



SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

English ▾

Option 1: Amazon SageMaker Data Wrangler and Feature Store

- [Overview](#)
- [Dataset upload to S3 bucket](#)
- [Amazon SageMaker Data Wrangler initial setup](#)
- [Add a S3 data file as Data Wrangler source](#)
- [Dataset analysis](#)
- [Data Transformation](#)
- [Data Export Options](#)
- [Model training with Amazon SageMaker Autopilot \(Optional\)](#)
- [Conclusion](#)

Overview

In this lab, you will learn how to use [Amazon SageMaker Data Wrangler](#) to prepare data for machine learning. You will also use [Amazon SageMaker Feature](#) to store, retrieve, and share machine learning (ML) features.

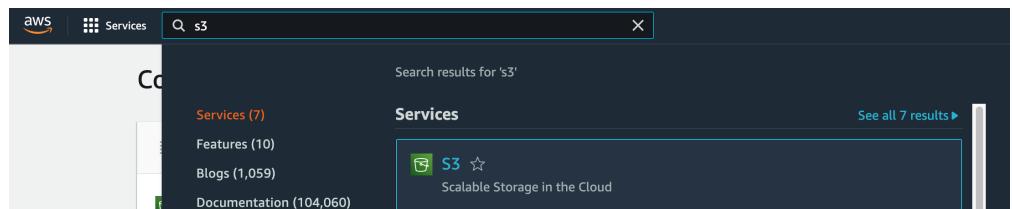
Dataset upload to S3 bucket

1. Type the following URL in your browser to download the dataset that we are going to use in the lab:

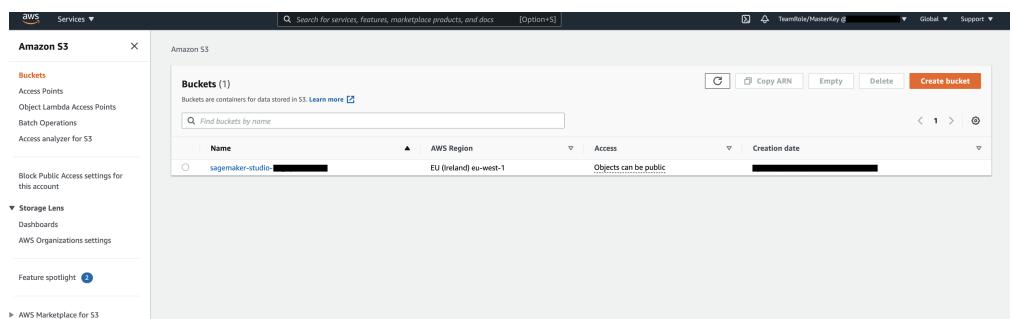


2. Unzip it on your computer.

3. Go to the AWS Management Console, search **S3** in the searchbox on top of your console, then go to **S3 service console**.



4. In the S3 console, click on the **sagemaker-studio-*** bucket.



SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

ⓘ The `sagemaker-studio-*` bucket was created automatically when you created the SageMaker Studio domain in the **Prerequisites** section. If you follow the **Event Engine** track, the bucket was preprovisioned by your instructor.

5. Click **Upload**.

The screenshot shows the Amazon S3 console. A breadcrumb navigation bar at the top left says "Amazon S3 > Buckets > sagemaker-studio-[REDACTED]". The main title "sagemaker-studio-[REDACTED] info" is centered above a navigation bar with tabs: Objects (highlighted in orange), Properties, Permissions, Metrics, Management, and Access Points. Below the navigation bar is a search bar with placeholder text "Find objects by prefix" and a "Upload" button. A table header for "Objects (0)" includes columns for Name, Type, Last modified, Size, and Storage class. A message below the table states "No objects" and "You don't have any objects in this bucket.".

6. On the Upload page, drag and drop the unzipped **bank-additional** folder into the drag and drop area.

The screenshot shows the "Upload" page of the Amazon S3 console. At the top, it says "Amazon S3 > sagemaker-studio-[REDACTED] > Upload". The main heading is "Upload". Below it is a large dashed box with the text "Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.". Underneath this is a section titled "Files and folders (0)". It contains a sub-instruction "All files and folders in this table will be uploaded." and a search bar with placeholder text "Find by name". A table header for "Files and folders (0)" includes columns for Name, Folder, Type, and Size. A message below the table says "No files or folders" and "You have not chosen any files or folders to upload.".

The 3 files inside the bank-additional folder will appear in the **Files and folders** area.

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Amazon S3 > sagemaker-studio-[REDACTED] > Upload

Upload

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](#)

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folder**.

Files and folders (3 Total, 5.4 MB)					
All files and folders in this table will be uploaded.					
	Name	Folder	Type	Size	
<input type="checkbox"/>	bank-additional-full.csv	bank-additional/	-	4.9 MB	
<input type="checkbox"/>	bank-additional-names.txt	bank-additional/	text/plain	5.3 KB	
<input type="checkbox"/>	bank-additional.csv	bank-additional/	-	503.2 KB	

7. Click **Upload** at the bottom of the page.

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
- Option 1: Amazon SageMaker Data Wrangler and Feature Store**
- Option 2: Numpy and Pandas
- Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Amazon S3 > sagemaker-studio-<REDACTED> > Upload

Upload

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](#)

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folder**.

Files and folders (3 Total, 5.4 MB)						Remove	Add files	Add folder
All files and folders in this table will be uploaded.								
	Name	Folder	Type	Size				
<input type="checkbox"/>	bank-additional-full.csv	bank-additional/	-	4.9 MB				
<input type="checkbox"/>	bank-additional-names.txt	bank-additional/	text/plain	5.3 KB				
<input type="checkbox"/>	bank-additional.csv	bank-additional/	-	503.2 KB				

Destination

Destination
[s3://sagemaker-studio-<REDACTED>](#)

Destination details

The following bucket settings impact new objects stored in the specified destination.

Bucket Versioning When enabled, multiple variants of an object can be stored in the bucket to easily recover from unintended user actions and application failures. Learn more	Default encryption When enabled, new objects stored in this bucket are automatically encrypted. Learn more	Object Lock When enabled, objects in this bucket might be prevented from being deleted or overwritten for a fixed amount of time or indefinitely. Learn more
⚠ Disabled	Disabled	Disabled

⚠ We recommend that you enable Bucket Versioning to help protect against unintentionally overwriting or deleting objects. [Learn more](#)

[Enable Bucket Versioning](#)

Additional upload options

[Cancel](#) [Upload](#)

Once the upload is finished, you will see a green band with a message **Upload succeeded**.

8. Click on **Close**.

Upload succeeded

View details below.

Upload: status

✔ The information below will no longer be available after you navigate away from this page.

Summary		
Destination s3://sagemaker-studio-<REDACTED>	Succeeded ✔ 3 files, 5.4 MB (100.00%)	Failed ✖ 0 files, 0 B (0%)
Files and folders	Configuration	

Files and folders (3 Total, 5.4 MB)

Files and folders (3 Total, 5.4 MB)					
	Name	Folder	Type	Size	Status
<input checked="" type="checkbox"/>	bank-additional-full.csv	bank-additional/	-	4.9 MB	✔ Succeeded
<input checked="" type="checkbox"/>	bank-additional-names.txt	bank-additional/	text/plain	5.3 KB	✔ Succeeded
<input checked="" type="checkbox"/>	bank-additional.csv	bank-additional/	-	503.2 KB	✔ Succeeded

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

You can verify that the folder is present in the S3 bucket.

The screenshot shows the AWS S3 console interface. The top navigation bar includes 'Amazon S3', 'Buckets', 'sagemaker-studio-[REDACTED]', and tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. Below the tabs, there's a toolbar with actions like 'Upload' and 'Actions'. A search bar says 'Find objects by prefix'. A table lists one object: 'Name' (bank-additional/), 'Type' (Folder), 'Last modified' (empty), 'Size' (empty), and 'Storage class' (empty).

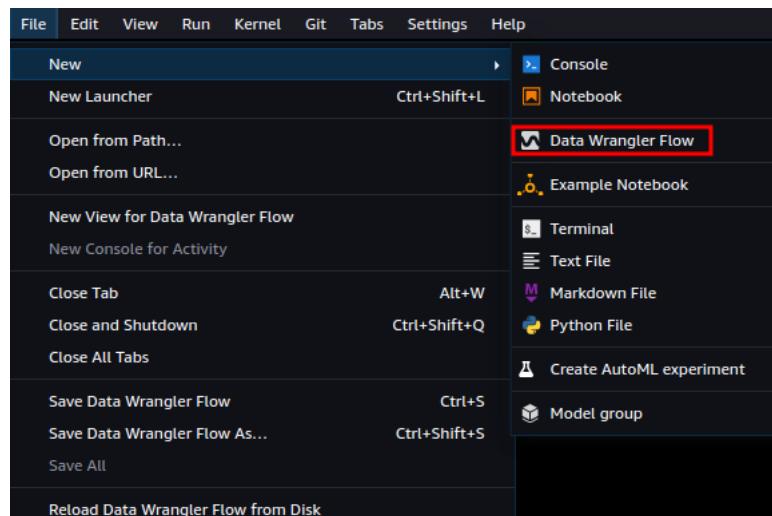
You will see the 3 files if you click on the **bank-additional** folder in the S3 bucket.

This screenshot shows the contents of the 'bank-additional/' folder within the 'sagemaker-studio-[REDACTED]' bucket. The 'Objects' table lists three items: 'bank-additional-full.csv' (Type: csv, Last modified: April 6, 2022, Size: 4.9 MB, Storage class: Standard), 'bank-additional.names.txt' (Type: txt, Last modified: April 6, 2022, Size: 5.3 KB, Storage class: Standard), and 'bank-additional.csv' (Type: csv, Last modified: April 6, 2022, Size: 503.2 KB, Storage class: Standard). A 'Copy S3 URI' button is visible at the top right.

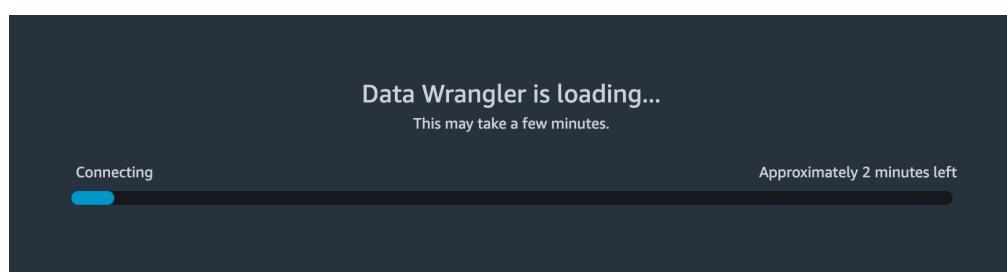
Congratulations!! You have successfully uploaded the dataset to Amazon S3.

Amazon SageMaker Data Wrangler initial setup

1. Go back to the SageMaker Studio Notebook that you created in the **Prerequisites** section.
2. From the top menu, select **File → New → Data Wrangler Flow**



An instance will be launched for the newly created Data Wrangler workflow, which may take a few minutes.



After a while, the loading message disappears, data source screen is shown:

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

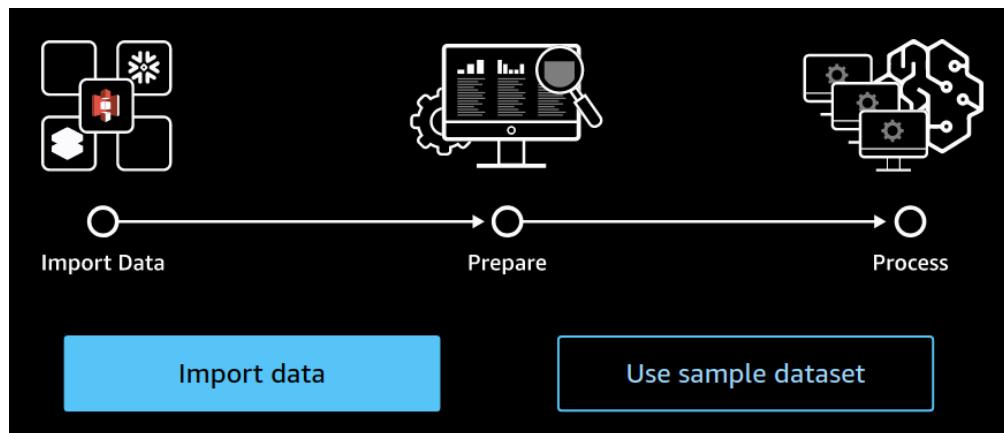
► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

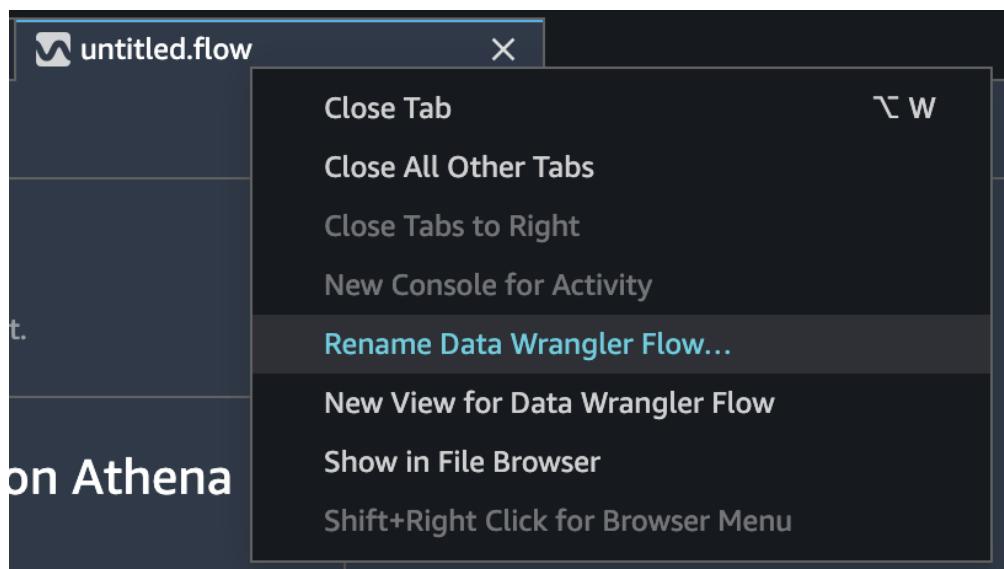
Language



Add a S3 data file as Data Wrangler source

ⓘ To use Data Wrangler, you need access to a m5.4xlarge Amazon Elastic Compute Cloud (Amazon EC2) instance in your account. If the AWS account you use has been provided by an instructor, you will have access to this instance type.

