

Hands-on tutorials

# Prepare Training Data for Machine Learning with Minimal Code



# Prepare Training Data for Machine Learning with Minimal Code:

## Hands-on tutorials

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

# Table of Contents

**Prepare Training Data for Machine Learning with Minimal Code** ..... **i**

Overview ..... 1

What you will accomplish ..... 1

Prerequisites ..... 2

Implementation ..... 2

Conclusion ..... 36

# Prepare Training Data for Machine Learning with Minimal Code

<b>AWS experience</b>	Beginner
<b>Time to complete</b>	30 minutes
<b>Cost to complete</b>	See <a href="#">Amazon SageMaker AI pricing</a> to estimate cost for this tutorial.
<b>Services used</b>	Amazon SageMaker AI Data Wrangler
<b>Last updated</b>	March 7, 2023

## Overview

In this tutorial, you will learn how to prepare data for machine learning (ML) using [Amazon SageMaker AI Data Wrangler](#).

Amazon SageMaker AI Data Wrangler reduces the time it takes to aggregate and prepare data for ML from weeks to minutes. Using SageMaker AI Data Wrangler, you can simplify the process of data preparation and feature engineering and complete each step of the data preparation workflow, including data selection, cleansing, exploration, and visualization from a single visual interface.

In this tutorial, you will use Amazon SageMaker AI Data Wrangler to prepare data to train a rental prediction model. You will use a version of the Brazil house rental dataset found in the Kaggle Data Repository. The data consists of thousands of records, each containing thirteen different features including area, rooms, parking, and other attributes. In addition, each record includes the target feature called rent amount. You will upload the data into Amazon Simple Storage Service (Amazon S3), create a new SageMaker AI Data Wrangler flow, transform the data, check the data for bias, and lastly save the output to Amazon S3 to be used later for ML training.

## What you will accomplish

In this guide, you will:

- Visualize and analyze data to understand key relationships
- Apply transformations to clean up the data and generate new features
- Automatically generate notebooks for repeatable data preparation workflows

## Prerequisites

Before starting this tutorial, you will need:

- An AWS account: If you don't already have an account, follow the [Setting Up Your AWS Environment](#) getting started guide for a quick overview.

## Implementation

### Step 1: Set up your Amazon SageMaker AI Studio domain

With Amazon SageMaker AI, you can deploy a model visually using the console or programmatically using either SageMaker AI Studio or SageMaker AI notebooks. In this tutorial, you deploy the model programmatically using a SageMaker AI Studio notebook, which requires a SageMaker AI Studio domain.

An AWS account can have only one SageMaker AI Studio domain per Region. If you already have a SageMaker AI Studio domain in the US East (N. Virginia) Region, follow the [SageMaker AI Studio setup guide](#) to attach the required AWS IAM policies to your SageMaker AI Studio account, then skip Step 1, and proceed directly to Step 2.

If you don't have an existing SageMaker AI Studio domain, continue with Step 1 to run an AWS CloudFormation template that creates a SageMaker AI Studio domain and adds the permissions required for the rest of this tutorial.

Choose the [AWS CloudFormation stack](#) link. This link opens the AWS CloudFormation console and creates your SageMaker AI Studio domain and a user named **studio-user**. It also adds the required permissions to your SageMaker AI Studio account. In the CloudFormation console, confirm that **US East (N. Virginia)** is the **Region** displayed in the upper right corner. **Stack name** should be **CFN-SM-IM-Lambda-catalog**, and should not be changed. This stack takes about 10 minutes to create all the resources.

This stack assumes that you already have a public VPC set up in your account. If you do not have a public VPC, see [VPC with a single public subnet](#) to learn how to create a public VPC.

## 1. Create the stack

Choose the [AWS CloudFormation stack](#) link. This link opens the AWS CloudFormation console and creates your SageMaker AI Studio domain and a user named **studio-user**. It also adds the required permissions to your SageMaker AI Studio account. In the CloudFormation console, confirm that **US East (N. Virginia)** is the **Region** displayed in the upper right corner. **Stack name** should be **CFN-SM-IM-Lambda-catalog**, and should not be changed. This stack takes about 10 minutes to create all the resources.

This stack assumes that you already have a public VPC set up in your account. If you do not have a public VPC, see [VPC with a single public subnet](#) to learn how to create a public VPC.

The screenshot shows the AWS CloudFormation console's 'Quick create stack' page. The 'Stack name' field is highlighted with a red box and contains the text 'CFN-SM-IM-Lambda-catalog'. Below it, a note states: 'Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-)'. The 'Template' section displays the 'Template URL' as 'https://sagemaker-sample-files.s3.amazonaws.com/libraries/sagemaker-user-journey-tutorials/CFN-SM-IM-Lambda-catalog.yaml' and a 'Stack description' explaining that the template is for an account without a pre-existing SageMaker Studio Domain & SageMaker User Profile. The 'Parameters' section shows 'DomainName' as 'StudioDomain' and 'UserProfileName' as 'studio-user'.

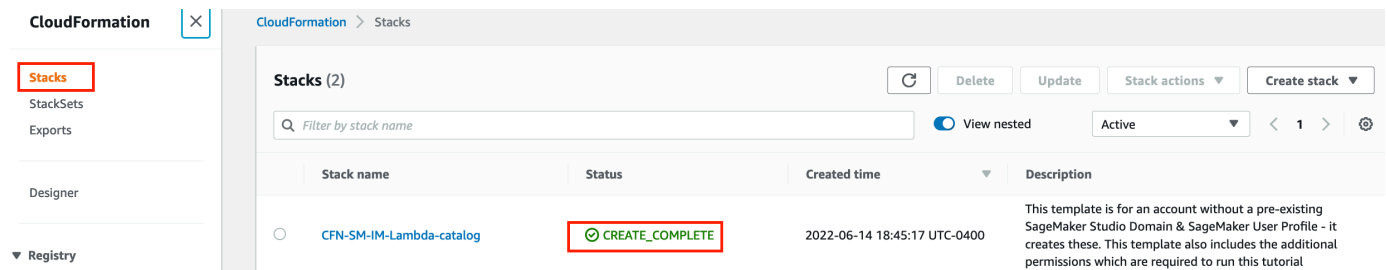
## 2. Acknowledge IAM resource creation

Select **I acknowledge that AWS CloudFormation might create IAM resources**, and then choose **Create stack**.

The screenshot shows the 'Capabilities' section in the AWS CloudFormation console. A blue box contains the text: 'The following resource(s) require capabilities: [AWS::IAM::Role]. This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. Learn more'. Below this, the checkbox 'I acknowledge that AWS CloudFormation might create IAM resources.' is checked and highlighted with a red box. At the bottom, there are buttons for 'Cancel', 'Create change set', and 'Create stack'.

## 3. Monitor stack creation progress

On the **CloudFormation** pane, choose **Stacks**. It takes about 10 minutes for the stack to be created. When the stack is created, the status of the stack changes from **CREATE\_IN\_PROGRESS** to **CREATE\_COMPLETE**.

