Below this, a table lists the uploaded file: 'titanic.csv' (1 Total, 59.8 KB). The table has columns for Name, Folder, Type, and Size. Buttons for 'Remove', 'Add files', and 'Add folder' are available above the table. A search bar labeled 'Find by name' is present. The 'Destination' section shows the target bucket as 's3://sagemaker-studio-694649038895-ylwtdvvsaa'. Under 'Destination details', it says 'Bucket settings that impact new objects stored in the specified destination.' The 'Permissions' section allows granting public access and access to other AWS accounts. The 'Properties' section lets you specify storage class, encryption settings, tags, and more. At the bottom right are 'Cancel' and 'Upload' buttons.

Name	Folder	Type	Size
titanic.csv	-	text/csv	59.8 KB

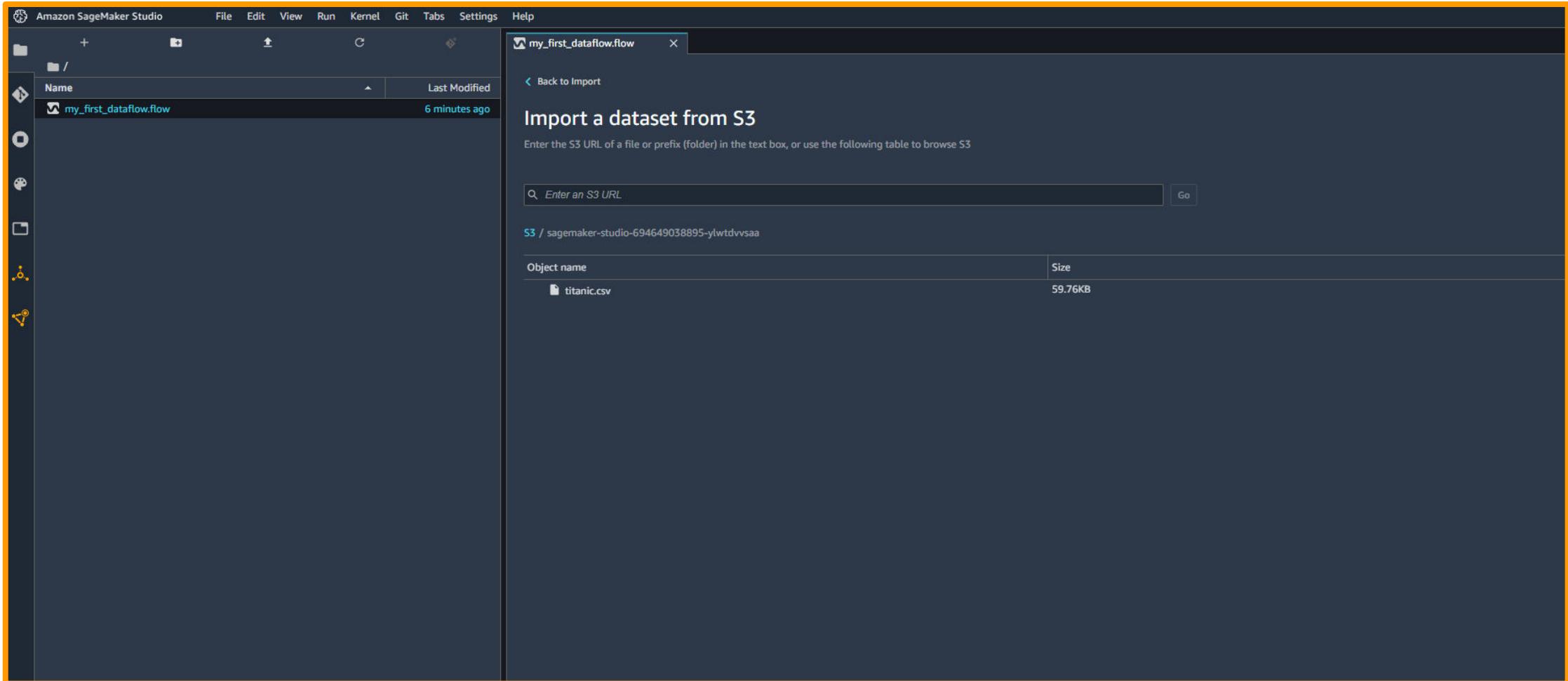
DATA WRANGLER DEMO

SELECT DATA FROM S3



DATA WRANGLER DEMO

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



DATA WRANGLER

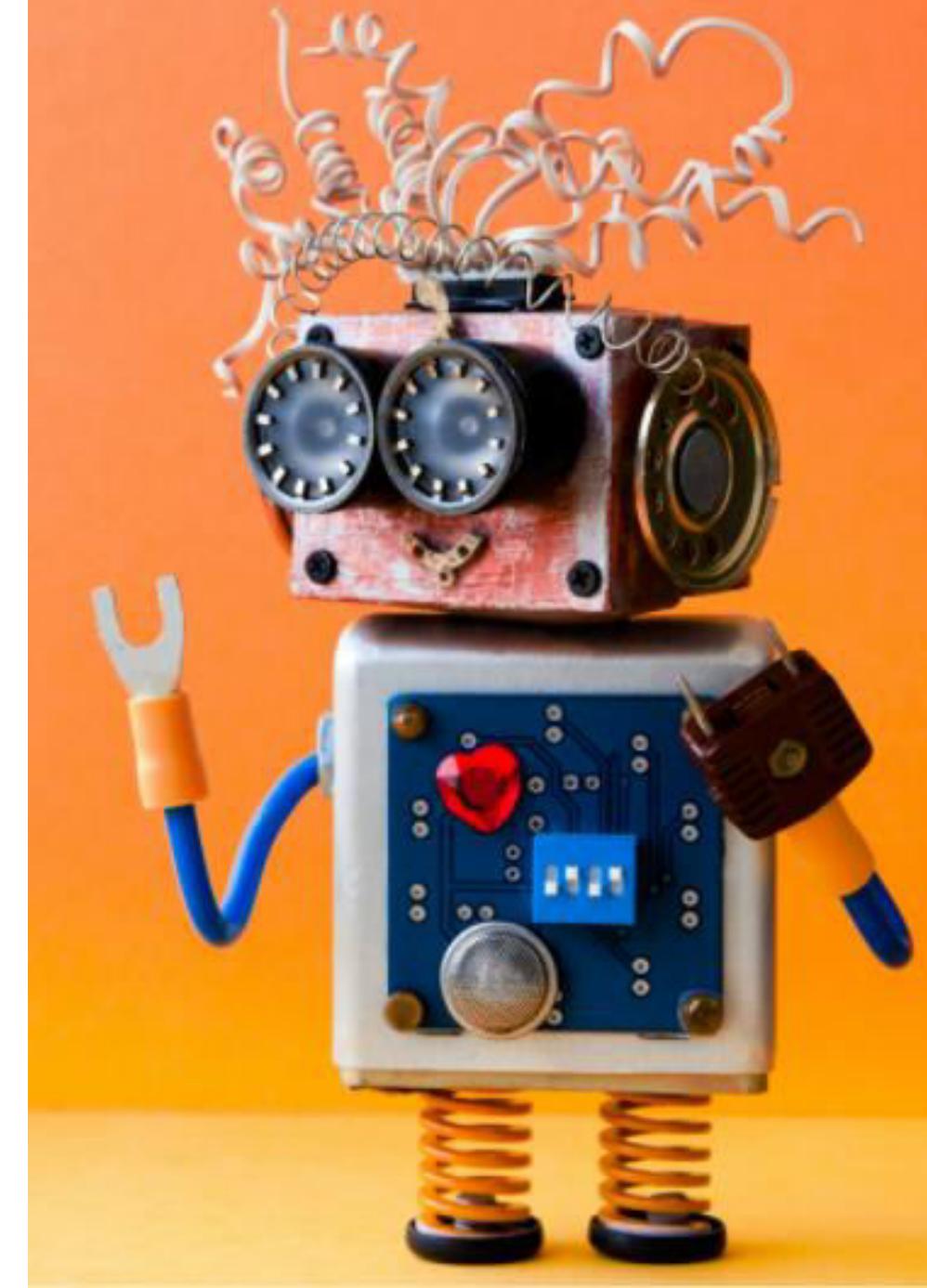
DEMO #2:

PERFORM EDA (CHANGE DATA TYPE & SUMMARY)



EASY

ADVANCED



DATA WRANGLER DEMO

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

The screenshot shows the 'my_first_dataflow.flow' interface with the 'Import a dataset from S3' step selected. A modal window displays the 'titanic.csv' file from the S3 bucket 'sagemaker-studio-694649038895-ylwtvdvsa'. The modal includes a search bar, a table of object details, and a preview of the first 100 rows of the CSV file.

Object name: titanic.csv

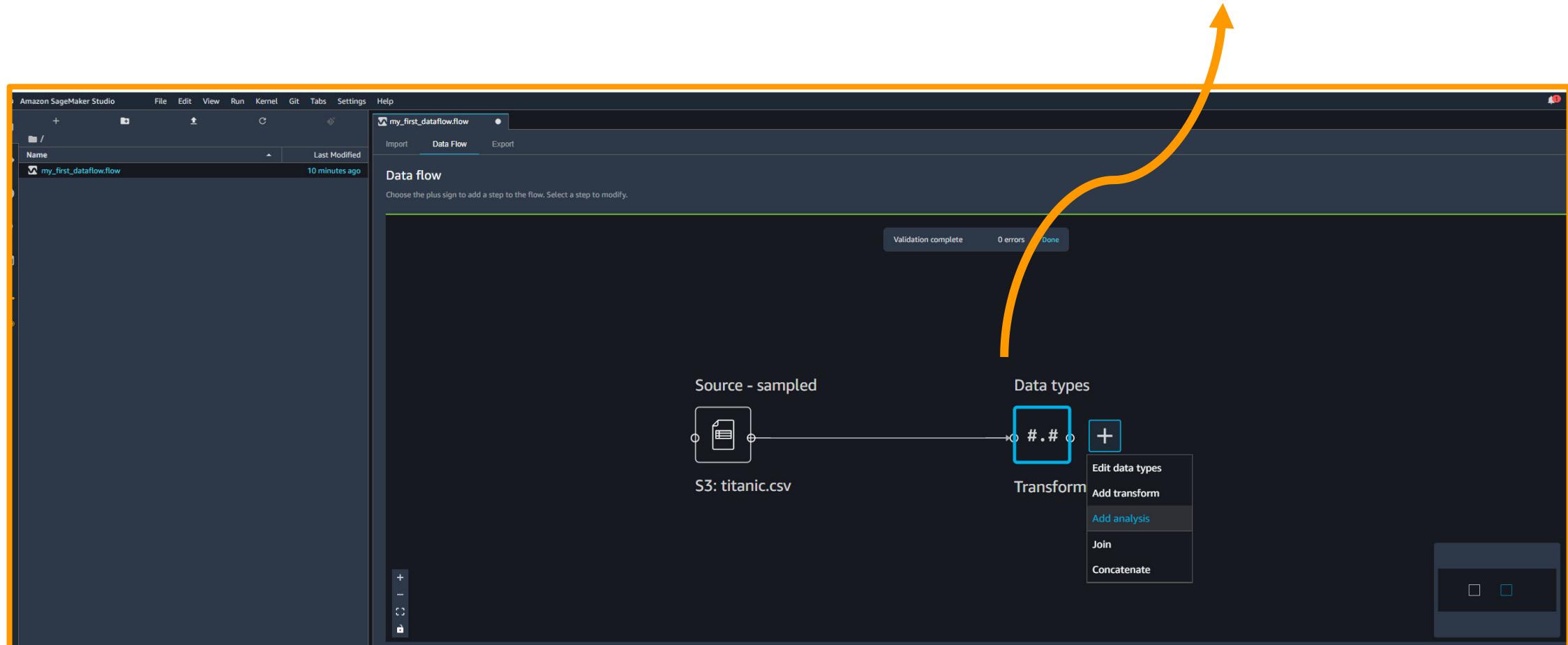
Object name	Size
titanic.csv	59.76KB

PREVIEW • titanic.csv (first 100 rows shown)

PassengerId	Survived	Pclass	Name	Sex	Age
1	0	3	Braund, Mr. Owen Harris	male	22
2	1	1	Cumings, Mrs. John Bra...	female	38
3	1	3	Heikkinen, Miss. Laina	female	26
4	1	1	Futrelle, Mrs. Jacques H...	female	35
5	0	3	Allen, Mr. William Henry	male	35
6	0	3	Moran, Mr. James	male	
7	0	1	McCarthy, Mr. Timothy J	male	54
8	0	3	Palsson, Master. Gosta ...	male	2
9	1	3	Johnson, Mrs. Oscar W (...	female	27
10	1	2	Nasser, Mrs. Nicholas (A...	female	14
11	1	3	Sandstrom, Miss. Margu...	female	4
12	1	1	Bonnell, Miss. Elizabeth	female	58

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

The screenshot shows the Amazon SageMaker Studio interface with the 'my_first_dataflow.flow' file open. The main area displays the 'Data types - Transform: titanic.csv' step. On the right, there's a 'CONFIGURE TYPES' panel where the 'Fare' column is being edited from 'Float' to 'Long'. The 'Fare' column currently has a value of 7.25.