Right click on the file tab to rename it to **ImmersionDay.flow**.



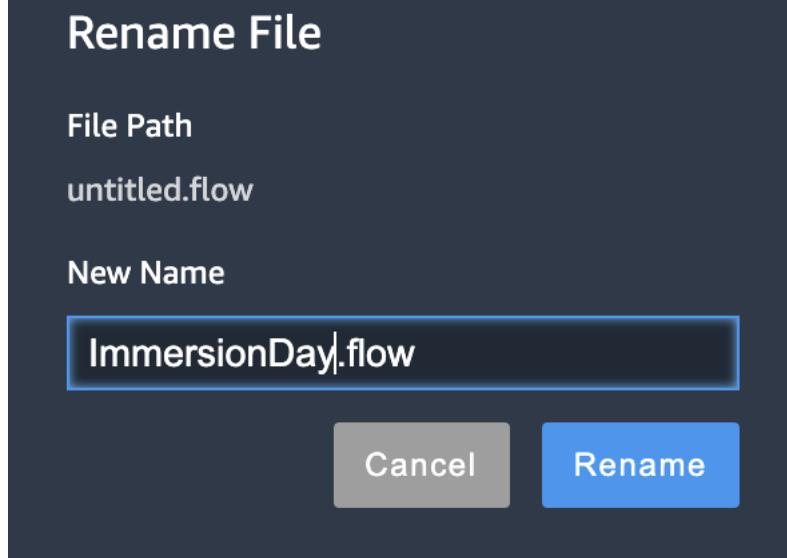
Then click the Rename button.

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

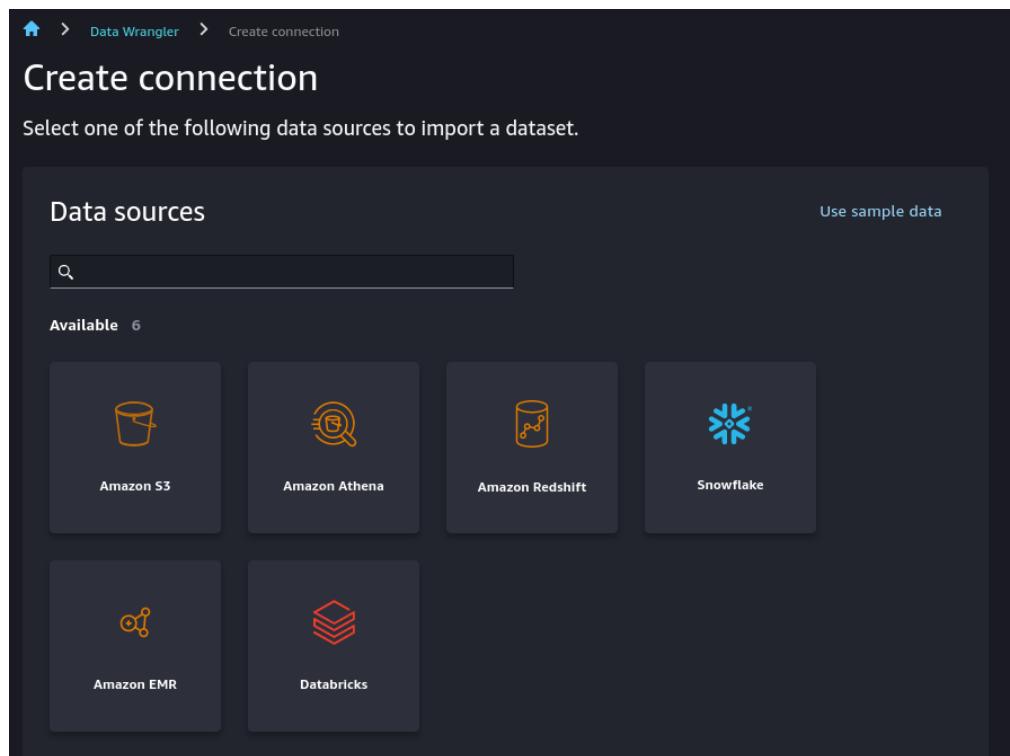
▼ Content preferences

Language



Click **Import data**, then click **Amazon S3**.

Double-click on the **sagemaker-*** bucket.



Double-click on the **bank-additional** folder to open it.

Click on the **bank-additional-full.csv** file.

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Import a dataset from S3

Enter the S3 URL of a file or prefix (folder) in the text box, or use the following table to browse S3

Advanced configuration

S3 URI path Go

S3 / sagemaker-us-east-1-327883354475 / bank-additional

Object name	Size	Last modified
bank-additional-full.csv	4.66MB	2023-02-16 18:11:14+00:00
bank-additional-names.txt	5.33KB	2023-02-16 18:11:15+00:00
bank-additional.csv	503.17KB	2023-02-16 18:11:15+00:00

A preview of the data is displayed:

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Import data

Import a dataset from S3

Enter the S3 URL of a file or prefix (folder) in the text box, or use the following table to browse S3

Advanced configuration

S3 URI path

Enter an S3 URI

S3 / sagemaker-us-east-1-327883354475 / bank-additional / bank-additional-full.csv

Object name	Size	Last modified
bank-additional-full.csv	4.66MB	2023-02-16 18:11:14+00:00

Go

Displaying 1 - 1

PREVIEW • bank-additional-full.csv (First 100 rows shown. The preview doesn't reflect your sampling configuration.)

age	job	marital	education	default
56	housemaid	married	basic.4y	no
57	services	married	high.school	unknown
37	services	married	high.school	no
40	admin.	married	basic.6y	no
56	services	married	high.school	no
45	services	married	basic.9y	unknown
59	admin.	married	professional.course	no
41	blue-collar	married	unknown	unknown
24	technician	single	professional.course	no
25	services	single	high.school	no
41	blue-collar	married	unknown	unknown
25	services	single	high.school	no
29	blue-collar	single	high.school	no
57	housemaid	divorced	basic.4y	no
35	blue-collar	married	basic.6y	no
54	retired	married	basic.9y	unknown
35	blue-collar	married	basic.6y	no
46	blue-collar	married	basic.6y	unknown
50	blue-collar	married	basic.6y	no

For the **Sampling** option select **None**. This is a relatively small dataset, no need to sample.

Click on **Import dataset**.

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy

XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

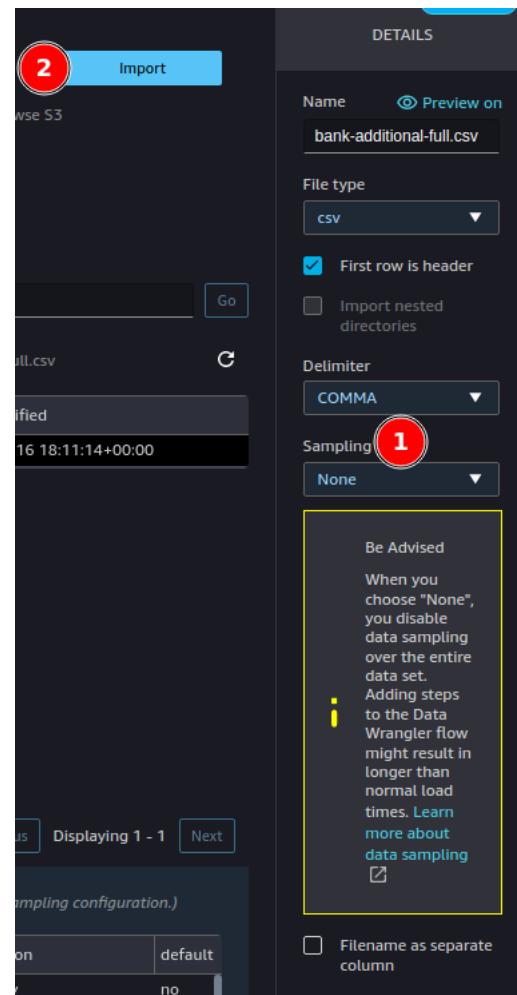
► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language



The **Data types** screen will appear. Click the arrow at the top next to **Data Flow** to see the new flow file.

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

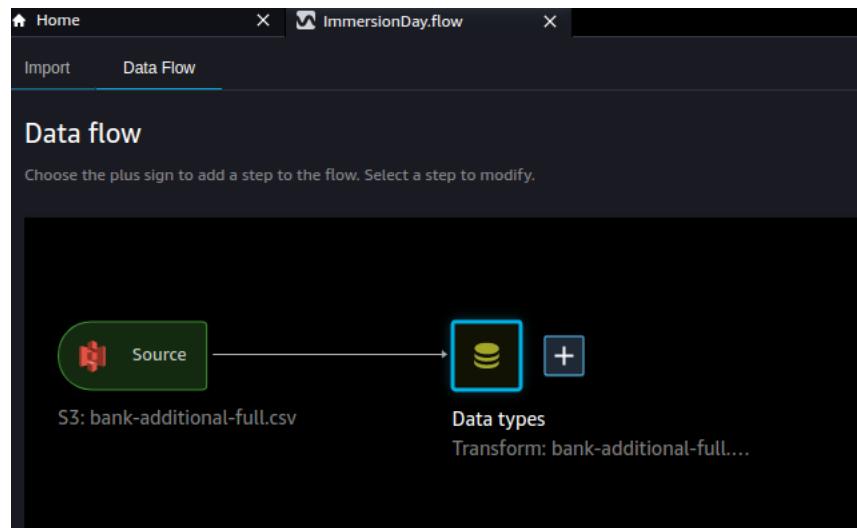
► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

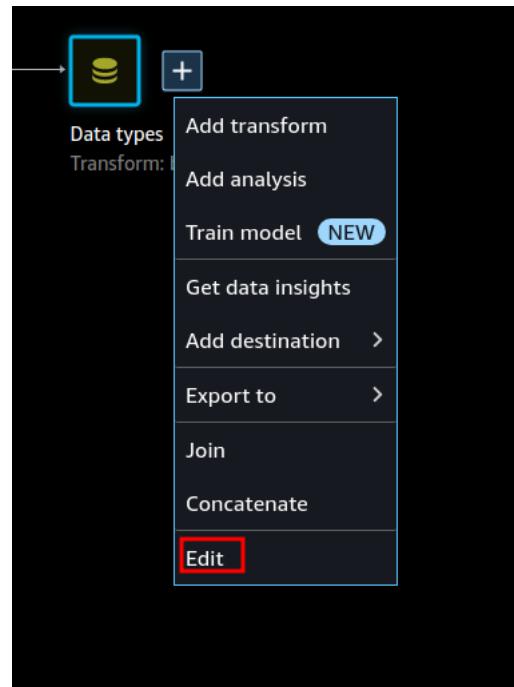
Language



Dataset analysis

In this section, we are going to analyze the dataset.

To get an overview of the data types, click on the + sign next to the **Data Types** node, and select **Edit**:



You get an overview of the dataset with the column names and types:

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy

XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

We don't want to change any type, so we click on < Data Flow:

The specifics on each feature are the following: **Demographics:**

- **age:** Customer's age (numeric)

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

- **job:** Type of job (categorical: 'admin.', 'services', ...)
- **marital:** Marital status (categorical: 'married', 'single', ...)
- **education:** Level of education (categorical: 'basic.4y', 'high.school', ...)

Past customer events:

- **default:** Has credit in default? (categorical: 'no', 'unknown', ...)
- **housing:** Has housing loan? (categorical: 'no', 'yes', ...)
- **loan:** Has personal loan? (categorical: 'no', 'yes', ...)

Past direct marketing contacts:

- **contact:** Contact communication type (categorical: 'cellular', 'telephone', ...)
- **month:** Last contact month of year (categorical: 'may', 'nov', ...)
- **day_of_week:** Last contact day of the week (categorical: 'mon', 'fri', ...)
- **duration:** Last contact duration, in seconds (numeric). Important note: If duration = 0 then y = 'no'.

Campaign information:

- **campaign:** Number of contacts performed during this campaign and for this client (numeric, includes last contact)
- **pdays:** Number of days that passed by after the client was last contacted from a previous campaign (numeric)
- **previous:** Number of contacts performed before this campaign and for this client (numeric)
- **poutcome:** Outcome of the previous marketing campaign (categorical: 'nonexistent','success', ...)