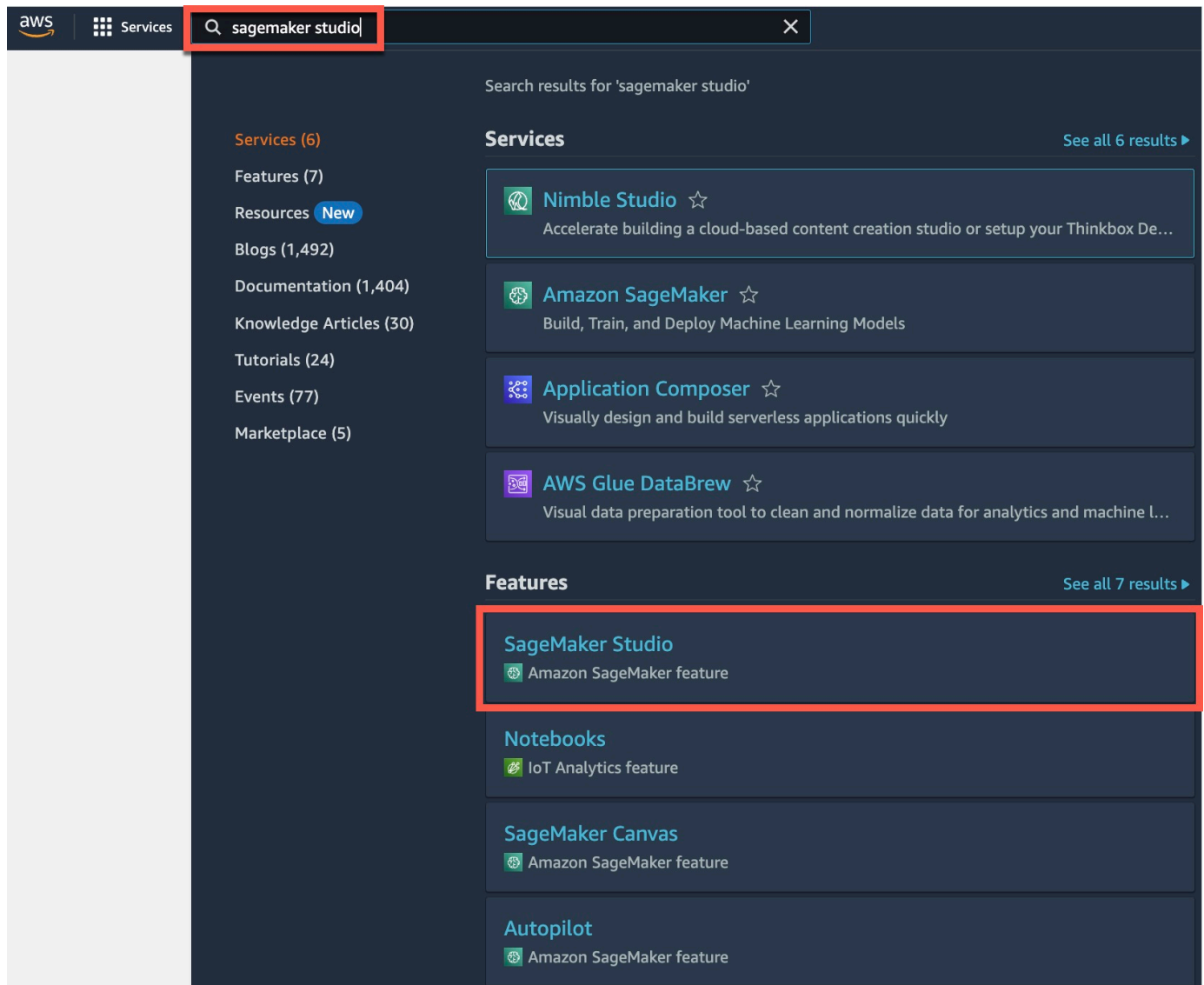


## Step 2: Create a new SageMaker AI Data Wrangler flow

SageMaker AI accepts data from a wide variety of sources, including Amazon S3, Amazon Athena, Amazon Redshift, Snowflake, Databricks, and SaaS data sources. In this step, you will create a new SageMaker AI Data Wrangler flow using the Kaggle Brazil house rental dataset stored in Amazon S3. This dataset contains demographic and financial information about homes along with a target column indicating the rental amount of the property.

### 1. Open SageMaker AI Studio

Enter **SageMaker AI Studio** into the console search bar, and then choose **SageMaker AI Studio**.



## 2. Open Studio

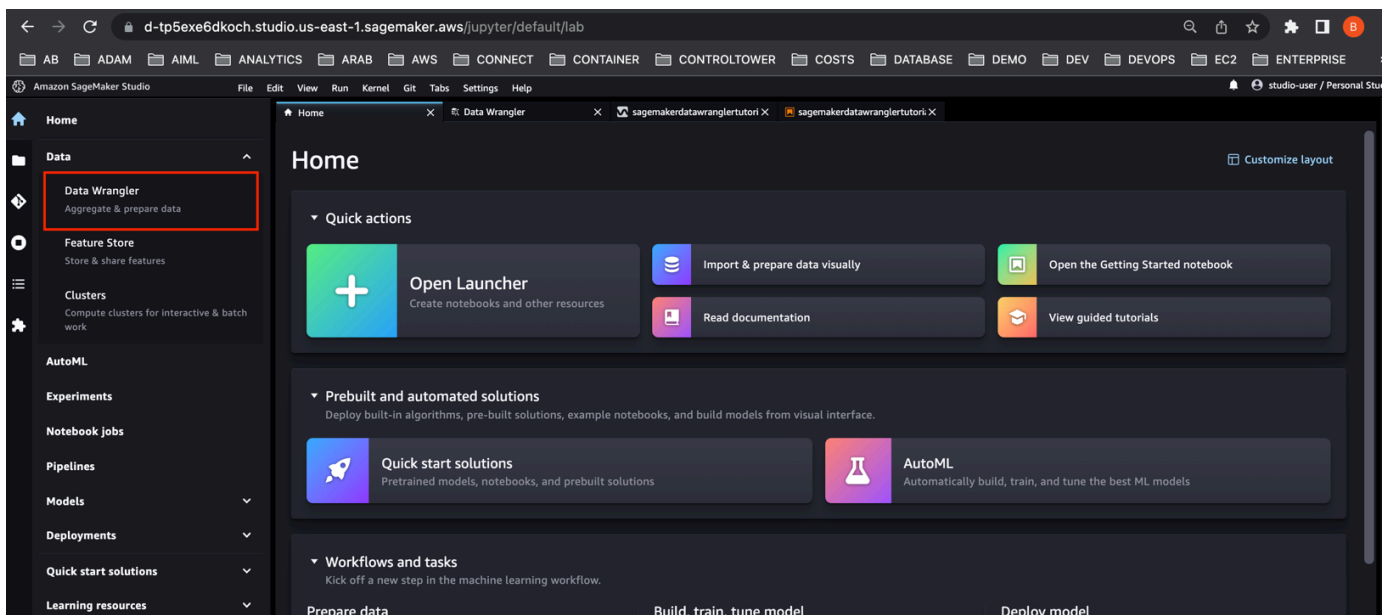
Choose **US East (N. Virginia)** from the Region dropdown list on the upper right corner of the SageMaker AI console. Browse to the **Getting Started** section in the left-hand navigation and then choose **Studio**. Then select the studio-user profile and then choose the **Open Studio** button.