PassengerId (long)	Survived (long)	Pclass (long)	Name (string)	Sex (string)	Age (long)	SibSp (long)	Parch (long)	Ticket (string)	Fare (float)
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25
2	1	1	Cumings, Mrs. John Bra... Heikkinen, Miss. Laina	female	38 26	1 0	0	PC 17599 STON/O2. 3101282	71.28 7.925
3	1	3	Futrelle, Mrs. Jacques H... Allen, Mr. William Henry	female male	35 35	1 0	0	113803 373450	53.1 8.05
4	1	1	Moran, Mr. James	male	35	0	0	350877	8.45
5	0	3	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.88
6	0	3	Palsson, Master. Gosta ...	male	2	3	1	349909	21.0
7	0	1	Johnson, Mrs. Oscar W ...	female	27	0	2	347742	11.13
8	0	3	Nasser, Mrs. Nicholas (A... Sandstrom, Miss. Margu... Bonnell, Miss. Elizabeth	female female female	14 4 58	1 1 0	0	237736 PP 9549 113783	30.0 16.7 26.5!
9	1	2	Saundercock, Mr. Willia... Andersson, Mr. Anders J... Vestrom, Miss. Hulda A...	male male female	20 39 14	0 1 0	5	347082 350406 248706	31.25 7.854 16
10	1	3	Hewlett, Mrs. (Mary D K... Rice, Master. Eugene	female male	55 2	0 4	1	382652	29.13
11	1	1	Williams, Mr. Charles Eu... Vander Planke, Mrs. Juli...	male female	31	1	0	244373 345765	13 18
12	0	3	Masselmann, Mrs. Fatima	female	0	0	0	2649	7.225
13	0	3	Fynney, Mr. Joseph J	male	35	0	0	239865	26
14	0	2	Beesley, Mr. Lawrence	male	34	0	0	248698	13
15	1	2	McGowan, Miss. Anna "	female	15	0	0	330923	8.025
16	1	1	Sloper, Mr. William Tho...	male	28	0	0	113788	35.5
17	0	3	Palsson, Miss. Torborg ...	female	8	3	1	349909	21.0
18	1	1	Alvarez, Mr. G. L. G. ...	male	70	1	0	244373	13

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"

The screenshot shows the Amazon SageMaker Studio interface. On the left, there's a sidebar with various icons and a file tree showing a folder named 'my_first_dataflow.flow'. The main area is titled 'my_first_dataflow.flow' and contains a 'Data types - Transform: titanic.csv' section. Below this, there are two tabs: 'Data' (which is currently active) and 'Analysis'. The 'Analysis' tab displays a 'Table Summary' table for the 'Untitled' dataset. The table includes columns for summary statistics (count, mean, stddev, min, max) and specific passenger details (PassengerId, Survived, Pclass, Name, Sex, Age, SibSp). The 'Age' and 'SibSp' columns show missing values (None). Below the summary table is a 'Data table' view showing the first 11 rows of the titanic.csv dataset.

summary	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp
count	891	891	891	891	891	714	891
mean	446.0	0.3838383838383838	2.308641975308642	None	None	29.679271708683473	0.523007856341
stddev	257.3538420152301	0.48659245426485753	0.8360712409770491	None	None	14.536482769437564	1.10274343229
min	1	0	1	Abbing, Mr. Anthony	female	0	0
max	891	1	3	van Melkebeke, Mr. Phil...	male	80	8

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
4	1	1	Futrelle, Mrs. Jacques H...	female	35	1	0
5	0	3	Allen, Mr. William Henry	male	35	0	0
6	0	3	Moran, Mr. James	male		0	0
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0
8	0	3	Palsson, Master. Gosta ...	male	2	3	1
9	1	3	Johnson, Mrs. Oscar W. (...)	female	27	0	2
10	1	2	Nasser, Mrs. Nicholas (A...)	female	14	1	0
11	1	2	C尤尔波恩, Miss. M...	female	4	1	1

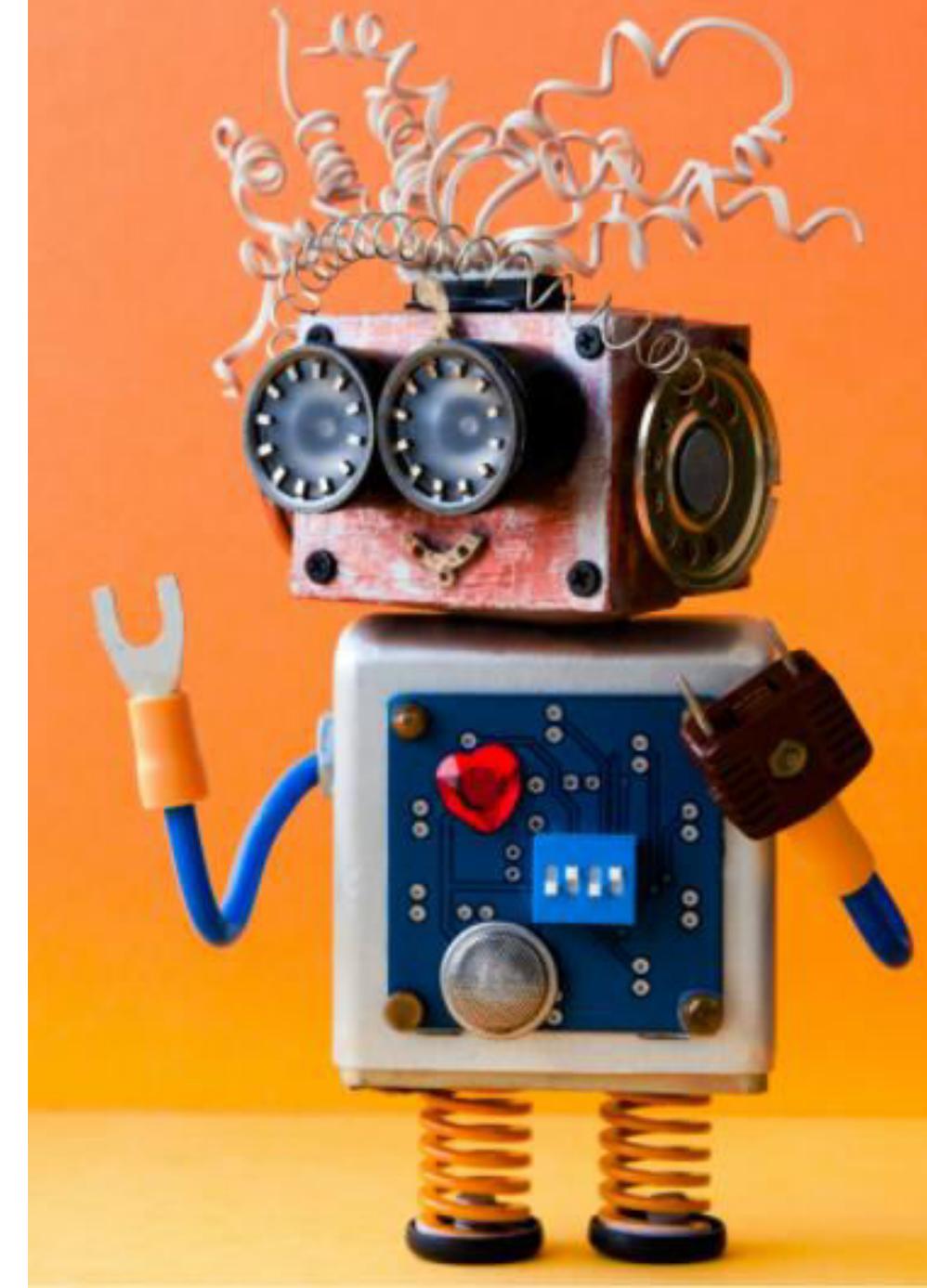
DATA WRANGLER

DEMO #3: PERFORM DATA VISUALIZATION



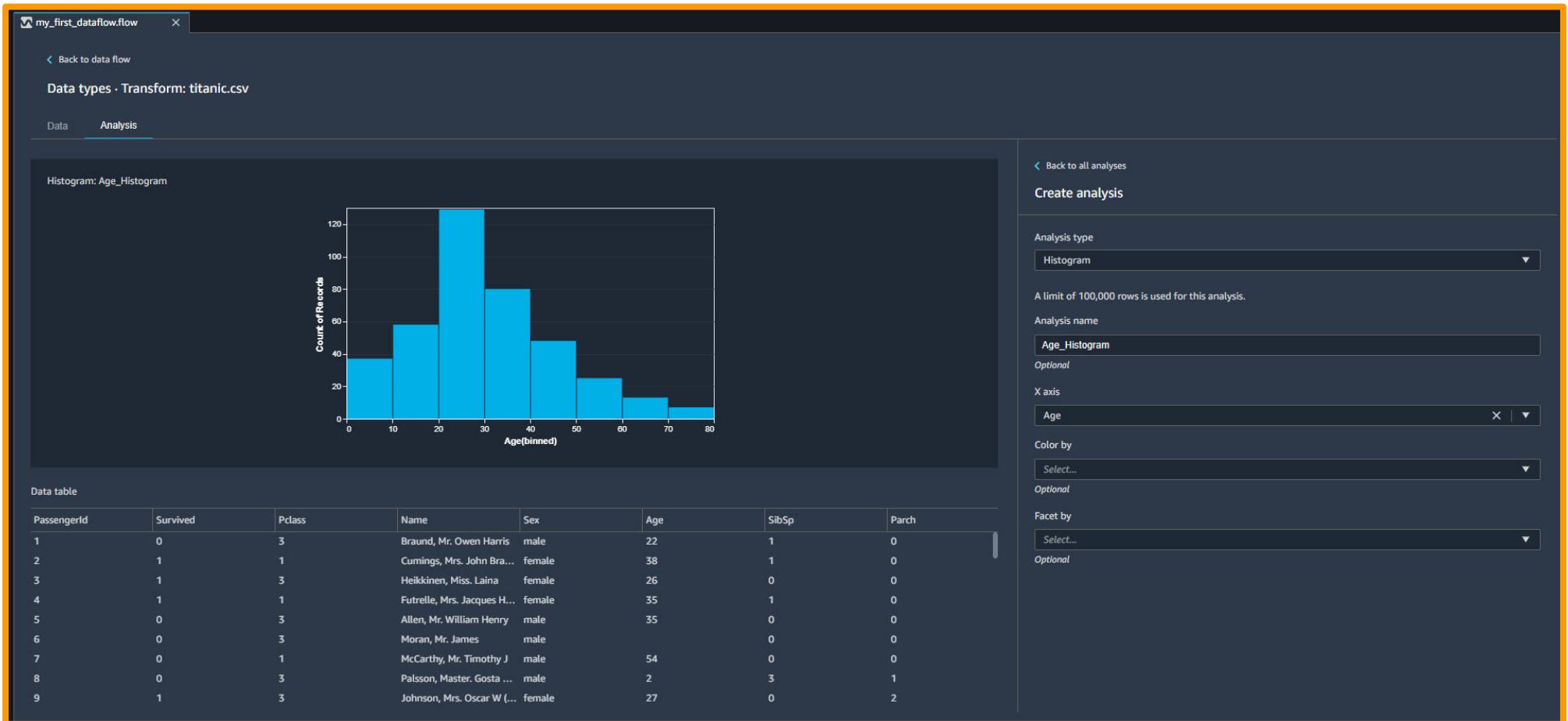
EASY

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 categorical 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:

correlation

PassengerId
Survived
Pclass
Age
SibSp
Parch
Fare
Sex
Embarked

PassengerId
Survived
Pclass
Age
SibSp
Parch
Fare
Sex
Embarked

Optional

Create analysis

Analysis type: Feature Correlation

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

Analysis name: Feature_Correlations

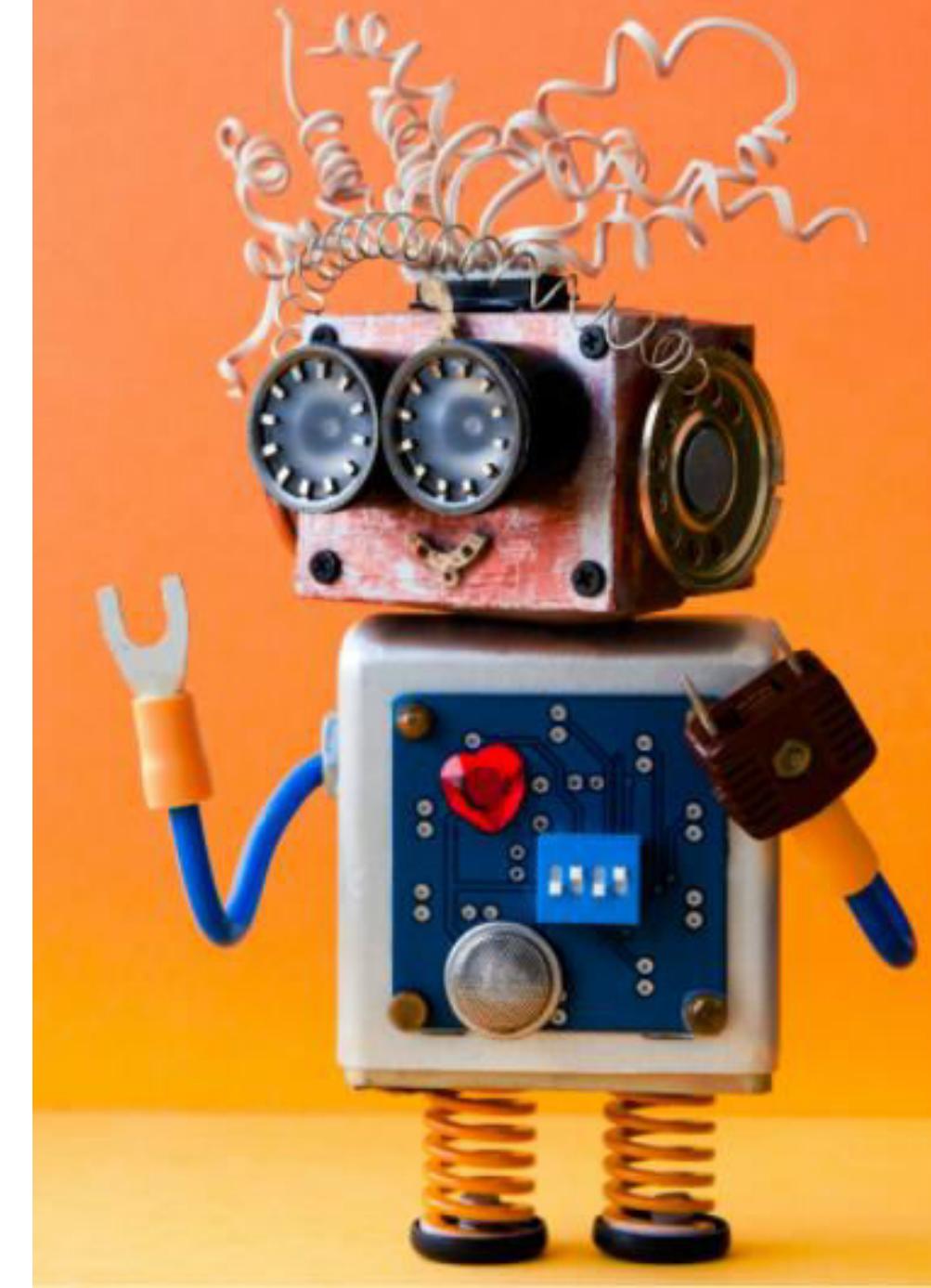
Correlation type: linear

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

EASY



ADVANCED



DATA WRANGLER DEMO

LET'S SEE IF THERE ARE ANY DUPLICATED ROWS

my_first_dataflow.flow

Back to data flow

Data types - Transform: titanic.csv

Data Analysis

Duplicate rows: Duplicated

No duplicate rows were found in the 891 rows tested.