External environment factors:

- **emp.var.rate:** Employment variation rate - quarterly indicator (numeric)
- **cons.price.idx:** Consumer price index - monthly indicator (numeric)
- **cons.conf.idx:** Consumer confidence index - monthly indicator (numeric)
- **euribor3m:** Euribor 3 month rate - daily indicator (numeric)
- **nr.employed:** Number of employees - quarterly indicator (numeric)

Target variable:

- **y:** Has the client subscribed a term deposit? (binary: 'yes','no')

Data Quality and Insights Report

Data Wrangler's built-in **Data Quality and insights Report** provides an automated Exploratory Data Analysis (EDA) report on your imported dataset. Included in this report is feature summary, a quick model review, and much more information that would help in data processing and cleaning.

Click on the + sign again and then on **Add analysis**:

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

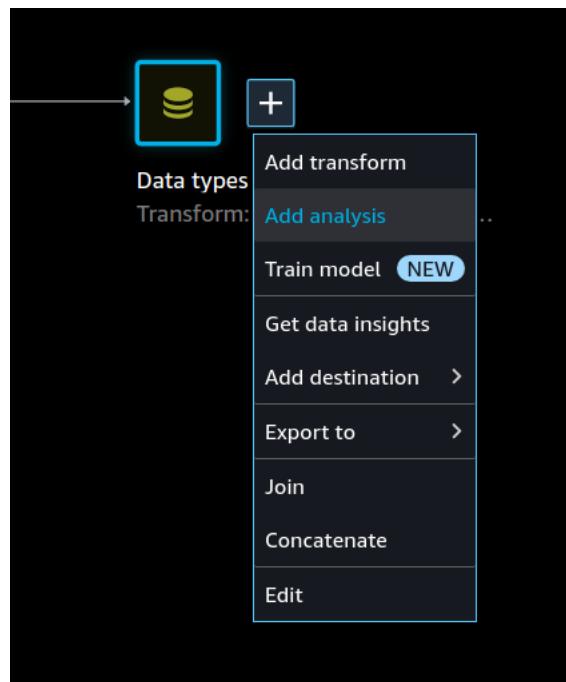
► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language



You are presented with the **Analysis** screen.

1. Under **Analysis type**, select **Data Quality and Insights Report**, and for **Target column**, select **y**.
2. Select **Classification** for **Problem type**.
3. Click **Create**.

A screenshot of the 'Analysis' screen in the Amazon SageMaker Data Wrangler. On the left, there's a 'Data table' preview showing a sample of 41188 rows across 21 columns with various categorical and numerical values. On the right, there are configuration panels: 'Analysis type' set to 'Data Quality And Insights Report', 'Target column' set to 'y' (optional), and 'Problem type' set to 'Classification'. There are also buttons for 'Create' and 'Clear'.

After some time, the **Data Quality and Insights Report** content will populate. Scroll down the generated report to see all populated content. Lets review some of the topics in these report.

• Data Statistics

- Our dataset contains 41188 rows and 21 features.
- No missing rows
- A few duplicate rows. We can drop duplicate rows using the **Drop duplicates** built-in transform.
- Breakdown of the type of features

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Dataset statistics

Key	Value
Number of features	21
Number of rows	41188
Missing	0%
Valid	100%
Duplicate rows	0.0583%

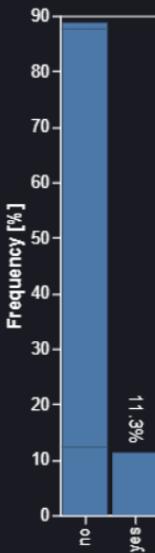
Feature type	Count
numeric	10
categorical	9
text	0
datetime	0
binary	1
unknown	0

• Target Column

- This report clearly shows the imbalance in our dataset target label.
- no class compromises of ~88% of the entire data.
- This imbalanced data will pose problems during model training. One way to remediate imbalanced data is to use the built-in **Balance data** transform.

TARGET COLUMN

key	value
Number of classes	2
Valid	100%
Missing	0%



Histogram of the frequent values of the target column.

• Quick model

- Our dataset achieves a validation accuracy of 91.7% and a decent balanced accuracy score of 75.4%
- Minority class, yes, performs poorly on all metrics including Recall, Precision and F1-Score.
- Quick model metrics will serve as a benchmark for any training job down the line to compare model performance.

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

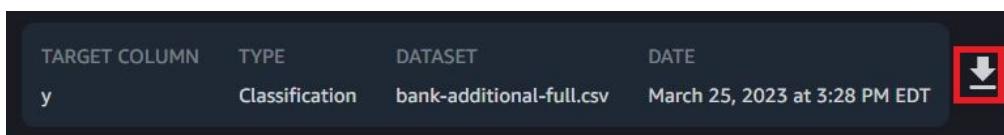
▼ Content preferences

Language

Metric	Validation scores	Train scores
Accuracy	0.917	0.932
Balanced accuracy	0.754	0.792
ROC-AUC	0.95	0.965
F1	0.596	0.669
Precision	0.661	0.739
Recall	0.543	0.612

class	precision	recall	f1-score	support
no	0.9432775919732441	0.9645690834473324	0.953804531619885	7310.0
yes	0.6605504587155964	0.5431034482758621	0.5960969840331165	928.0

To download this report, click on the download icon on the upper-right corner of the report page.

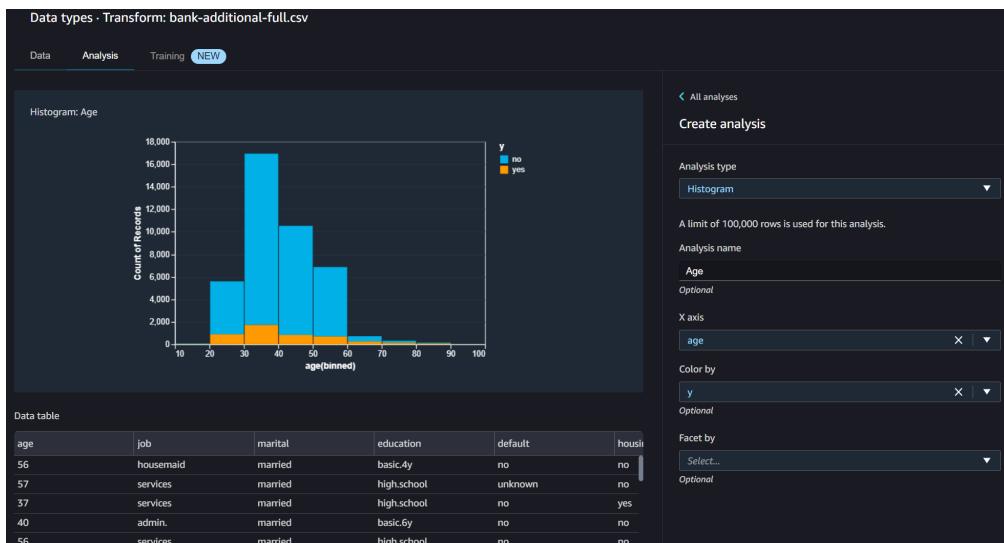


Now that we've viewed a good general overview of the data, let's create a second analysis to analyze the correlation between the target variable, and other variables.

Histogram

1. Create a new Analysis as you did before.
2. Under the **Analysis type** drop-down, select **Histogram**
3. Enter **Age** for **Analysis name**
4. Choose **age** as **X-axis** and **Color by** **y** as shown below and click on **Preview**.

You will see the following histogram and will have a visual representation of the potential influence of the **age** feature of a person on the target variable **y**, which represents if the person will accept a marketing offer:



Once you click on **Save**, you will be able to access your analysis from the **Transform** screen **Analysis** tab:

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

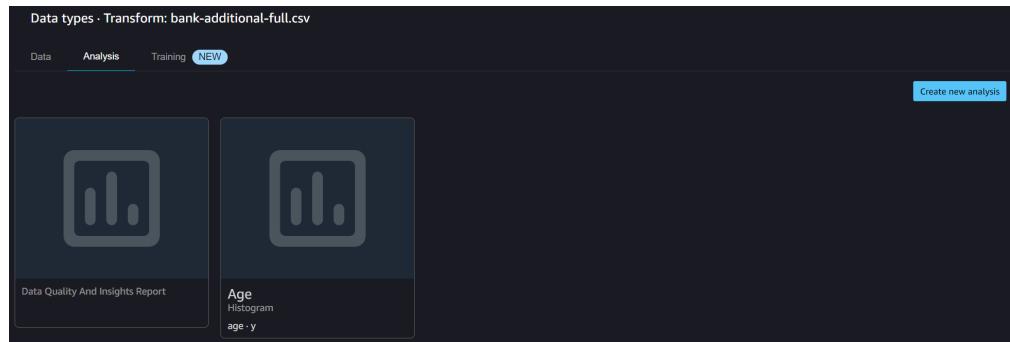
► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language



You can repeat the operation for job, marital status and even other features to visualize the influence on the target variable.

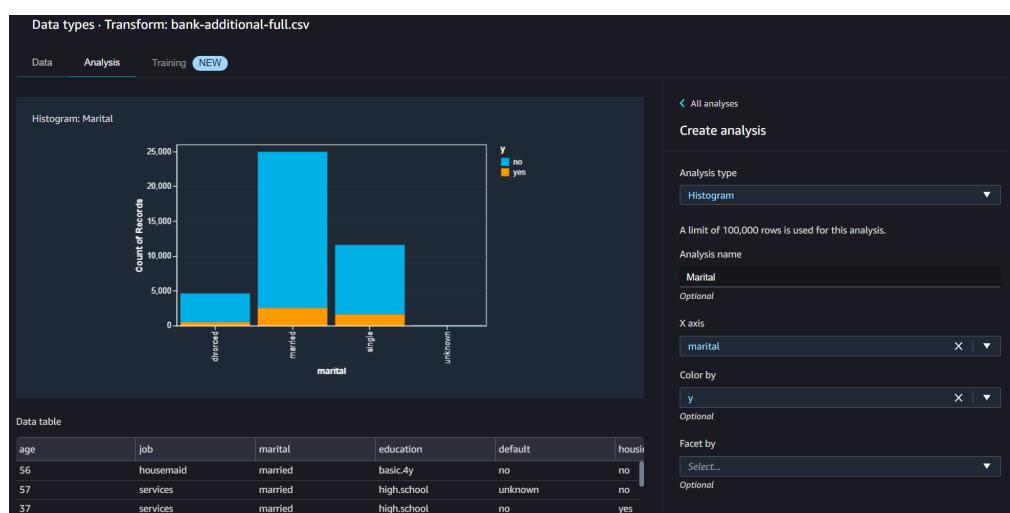
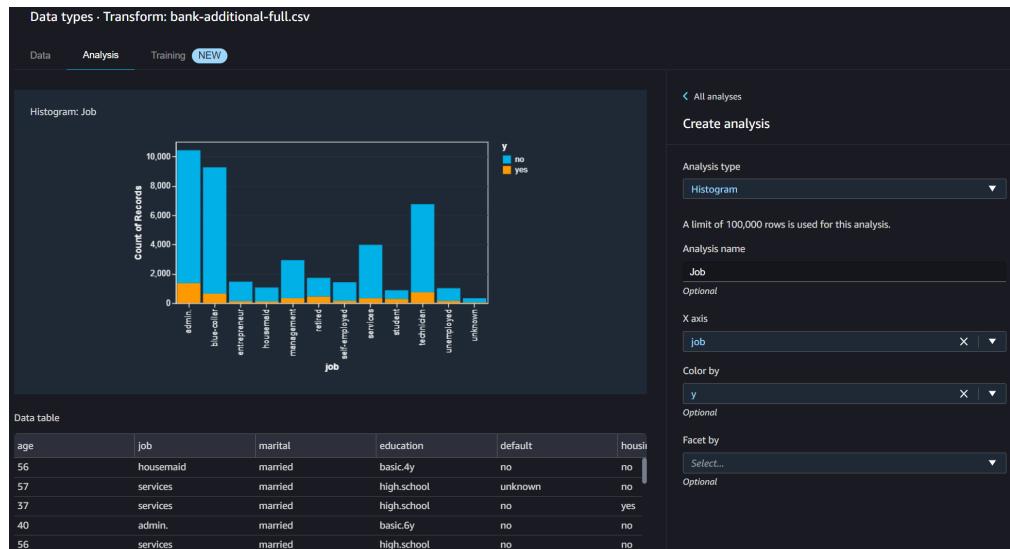


Table Summary

For a more general table statistics overview, we can run a **Table Summary** analysis.

1. Create another Analysis.

2. Under **Analysis type**, select **Table Summary**, and give the analysis a name (e.g. **Summary**).

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

All analyses

Create analysis

Analysis type

Table Summary

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

Analysis name

Summary

Optional

3. Click on **Preview**:

Table Summary: Summary				
summary	age	job	marital	education
count	41188	41188	41188	41188
mean	40.02406040594348	None	None	None
stddev	10.421249980934045	None	None	None
min	17	admin.	divorced	basic.4y
max	98	unknown	unknown	unknown

A table summary appears. Click on **Save** to save your analysis.

In addition to the dataset feature and row count, you have a view on different statistics of the dataset (e.g. mean, standard deviation). Once saved, you can access your analysis from the **Transform** screen **Analysis** tab.

Bias Report

Next, lets gain some insight into any potential bias at the different `marital` groups by creating a **Bias Report**.

1. Create a new Analysis.
2. For Analysis Type, select Bias Report.
3. For name, enter **Bias**.
4. For target, select **y**.
5. For predicted column value or threshold, select **Value**.
6. For predicted value(s) enter **yes**.
7. For Column to analyze for bias, enter `marital`.
8. For Column value(s) to analyze for bias, leave empty. This will provide bias for all the values in the `marital` facet.
9. For Bias metrics, check all **EXCEPT CDDL**.
10. Select **No** for analyzing additional metrics.
11. Click **Check For Bias**.