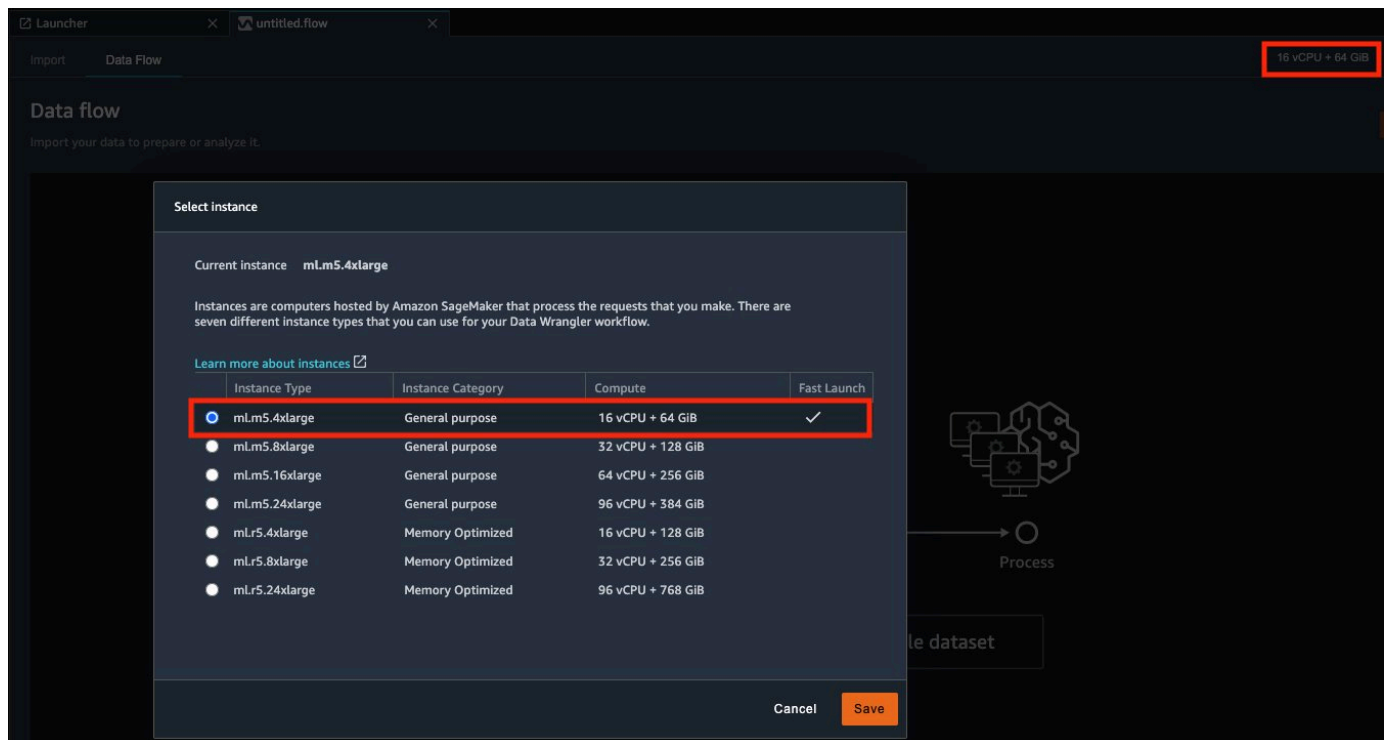
### 3. Start Data Wrangler

Open the **SageMaker AI Studio** interface. On the navigation bar, choose **Data Wrangler** on the left-hand side, and then choose the **Import Data** button.



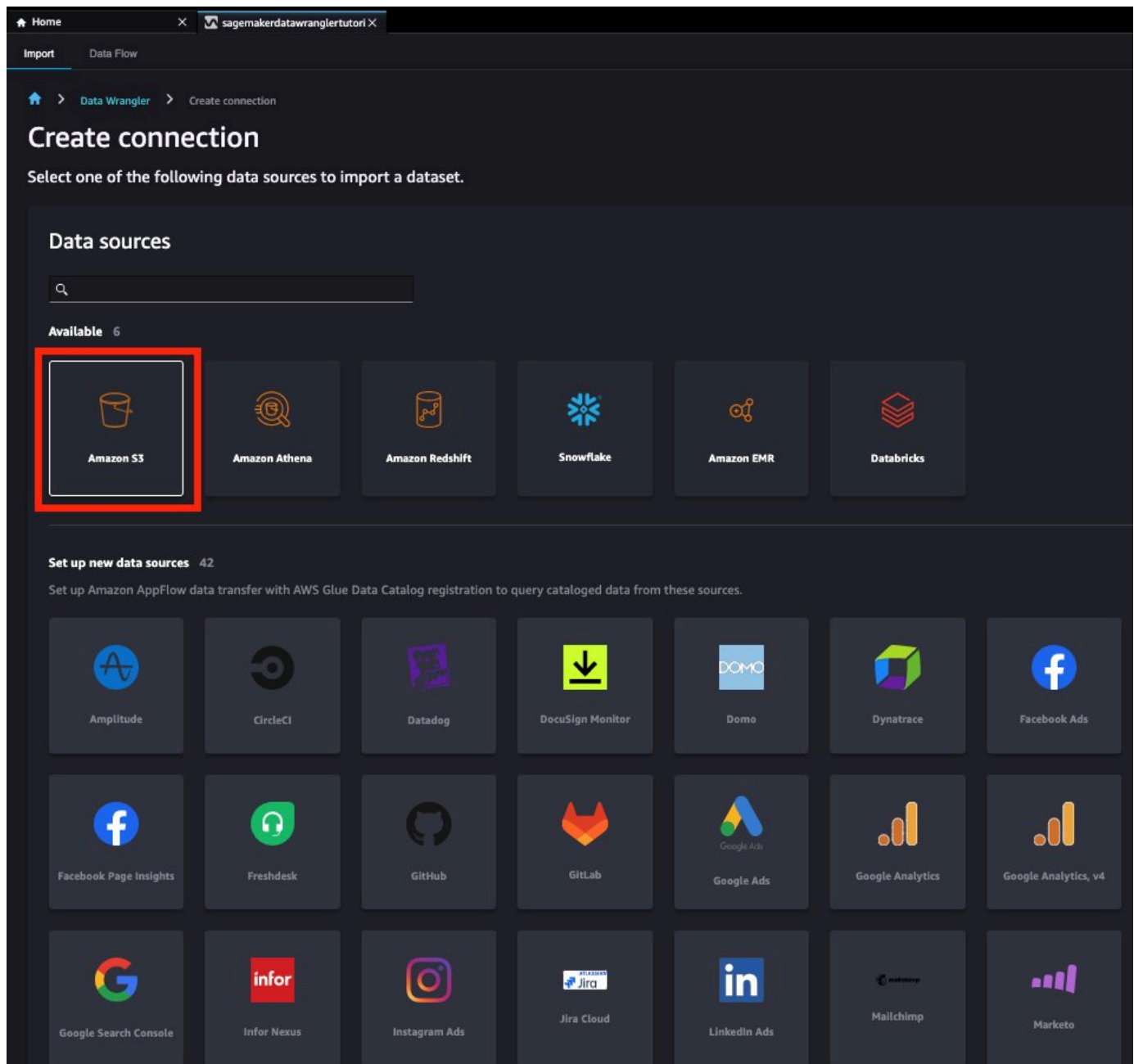
### 4. Choose instance type

Note that you can change the Flow's compute instance type using the upper right button showing the current Compute instance. You may decide to change the compute instance type based on your scenario's dataset size and can scale it up or down when your requirements change. For the purposes of this tutorial, you can use the default **ml.m5.4xlarge**.