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
4	1	1	Futrelle, Mrs. Jacques H...	female	35	1	0
5	0	3	Allen, Mr. William Henry	male	35	0	0
6	0	3	Moran, Mr. James	male		0	0
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0
8	0	3	Palsson, Master. Gosta ...	male	2	3	1
9	1	3	Johnson, Mrs. Oscar W (...	female	27	0	2
10	1	2	Nasser, Mrs. Nicholas (A...	female	14	1	0

Back to all analyses

Create analysis

Analysis type: Duplicate rows

Count the number of unique vs duplicate rows and present the most common duplicate rows

Analysis name: Duplicated

Optional

Clear

Preview Save

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.

Importance

column

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

my_first_dataflow.flow

Data Analysis

Bias Report: Bias

The computed bias metrics are below:

Predicted column: Survived

Predicted value or threshold: 0

Column analyzed for bias: Sex

Column value or threshold analyzed for bias: female

Class Imbalance (CI): 0.5

Detects if the advantaged group is represented in the dataset at a substantially higher rate than the disadvantaged group, or vice versa.

Difference in Positive Proportions in Labels (DPL): 0.55

Detects if one class has a significantly higher proportion of desirable (or, alternatively, undesirable) outcomes in the training data.

Jensen-Shannon Divergence (JS): 0.16

JS measures how much the label distributions of different classes diverge from each other. If the average label distribution across all of the classes is P, the JS divergence is the average of the KL divergences of the probability distributions for each class from the average distribution P. This entropic measure also generalizes to multiple label and continuous cases.

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
4	1	1	Futrelle, Mrs. Jacques H...	female	35	1	0

Back to all analyses

Create analysis

Analysis type: Bias Report

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

Analysis name: Bias

Select the column your model predicts (target): Survived

Is your predicted column a value or threshold? Value (radio button selected)

Predicted value(s): 0

Select the column to analyze for bias: Sex

Is your column a value or threshold? Value (radio button selected)

Column value(s) to analyze for bias: female

Optional

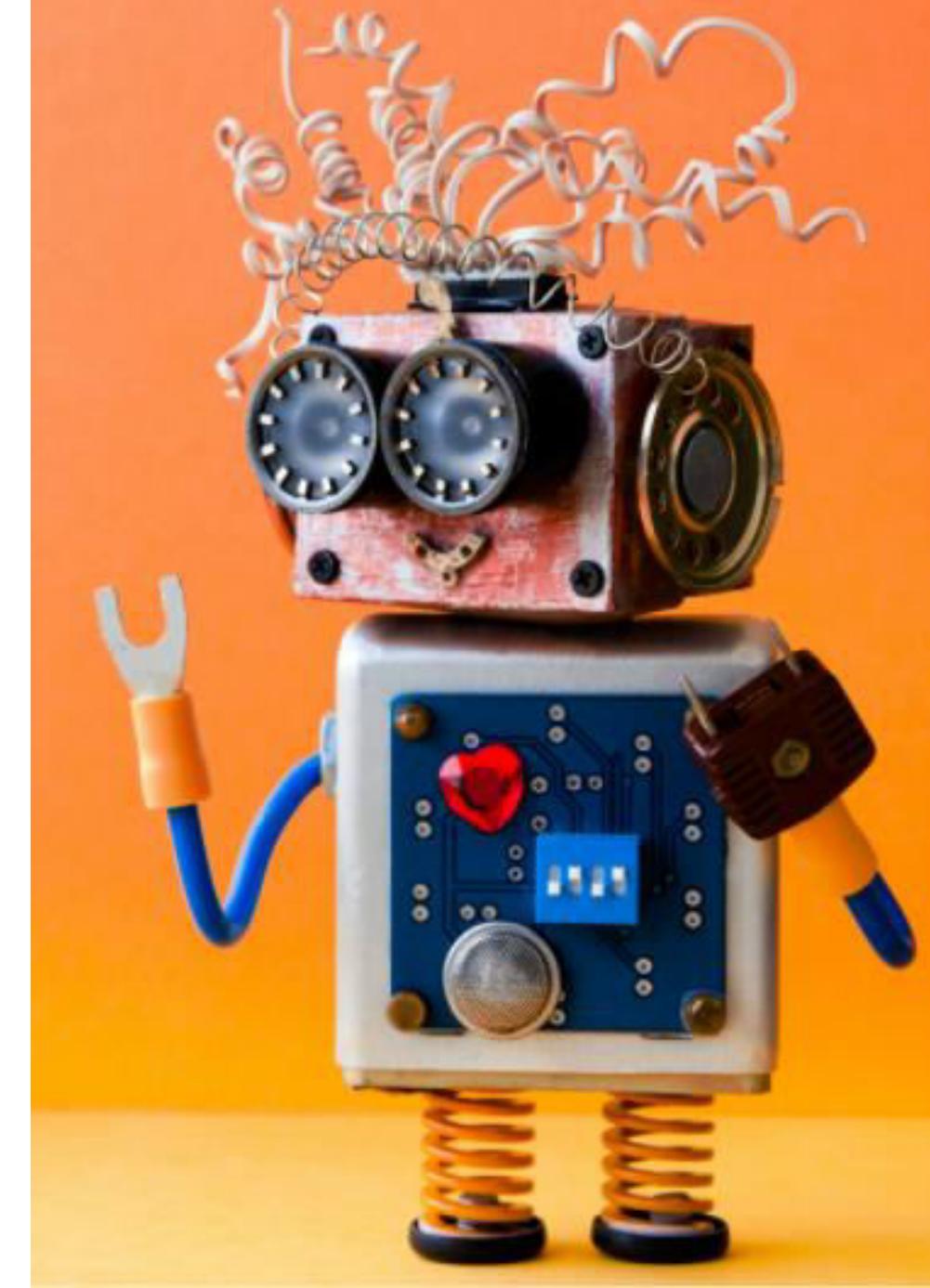
Choose bias metrics

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



EASY

ADVANCED



DATA WRANGLER DEMO

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

Step 2. Data types										Export data		Add transform	X
PassengerId (long)	Survived (long)	Pclass (long)	Name (string)	Sex (string)	Age (long)	SibSp (long)	Parch (long)	Ticket (string)	Fare (float)				
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	Saunderscock, 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

The screenshot shows the Data Wrangler interface with a dark theme. On the left, there's a preview of the 'titanic.csv' dataset with columns: Age (long), SibSp (long), Parch (long), Ticket (string), Fare (long), Cabin (string), Embarked (string), Sex_male (float), and Sex_female (float). The preview shows several rows of passenger data, including names like Isidorus, M. I. Alf., and Ed... The right side of the interface has a sidebar titled 'my_first_dataflow.flow' with a 'Data types - Transform: titanic.csv' section. A modal window titled 'ENCODE CATEGORICAL' is open, containing settings for transforming the 'Sex' column. The 'Transform' section is set to 'One-hot encode'. The 'Input columns' section lists 'Sex' with a dropdown menu. Under 'Optional', there are checkboxes for 'Input already ordinal encoded', 'Drop last', and 'Output style'. Buttons for 'Preview' and 'Add' are at the bottom right of the modal.

my_first_dataflow.flow

Back to data flow

Data types - Transform: titanic.csv

Data Analysis

Previewing: Encode categorical

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

ENCODE CATEGORICAL

Convert categorical variables to numeric or vector representations. [Learn more](#).

Transform [?](#)

One-hot encode

Input columns [?](#)

Sex [X](#)

Input already ordinal encoded [?](#)

Invalid handling strategy [?](#)

Keep

Drop last [?](#)

Output style [?](#)

Columns [X | ▾](#)

Output column [?](#)

Optional

Clear

Preview Add

DATA WRANGLER DEMO

LET'S DROP THE CABIN COLUMN

my_first_dataflow.flow

Back to data flow

One-hot encode · Transform: titanic.csv

Data Analysis

Previewing: Manage columns

PassengerId (long)	Survived (long)	Pclass (long)	Name (string)	Age (long)	SibSp (long)	Parch (long)	Ticket (string)	Fare (long)	Embarked (string)
1	0	3	Braund, Mr. Owen Harris	22	1	0	A/5 21171	7	S
2	1	1	Cumings, Mrs. John Bra... 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... 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 ... Palsson, Master. Gosta ...	2	3	1	349909	21	S
9	1	3	Johnson, Mrs. Oscar W (... Johnson, Mrs. Oscar W (...	27	0	2	347742	11	S
10	1	2	Nasser, Mrs. Nicholas (A... Nasser, Mrs. Nicholas (A...	14	1	0	237736	30	C
11	1	3	Sandstrom, Miss. Margu... 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	Saunderscock, Mr. Willia... Saunderscock, Mr. Willia...	20	0	0	A/5. 2151	8	S
14	0	3	Andersson, Mr. Anders J... Andersson, Mr. Anders J...	39	1	5	347082	31	S
15	0	3	Vestrom, Miss. Hulda A... Vestrom, Miss. Hulda A...	14	0	0	350406	7	S
16	1	2	Hewlett, Mrs. (Mary D K... 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... Williams, Mr. Charles Eu...		0	0	244373	13	S
19	0	3	Vander Planke, Mrs. Juli... 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 " ...<br/ McGowan, Miss. Anna "	15	0	0	330923	8	Q
24	1	1	Sloper, Mr. William Tho... Sloper, Mr. William Tho...	28	0	0	113788	35	S
25	0	3	Palsson, Miss. Torborg ... Palsson, Miss. Torborg ...	8	3	1	349909	21	S
26	1	3	Asplund, Mrs. Carl Osca... Asplund, Mrs. Carl Osca...	38	1	5	347077	31	S

Manage Columns

Move, drop, duplicate or rename columns in the dataset. [Learn more.](#)

Transform [?](#)

Drop column [X](#) | [▼](#)

Columns to drop

Cabin [X](#) [▼](#)

Clear [Preview](#) [Add](#)

DATA WRANGLER DEMO

FILL OUT MISSING AGE VALUES

The screenshot shows the Data Wrangler interface for a flow named "my_first_dataflow.flow". The current step is "Drop column · Transform: titanic.csv". The interface has two tabs: "Data" (selected) and "Analysis".

Data Preview: The preview shows the first 20 rows of the titanic.csv dataset. The columns are: Age (long), SibSp (long), Parch (long), Ticket (string), Fare (long), Embarked (string), Sex_male (float), Sex_female (float), and New_Age (float). The "New_Age" column contains the value 29.622191011235955 for all rows where the original "Age" value is missing.

	Age (long)	SibSp (long)	Parch (long)	Ticket (string)	Fare (long)	Embarked (string)	Sex_male (float)	Sex_female (float)	New_Age (float)
1	22	1	0	A/5 21171	7	S	1	0	22
2	38	1	0	PC 17599	71	C	0	1	38
3	26	0	0	STON/O2. 3101282	7	S	0	1	26
4	35	1	0	113803	53	S	0	1	35
5	35	0	0	373450	8	S	1	0	35
6	0	0	0	330877	8	Q	1	0	29.622191011235955
7	54	0	0	17463	51	S	1	0	54
8	2	3	1	349909	21	S	1	0	2
9	27	0	2	347742	11	S	0	1	27
10	14	1	0	237736	30	C	0	1	14
11	4	1	1	PP 9549	16	S	0	1	4
12	58	0	0	113783	26	S	0	1	58
13	20	0	0	A/5. 2151	8	S	1	0	20
14	39	1	5	347082	31	S	1	0	39
15	14	0	0	350406	7	S	0	1	14
16	55	0	0	248706	16	S	0	1	55
17	2	4	1	382652	29	Q	1	0	2
18	0	0	0	244373	13	S	1	0	29.622191011235955
19	31	1	0	345763	18	S	0	1	31
20	0	0	0	2649	7	C	0	1	29.622191011235955
21	35	0	0	239865	26	S	1	0	35
22	34	0	0	248698	13	S	1	0	34
23	15	0	0	330923	8	Q	0	1	15
24	28	0	0	113788	35	S	1	0	28
25	8	3	1	349909	21	S	0	1	8
26	38	1	5	347077	31	S	0	1	38

Handle missing transform: A modal dialog titled "HANDLE MISSING" is open. It says: "Replace, drop, or add indicators for missing values. [Learn more.](#)".
Transform: **Impute**
Column type: **Numeric**
Input column: **Age**
Imputing strategy: **Mean**
Output column: **New_Age**
Optional:
Clear Preview Add

DATA WRANGLER

DEMO #6:

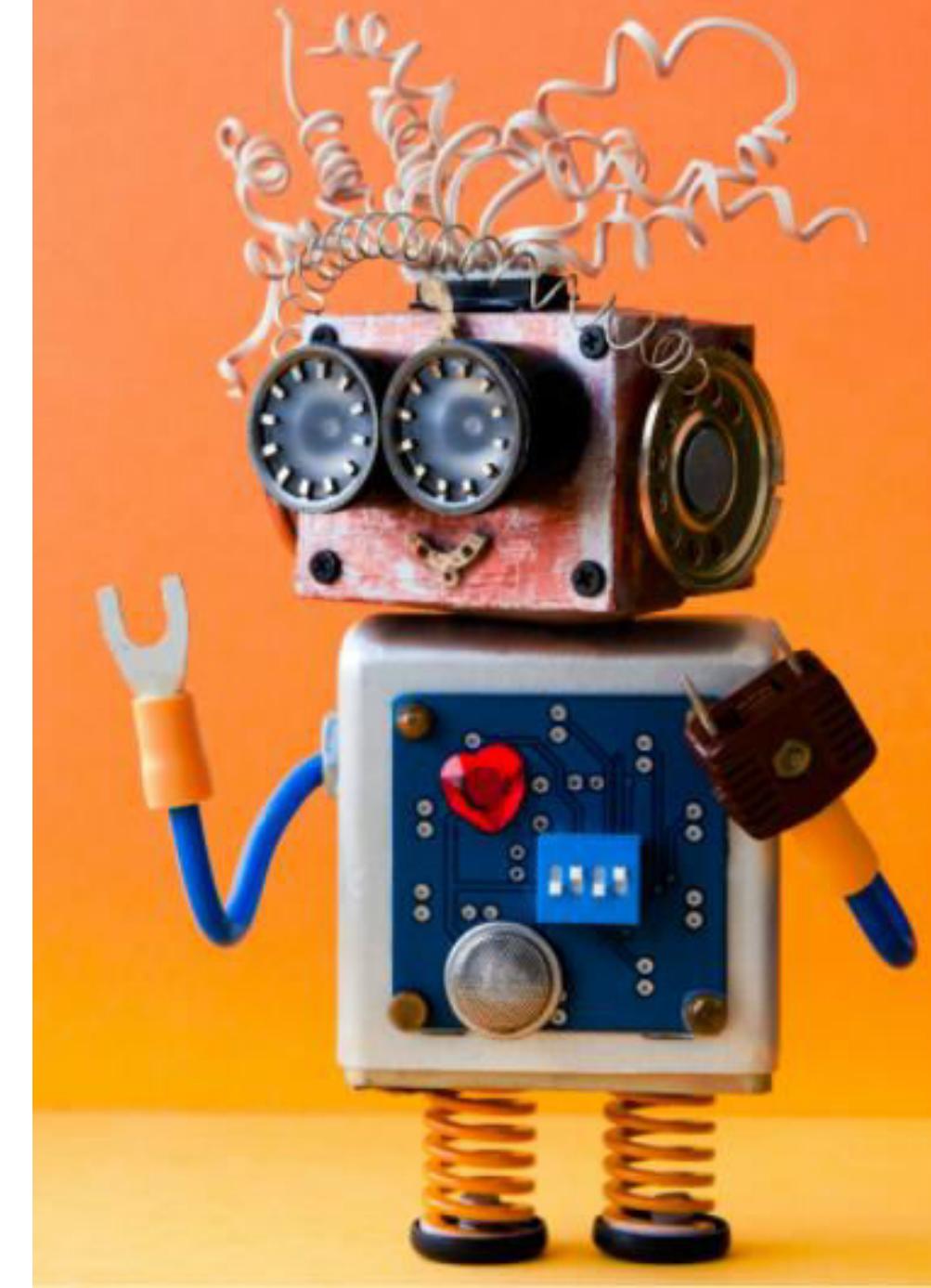
TRANSFORMATIONS

(CUSTOM TRANSFORM & SCALING)



EASY

ADVANCED



DATA WRANGLER DEMO

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

The screenshot shows the Data Wrangler interface for a flow named "my_first_dataflow.flow". The main area displays a preview of the "titanic.csv" dataset. The preview table has 26 rows and 10 columns, with the first few rows of data visible. The columns are: PassengerId (long), Survived (long), Pclass (long), Name (string), Age (float), SibSp (long), Parch (long), Ticket (string), Fare (long), and Embarked.

To the right of the preview, there is a "CUSTOM PANDAS" dialog box. This dialog is titled "CUSTOM PANDAS" and contains instructions for defining custom transformations using PySpark, Pandas, or PySpark (SQL). It includes a warning message about using Python (Pandas) for smaller datasets and recommends PySpark for production use-cases. Below the message, there is a "Python (Pandas)" section with a code editor containing the following Python code:

```
1 # Table is available as variable `df`
2 df.info()
```

At the bottom of the dialog, there are "Preview" and "Add" buttons. The "Preview" button is highlighted with a red border. The "Add" button is located at the bottom right of the dialog window.

PassengerId (long)	Survived (long)	Pclass (long)	Name (string)	Age (float)	SibSp (long)	Parch (long)	Ticket (string)	Fare (long)	Embarked
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	Stoler, 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!

The screenshot shows the Data Wrangler interface with a dark theme. On the left, a preview of the 'titanic.csv' dataset is displayed in a table format. The columns include Age (float), SibSp (long), Parch (long), Ticket (string), Fare (long), Embarked (string), Sex_male (float), Sex_female (float), and New_Age (long). The 'Data' tab is selected. On the right, a 'CUSTOM PANDAS' section allows users to define custom transformations using Python (Pandas). A note states: "Use PySpark, Pandas, or PySpark (SQL) to define custom transformations. [Learn more.](#)" Below this is a warning message: "Using Python (Pandas) requires your dataset to fit in memory and only uses a single instance in batch computation. It is ideal for smaller datasets less than 2GB and experimentation but we recommend PySpark for production use-cases". The 'Python (Pandas)' code editor contains the following code:

```
1 # Table is available as variable `df`
2 df['New_Age'] = df['New_Age'].astype('int')
```

Buttons for 'Preview' and 'Add' are visible at the bottom right of the code editor.

	Age (float)	SibSp (long)	Parch (long)	Ticket (string)	Fare (long)	Embarked (string)	Sex_male (float)	Sex_female (float)	New_Age (long)
1	22	1	0	A/5 21171	7	S	1	0	22
2	38	1	0	PC 17599	71	C	0	1	38
3	26	0	0	STON/O2. 3101282	7	S	0	1	26
4	35	1	0	113803	53	S	0	1	35
5	35	0	0	373450	8	S	1	0	35
6		0	0	330877	8	Q	1	0	29
7	54	0	0	17463	51	S	1	0	54
8	2	3	1	349909	21	S	1	0	2
9	27	0	2	347742	11	S	0	1	27
10	14	1	0	237736	30	C	0	1	14
11	4	1	1	PP 9549	16	S	0	1	4
12	58	0	0	113783	26	S	0	1	58
13	20	0	0	A/5. 2151	8	S	1	0	20
14	39	1	5	347082	31	S	1	0	39
15	14	0	0	350406	7	S	0	1	14
16	55	0	0	248706	16	S	0	1	55
17	2	4	1	382652	29	Q	1	0	2
18		0	0	244373	13	S	1	0	29
19	31	1	0	345763	18	S	0	1	31
20		0	0	2649	7	C	0	1	29
21	35	0	0	239865	26	S	1	0	35
22	34	0	0	248698	13	S	1	0	34
23	15	0	0	330923	8	Q	0	1	15
24	28	0	0	113788	35	S	1	0	28
25	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!)

The screenshot shows the Data Wrangler interface with a dark theme. At the top left, it says "my_first_dataflow.flow". Below that is a breadcrumb trail: "Back to data flow". The main title is "Drop column - Transform: titanic.csv". There are two tabs: "Data" (which is selected) and "Analysis". Under "Data", there's a table titled "Previous step 8. Drop column" with columns: Survived (long), Pclass (long), SibSp (long), Parch (long), Fare (long), Sex_male (float), Sex_female (float), and New_Age (long). The table contains 12 rows of passenger data. To the right of the table is a sidebar titled "TRANSFORMS" with a list of steps: 1. S3 Source, 2. Data types, 3. One-hot encode, 4. Drop column (which is expanded to show "Move, drop, duplicate or rename columns in the dataset"), 5. Impute, 6. Custom Pandas, 7. Custom Pandas, and 8. Drop column. Below this is a "Transform" section with a dropdown menu set to "Drop column". Under "Columns to drop", there are four items: PassengerId, Age, Ticket, and Name, all with red X icons indicating they are selected for dropping. At the bottom of the sidebar are "Clear", "Preview", and "Update" buttons.

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

The screenshot shows the Data Wrangler interface with a flow named "my_first_dataflow.flow". The main area displays a preview of the "titanic.csv" dataset, specifically the "Process numeric" step. The preview table includes columns: Survived (long), Pclass (long), SibSp (long), Parch (long), Fare (float), Sex_male (float), Sex_female (float), New_Age (long), and Scaled_Age (float). The Scaled_Age column shows scaled values ranging from approximately 1.690 to 2.919. To the right of the preview is a detailed configuration panel for the "PROCESS NUMERIC" step. It includes sections for Transform (Scale values), Scaler (Standard scaler), Input column (Fare), and Output column (Fare). There are also optional checkboxes for Center and Scale, and a "Preview" button.

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

The screenshot shows the Google Data Studio Data Wrangler interface. The left side displays a preview of the dataset 'titanic.csv' with 42 rows and 8 columns. The columns are: Survived (long), Pclass (long), SibSp (long), Parch (long), Fare (float), Sex_male (float), Sex_female (float), and Scaled_Age (float). The right side shows the 'TRANSFORMS' panel, which is currently set to 'Drop column'. Under 'Transform', it says 'Drop column' and lists 'Columns to drop' as 'New_Age X'. The 'TRANSFORMS' panel also includes a sidebar with numbered steps from 1 to 11, where step 11 is '11. Drop column'.

Survived (long)	Pclass (long)	SibSp (long)	Parch (long)	Fare (float)	Sex_male (float)	Sex_female (float)	Scaled_Age (float)
0	3	1	0	0.14083450185392965	1	0	1.6905159661328477
1	1	1	0	1.4284642330898578	0	1	2.9199821233203735
1	3	0	0	0.14083450185392965	0	1	1.9978825054297291
1	1	1	0	1.066318371179753	0	1	2.6894572188477124
0	3	0	0	0.16095371640449102	1	0	2.6894572188477124
0	3	0	0	0.16095371640449102	1	0	2.22840740990239
0	1	0	0	1.0260799420786302	1	0	4.149448269507899
0	3	3	1	0.42250350556178895	1	0	0.1536832696484407
1	3	0	2	0.22131136005617516	0	1	2.0747241402539496
1	2	1	0	0.6035764365168413	0	1	1.075782887539085
1	3	1	1	0.32190743280898204	0	1	0.3073665392968814
1	1	0	0	0.5230995783145959	0	1	4.45681481980478
0	3	0	0	0.16095371640449102	1	0	1.536832696484407
0	3	1	5	0.6236956510674027	1	0	2.996823758144594
0	3	0	0	0.14083450185392965	0	1	1.075782887539085
1	2	0	0	0.32190743280898204	0	1	4.22628915332119
0	3	4	1	0.5834572219662799	1	0	0.1536832696484407
1	2	0	0	0.26154978915729793	1	0	2.22840740990239
0	3	1	0	0.3621458619101048	0	1	2.382090679550831
1	3	0	0	0.14083450185392965	0	1	2.22840740990239
0	2	0	0	0.5230995783145959	1	0	2.6894572188477124
1	2	0	0	0.26154978915729793	1	0	2.6126155840234917
1	3	0	0	0.16095371640449102	0	1	1.1526245223633054
1	1	0	0	0.7041725092696483	1	0	2.15156577507817
0	3	3	1	0.42250350556178895	0	1	0.6147330785937628
1	3	1	5	0.6236956510674027	0	1	2.9199821233203735
n	z	n	n	0.14083450185392965	1	n	2.22840740990239

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

my_first_dataflow.flow

Back to data flow

Drop column - Transform: titanic.csv

Data Analysis

Previous step 11. Drop column

Survived (long)	Pclass (long)	SibSp (long)	Parch (long)	Fare (float)	Sex_male (float)	Sex_female (float)	Scaled_Age (float)
0	3	1	0	0.14083450185392965	1	0	1.6905159661528477
1	1	1	0	1.4284642330898578	0	1	2.9199821233203735
1	3	0	0	0.14083450185392965	0	1	1.9978825054297291
1	1	1	0	1.066318371179753	0	1	2.6894572188477124
0	3	0	0	0.16095371640449102	1	0	2.6894572188477124
0	3	0	0	0.16095371640449102	1	0	2.22840740990239
0	1	0	0	1.0260799420786302	1	0	4.149448280507899
0	3	3	1	0.42250350556178895	1	0	0.1536832696484407
1	3	0	2	0.22131136005617516	0	1	2.0747241402539496
1	2	1	0	0.6035764365168413	0	1	1.075782887539085
1	3	1	1	0.32190745280898204	0	1	0.3073665392968814
1	1	0	0	0.5230995783145959	0	1	4.45681481980478
0	3	0	0	0.16095371640449102	1	0	1.536832696484407
0	3	1	5	0.6236956510674027	1	0	2.996823758144594
0	3	0	0	0.14083450185392965	0	1	1.075782887539085
1	2	0	0	0.32190745280898204	0	1	4.226289915352119
0	3	4	1	0.5834572219662799	1	0	0.1536832696484407
1	2	0	0	0.26154978915729793	1	0	2.22840740990239
0	3	1	0	0.3621458619101048	0	1	2.382090679550831
1	3	0	0	0.14083450185392965	0	1	2.22840740990239
0	2	0	0	0.5230995783145959	1	0	2.6894572188477124
1	2	0	0	0.26154978915729793	1	0	2.6126155840234917
1	3	0	0	0.16095371640449102	0	1	1.1526245223633054
1	1	0	0	0.7041725092696483	1	0	2.15156577507817
0	3	3	1	0.42250350556178895	0	1	0.6147330785937628
1	3	1	5	0.6236956510674027	0	1	2.9199821233203735
0	3	0	0	0.14083450185392965	1	0	2.22840740990239

Export data

TRANSFORMS

- + Add step
- 1. S3 Source
- 2. Data types
- 3. One-hot encode
- 4. Drop column
- 5. Impute
- 6. Custom Pandas
- 7. Custom Pandas
- 8. Drop column
- 9. Scale values
- 10. Scale values

11. Drop column

Move, drop, duplicate or rename columns in the dataset. [Learn more](#)

Transform

Drop column

Columns to drop

New_Age

Clear Preview Update

DATA WRANGLER DEMO

RENAME COLUMN

The screenshot shows the Data Wrangler interface with the following details:

- Left Panel:** Shows a preview of the "titanic.csv" dataset with 13 columns: Survived, Pclass, SibSp, Parch, Fare, Sex, and Scaled_Age.
- Right Panel:** A "MANAGE COLUMNS" dialog is open, titled "Move, drop, duplicate or rename columns in the dataset. [Learn more](#)".
 - Transform:** Set to "Rename column".
 - Input column:** Set to "Sex_female".
 - New name:** Set to "Sex".
- Bottom Right:** Buttons for "Preview" and "Add".