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy

XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Bias Report: Bias
The computed bias metrics are below:
Predicted column: y
Predicted value or threshold: yes
Column analyzed for bias: marital Column value or threshold analyzed for bias: married

Class Imbalance (CI)
Detects if the advantaged group is represented in the dataset at a substantially higher rate than the disadvantaged group, or vice versa.
Value: -0.21
Range: -1 to 1

Difference in Positive Proportions in Labels (DPL)
Detects if one class has a significantly higher proportion of desirable (or, alternatively, undesirable) outcomes in the training data.
Value: 0.028
Range: -1 to 1

Jensen-Shannon Divergence (JS)
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 metric also generalizes to multiple label and continuous cases.
Value: 0.00097
Range: 0 to =

Data table

age	job	marital	education	default	housing	loan
56	housemaid	married	basic.4y	no	no	no
57	services	married	high.school	unknown	no	no
57	services	married	high.school	no	yes	no
40	admin.	married	basic.4y	no	no	no
56	services	married	high.school	no	no	yes
45	services	married	basic.4y	unknown	no	no
59	admin.	married	professional.course	no	no	no
41	blue-collar	married	unknown	unknown	no	no
24	technician	single	professional.course	no	yes	no
25	services	single	high.school	no	yes	no
41	blue-collar	married	unknown	unknown	no	no
25	services	single	high.school	no	yes	no
29	blue-collar	single	high.school	no	no	yes

Edit 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): y

Is your predicted column a value or threshold? Value

Predicted value(s): yes

Select the column to analyze for bias: marital

Is your column a value or threshold? Value

Choose bias metrics:

- Class imbalance (CI) ⓘ
- Difference in Positive Proportions in Labels (DPL) ⓘ
- JS divergence (JS) ⓘ
- Conditional Demographic Disparity in Labels (CDOL) ⓘ

Would you like to analyze additional metrics? Yes

From the results, we observe a class imbalance where the `married` class is 21% more represented than other classes. We also observe that the `married` class is 2.8% less likely to subscribe to a bank term deposit. For more information on all the bias metrics supported by Data Wrangler, [learn how Amazon SageMaker Clarify helps detect bias](#).

12. Click Save.

Target Leakage

For our final analysis, let's create a **Target Leakage** analysis. Target leakage occurs when there is data in a machine learning training dataset that is strongly correlated with the target label, but is not available in real-world data.

1. Create a new Analysis.

2. For Analysis Type, select **Target Leakage**.

3. For name, enter *Target Leakage*.

4. For Problem Type, select **Classification**.

5. For Target, select `y`.

6. Click Preview.

Target Leakage: Target Leakage

The provided predictive metric is roc, computed individually for each column via cross validation, on a sample of 2036 rows. A score of 1 indicates perfect predictive abilities, which often indicates an error called target leakage. The cause is typically a column that will not be available at prediction time such as a duplicate of the target column. A score of 0.5 indicates that the information on the column could not tell us anything useful towards predicting the target. Although it can happen that a column is uninformative on its own but is useful in predicting the target when used in tandem with other features, a low score could indicate the feature is redundant.

Data table

age	job	marital	education	default	housing	loan
56	housemaid	married	basic.4y	no	no	no
57	services	married	high.school	unknown	no	no
57	services	married	high.school	no	yes	no

Edit analysis

Analysis type: Target Leakage

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

Analysis name: Target Leakage

Max features: 20

Problem Type: classification

Target: y

From the result, we see that the `cons.conf.idx`, `cons.price.idx`, `nr.employed`, `loan` and `month` features have low predictive abilities and are possibly redundant features. We would drop some of these features later on.

7. Click Save.

SageMaker Immersion Day

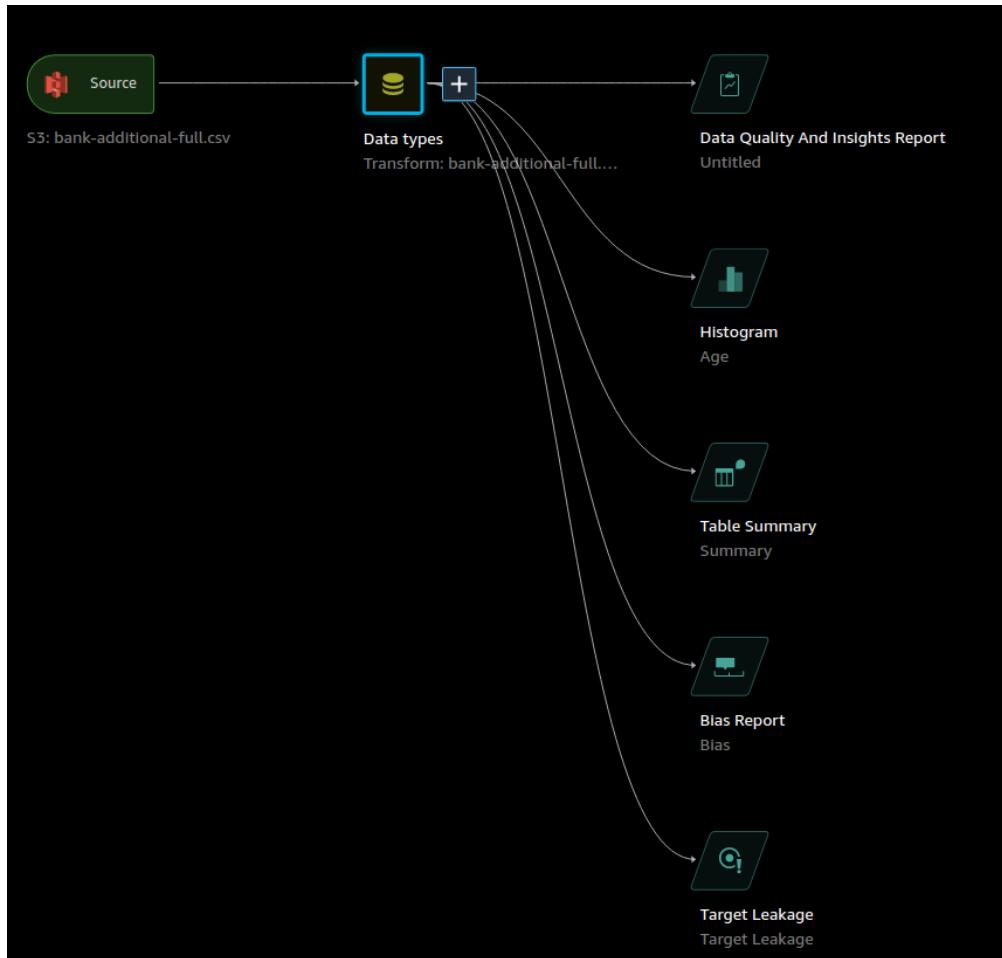
- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

 Data Wrangler samples your data to perform Target Leakage analysis, therefore, you may see discrepancies in the charts for each successive analysis.

At this point, your flow should look similar to the image below:



We've performed an import, and analyzed our data. Now we can take some actions based on our analyses and domain knowledge to transform the data to make it ready for training.

Data Transformation

Cleaning up data is part of nearly every machine learning project. It arguably presents the biggest risk if done incorrectly and is one of the more subjective aspects in the process.

Handle Class Imbalance

From the Data Quality and Insights Report above, we discovered an imbalanced representation of the target class y . Let's make use of the [Synthetic Minority Over-sampling Technique \(SMOTE\)](#) to balance our data target class representation.

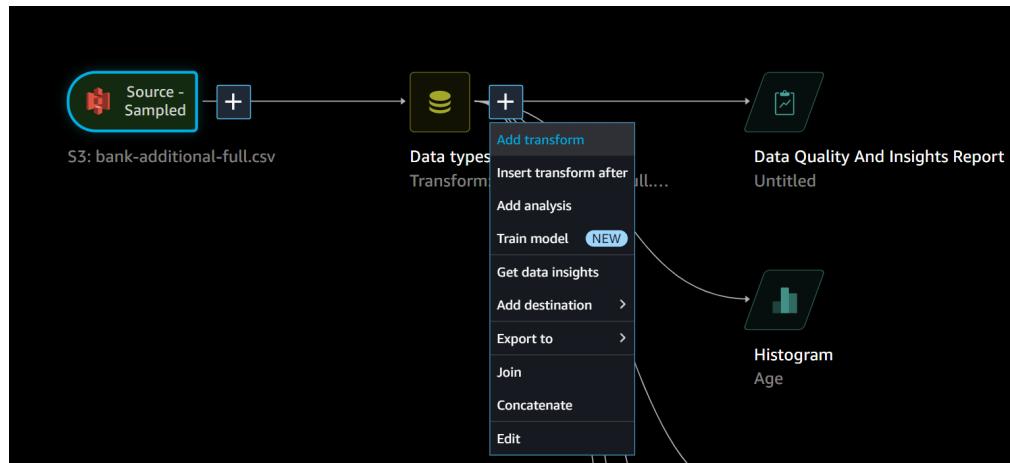
In order to add transformation, go to the **Data flow** tab and click on the **+** sign next to **Data types** and select **Add transform**.

SageMaker Immersion Day

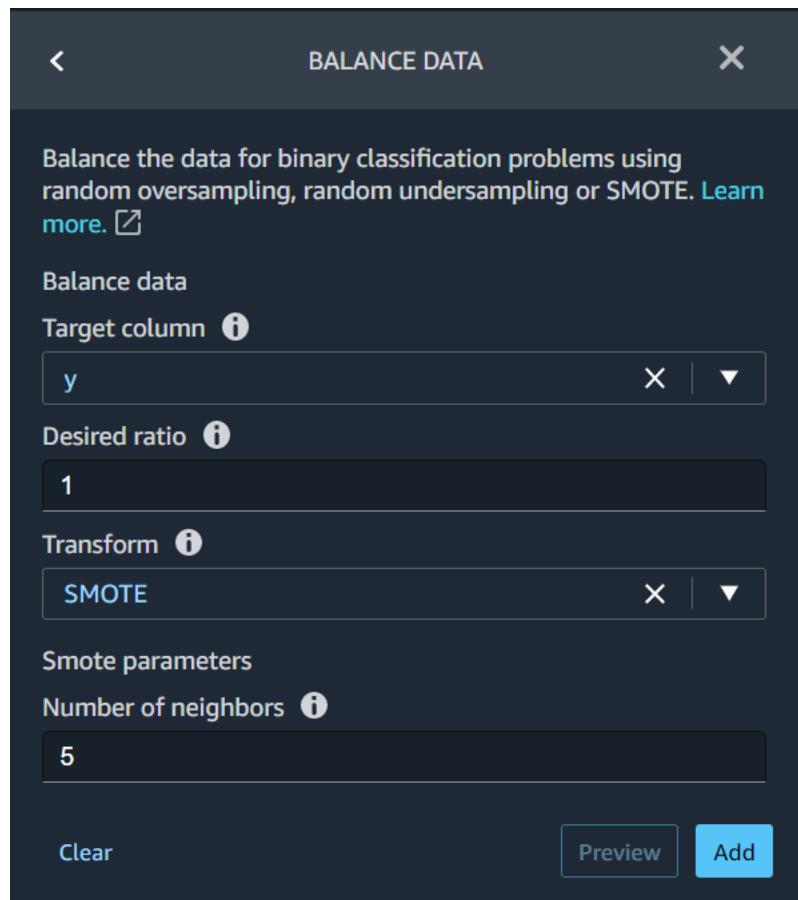
- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language



1. Choose + Add step -> Balance data transform
2. Choose SMOTE as the transform type.
3. Select y as Target column and keep rest parameters as default.
4. Click Preview and Add.



Drop Columns

We would remove certain features from our dataset.

1. Under ALL STEPS, click + Add Step.
2. Click on Manage columns. (Can also search for Manage columns in the search bar instead of scrolling)
3. Select Drop column and choose the following:
 - duration
 - emp.var.rate

SageMaker Immersion Day

▶ Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy

XGBoost

▶ Lab 3. Bring your own model

▶ Lab 4. Autopilot, Debugger and Model Monitor

▶ Lab 5. Bias and Explainability

▶ Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

▶ Lab 9. Amazon SageMaker JumpStart

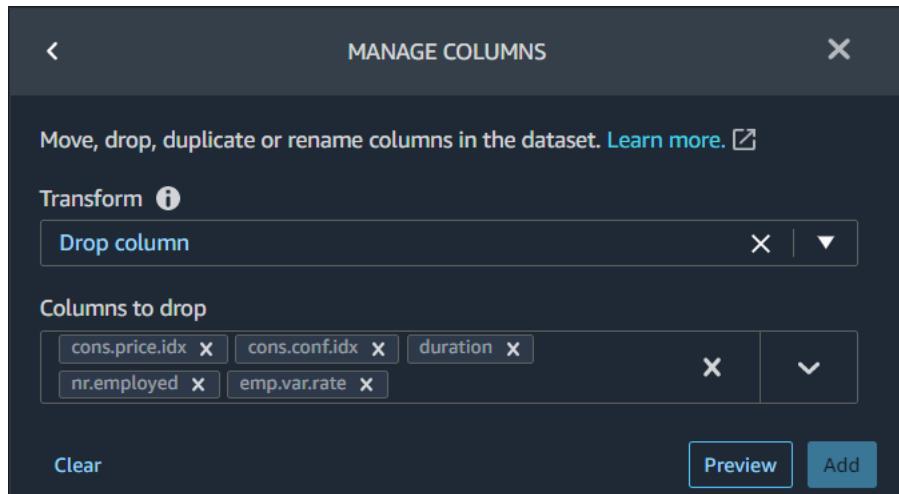
▶ Lab 10. ML Governance Tools for Amazon SageMaker

▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

- **cons.price.idx**
 - **cons.conf.idx**
 - **nr.employed**
4. choose **Preview and Add**



Handle Feature Outliers

We also noticed, from the Data Quality and Insights Report above, some anomalous data samples which may be a result of outliers in our data. Let's remove outliers from numerical features in our dataset.