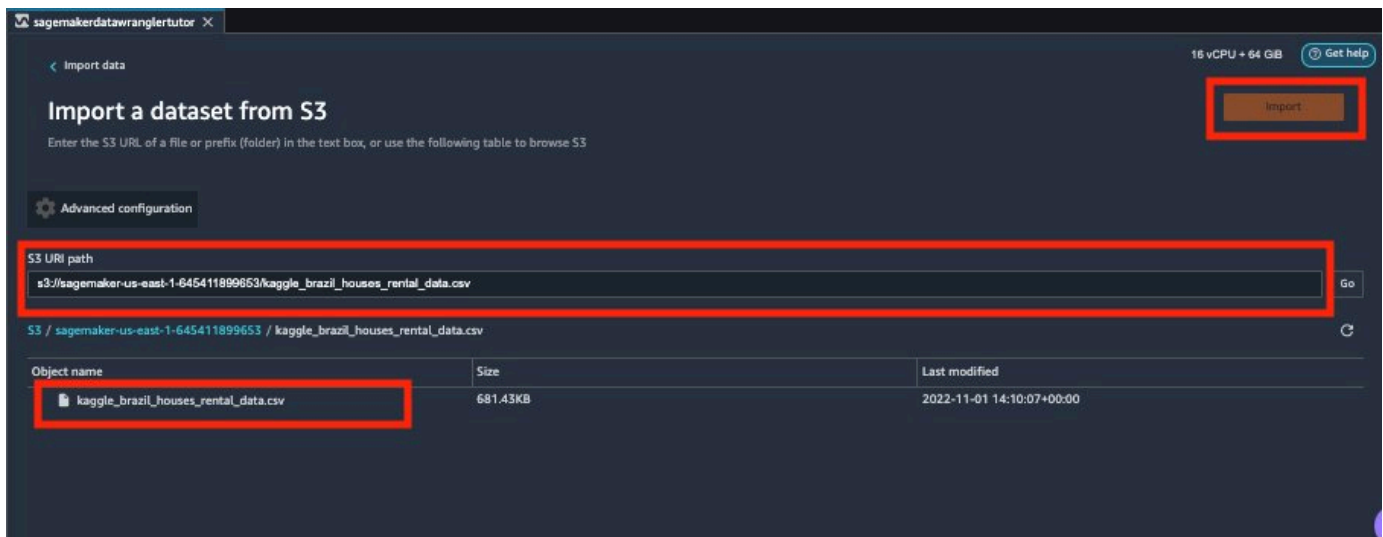
## 5. Import data from S3

In the **Data Import** tab, under **Import data**, choose **Amazon S3**.



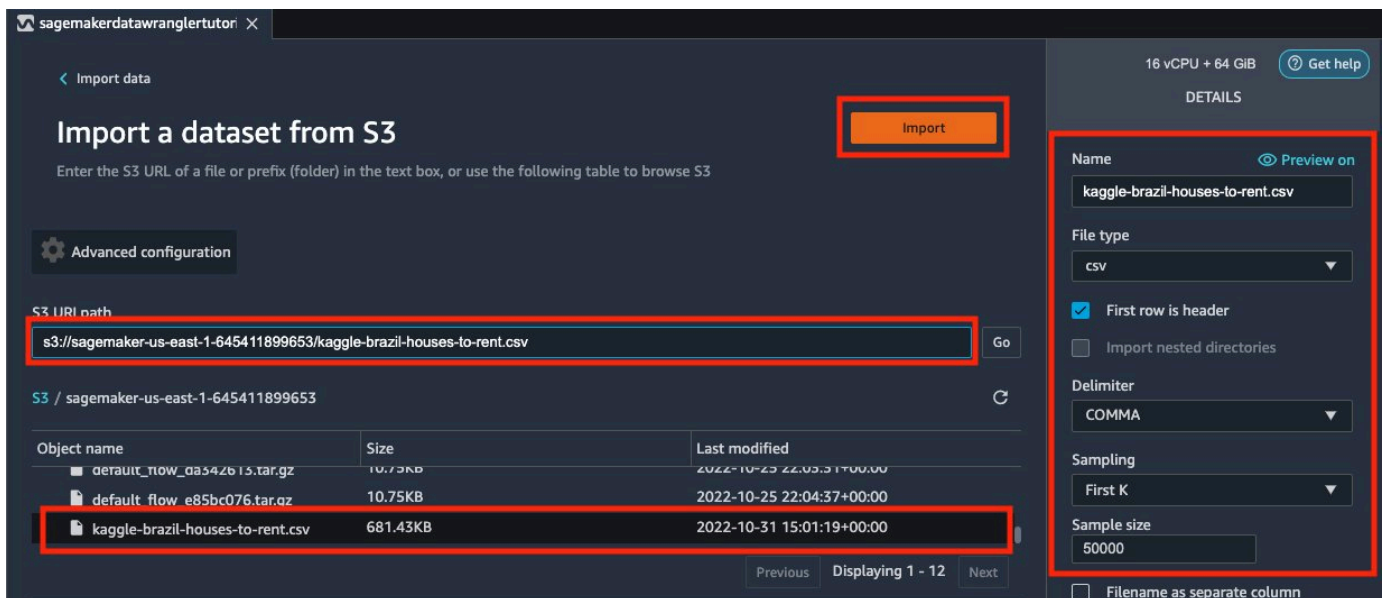
## 6. Specify S3 location

In the S3 URI Path field, enter **s3://sagemaker-sample-files/datasets/tabular/brazil\_houses/kaggle\_brazil\_houses\_rental\_data.csv**, and then choose **Go**. Under **Object name**, select **kaggle\_brazil\_houses\_rental\_data.csv**.



## 7. Import the dataset

In the S3 import details panel, note that you can change the default delimiter and the sampling method when necessary. For the purposes of this tutorial, you can use the default **comma delimiter** and **First K sampling method**. Then choose **Import**.

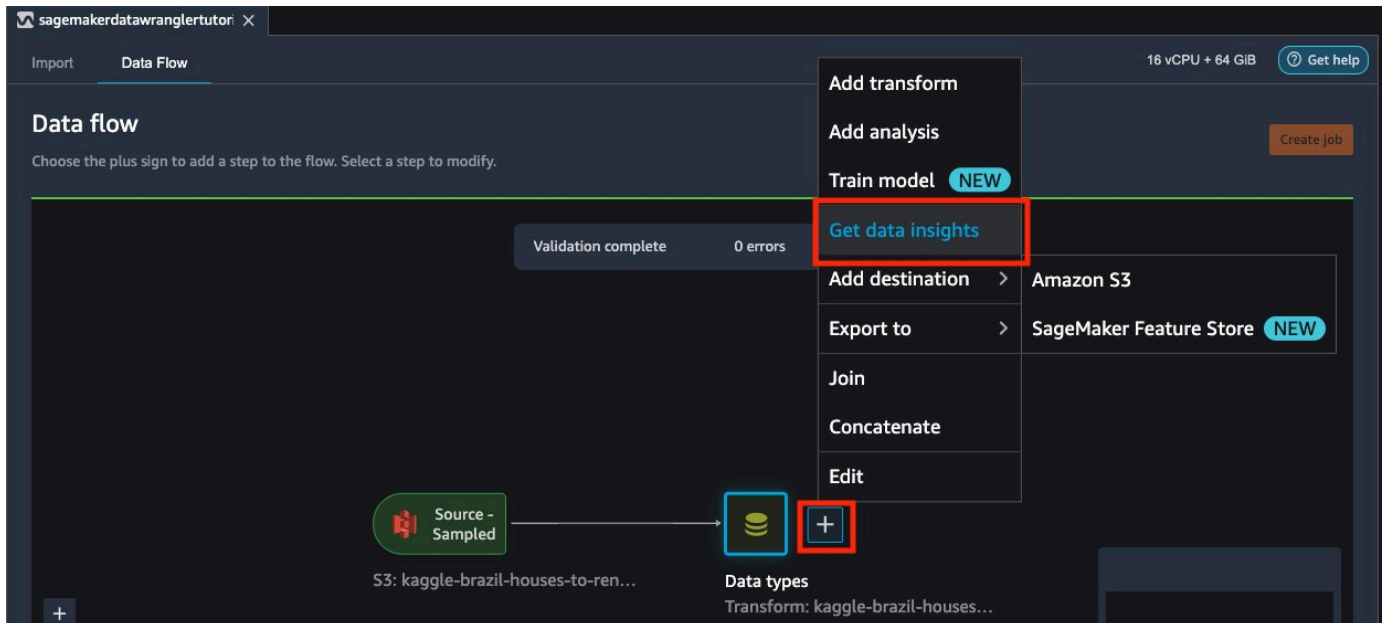


## Step 3: Explore the data

In this step, you use SageMaker AI Data Wrangler to assess and explore the quality of the training dataset for building machine learning models. Use the Data Quality and Insights report feature to understand your dataset quality, and then use the Quick Model feature to estimate the expected prediction quality and the predictive power of the features in your dataset.