DATA WRANGLER DEMO

NOW YOU'RE GOOD TO GO!

The screenshot shows the Data Wrangler interface for a flow named "my_first_dataflow.flow". The main area displays a preview of the "titanic.csv" dataset with 893 rows and 7 columns. The columns are: Survived (long), Pclass (long), SibSp (long), Parch (long), Fare (float), Sex_female (float), and Scaled_Age (float). The preview shows various numerical values for each row.

To the right, a modal window titled "MANAGE COLUMNS" is open, specifically the "Drop column" section under the "Transform" tab. It lists "Sex_male" as a column to drop. There are "Preview" and "Add" buttons at the bottom of this modal.

DATA WRANGLER

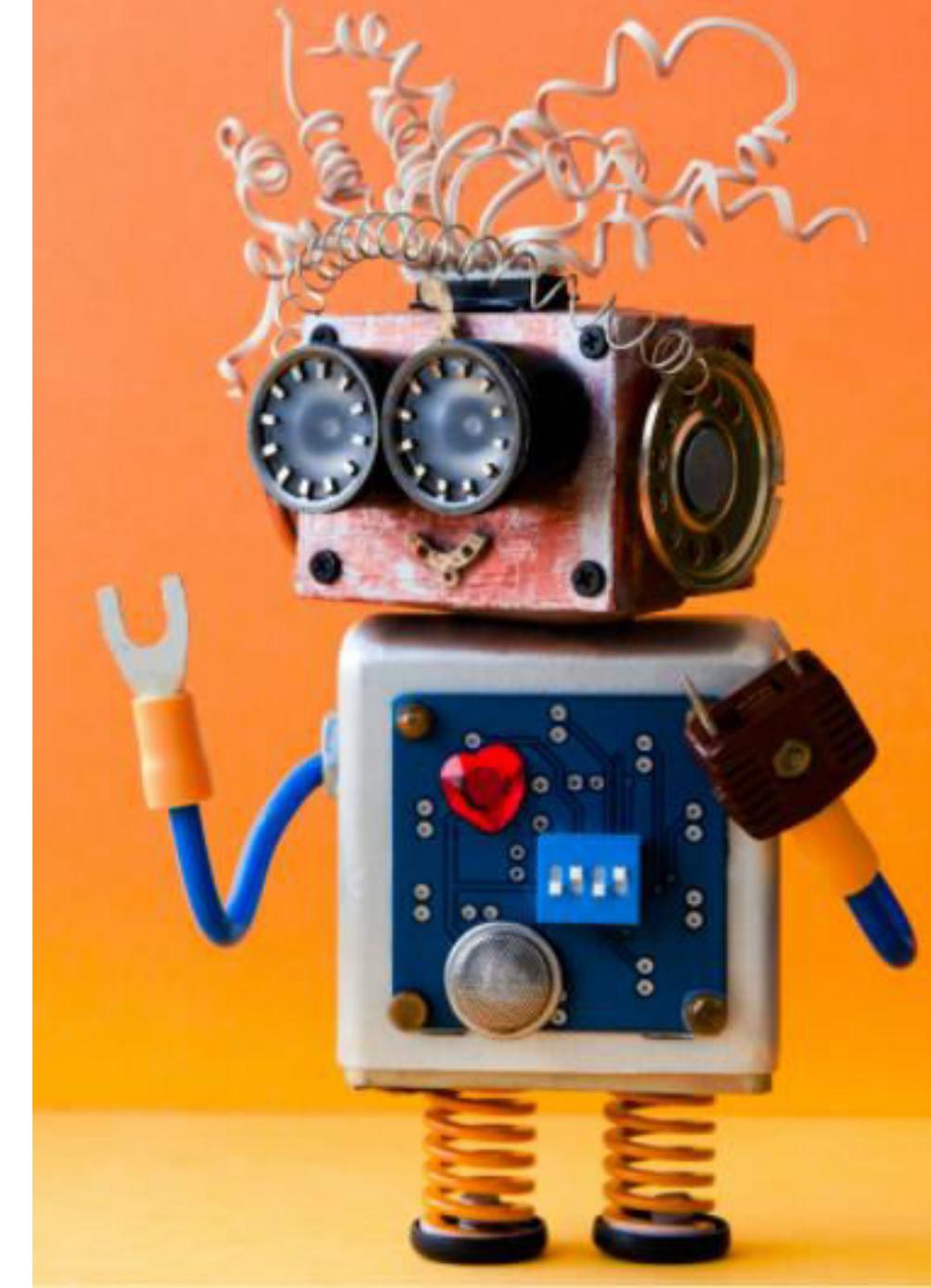
DEMO #7:

EXPORT WORKFLOW



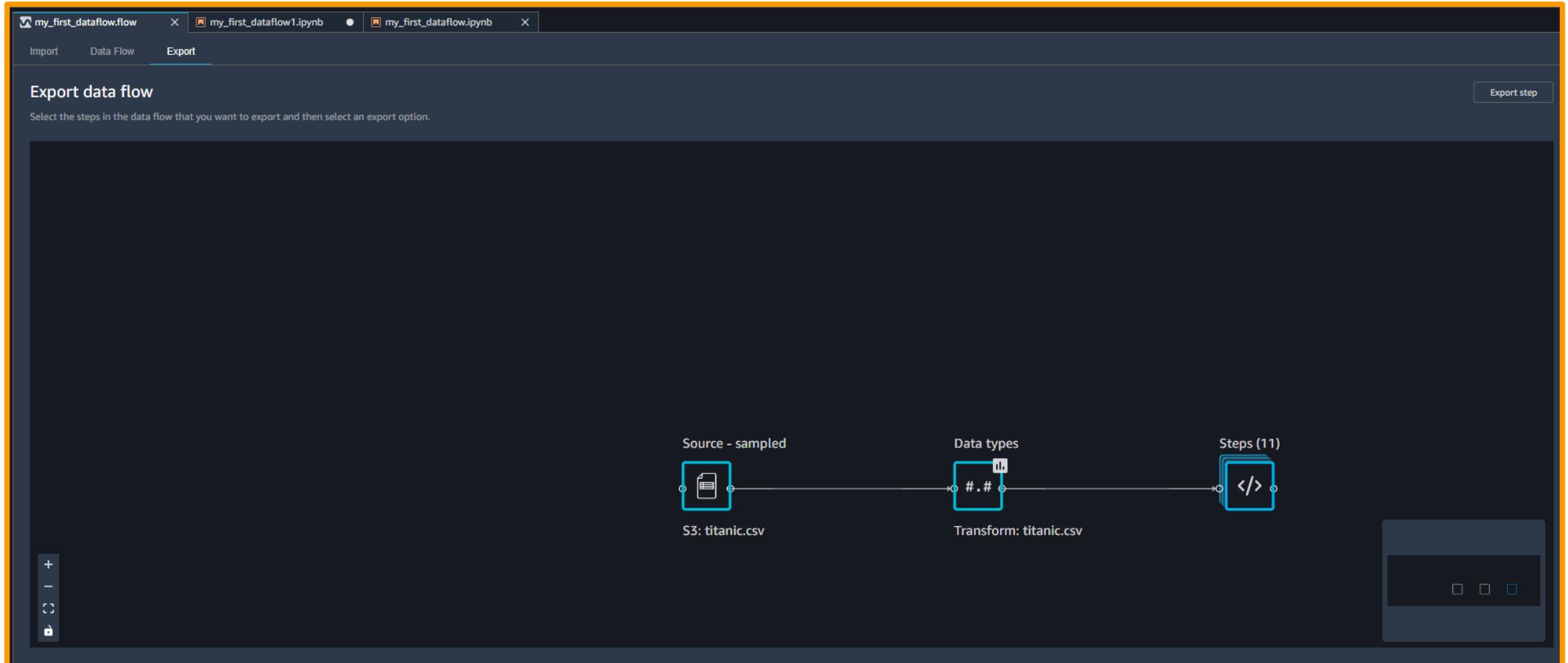
EASY

ADVANCED



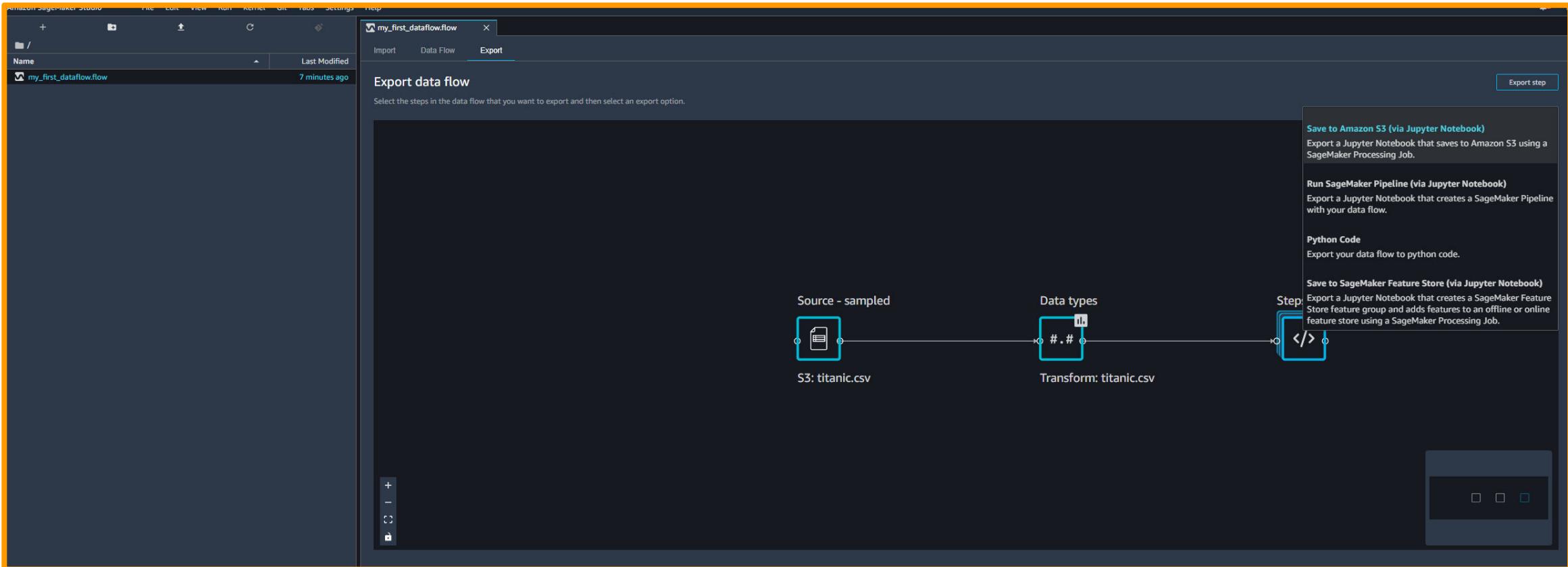
DATA WRANGLER DEMO

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 WORKFLOW



DATA WRANGLER DEMO

SELECT THE S3 OPTION AND RUN THE GENERATED NOTEBOOK

The screenshot shows a Jupyter Notebook interface with the title bar "my_first_dataflow.ipynb". The main content area has a dark background with light-colored text. It starts with a section titled "Save to S3 with a SageMaker Processing Job". Below this, a blue header bar contains a lightbulb icon and the text "Quick Start To save your processed data to S3, select the Run menu above and click Run all cells. [View the status of the export job and the output S3 location.](#)". The main text explains that the notebook executes a Data Wrangler Flow named "my_first_dataflow.flow" on the entire dataset using a SageMaker Processing Job to save the processed data to S3. It also notes that the notebook saves data from the step "Manage Columns" from "Source: Titanic.Csv". A "Contents" section lists several steps and optional next steps. The "Inputs and Outputs" section details how to configure inputs and outputs for the flow export, mentioning S3 sources and DatasetDefinition objects. A note at the bottom states that modified inputs must have the same schema and format as the data used in the Flow.

Save to S3 with a SageMaker Processing Job

Quick Start To save your processed data to S3, select the Run menu above and click Run all cells. [View the status of the export job and the output S3 location.](#)

This notebook executes your Data Wrangler Flow `my_first_dataflow.flow` on the entire dataset using a SageMaker Processing Job and will save the processed data to S3.

This notebook saves data from the step `Manage Columns` from `Source: Titanic.Csv`. To save from a different step, go to Data Wrangler to select a new step to export.

Contents

1. Inputs and Outputs
2. Run Processing Job
 - A. Job Configurations
 - B. Create Processing Job
 - C. Job Status & S3 Output Location
3. Optional Next Steps
 - A. Load Processed Data into Pandas
 - B. Train a model with SageMaker

Inputs and Outputs

The below settings configure the inputs and outputs for the flow export.

Configurable Settings

In Input - Source you can configure the data sources that will be used as input by Data Wrangler

1. For S3 sources, configure the source attribute that points to the input S3 prefixes
2. For all other sources, configure attributes like query_string, database in the source's `DatasetDefinition` object.

If you modify the inputs the provided data must have the same schema and format as the data used in the Flow. You should also re-execute the cells in this section if you have modified the settings in any data sources.