1. Under **ALL STEPS**, click **+ Add Step**.
2. Click on **Handle Outliers**. (Can use the search bar)
3. Select **Standard deviation numeric outliers** under **Transforms** and choose the following under **Input columns**:
 - **previous**
 - **age**
 - **campaign**
4. Type **3** for **Standard deviations** (the criteria of outliers).
5. choose **Preview and Add**

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

HANDLE OUTLIERS

Remove or replace outlier numeric and categorical values. [Learn more.](#)

Transform

Standard deviation numeric outliers X | ▾

Detect and fix outliers in numeric features using the mean and standard deviation.

Input columns

previous X campaign X age X X ▼

Output column i

Optional

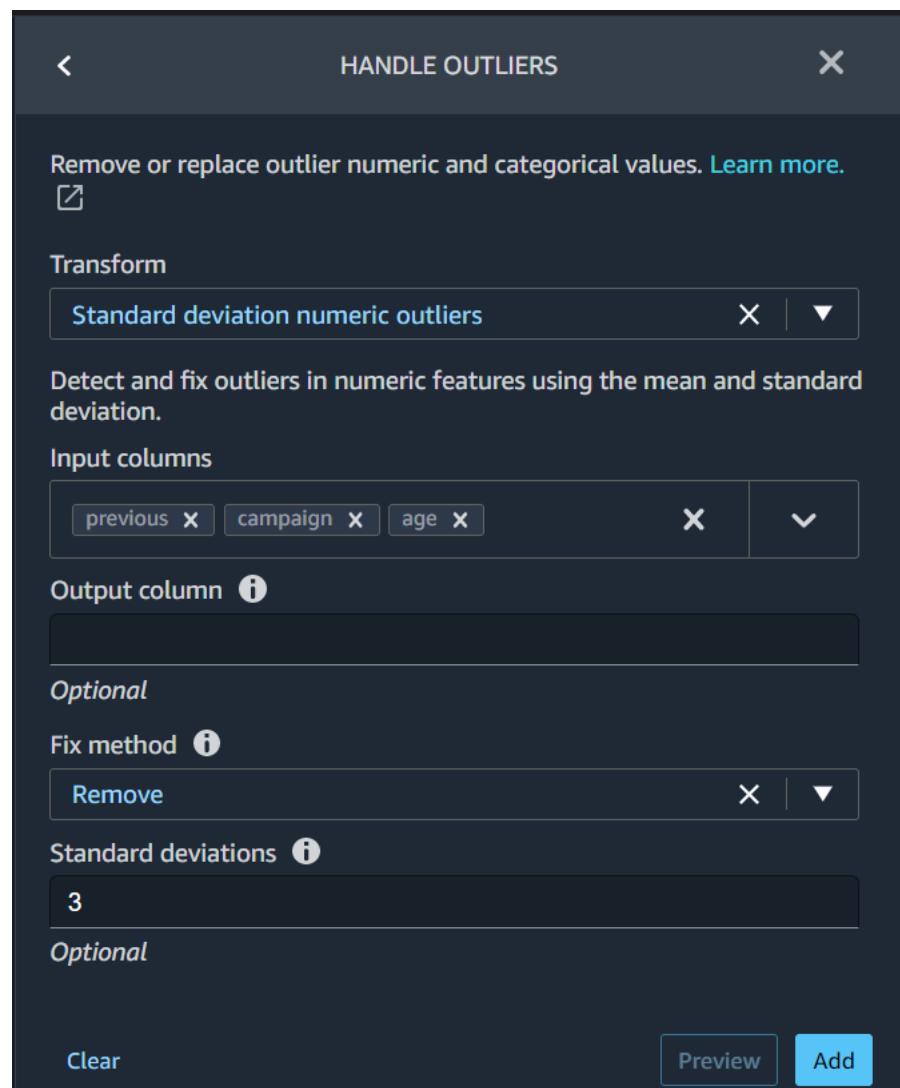
Fix method i

Remove X ▼

Standard deviations i

3 Optional

Clear Preview Add



Scale Numerical Features

Content preferences

Language

Scaling is required to rescale the data and it's used when we want features to be compared on the same scale for our algorithm. And, when all features are in the same scale, it also helps algorithms to understand the relative relationship better and converge faster.

Let's use min-max scaler to scale all numeric columns to values between 0 and 1.

1. Under **ALL STEPS**, click **+ Add Step**.
2. Click on **Process Numeric**. (Can use the search bar)
3. Select **Min-max scaler** under **Scaler** and choose the following under **Input columns**:
 - o previous
 - o age
 - o campaign
4. choose **Preview** and **Add**

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

PROCESS NUMERIC

Transform numeric values to improve machine learning model performance. [Learn more.](#)

Scale values

Scaler

Min-max scaler

Rescale the column to a specific range ([0, 1] by default).

Input columns

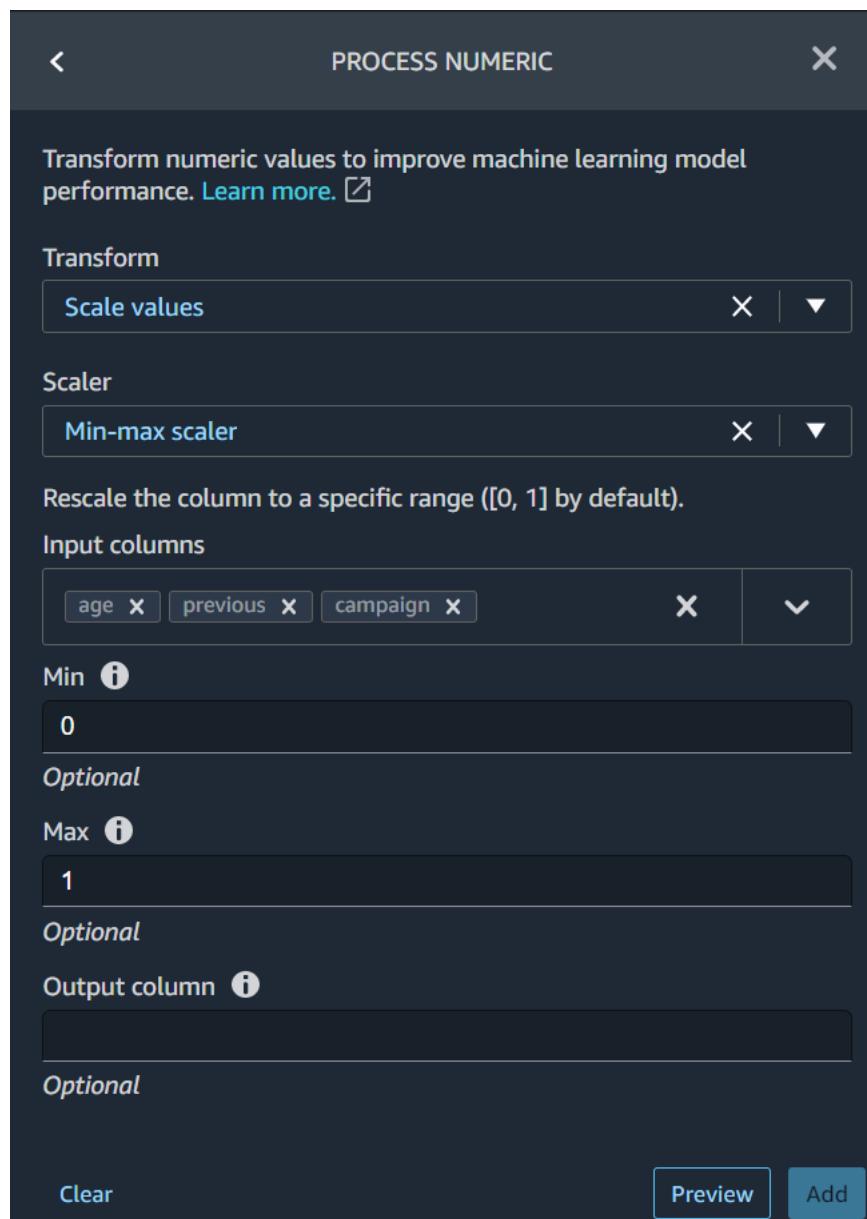
age ✖ previous ✖ campaign ✖

Min ⓘ 0

Max ⓘ 1

Output column ⓘ

Clear **Preview** **Add**



Replace Feature Characters

We would replace the dot . values into _ values for all our categorical features. This is important when exporting data to Feature Store, as feature names with . are not currently supported by Feature Store.

1. Under **ALL STEPS**, click + Add Step.
2. Click on **Search and edit**. (Can use the search bar)
3. Select **Find and replace substring** under **Transform** and choose all string type columns under **Input columns**
4. For **Pattern**, type in \. and _ for **Replacement string**
5. choose **Preview** and **Add**

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

SEARCH AND EDIT

Find, replace, split, and otherwise transform input string values using search and edit functions. [Learn more.](#)

Transform

Find and replace substring

Replace a substring matching the given regex with a new one.

Input columns

contact X day_of_week X default X education X X | ▾
housing X + 6 items selected

Pattern ⓘ

_

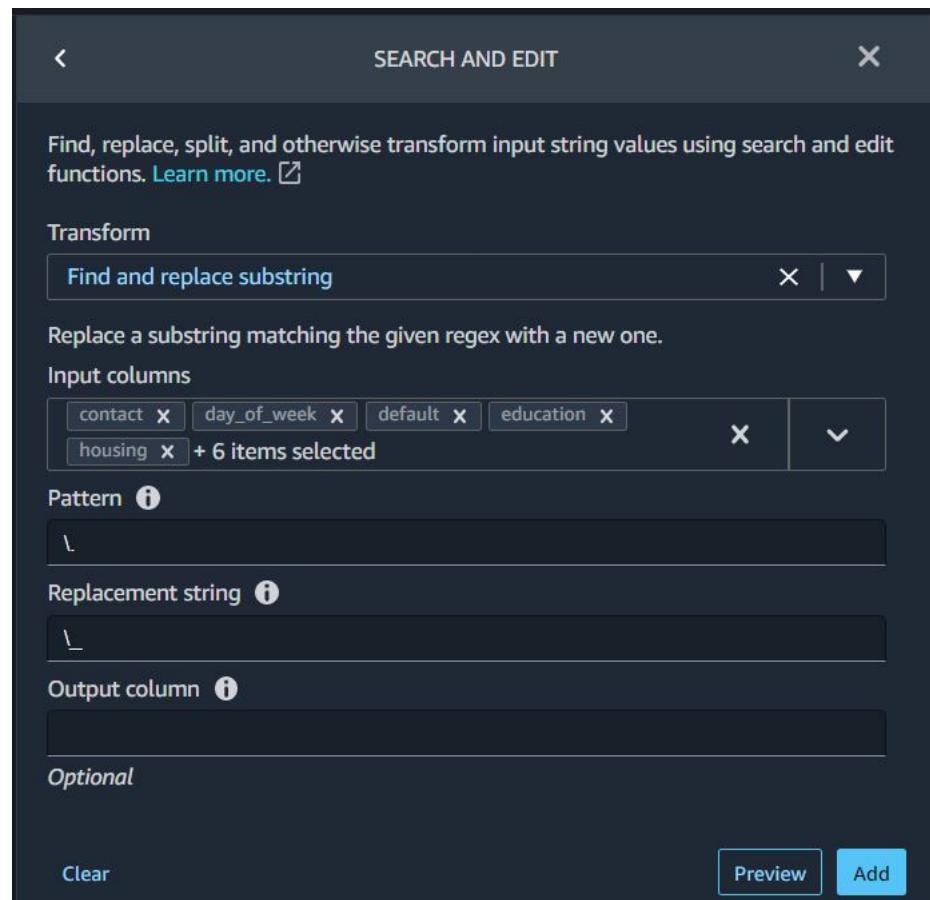
Replacement string ⓘ

_

Output column ⓘ

Optional

Clear **Preview** **Add**



We also notice that certain feature values end with a `_`, lets remove this appendix. Repeat the same process above, however, input `\$_` for **Symbols** and leave **Replacement string** empty (put an S and delete it to be able to proceed with this field empty).

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

Content preferences

Language

SEARCH AND EDIT

Find, replace, split, and otherwise transform input string values using search and edit functions. [Learn more.](#)

Transform

Find and replace substring

Replace a substring matching the given regex with a new one.

Input columns

contact **x** day_of_week **x** default **x**
education **x** housing **x** + 6 items selected **x** **v**

Pattern ⓘ

\\$_

Replacement string ⓘ

Output column ⓘ

Optional

Clear **Preview** **Add**

Custom Transforms

The Custom Transforms step allows you to define custom transformations in **Python (User-Defined Function)**, **Python (PySpark)**, **Python (Pandas)**, and/or **SQL (PySpark SQL)**. We will add a custom transformation step to do the following.

- Add an indicator variable to capture when pdays takes a value of 999.
- Add an indicator for individuals not actively employed
- Add unique ID and date for features store
- Compute the [Variance Inflation factor \(VIF\)](#) of various numerical features in the dataset and drop columns with a VIF value greater than 1.2. This helps us mitigate any multicollinearity in our dataset.

Follow the step below to implement this custom transformation in either **Python (Pandas)** or **Python (PySpark)** by choosing the appropriate tab.