## 1. Generate data insights

When exploring your dataset, begin by using the Data Quality and Insights report to help you quickly understand your dataset, identify possible issues, and focus your attention on the most important areas to improve the data. On the **Data flow** tab, in the data flow diagram, choose the **+** icon, then choose **Add analysis**. Then choose **Get data insights**.



## 2. Set analysis parameters

From the **Data Insights** pane, choose **rent amount** as the Target column. Then choose **Regression** as the **Problem type**. Then choose **Create**.

The screenshot shows the Amazon SageMaker Data Wrangler interface. On the left, a 'Data table' is displayed with columns: rent amount (R\$), property tax (R\$), fire insurance (R\$), and total (R\$). The table contains 15 rows of data. On the right, the 'Create analysis' panel is visible. It has a tab 'All analyses' and a 'Create analysis' section. In this section, the 'Analysis type' is set to 'Data Quality And Insights Report'. The 'Target column' is set to 'rent amount (R\$)'. Under the 'Optional' section, the 'Problem type' is set to 'Regression' (selected with a radio button). At the bottom right of the panel, there is a 'Create' button.

	rent amount (R\$)	property tax (R\$)	fire insurance (R\$)	total (R\$)
3300	211	42	5618	
4960	1750	63	7973	
2800	0	41	3841	
1112	22	17	1421	
800	25	11	836	
8000	834	121	8955	
1900	85	25	2750	
3223	1735	41	7253	
15000	250	191	16440	
2300	35	30	2955	
2100	150	27	2747	
580	43	8	1181	
2100	70	27	2556	
4200	224	54	5268	

### 3. View insights report

You may need to wait a minute while the report is generated. Once completed, review the Data Quality and Insights report sections to improve the dataset further before building the ML model. For this specific dataset, the Data Insights report has highlighted two possible issues: the first is related to **duplicate rows** in the dataset and the second is related to possible **target leakage** such that one feature is highly correlated with the output and may indicate a duplicate of the target **rent** column. The report can also be downloaded to a PDF file and shared with colleagues on your team.

The screenshot shows the Amazon SageMaker Data Wrangler interface. At the top, the title bar reads 'sagemakerdatawranglertutorial'. Below it, the 'Data flow' section shows 'Transform: kaggle-brazil-houses-to-rent.csv'. On the right, a table lists the target column, type, dataset, and date. A download icon is highlighted with a red box. The main content area is divided into two sections: 'SUMMARY' and 'High Priority Warnings'.

**SUMMARY**

Dataset statistics

Key	Value	Feature type	Count
Number of features	13	numeric	8
Number of rows	10692	categorical	2
Missing	0%	text	0
Valid	100%	datetime	0
Duplicate rows	5.65%	binary	2
		unknown	0

**High Priority Warnings**

2 high severity warnings were detected. See the list below.

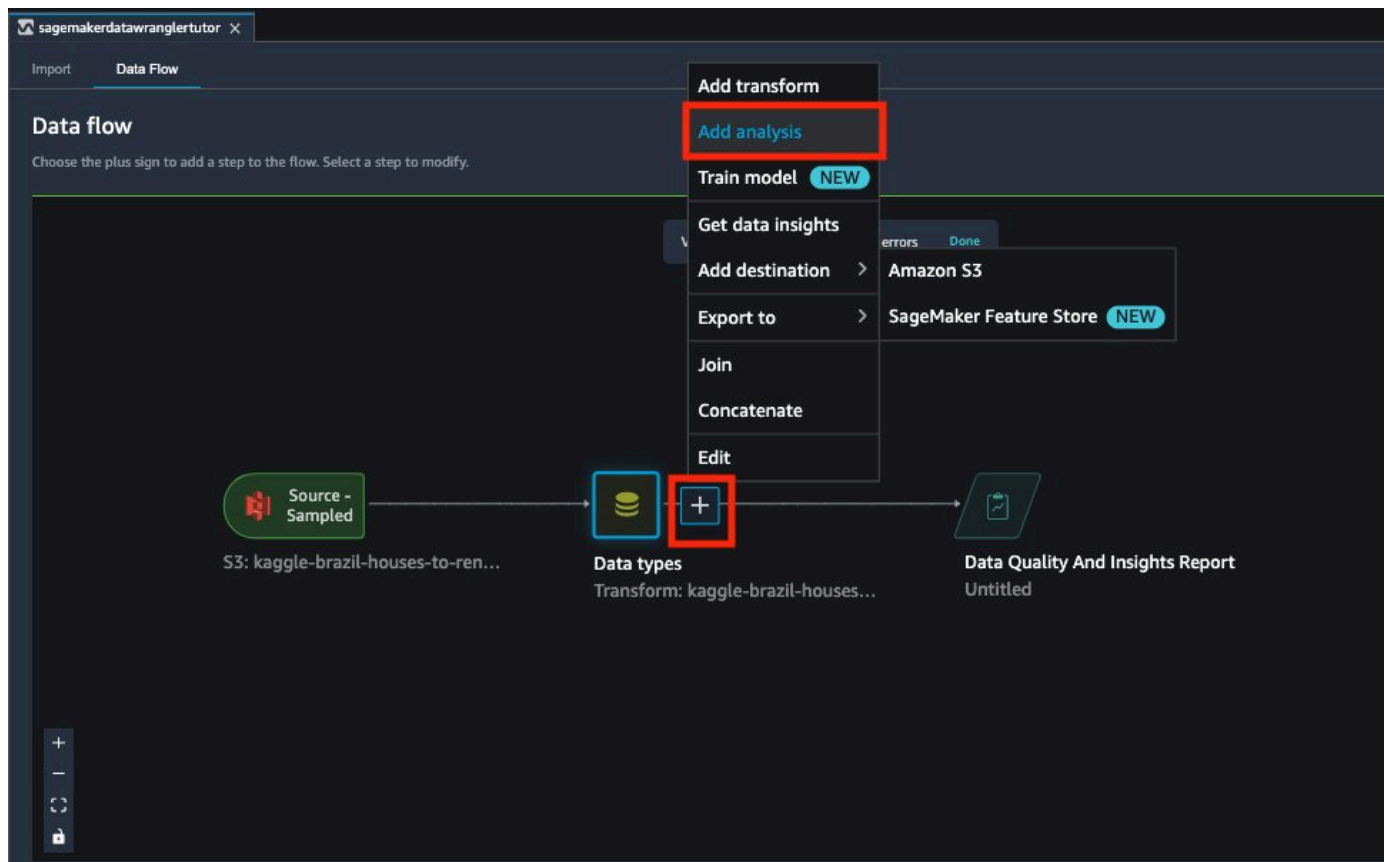
- Duplicate rows** (High)
 

We found that 5.65% of the data are duplicate. Some data sources could include valid duplicates and in other cases these duplicates could point to problems in data collection. Duplicate samples resulting from faulty data collection, could derail machine learning processes that rely on splitting to independent training and validation folds. For example quick model scores, prediction power estimation and automatic hyper parameter tuning. Duplicate samples could be removed from the dataset using the **Drop duplicates** transform under **Manage rows**.
- Target leakage** (High)
 

The feature **fire insurance (R\$)** predicts the target extremely well on its own. A feature this predictive often indicates an error called target leakage. The cause is typically data that is not available at time of prediction. For example, a duplicate of the target column in the dataset can result in target leakage. Alternatively, if the machine learning task is "easy", then a single feature can have legitimately high prediction power. If you think that a single feature is very highly predictive, you don't need to do anything further. However, if you think there's target leakage, we recommended that you remove the highly predictive column from the dataset using the **Drop column** transform under **Manage columns**.