DATA WRANGLER DEMO

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

```
my_first_dataflow.ipynb  my_first_dataflow.ipynb  ●  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': 'File', 'S3DataDistributionType': 'FullyReplicated', 'S3CompressionType': 'None'}}, {'InputName': 'titanic.csv', 'AppManaged': False, 'S3Input': {'S3Uri': 's3://sagemaker-studio-694649038895/ylwtdvssaa/titanic.csv', 'LocalPath': '/opt/ml/processing/titanic.csv', 'S3DataType': 'S3Prefix', 'S3InputMode': 'File', 'S3DataDistributionType': 'FullyReplicated', 'S3CompressionType': 'None'}}]
Outputs: [{"OutputName": "67e10d8a-7319-4add-bf94-2fe248ea25d.default", "AppManaged": False, "S3Output": {"S3Uri": "s3://sagemaker-us-east-1-694649038895/export-flow-02-10-19-38-966039ef/output", "LocalPath": '/opt/ml/processing/output', 'S3UploadMode': '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

The screenshot shows the Amazon S3 console interface. The URL in the address bar is: `Amazon S3 > sagemaker-us-east-1-694649038895 > export-flow-02-10-19-38-966039ef/ > output/ > data-wrangler-flow-processing-02-10-19-38-966039ef/ > 67e10d8a-7319-4add-bf94-2fe248ea255d/ > default/`. The current view is the 'default/' folder. On the left, there are 'Objects' and 'Properties' tabs, with 'Objects' selected. The main area displays 'Objects (1)'. A message states: 'Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)'. Below this are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar says 'Find objects by prefix'. The table lists one object: 'part-00000-2a86eb8b-4281-4966-8157-69dfc79f949a-c000.csv'. The table columns are 'Name', 'Type', 'Size', and 'Storage class'. The file is a CSV type, 46.6 KB in size, and stored in the Standard storage class. The table has sorting arrows for 'Name', 'Type', 'Size', and 'Storage class'.

Name	Type	Size	Storage class
part-00000-2a86eb8b-4281-4966-8157-69dfc79f949a-c000.csv	csv	46.6 KB	Standard

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:

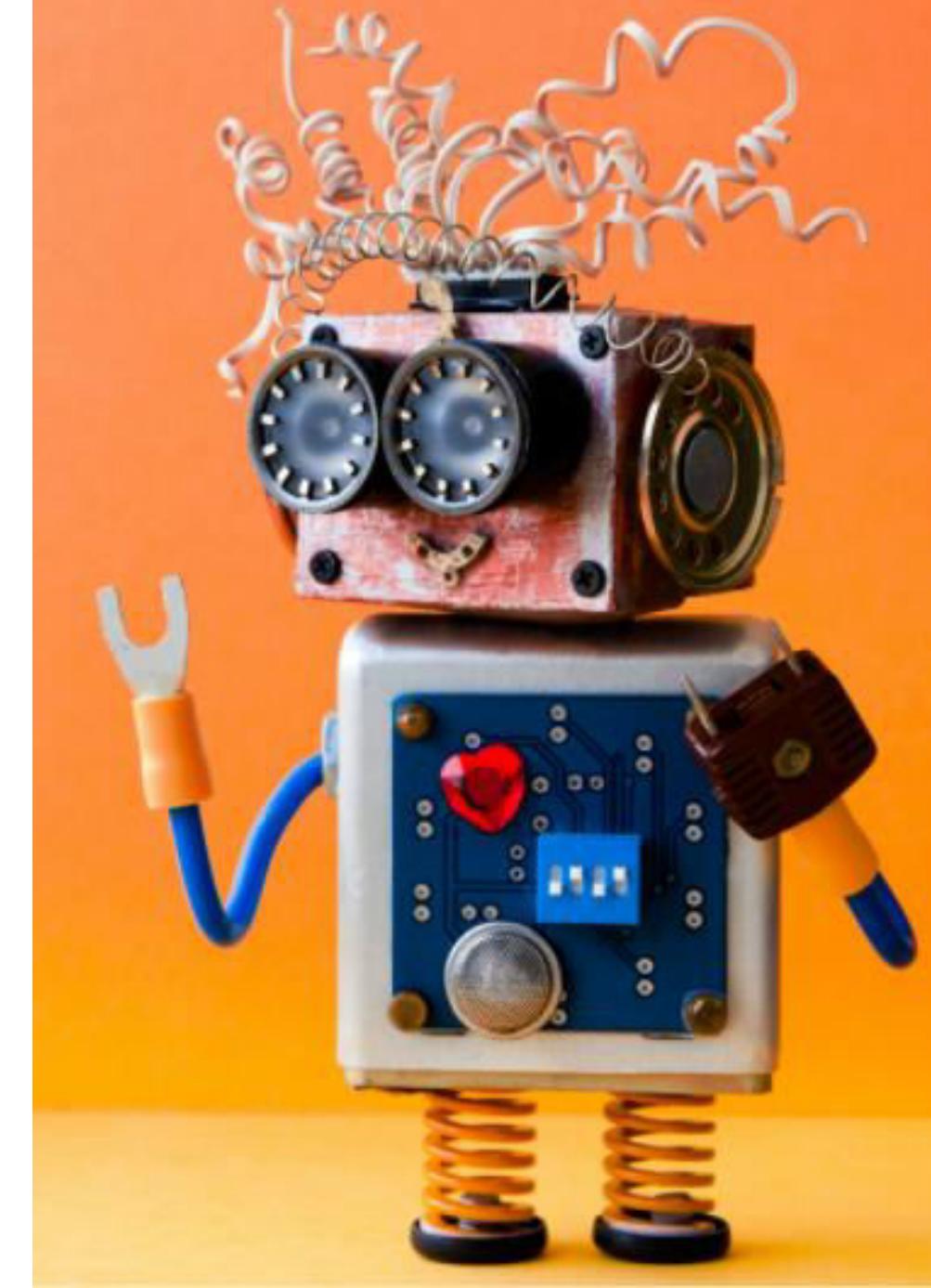
SHUTDOWN RESOURCES

[IMPORTANT]



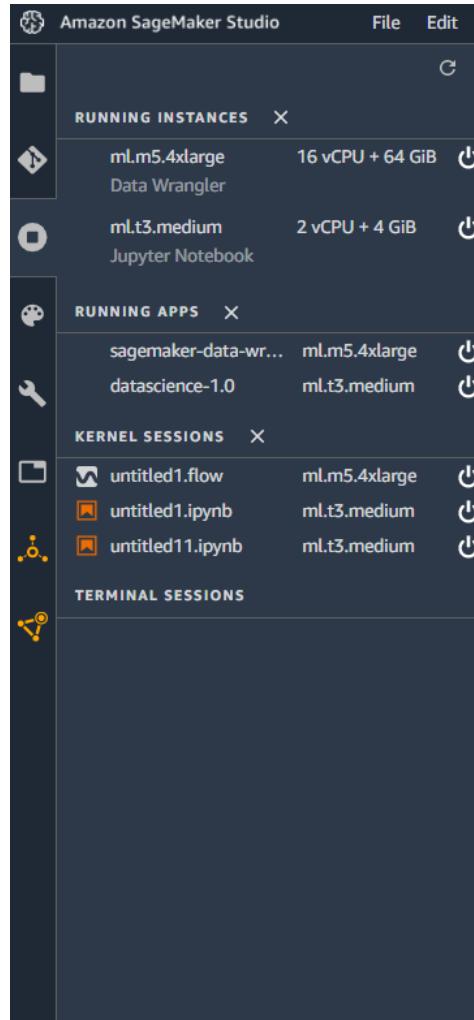
EASY

ADVANCED

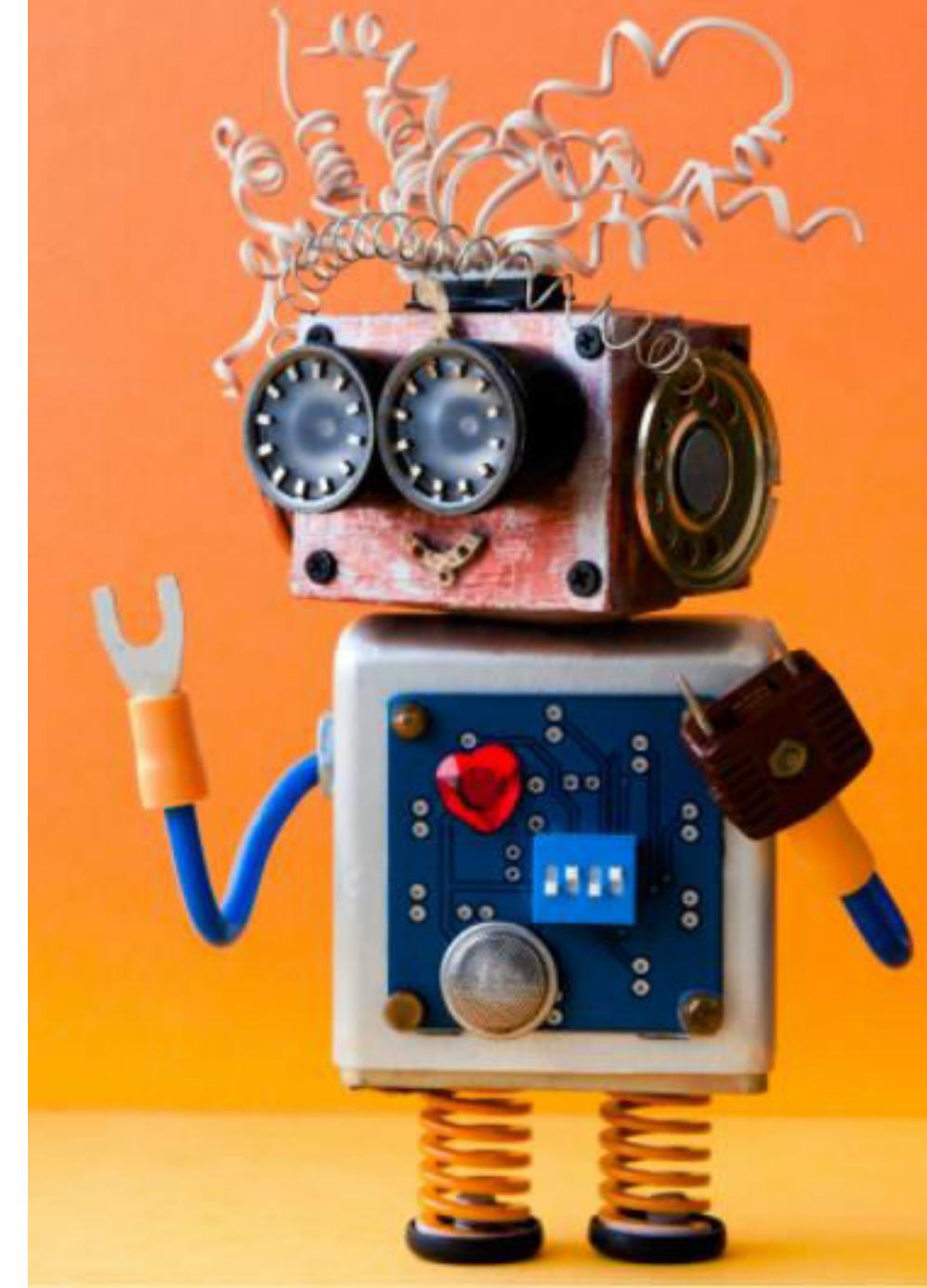


DATA WRANGLER DEMO [IMPORTANT]

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



FINAL END-OF-DAY CAPSTONE PROJECT

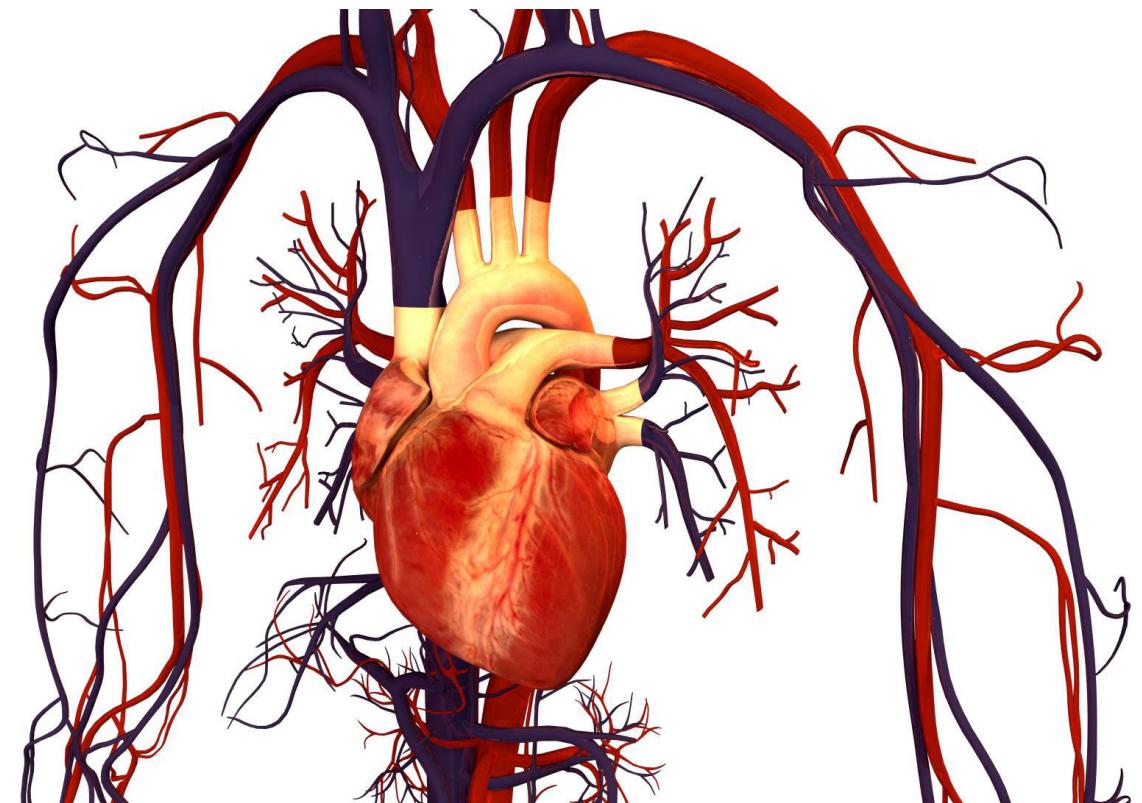


FINAL PROJECT OVERVIEW

- 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

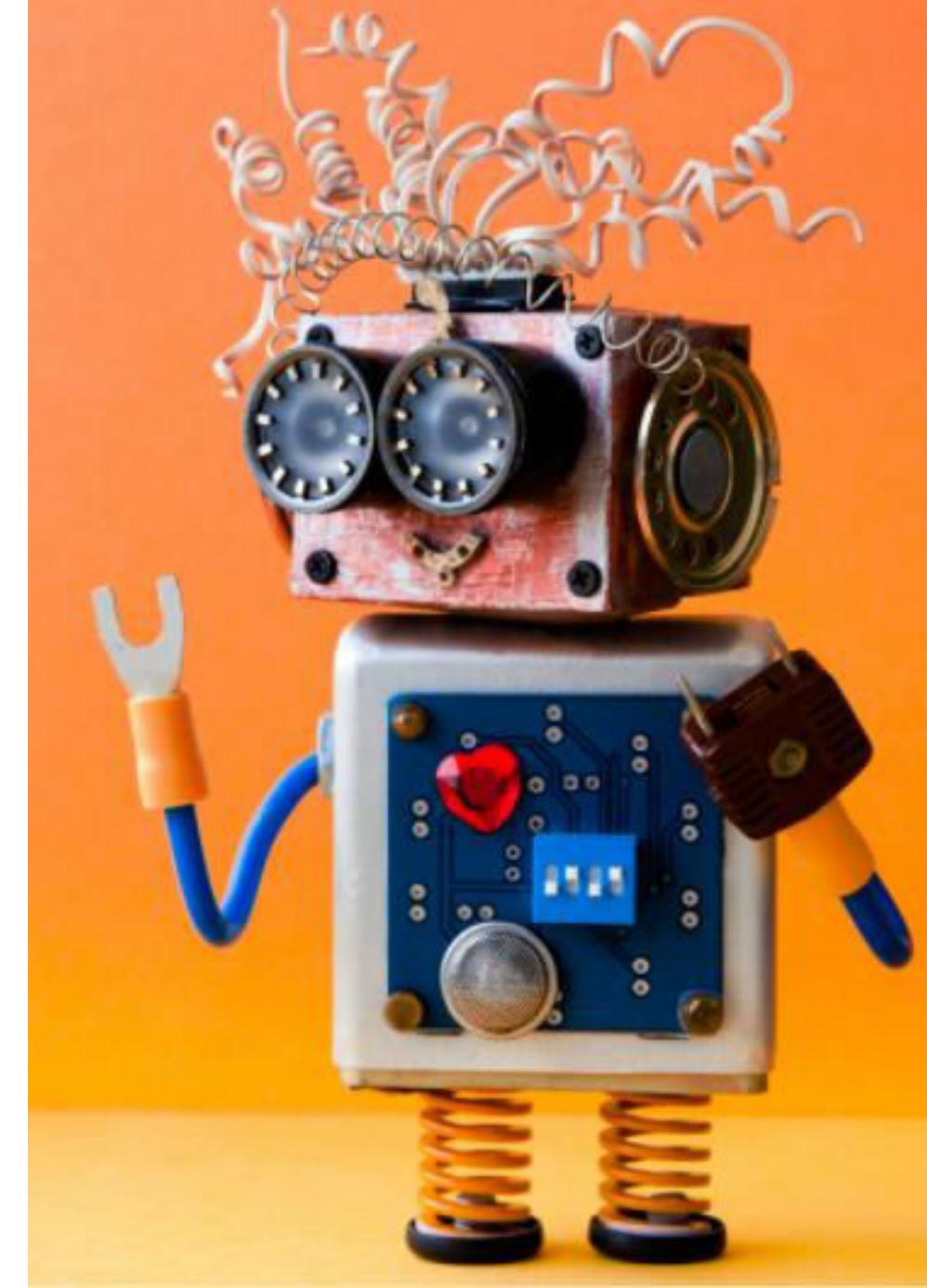
PROJECT SOLUTION



EASY



ADVANCED



PROJECT

UPLOAD THE DATA TO S3

The screenshot shows the AWS S3 console interface for uploading data. The top navigation bar includes the AWS logo, a 'Services' dropdown, a search bar ('Search for services, features, blogs, docs, and more'), and a keyboard shortcut ('[Alt+S]'). Below the navigation is a secondary header with 'S3' and 'Amazon SageMaker' tabs.

The left sidebar, titled 'Amazon S3', contains the following sections:

- Buckets**: Includes links for Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and Access analyzer for S3.
- Block Public Access settings for this account
- Storage Lens**: Includes links for Dashboards and AWS Organizations settings.
- Feature spotlight (3)
- AWS Marketplace for S3

The main content area shows the 'Upload' step in the process flow. The breadcrumb navigation indicates the path: Amazon S3 > Buckets > demo-datawrangler > Upload.

The 'Upload' section includes the following components:

- An instruction: "Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more" with a link icon.
- A central area with a dashed border for dragging and dropping files, containing the text: "Drag and drop files and folders you want to upload here, or choose Add files, or Add folders."
- A table titled "Files and folders (1 Total, 3.3 MB)" showing one item: "cardio.csv" (text/csv, 3.3 MB). It includes "Remove", "Add files", and "Add folder" buttons.
- A "Destination" section with the URL "s3://demo-datawrangler". It includes a "Destination details" expandable section describing bucket settings for new objects.
- Two additional expandable sections: "Permissions" (describing public access and other accounts) and "Properties" (describing storage class, encryption, and tags).
- At the bottom right are "Cancel" and "Upload" buttons.

PROJECT

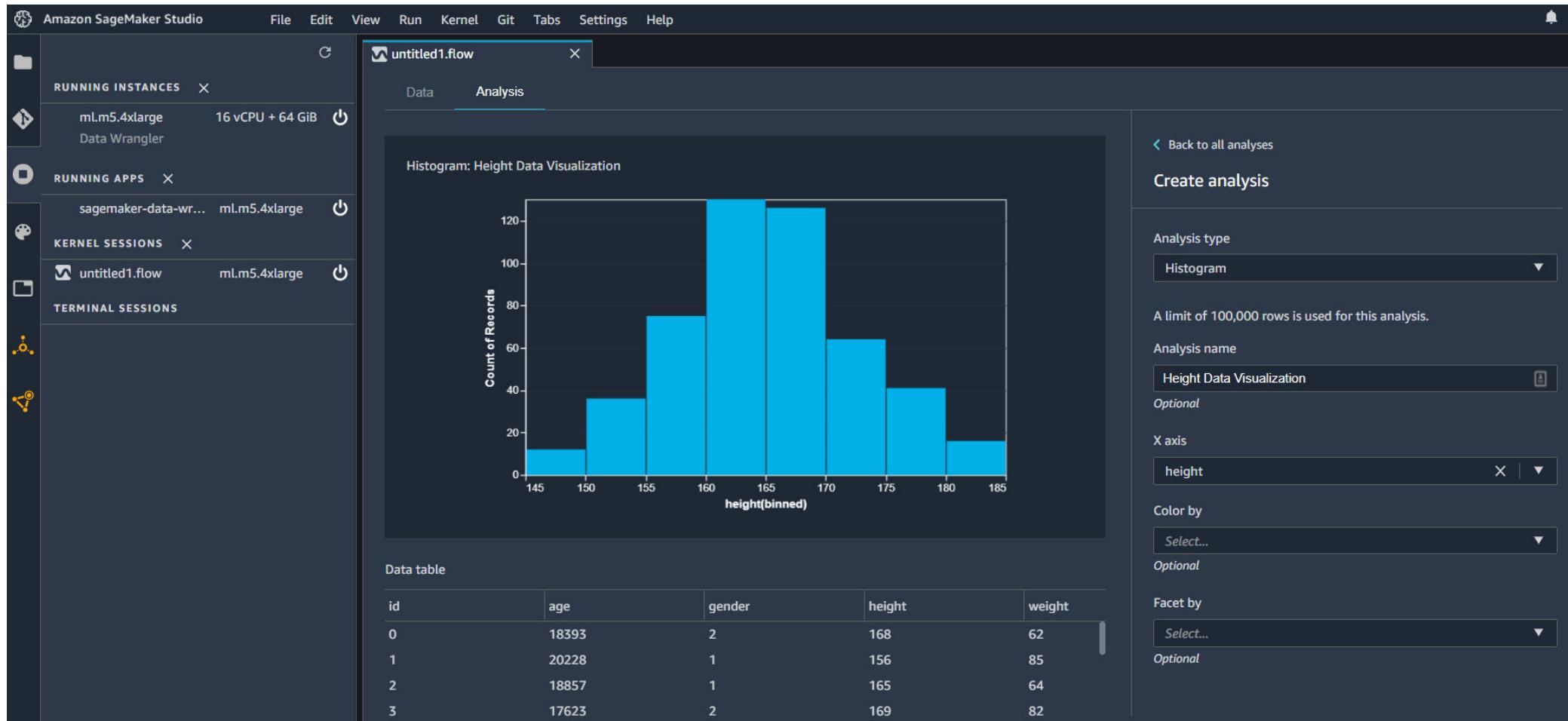
UPLOAD THE DATA TO S3

The screenshot shows the Amazon SageMaker Studio interface with the following details:

- Left Sidebar:** Displays "RUNNING INSTANCES" (ml.m5.4xlarge, 16 vCPU + 64 GB), "RUNNING APPS" (sagemaker-data-wrangler, ml.m5.4xlarge), "KERNEL SESSIONS" (untitled.flow, ml.m5.4xlarge), and "TERMINAL SESSIONS".
- Middle Panel:** A modal window titled "Import a dataset from S3" is open. It contains:
 - A "S3 URI path" input field with placeholder "Enter an S3 URI" and a "Go" button.
 - A preview table showing one object: "cardio.csv" (Size: 3.26MB).
 - A "PREVIEW • cardio.csv (first 100 rows shown)" section showing a sample of the CSV data with columns: id, age, gender, height.
- Right Panel:** A "DETAILS" panel on the right shows the imported file: "Name: cardio.csv", "File type: csv", "First row is header: checked", and "Import nested directories: unchecked". There is also an "Advanced configuration" link.

PROJECT

PLOT THE HEIGHT HISTOGRAM



PROJECT

PLOT SCATTERPLOT BETWEEN HEIGHT AND WEIGHT



PROJECT

PLOT CORRELATION BETWEEN FEATURES

Amazon SageMaker Studio

File Edit View Run Kernel Git Tabs Settings Help

RUNNING INSTANCES X
mLm5.4xlarge 16 vCPU + 64 GiB Data Wrangler

RUNNING APPS X
sagemaker-data-wr... mLm5.4xlarge

KERNEL SESSIONS X
untitled1.flow mLm5.4xlarge

TERMINAL SESSIONS

untitled1.flow X

Feature Correlation: Untitled

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 categorical 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. We display a correlation matrix for a dataset with up to 20 columns

	Most correlated feature	Correlation
gender (numeric)	height (numeric)	0.4999
height (numeric)	gender (numeric)	0.4999
cholesterol (numeric)	gluc (numeric)	0.452663
gluc (numeric)	cholesterol (numeric)	0.452663
smoke (numeric)	gender (numeric)	0.341212
alco (numeric)	smoke (numeric)	0.33561
weight (numeric)	height (numeric)	0.284715
age (numeric)	cardio (numeric)	0.236921
cardio (numeric)	age (numeric)	0.236921
ap_lo (numeric)	cardio (numeric)	0.0624712
ap_hi (numeric)	cardio (numeric)	0.0558411
active (numeric)	cardio (numeric)	-0.0380832
id (numeric)	smoke (numeric)	-0.00770905

Correlation matrix:

PROJECT

DROP THE ID COLUMN

The screenshot shows the Amazon SageMaker Studio interface. On the left, there's a sidebar with sections for RUNNING INSTANCES, RUNNING APPS, KERNEL SESSIONS, and TERMINAL SESSIONS. The main area displays a data wrangling session titled "untitled1.flow". The session details show it's running on an m1.m5.4xlarge instance with 16 vCPU + 64 GiB of memory. The current step is "Data types · Transform: cardio.csv". The "Data" tab is selected, showing a preview of the "cardio.csv" dataset. The dataset has 32 rows and 7 columns: id (long), age (long), gender (long), height (long), weight (long), ap_hi (long), and ap_lo (long). The "Analysis" tab is also visible. To the right of the data preview, a "MANAGE COLUMNS" panel is open, showing a "Drop column" section with "id" selected for removal. A "Columns to drop" section also lists "id". Buttons for "Preview" and "Add" are at the bottom of the panel.

	id (long)	age (long)	gender (long)	height (long)	weight (long)	ap_hi (long)	ap_lo (long)
0	18393	2	168	62	110	80	
1	20228	1	156	85	140	90	
2	18857	1	165	64	130	70	
3	17623	2	169	82	150	100	
4	17474	1	156	56	100	60	
8	21914	1	151	67	120	80	
9	22113	1	157	93	130	80	
12	22584	2	178	95	130	90	
13	17668	1	158	71	110	70	
14	19834	1	164	68	110	60	
15	22530	1	169	80	120	80	
16	18815	2	173	60	120	80	
18	14791	2	165	60	120	80	
21	19809	1	158	78	110	70	
23	14532	2	181	95	130	90	
24	16782	2	172	112	120	80	
25	21296	1	170	75	130	70	
27	16747	1	158	52	110	70	
28	17482	1	154	68	100	70	
29	21755	2	162	56	120	70	
30	19778	2	163	83	120	80	
31	21413	1	157	69	130	80	
32	23046	1	158	90	145	85	

PROJECT

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

The screenshot shows the Amazon SageMaker Studio interface. On the left, there's a sidebar with sections for RUNNING INSTANCES, RUNNING APPS, KERNEL SESSIONS, and TERMINAL SESSIONS. A central panel displays a data preview titled "Previewing: Custom transform" from a file named "cardio.csv". The data table has columns: age (float), gender (long), height (long), weight (long), ap_hi (long), ap_lo (long), and cholesterol (long). The first few rows of data are:

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

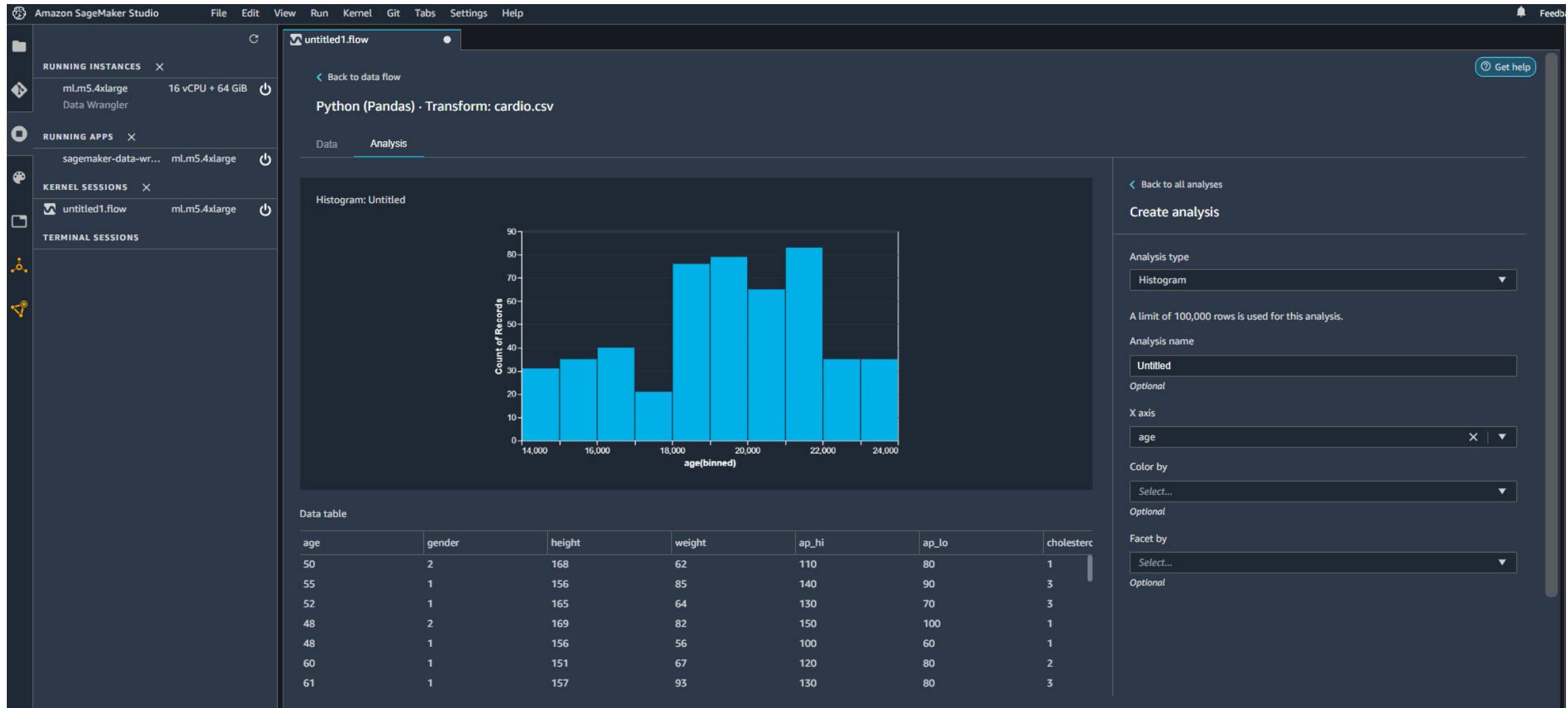
To the right of the data preview is a "CUSTOM TRANSFORM" dialog. It contains a note about using Python (Pandas) for custom transformations, a warning message, and a code editor with the following Python code:

```
1 # Table is available as variable `df`
2 df['age'] = df['age']/365
3 df['age'] = round(df['age'])
```

Buttons at the bottom of the dialog include "Clear", "Preview", and "Add".

PROJECT

PLOT THE HISTOGRAM FOR THE AGE COLUMN



PROJECT

GENERATE A SUMMARY TABLE.

The screenshot shows the Amazon SageMaker Studio interface. On the left, there's a sidebar with sections for RUNNING INSTANCES, RUNNING APPS, KERNEL SESSIONS, and TERMINAL SESSIONS. The main area displays a flow named 'untitled1.flow' for transforming 'cardio.csv'. The 'Analysis' tab is selected, showing a 'Table Summary: Untitled' table with statistics for columns: summary, age, gender, height, weight, ap_hi, and ap_lo. Below this is a 'Data table' section showing a subset of the cardio.csv data.

Table Summary: Untitled

summary	age	gender	height	weight	ap_hi	ap_lo
count	50000	50000	50000	50000	50000	50000
mean	53.32882	1.34702	164.36694	74.22956	128.74478	96.98146
stddev	6.767739168546063	0.47602694423008574	8.188911783743025	14.32572534758833	154.45595390137987	200.208980690146!
min	30.0	1	55	11	-150	0
max	65.0	2	250	200	16020	11000

Data table

age	gender	height	weight	ap_hi	ap_lo	cholesterol
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

PROJECT

SORT DATAFRAME IN AN ASCENDING ORDER USING AGE COLUMN

The screenshot shows the Amazon SageMaker Studio Data Wrangler interface. On the left, the sidebar displays 'RUNNING INSTANCES' (ml.M5.4xlarge, 16 vCPU + 64 GiB), 'RUNNING APPS' (sagemaker-data-wrangler, ml.m5.4xlarge), and 'KERNEL SESSIONS' (untitled1.flow, ml.m5.4xlarge). The main area shows a Python (Pandas) transform titled 'Transform: cardio.csv'. A preview of the data is shown in a table with columns: age (float), gender (long), height (long), weight (long), ap_hi (long), ap_lo (long), cholesterol (long). To the right, a 'MANAGE ROWS' panel is open, showing a 'Sort' operation is being applied to the 'age' column in ascending order. The data preview shows the first 20 rows of the 'cardio.csv' dataset.

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

The screenshot shows the Amazon SageMaker Studio interface with the following details:

- Left Sidebar:** Displays "RUNNING INSTANCES" (ml.m5.4xlarge, 16 vCPU + 64 GiB), "RUNNING APPS" (sagemaker-data-wrangler, ml.m5.4xlarge), and "TERMINAL SESSIONS" (untitled1.flow, ml.m5.4xlarge).
- Central Area:** A tab titled "Sort - Transform: cardio.csv" is open. It shows the "Analysis" tab selected under "Bias Report: Untitled".
 - Predicted column:** cardio
 - Predicted value or threshold:** 0
 - Column analyzed for bias:** cholesterol
 - Column value or threshold analyzed for bias:** 1

Three bias metrics are listed:

 - Class Imbalance (CI):** Score: -0.5. Detects if the advantaged group is represented in the dataset at a substantially higher rate than the disadvantaged group, or vice versa.
 - Difference in Positive Proportions in Labels (DPL):** Score: -0.24. Detects if one class has a significantly higher proportion of desirable (or, alternatively, undesirable) outcomes in the training data.
 - Jensen-Shannon Divergence (JS):** Score: 0.029. JS measures how much the label distributions of different classes diverge from each other. If the average label distribution across all of the classes is P, the JS divergence is the average of the KL divergences of the probability distributions for each class from the average distribution P. This entropic measure also generalizes to multiple label and continuous cases.

Data table: Shows columns ap_lo, cholesterol, gluc, smoke, alco, active, and cardio. The first row has values 0, 1, 2, 0, 0, 1, 0.
- Right Panel:** A "Create analysis" form is visible, set to "Bias Report" type, named "Untitled". It includes fields for selecting the target column (cardio), predicted value (0), and the column to analyze for bias (cholesterol). It also asks if the predicted column is a value or threshold.

PROJECT

SCALE AGE, HEIGHT, WEIGHT, AP_HI AND AP_LO

The screenshot shows the Amazon SageMaker Studio interface. On the left, there's a sidebar with sections for RUNNING INSTANCES, RUNNING APPS, KERNEL SESSIONS, and TERMINAL SESSIONS. In the main area, there are two tabs: 'untitled1.flow' and 'untitled1.ipynb'. The 'untitled1.flow' tab is active and displays a 'Sort · Transform: cardio.csv' process. The 'Data' tab is selected, showing a preview of the 'Process numeric' step. The preview table has columns: age (float), gender (long), height (float), weight (float), ap_hi (float), and ap_lo (float). The right side of the screen shows the configuration for the 'PROCESS NUMERIC' step, which is a 'Scale values' operation using a 'Min-max scaler'. It lists input columns (age, height, weight, ap_hi, ap_lo) and specifies min=0 and max=1.

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	0.620512805128205	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	0.528205128051282	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

The screenshot shows the Amazon SageMaker Studio interface. On the left, there's a sidebar with sections for **RUNNING INSTANCES**, **RUNNING APPS**, **KERNEL SESSIONS**, and **TERMINAL SESSIONS**. The main area has two tabs: **untitled1.flow** and **untitled1.ipynb**. The **untitled1.ipynb** tab is active.

Save to S3 with a SageMaker Processing Job

Quick Start: To save your processed data to S3, select the Run menu above and click **Run all cells**. [View the status of the export job and the output S3 location.](#)

This notebook executes your Data Wrangler Flow `untitled1.flow` on the entire dataset using a SageMaker Processing Job and will save the processed data to S3.

This notebook saves data from the step `Manage Rows` from Source: `Cardio.Csv`. To save from a different step, go to Data Wrangler to select a new step to export.

Contents

- 1. Inputs and Outputs
- 2. Run Processing Job
 - A. Job Configurations
 - B. Create Processing Job
 - C. Job Status & S3 Output Location
- 3. Optional Next Steps
 - A. Load Processed Data into Pandas
 - B. Train a model with SageMaker

Inputs and Outputs

The below settings configure the inputs and outputs for the flow export.

Configurable Settings

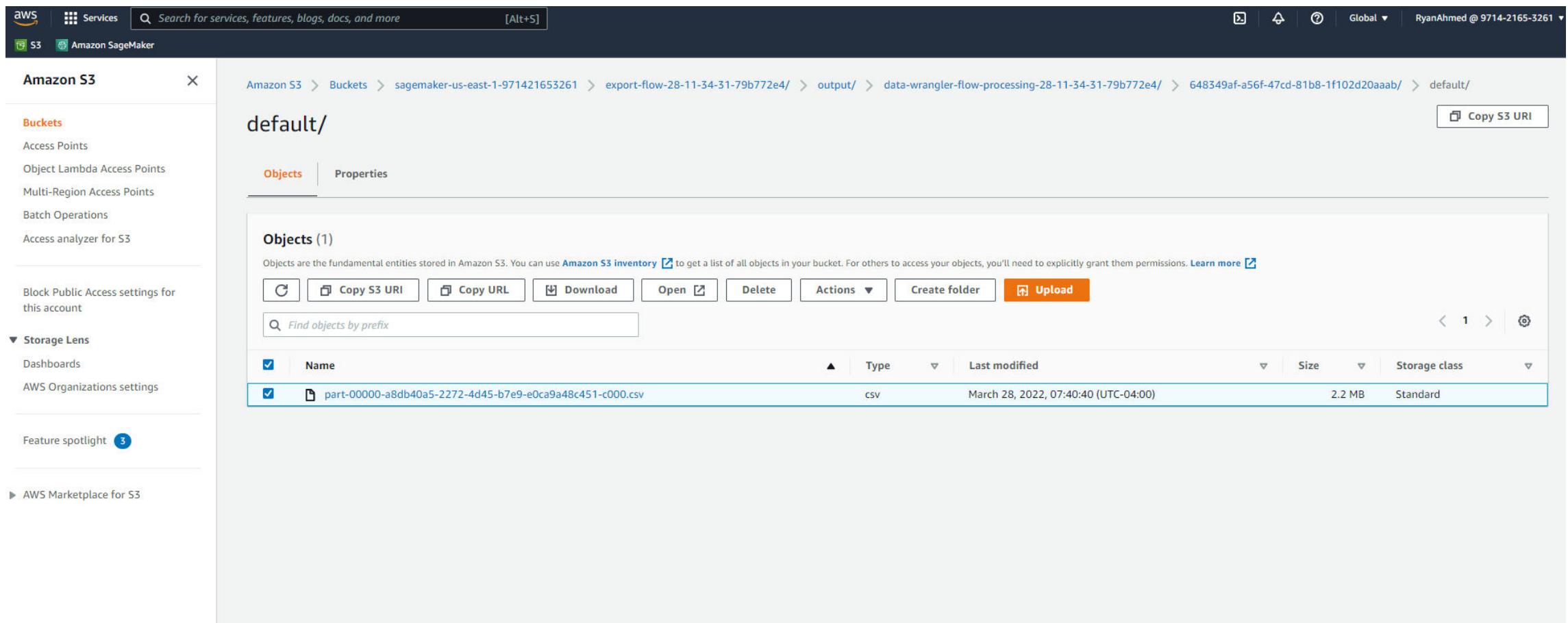
In `Input - Source` you can configure the data sources that will be used as input by Data Wrangler

1. For S3 sources, configure the `source` attribute that points to the input S3 prefixes
2. For all other sources, configure attributes like `query_string`, `database` in the source's `DatasetDefinition` object.

If you modify the inputs the provided data must have the same schema and format as the data used in the Flow. You should also re-execute the cells in this section if you have modified the settings in any data sources.

PROJECT

NAVIGATE TO THE PATH SHOWN BELOW AND DOWNLOAD THE FILE



The screenshot shows the AWS S3 console interface. The left sidebar includes links for Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, Access analyzer for S3, Block Public Access settings for this account, Storage Lens, Dashboards, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main content area displays a breadcrumb path: Amazon S3 > Buckets > sagemaker-us-east-1-971421653261 > export-flow-28-11-34-31-79b772e4/ > output/ > data-wrangler-flow-processing-28-11-34-31-79b772e4/ > 648349af-a56f-47cd-81b8-1f102d20aaab/ > default/. Below this, a folder named "default/" is shown. The "Objects" tab is selected in the navigation bar. A single object, "part-00000-a8db40a5-2272-4d45-b7e9-e0ca9a48c451-c000.csv", is listed in the table. The table columns are Name, Type, Last modified, Size, and Storage class. The file is a CSV type, last modified on March 28, 2022, at 07:40:40 (UTC-04:00), is 2.2 MB in size, and is stored in the Standard storage class.

Name	Type	Last modified	Size	Storage class
part-00000-a8db40a5-2272-4d45-b7e9-e0ca9a48c451-c000.csv	csv	March 28, 2022, 07:40:40 (UTC-04:00)	2.2 MB	Standard

DATA WRANGLER DEMO [IMPORTANT]

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