Python (Pandas) **Python (PySpark)**

Under **ALL STEPS** on the right select **+ Add Step** then **Custom Transform**. Change to **Python (Pandas)**. You will get a warning, which is not a problem. Copy and paste this script:

```
1 import time
2 import pandas as pd
3 from statsmodels.stats.outliers_influence import variance_inflation_factor
4 from statsmodels.tools.tools import add_constant
5 import numpy as np
6
7 # Add two new indicators
8 df["no_previous_contact"] = (df["pdays"] == 999).astype(int)
```

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

```
9     df["not_working"] = df["job"].isin(["student", "retired", "unemployed"]).astype(int)
10    df['pdays']=df['pdays'].astype(np.float64) #cast pdays column type to double precision
11
12    # Add unique ID and event time for features store
13    df['FS_ID'] = df.index + 1000
14    current_time_sec = int(round(time.time()))
15    df['FS_time'] = pd.Series([current_time_sec]*len(df), dtype="float64")
16
17    # compute the vif and drop columns greater than 1.2 threshold
18    def compute_vif(df, threshold):
19        names=['age','euribor3m','campaign','not_working', 'no_previous_contact']
20        considered_features= [name for name in names if name in df.columns]
21        subset=df[considered_features]
22        subset=subset.dropna()
23        subset['intercept'] = 1
24        vif = pd.DataFrame()
25        vif["Variable"] = subset.columns
26        vif["VIF"] = [variance_inflation_factor(subset.values, i) for i in range(subset.shape[1])]
27        vif = vif[vif['Variable']!= 'intercept']
28        vif=vif.sort_values('VIF', ascending=False)
29        print(vif)
30        drop_clm=vif.index[vif.VIF.gt(threshold)].tolist()
31        drop_clm_names=vif.loc[drop_clm]['Variable'].tolist()
32        df.drop(drop_clm_names, axis=1, inplace=True)
33        return df
34    df=compute_vif(df, 1.2)
```

Choose **Preview**, then choose **Add**.

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy

XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

CUSTOM TRANSFORM

Python (Pandas)

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 Python (PySpark) or Python (User-Defined Function) for production use-cases

```
1 import time
2 import pandas as pd
3 from statsmodels.stats.outliers_influence import variance_inflation_factor
4 from statsmodels.tools.tools import add_constant
5 import numpy as np
6
7 # Add two new indicators
8 df["no_previous_contact"] = (df["pdays"] == 999).astype(int)
9 df["not_working"] = df["job"].isin(["student", "retired", "unemployed"]).astype(int)
10 df['pdays']=df['pdays'].astype(np.float64) #cast pdays column type to double precision
11
12 # Add unique ID and event time for features store
13 df['FS_ID'] = df.index + 1000
14 current_time_sec = int(round(time.time()))
15 df['fs_time'] = pd.Series([current_time_sec]*len(df), dtype="float64")
16
17 # compute the vif and drop columns greater than 1.2 threshold
18 def compute_vif(df, threshold):
19     names=['age','euribor3m','campaign','not_working', 'no_previous_contact']
20     considered_features= [name for name in names if name in df.columns]
21     subset=df[considered_features]
22     subset=subset.dropna()
23     subset['intercept'] = 1
24     vif = pd.DataFrame()
25     vif["Variable"] = subset.columns
26     vif["VIF"] = [variance_inflation_factor(subset.values, i) for i in range(subset.shape[1])]
27     vif = vif[vif['Variable']!= 'intercept']
28     vif=vif.sort_values('VIF', ascending=False)
29     print(vif)
30     drop_clm=vif.index[vif.VIF.gt(threshold)].tolist()
31     drop_clm_names=vif.loc[drop_clm]['Variable'].tolist()
32     df.drop(drop_clm_names, axis=1, inplace=True)
33     return df
34 df=compute_vif(df, 1.2)
```

Clear Output Preview Add

	Variable	VIF
2	1	euribor3m 1.206349
3	4	no_previous_contact 1.155552
4	3	not_working 1.109374
5	0	age 1.070811
6	2	campaign 1.029973

One-hot encoding the categorical variables

To convert categorical variables into sets of indicators, we use one hot encoding. This will encode the categorical features as a one-hot numeric array.

1. Click **Add Step**
2. Choose **Encode Categorical**.
3. Select **One-hot encode** as **Transform** and choose all string type columns as **Input columns**.
4. Make sure select **Columns** as **Output style**.
5. Click on **Preview**, then **Add** to add the transform to the data flow.

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

Content preferences

Language

Encode Categorical

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

Transform [i](#)

One-hot encode [X](#) | [▼](#)

Input columns [i](#)

contact [X](#) day_of_week [X](#) default [X](#) education [X](#) housing [X](#) + 6 items selected [X](#) [▼](#)

Input already ordinal encoded [i](#)

Invalid handling strategy [i](#)

Keep [X](#) | [▼](#)

Drop last [i](#)

Output style [i](#)

Columns [X](#) | [▼](#)

Output column [i](#)

Optional

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

After adding all the above transforms, you should see the 10 steps on the right of the screen under **Transforms**. Click on < Data flow on the top left of the screen.

Data flow

One-hot encode - Transform: bank-additional-full.csv

Data Analysis Training [NEW](#)

Step 10. One-hot encode [▼](#)

age (float)	campaign (float)	pdays (float)	previous (float)	no_previous_contact (float)	not_working (float)	FS_ID (long)
0.6659164857455301	0	999	0	1	0	1000
0.682991265380309	0	999	0	1	0	1001
0.3414956326900154	0	999	0	1	0	1002
0.3927199759351773	0	999	0	1	0	1003
0.6659164857455301	0	999	0	1	0	1004
0.47809588676602163	0	999	0	1	0	1005
0.7171408286490324	0	999	0	1	0	1006
0.40979475922801856	0	999	0	1	0	1007
0.11952347144150541	0	999	0	1	0	1008
0.13659925230700618	0	999	0	1	0	1009
0.40979475922801856	0	999	0	1	0	1010
0.13659925230700618	0	999	0	1	0	1011
0.204897375361400928	0	999	0	1	0	1012
0.682991265380309	0	999	0	1	0	1013

ALL STEPS

- + Add step
- 1. S3 Source
- 2. Data types
- 3. SMOTE
- 4. Drop column
- 5. Standard deviation numeric outliers
- 6. Scale values
- 7. Find and replace substring
- 8. Find and replace substring
- 9. Python (PySpark)
- 10. One-hot encode

Data Export Options

The ImmersionDay.flow file we created captures all of the transformations and analyses we have stitched together so far using the visual editor. This file persists every transformation step we have created and can be reused to modify/add new steps. After feature engineering we would like to persist the processed data in a storage that can be easily accessible or use the processed data for other downstream tasks including model training. We would look into a few of the export options and training capabilities integrated with Data Wrangler. To see a full list of the export options provided by Data Wrangler, please read this [documentation](#).

[Export to Amazon Feature Store](#)

[Export to Amazon S3](#)

[Export to SageMaker Inference Pipeline](#)

Amazon SageMaker Feature is a purpose-built repository where you can store and access features so it's much easier to name, organize, and reuse them across teams. SageMaker Feature

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

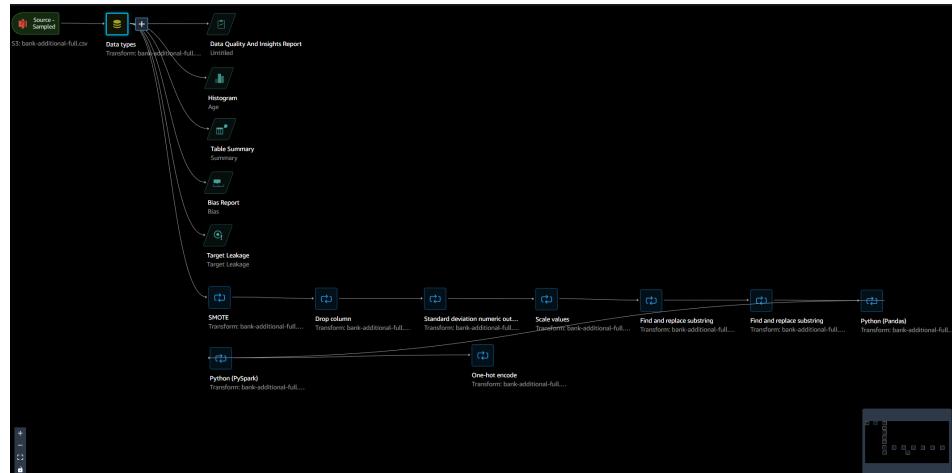
► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

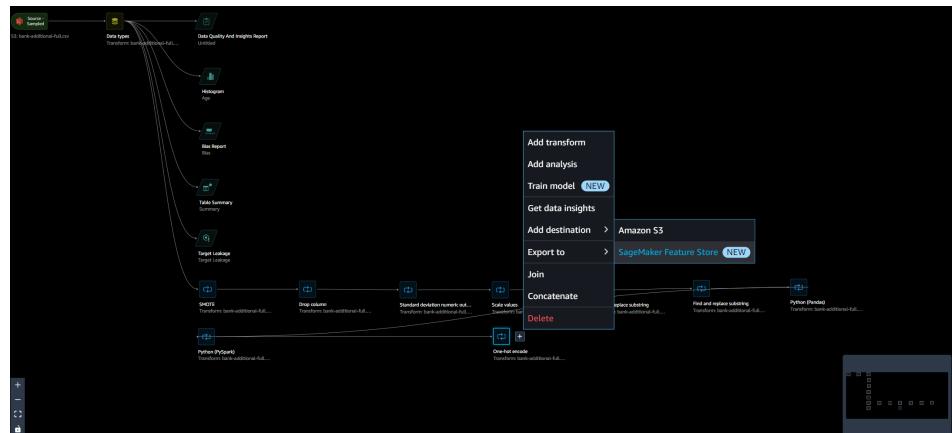
Store provides a unified store for features during training and real-time inference without the need to write additional code or create manual processes to keep features consistent. SageMaker Feature Store keeps track of the metadata of stored features (e.g. feature name or version number) so that you can query the features for the right attributes in batches or in real time using Amazon Athena, an interactive query service. SageMaker Feature Store also keeps features updated, because as new data is generated during inference, the single repository is updated so new features are always available for models to use during training and inference.

Data flow window should show all your transformation steps



Destination nodes allow you to choose destination sinks where your transformed features must go to. The destination sink can either be Amazon Simple Storage Service (S3) or SageMaker Feature Store. Once you create the node, you can configure and kick-off a SageMaker Processing job which will run a distributed fully managed Spark job to scale your flow recipe on larger amounts of dataset with the same schema.

Let's export all the features that have been created in Data Wrangler to Feature Store. This can be done by choosing the + sign on the **One-hot encode** transform tile. From there, choose **Add destination→ Sagemaker Feature Store**



Choose **Create Feature Group** button. On the pop-up window, leave **Create "EventTime" column** box unchecked and choose **Next**.

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

The screenshot shows the 'Data flow' interface with the 'Add a Feature Store destination' step selected. At the top right, there is a 'Create Feature Group' button, which is highlighted with a red box.

Choose **Copy JSON schema**, and choose **Create**. We would use the copied schema to create the Feature Store group.

The screenshot shows the 'Create Feature Group' wizard, specifically 'STEP 2 OF 2'. It displays a JSON schema for feature definitions:

```
[  
  {  
    "FeatureName": "age",  
    "FeatureType": "Fractional"  
  },  
  {  
    "FeatureName": "campaign",  
    "FeatureType": "Fractional"  
  },  
  {  
    "FeatureName": "pdays",  
    "FeatureType": "Fractional"  
  },  
  {  
    "FeatureName": "previous",  
    "FeatureType": "Fractional"  
  }]
```

At the bottom left, there is a 'Copy JSON schema' button, which is highlighted with a red box and has a red circle labeled '1' above it. At the bottom right, there is a 'Create' button, which is also highlighted with a red box and has a red circle labeled '2' above it.

Enter `ImmersionDay` for **Feature group name**, choose **Offline storage** under **Storage type**. We don't need the online store as we will use Feature Store to retrieve features for training (not for real-time inference). Select `sagemaker-us-east-1-*` (may not be `us-east-1` depending on AWS region) under **S3 bucket name** and select `*-SageMakerExecutionRole-*` (type `sagemakerexecutionrole` to populate the result) under **IAM Role ARN**. Choose **Continue**.

Make sure you select the appropriate IAM role ARN to prevent any permission errors from creating the Feature Group.

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy

XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Feature group details Learn more

Enter feature group name and description

Feature group name*
ImmersionDay

Description
A quick description of the Feature group (max 128 characters)

Feature group storage configuration Learn more

Select at least one of these options. Feature Store can be used in Online, Offline or Online and Offline modes.

Storage type*
Offline storage

Offline storage settings

SS bucket name* ?
sagemaker-us-east-1-771014164235

Dataset directory name ?
Select or type prefix

Table format* ?
Glue (Default)

IAM role ARN*
mod-6297809195fe4845-SageMakerExecutionRole-1OLSWT2VJKG01

Use the same encryption key as online store.

Offline store encryption key ?

Use AWS managed KMS key (default)

Data catalog Learn more

An AWS Glue data catalog is automatically created for your offline Feature Store data. You can write and execute SQL queries using Amazon Athena on an AWS Glue data catalog.

If you are using an tabular table format then you must use default values for your AWS Glue data catalog.

Use default values for your AWS Glue data catalog.

Cancel Continue

Choose **JSON** tab and paste the copied schema from the previous step. This is the schema of the transformed dataset from DataWrangler. The feature group we are creating needs to have the same data schema. Choose **Continue**.