#### 4. Create an analysis

For further data analysis and exploration, you can create additional analytical artifacts including correlation matrices, histograms, scatter plots, and summary statistics as well as custom visualizations. For example, choose the **+ icon**, then choose **Add analysis**.



## 5. Create a histogram

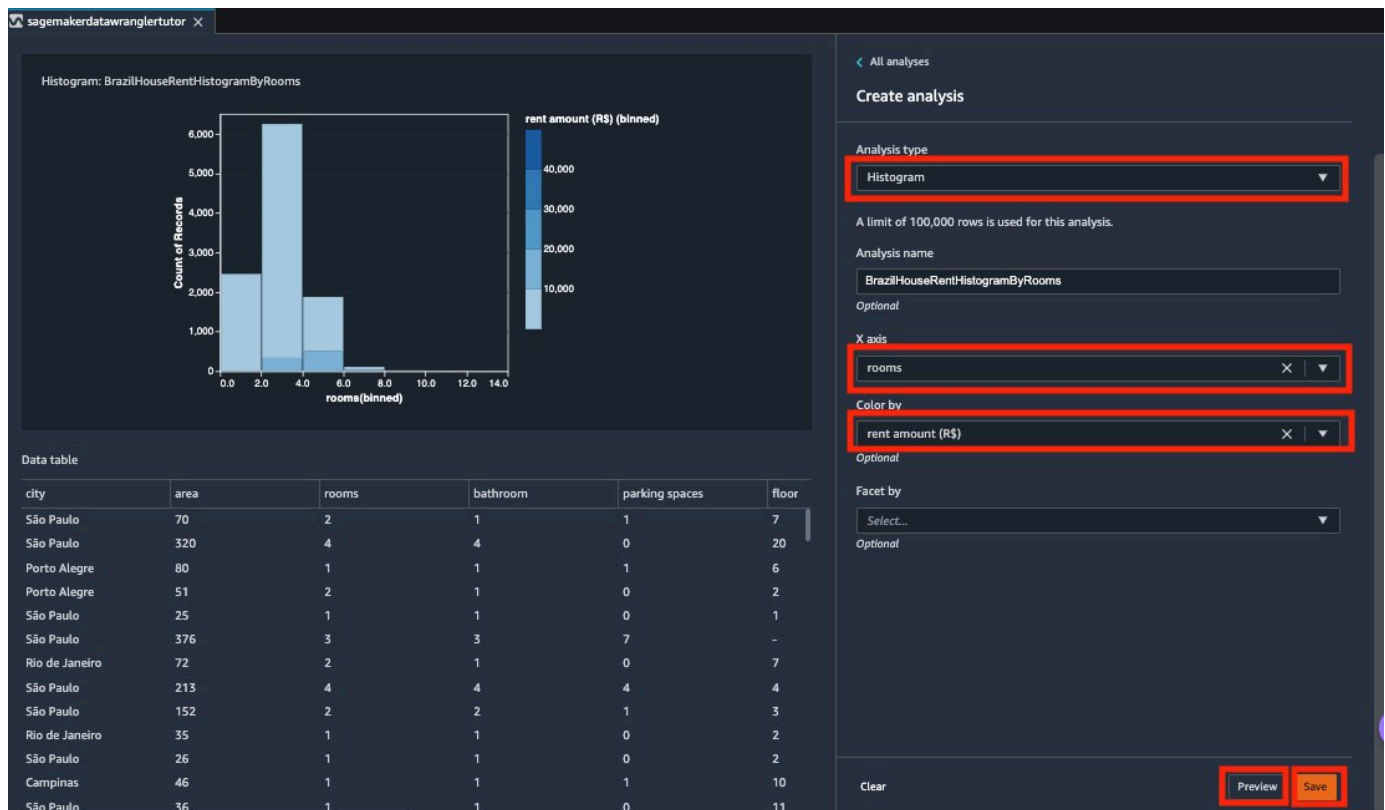
Under the **Create analysis** panel, for **Analysis type**, select **Histogram** and name it **RentHistogramByRooms**. For **X axis**, select **rooms**.

For **Color by**, select **Rent amount**.

Choose **Preview** to generate a **histogram** of the **rent amount** field, color-coded by the **rooms** variable.

Choose **Save** to save this analysis to the data flow.





## 6. Create a Quick Model

Next, to gain higher confidence that the underlying data has some predictive power, we are going to create a Quick Model. Under the **Create analysis** pane, for **Analysis type**, choose **Quick Model** and name it **RentQuickModel**.

Then for **Label**, select **rental amount** and then choose **Preview**.

The **Quick Model** may take several minutes to complete, then the pane shows a brief overview of the Random Cut Forest model built and trained with default hyperparameters. The model generated also displays some statistics, including the Mean Square Error (MSE) score and feature importance to help you evaluate the quality of the dataset. Choose **Save**.

sagemakerdatawranglertutorial X

Data types · Transform: kaggle-brazil-houses-to-rent.csv

Data Analysis Training **NEW**

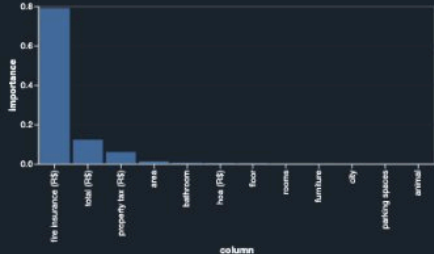
Quick Model: BrazilHouseRentQuickModel

We train a random forest with 10 trees on 7514 observations and measure prediction quality on the remaining 3178 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 mse of 6.17e+05 on the test set.

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



Data table

city	area	rooms	bathroom	parking spaces	floor	animal
São Paulo	70	2	1	1	7	accept
São Paulo	320	4	4	0	20	accept
Porto Alegre	80	1	1	1	6	accept
Porto Alegre	51	2	1	0	2	accept
São Paulo	25	1	1	0	1	not accept
São Paulo	376	3	3	7	-	accept
Rio de Janeiro	72	2	1	0	7	accept
São Paulo	213	4	4	4	4	accept
São Paulo	152	2	2	1	3	accept
Rio de Janeiro	35	1	1	0	2	accept
São Paulo	26	1	1	0	2	accept
Campinas	46	1	1	1	10	accept
São Paulo	36	1	1	0	11	accept
São Paulo	55	1	1	1	2	accept
São Paulo	100	2	2	2	24	accept
Campinas	330	4	6	6	-	accept
São Paulo	110	2	2	1	1	accept
Rio de Janeiro	88	2	3	1	9	not accept
Rio de Janeiro	56	2	1	0	8	accept

All analyses

Create analysis

Analysis type: Quick Model

Analysis name: BrazilHouseRentQuickModel

Optional

Label: rent amount (R\$)

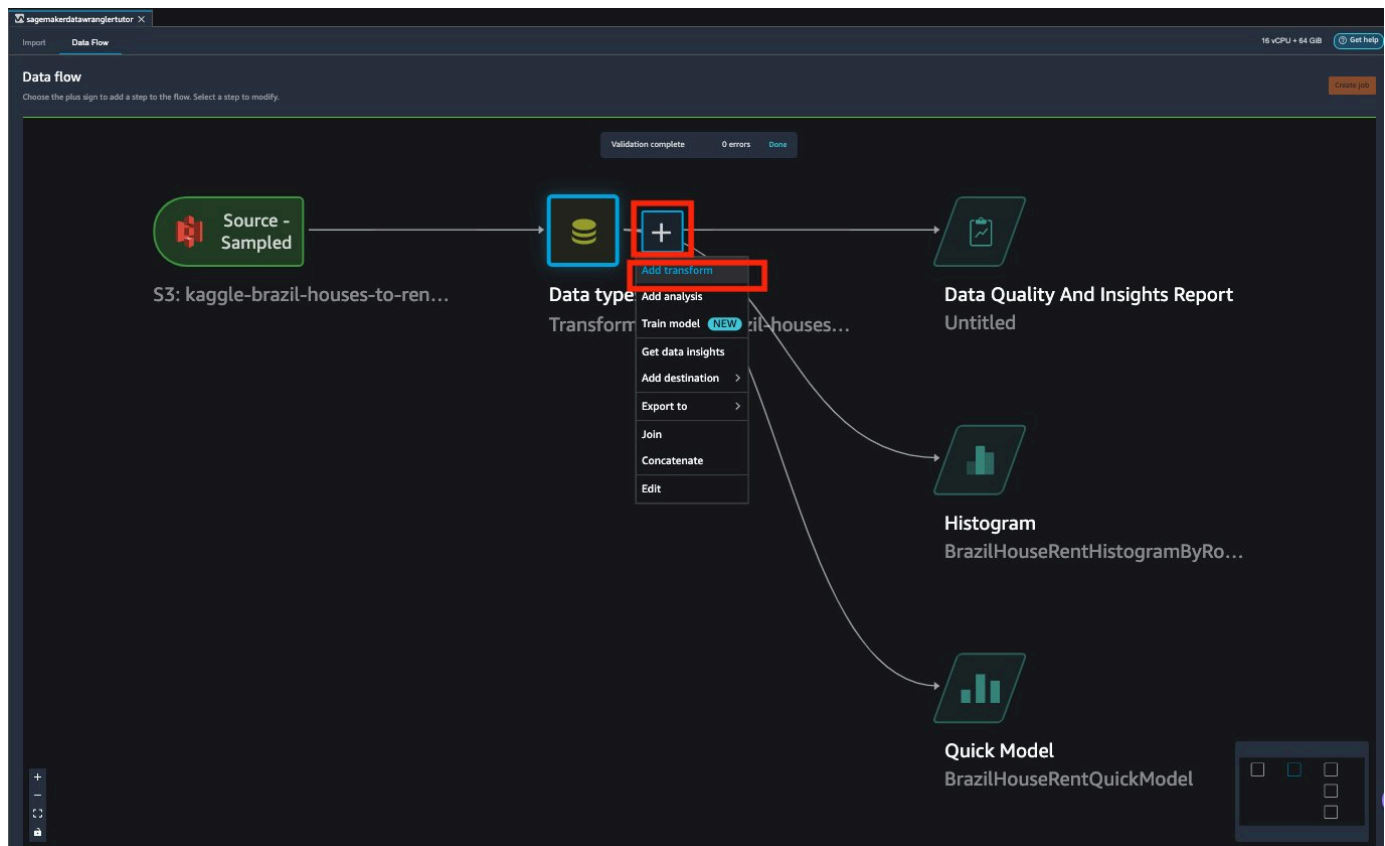
Clear Preview Save

## Step 4: Add transformations to the data flow

SageMaker AI Data Wrangler simplifies data processing by providing a visual interface with which you can add a wide variety of pre-built transformations. You can also write your custom transformations when necessary using SageMaker AI Data Wrangler. In this step, you change the type of a string column, rename columns, and drop unnecessary columns using the visual editor.

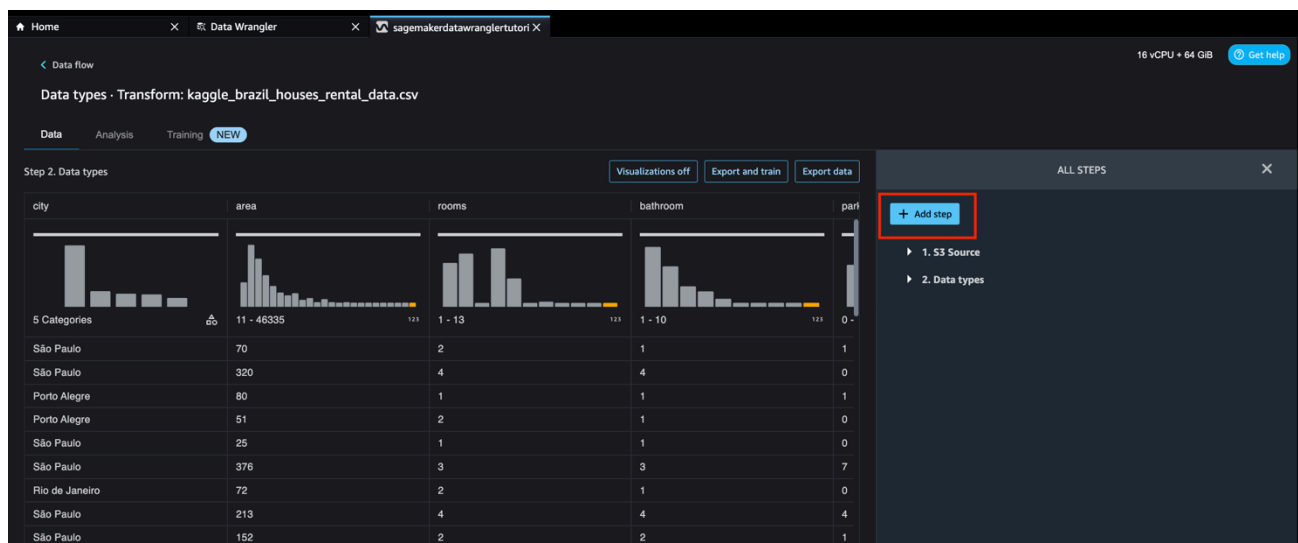
### 1. Open Data Wrangler flow

To navigate to the data flow diagram, choose **Data flow**. On the data flow diagram, choose the **+ icon**, then **Add transform**.



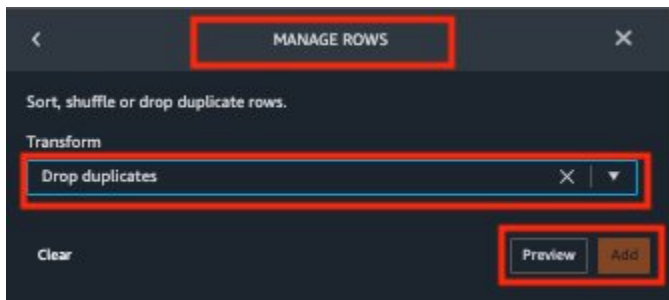
## 2. Add a transformation

Under the **ALL STEPS** pane, choose **Add step**.