Create feature group

Feature groups allow for the logical grouping of features, defined in the Feature Store, to describe records.

Specify feature definitions Learn more

Features are variables that are input to machine learning models. In Amazon SageMaker Feature Store, a feature is an attribute of a record. Choose unique names and data types for each feature in your group. Definitions can be added in JSON or entered in a table.

Two or more features are required to be used as the record identifier and event time feature in the following step. At least one string or fractional feature is required for the event time feature.

Table **JSON**

```

1  [
2    {
3      "FeatureName": "age",
4      "FeatureType": "Fractional"
5    },
6    {
7      "FeatureName": "campaign",
8      "FeatureType": "Fractional"
9    },
10   {
11     "FeatureName": "pdays",
12     "FeatureType": "Fractional"
13   },
14   {
15     "FeatureName": "previous",
16     "FeatureType": "Fractional"
17   },
18   {
19     "FeatureName": "no_previous_contact",
20     "FeatureType": "Fractional"
21   },
22   {
23     "FeatureName": "not_working",
24     "FeatureType": "Fractional"
25   },
26   {
27     "FeatureName": "FS_ID",
28     "FeatureType": "Integral"
29   },
30   {
31     "FeatureName": "FS_time",
32     "FeatureType": "Fractional"
33   },
34   {
35     "FeatureName": "contact_cellular",
36     "FeatureType": "Fractional"
37   },
38   {
39     "FeatureName": "contact_telephone",
40     "FeatureType": "Fractional"
41   }
42 ]

```

Cancel Back Continue

Select **FS_ID** (**Integral**) under **RECORD IDENTIFIER FEATURE NAME** and **FS_time** (**Fractional**) under **EVENT TIME FEATURE NAME**, choose **Continue**.

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Create feature group

Feature groups allow for the logical grouping of features, defined in the Feature Store, to describe records.

Step 1: Feature group information

Step 2: Feature definitions

Step 3: Required features

Step 4 - optional: Feature group tags

Step 5: Review feature group

Select required features [Learn more](#)

You must select a record identifier feature and an event time feature name. A combination of record identifier name and a timestamp uniquely identify a record within a feature group.

Record identifier feature name* Choose a record identifier from your list of feature definitions to uniquely identify your feature group.

FS_ID (Integral)

Event time feature name* The event time feature will be used to identify the latest feature values in online store. Choose a feature from your list of feature definitions to identify the event time. For standard Glue table format this can be string (ISO 8601) or fractional (UNIX time stamp). For Iceberg table format it is required to use string (ISO 8601).

|FS_time (Fractional)

Cancel Back Continue

Choose **Continue** and choose **Create feature group**. You would get a feature group created success notification.

Create feature group

Feature groups allow for the logical grouping of features, defined in the Feature Store, to describe records.

Step 1: Feature group information

Step 2: Feature definitions

Step 3: Required features

Step 4 - optional: Feature group tags

Step 5: Review feature group

Review feature group

Step 1: Feature group information

Feature group details

Feature group name: ImmersionDay

Description:

Feature group storage configuration

Storage type: Offline

SS bucket: s3://sagemaker-us-east-1-77014164235/mod-0297089195febd45-SageMakerExecutionRate-10LSWTZVXGG0/

IAM Role ARN: arn:aws:iam::77014164235:role/mod-0297089195febd45-SageMakerExecutionRate-10LSWTZVXGG0/

Step 2: Feature definitions

Specify feature definitions

Number of feature definitions: 63

Step 3: Required features

Select required features

Record identifier name: FS_ID

Event time feature name: FS_time

Step 4: Feature group tags

Add feature group tags

Number of feature group tags: 0

You successfully created your feature group. You can now view the new feature group or view all feature groups in the feature group catalog.

Exit View feature group View feature group catalog

Go back to your DataWrangler flow, choose the plus sign on your last tile and add feature store as destination again. You should see the newly created feature group populate. Select the **ImmersionDay** feature group under **Name** tab and click on the message under the **Validation** tab to validate the schema of the dataset to that of the feature group.

Add a Feature Store destination

Export the data that you've transformed by Amazon SageMaker Feature Store. Feature Store helps you store the features from your data into one place without needing to do additional processing. [Learn more](#)

Search for a Feature Group and choose one as the destination for your data flow.

Feature Group: Search for a Feature Group

Name: ImmersionDay Date created: 2023-05-21 01:06:21.916000+0000 Validation: Click this message to have Data Wrangler validate the schema of the dataset with the schema of the feature group.

Write to offline store only

Add Create Feature Group

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

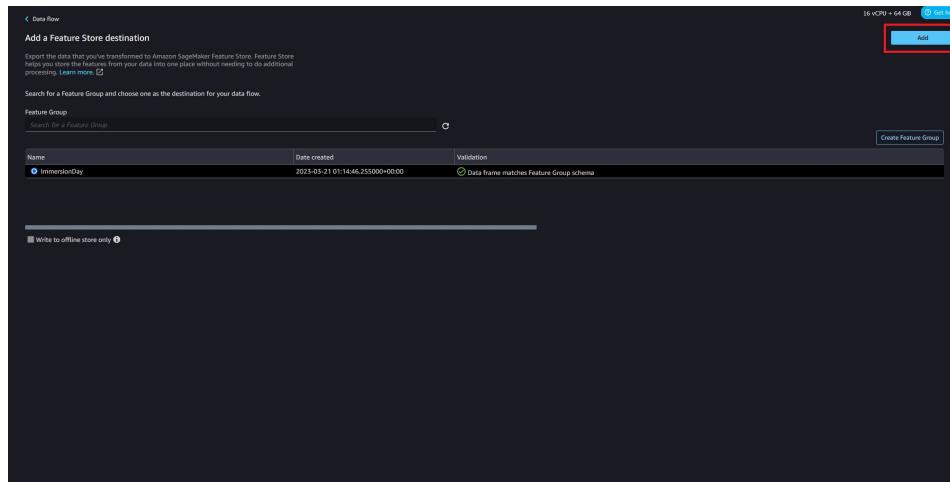
► Lab 10. ML Governance Tools for Amazon SageMaker

► Lab 11. SageMaker Notebook Instances

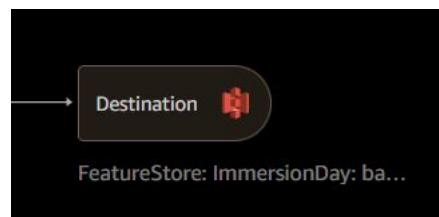
▼ Content preferences

Language

Once validation is successful, choose **Add**.



A feature store destination node will be added to your flow.

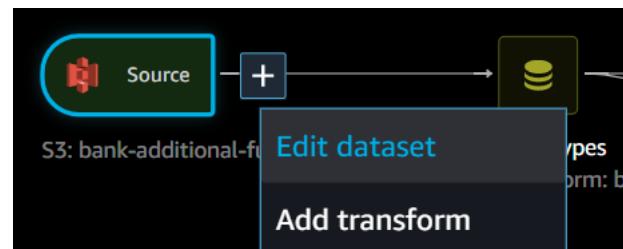


Before we move on to creating a SageMaker processing job, let's prepare the flow to accept a variable S3 file path, instead of the existing hardcoded one.

Parameterize the S3 data source path

When we imported the source dataset from S3, Data Wrangler registered the absolute path to that file. In this step, we'll parameterize that path, so we don't need to modify the import node/source data each time we wish to run the same transform flow against a different file.

1. Click the **+** from the **Source** node, and click **Edit dataset**.



You will see the *S3 URI path* at the top, ending in the basename of the dataset file, in our case, *bank-additional-full.csv*.

2. Highlight just the *bank-additional-full.csv* portion of the path.

The **Create custom parameter** drop-down appears. Click it to open it up, and create the parameter as shown in the figure below.

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
 - Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Data flow

Edit S3 source: bank-additional-full.csv

Enter the S3 URL of a file or prefix (folder) in the text box, or use the following table to browse S3

Advanced configuration

S3 URI path: s3://calibucket-aws/bank-additional-full.csv

Create custom parameter ▾

CREATE PARAMETER

Name: basename_param

Type: String

Value: bank-additional-full.csv

Description: The base name of the source dataset path

Cancel Create

Last modified: 2022-11-10 23:36:03+00:00

PREVIEW • bank-additional-full.csv

age	default	housing
56	no	no
57	unknown	no
37	no	yes
40	no	no

configuration.)

Previous Displaying 1 - 1 Next

3. Click **Create** to create the parameter.

4. Click **Apply** at the top of the screen to confirm the change.

Create Processing Job

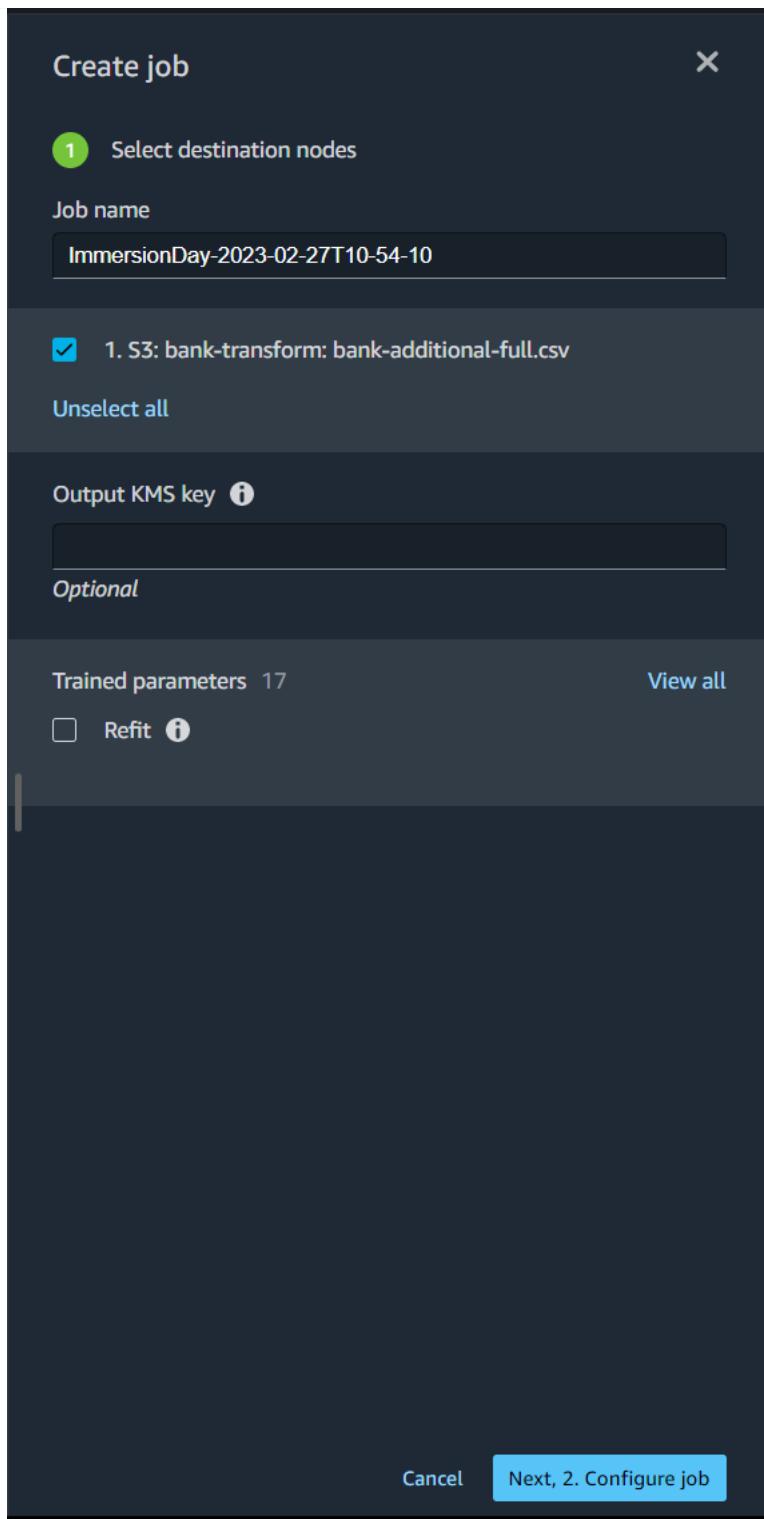
Now that we've parameterized the filename, and defined a destination node, we can move on to creating the actual processing job to execute the flow. To do this, choose **Create job** on the top-right corner of your **Data flow** screen.

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language



We will keep the **Output KMS key** empty as we will not be encrypting our data output for this example.

There are 17 trained parameters in our flow file. These trained parameters have been fitted on the available data in Data Wrangler with certain transforms including **One-hot encode**, **Scale values** and **Standard deviation numeric outliers**. These transforms depend on the available data and must be re-fitted when new data is available. Users can specify when they want to refit new transforms on their data by checking the **Refit** parameter when creating a Data Wrangler job. For a more in-depth walkthrough on the Refit feature see <https://aws.amazon.com/blogs/machine-learning/refit-trained-parameters-on-large-datasets-using-amazon-sagemaker-data-wrangler/>. We will keep the Refit parameter unchecked since we did not sample our dataset and imported the full dataset into Data Wrangler. Choose **Next, 2. Configure job**.

SageMaker Immersion Day

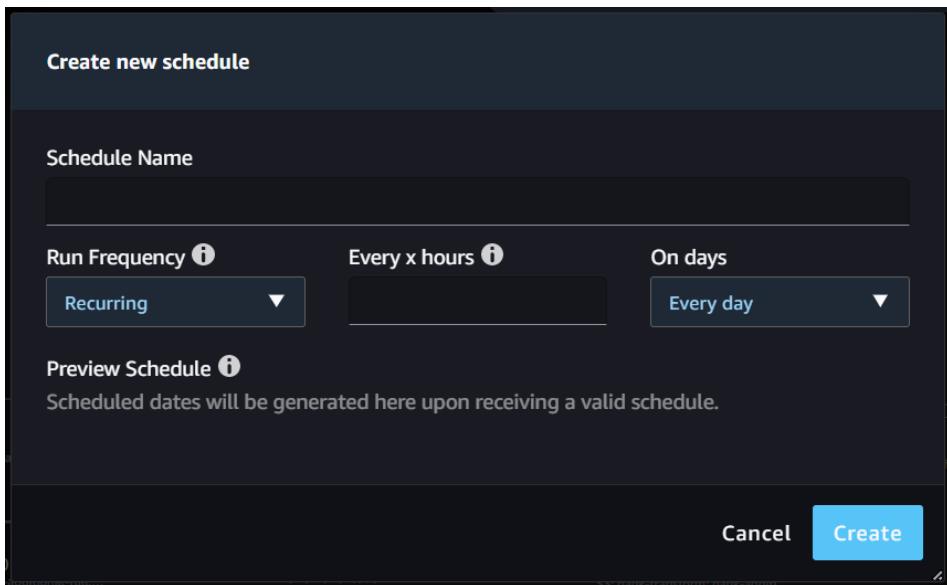
- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

The *Configure job* panel presents you with additional configurable parameters including **Instance type**, **instance count**, **Volume size**, etc. Some interesting features of Data Wrangler are:

1. **Associate Schedules:** You can schedule your Data Wrangler jobs to run at specific times and frequency. CRON expression provide additional flexibility in providing time schedules. To schedule Data Wrangler jobs, click on **Associate schedules** and **Create new schedule**. A pop-up appears where you can specify the Schedule Name and timing configurations. We will not be creating a schedule for this workshop.



2. **Parameters:** We parametrized our source data path, `basename_param`, earlier which is available under the **Parameters tab**. Here you can change the value of `basename_param` to point to a different file name. But for this workshop we will leave it same.

Click **Create** to create the processing job.

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Create job

2 Configure job

Instance type	ml.m5.4xlarge	Instance count	2
---------------	---------------	----------------	---

> Job configuration

> Spark memory configuration

> Network configuration

> Tags

▼ Parameters
basename_param

▼ Associate schedules
[Create new schedule](#)

[Previous](#) [Cancel](#) [Create](#)

The confirmation screen will appear. We have created a Sagemaker processing job to execute the Data Wrangler flow we created, implement all the transforms we added and store the output in the feature group we created

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

Content preferences

Language

Create job



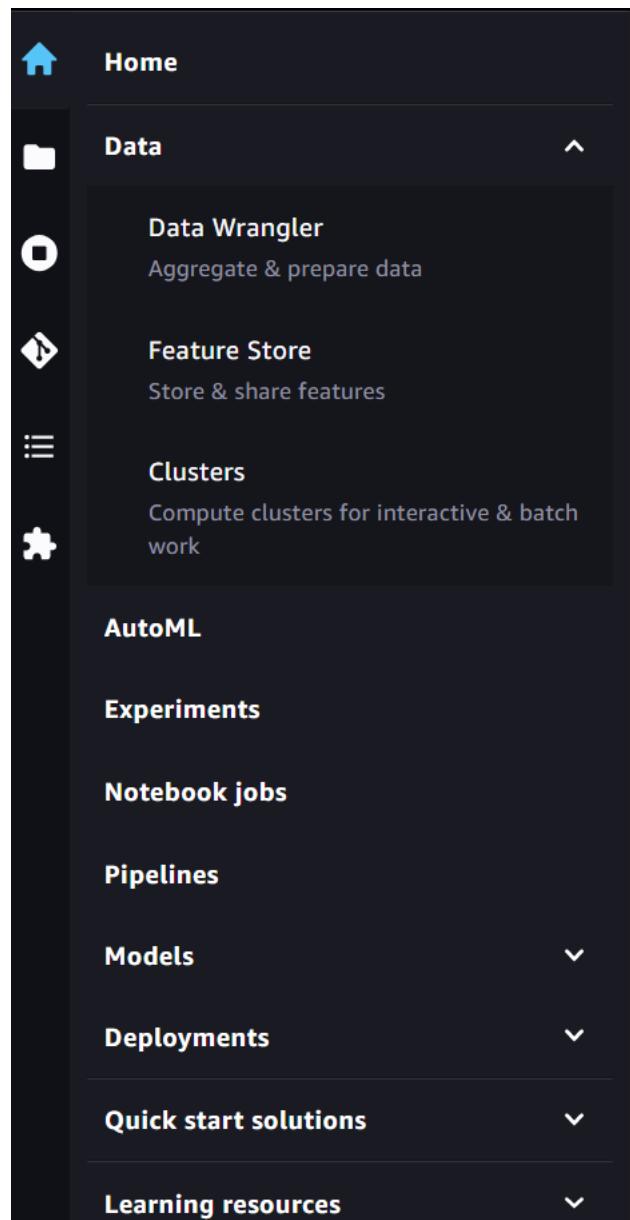
Job created successfully

To monitor the status of a job, select its name.

Processing Job name: **ImmersionDay-2023-02-27T12-43-05**

Processing Job ARN: **arn:aws:sagemaker:us-east-...**

Now, let's view the newly created feature group. In the SageMaker Studio Notebook environment, choose the 'home' icon, choose **Data** and select **Feature Store**.



You will then be able to see the feature group created inside the Feature store:

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Feature Store

Feature Group Catalog Feature Catalog

Search by Feature group name or Description or Re...

Show introduction

Create feature group

Feature group name	Description	Tags	Record identifier	Store type	Status	Created on	Offline store status
FG-ImmersionDay-3ea9a2c0		2 items	FS_ID	Online/Offline	Created	02/27/2023	Active

With the corresponding structure if you click on it:

Explore features / FG-ImmersionDay-3ea9a2c0					
FG-ImmersionDay-3ea9a2c0					
Features Details Sample queries					
Search by Feature name or Description or Parameters					
Feature name	Type	Description	Parameters	Created on	
default_yes	Fractional			02/27/2023	
education_iliterate	Fractional			02/27/2023	
housing_yes	Fractional			02/27/2023	
job_technician	Fractional			02/27/2023	
job_sef-employed	Fractional			02/27/2023	
loan_yes	Fractional			02/27/2023	
loan_unknown	Fractional			02/27/2023	
marital_married	Fractional			02/27/2023	
month_jan	Fractional			02/27/2023	
campaign	Fractional			02/27/2023	
no_previous_contact	Integral			02/27/2023	
contact_cellular	Fractional			02/27/2023	
day_of_week_fri	Fractional			02/27/2023	
education_university_degree	Fractional			02/27/2023	
education_unknown	Fractional			02/27/2023	
job_retired	Fractional			02/27/2023	
job_unemployed	Fractional			02/27/2023	
month_may	Fractional			02/27/2023	
month_aug	Fractional			02/27/2023	
month_apr	Fractional			02/27/2023	
poutcome_failure	Fractional			02/27/2023	
y_no	Fractional			02/27/2023	

💡 You can copy the feature group name to a notepad. You will need it later in the lab.

You can go to the main Amazon SageMaker page in the AWS console and click on **Processing jobs**:

The screenshot shows the Amazon SageMaker console with the 'Processing' section selected. On the left, there's a sidebar with links like 'Getting started', 'Studio', 'Canvas', 'RSStudio', 'Domains', 'SageMaker dashboard', 'Images', 'Lifecycle configurations', 'Search', 'JumpStart', 'Governance', 'Ground Truth', 'Notebook', 'Processing' (which is expanded to show 'Processing jobs'), 'Training', 'Inference', and 'Edge Manager'. The main content area has a dark header 'Amazon SageMaker' with the sub-header 'MACHINE LEARNING'. Below the header, there's a large call-to-action button 'Build, train, and deploy machine learning models at scale'. To the right of the button, there's a 'New to SageMaker?' section with a 'Get Started' button, and a 'Documentation' section with links to 'Getting started', 'Tutorials', 'Documentation', 'Developer Resources', 'AWS Developer Forum', and 'Contact us'. In the bottom left, there's a 'How it works' section with a 'What is Amazon SageMaker?' paragraph and a 'New user onboarding guide' link. In the bottom right, there's a 'Typical SageMaker workflow' section with a 'Label data' link.

You will see a processing job for data wrangler ingestion into features store:

Amazon SageMaker > Processing jobs					
Processing jobs					
Actions Create processing job					
Search processing jobs					
Name	ARN	Creation time	Duration	Status	
data-wrangler-flow-processing-27-19-54-22-cdf277bd	arn:aws:sagemaker:us-east-1:259508681668:processing-job/data-wrangler-flow-processing-27-19-54-22-cdf277bd	Feb 27, 2023 19:54 UTC	8 minutes	Completed	

Click on the job name under **Name** to reach the **Job settings** menu which shows a complete information about the processing job.

SageMaker Immersion Day

► Prerequisites

▼ Lab 1. Feature Engineering

Option 1: Amazon SageMaker Data Wrangler and Feature Store

Option 2: Numpy and Pandas

Option 3: Amazon SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

► Lab 3. Bring your own model

► Lab 4. Autopilot, Debugger and Model Monitor

► Lab 5. Bias and Explainability

► Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

► Lab 9. Amazon SageMaker JumpStart

► Lab 10. ML Governance Tools for Amazon SageMaker

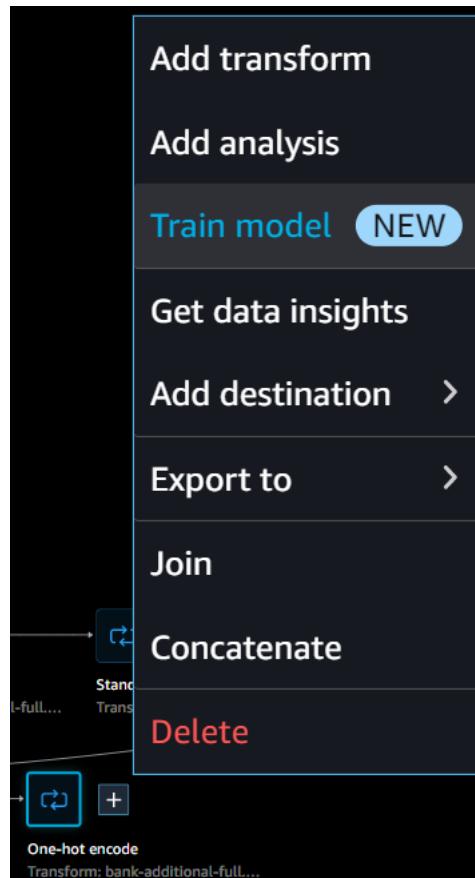
► Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

Model Training with Amazon SageMaker Autopilot (Optional)

Data Wrangler provides a unified experience enabling you to prepare data and seamlessly train an ML model, all from within the tool. With just a few clicks, you can automatically build, train, and tune ML models using Autopilot, XGBoost, or your own algorithm, directly from the Data Wrangler user interface (UI). In this lab, we will focus on training with AutoPilot. To get started, choose the + sign on the **One-hot encode** transform tile. From there, choose **Train model**.



On the **Training** console, you can change the S3 output location and the output file type. However, for this workshop we will go with the default parameters.

SageMaker Immersion Day

- ▶ Prerequisites
- ▼ Lab 1. Feature Engineering
 - Option 1: Amazon SageMaker Data Wrangler and Feature Store**
 - Option 2: Numpy and Pandas
 - Option 3: Amazon SageMaker Processing
- Lab 2. Train, Tune and Deploy XGBoost
- ▶ Lab 3. Bring your own model
- ▶ Lab 4. Autopilot, Debugger and Model Monitor
- ▶ Lab 5. Bias and Explainability
- ▶ Lab 6. SageMaker Pipelines
- Lab 7. Real Time ML inference on Streaming Data
- Lab 8. Build ML Model with No Code Using Sagemaker Canvas
- ▶ Lab 9. Amazon SageMaker JumpStart
- ▶ Lab 10. ML Governance Tools for Amazon SageMaker
- ▶ Lab 11. SageMaker Notebook Instances

▼ Content preferences

Language

◀ Data flow

One-hot encode · Transform: bank-additional-full.csv

Data Analysis Training **NEW**

Export data and train a model with SageMaker Autopilot

Amazon Sagemaker Autopilot trains models on data stored in an Amazon S3 bucket. Export your data to an S3 bucket to automatically train a model using SageMaker Autopilot.

Amazon S3 location i

s3://sagemaker-us-east-1-259508681668/ Browse

Your data is exported to the following S3 location: s3://sagemaker-us-east-1-259508681668/{processing-job-name}

File type

CSV (*.csv)

KMS key ID or ARN i

Optional

Export and train

Once export is successful, you are taken to the **Create an Autopilot experiment** page, with the **Input data** S3 location already filled in for you (as it was populated from the results of the previous screen.) More information on how to configure an Autopilot job is found in Lab 4 of this workshop.

Congratulations!!! You have successfully processed and cleaned your data, also saved it to a persistent storage.

Conclusion

In this lab you have walked through the process of required environment setup and data engineering to clean and prepare your data for model building and training. In the next lab you will learn how to Train, Tune and Deploy an XGBoost model using SageMaker's Built-in XGBoost algorithm.

[Previous](#)

[Next](#)