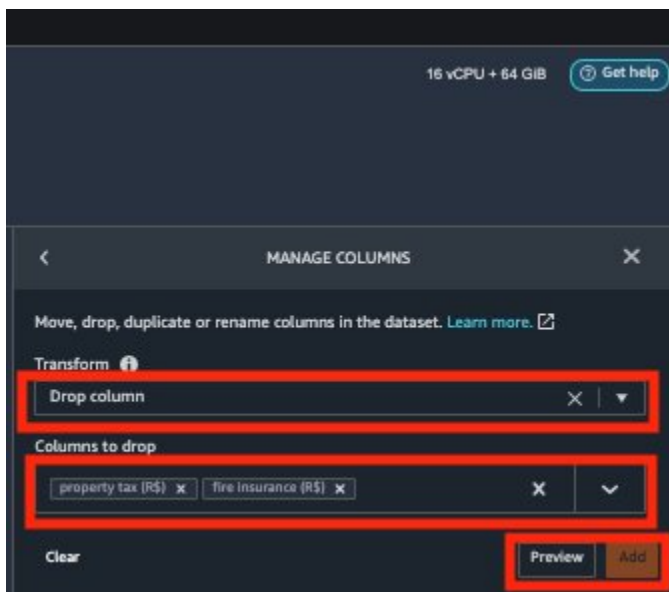
## 3. Remove duplicates

The first step is following the Data Insights Report recommendations regarding high risk items and removing the duplicate rows. So as the first transform step, choose **Manage Rows**, and then select the **Drop duplicates** operation. Then choose **Preview** and **Save**.



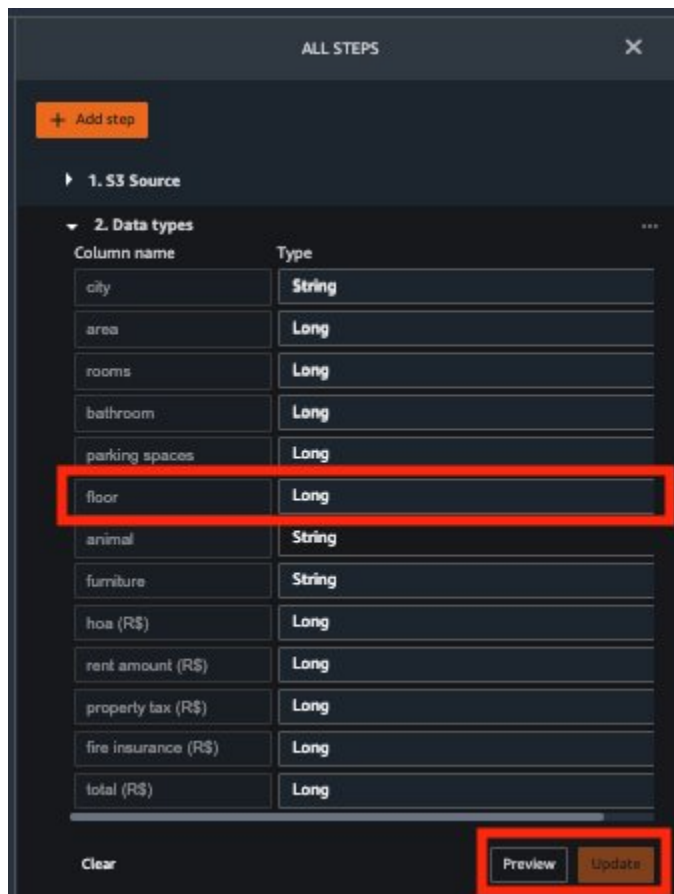
#### 4. Select columns to drop

Second, we are going to remove the dataset features highlighted as possible sources of target leakage and not appropriate for a machine learning model predicting the rental amount. From the **ADD TRANSFORM** list, choose **Manage columns**. Then choose **Drop column** and choose **property tax** and **fire insurance**. Choose **Preview** then **Save**.



#### 5. Change column type

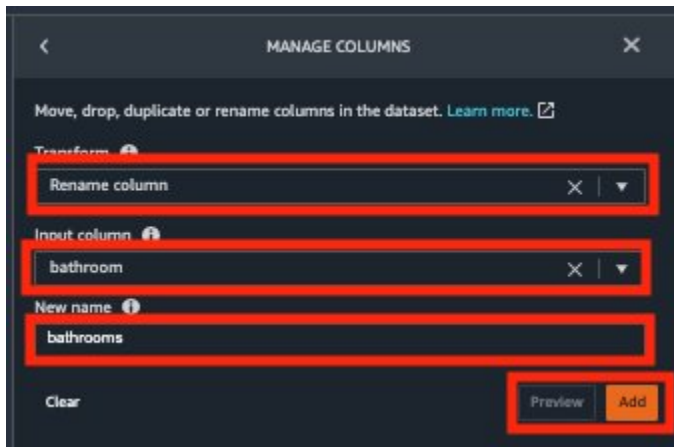
Next, change the data type of the **floor** column from **string** to **long**. Machine learning models can benefit from using numerically typed columns and this step will allow us to perform further processing later on.



## 6. Update column names

Then rename several columns to improve the readability of the input data set and later analysis.

From the **ADD TRANSFORM** list, choose **Manage columns**. Then choose **Rename column**. Then choose **bathroom** as the input column and **bathrooms** as the output column. Choose **Preview** then **Save**. Repeat this renaming column process for **hoa** [originally from **hoa (R\$)**], **rent** [originally from **rent amount (R\$)**], and **total** [originally from **total (R\$)**].



## Step 5: Add categorical encoding and numeric scaling transformations to data flow

In this step, you encode categorical variables and scale numerical variables. Categorical encoding transforms string data type categories into numerical features. It's a common preprocessing task because the numerical features can be used in a wide variety of machine learning model types.

### 1. Configure encoding

In the dataset, the rental property's **animal** and **furniture** classification is represented by various strings. In this step, you convert these string values to a binary representation, 0 or 1.

Under the **ALL STEPS** pane, choose **+ Add step**. From the **ADD TRANSFORM** list, choose **Encode categorical**. SageMaker AI Data Wrangler provides three transformation types: Ordinal encode, One hot encode, and Similarity encode.

Under the **ENCODE CATEGORICAL** pane, for **Transform**, use the default **Ordinal encode**. For **Input** columns, select **animal** and **furniture**. Ignore the **Invalid handling strategy** box for this tutorial. Choose **Preview**, then **Add**.

ENCODE CATEGORICAL

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

Transform ⓘ

Ordinal encode

Input columns ⓘ

animal x furniture x

Output column ⓘ

Optional

Invalid handling strategy ⓘ

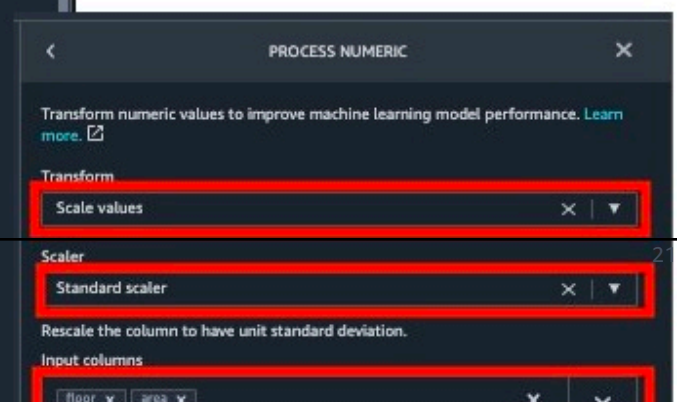
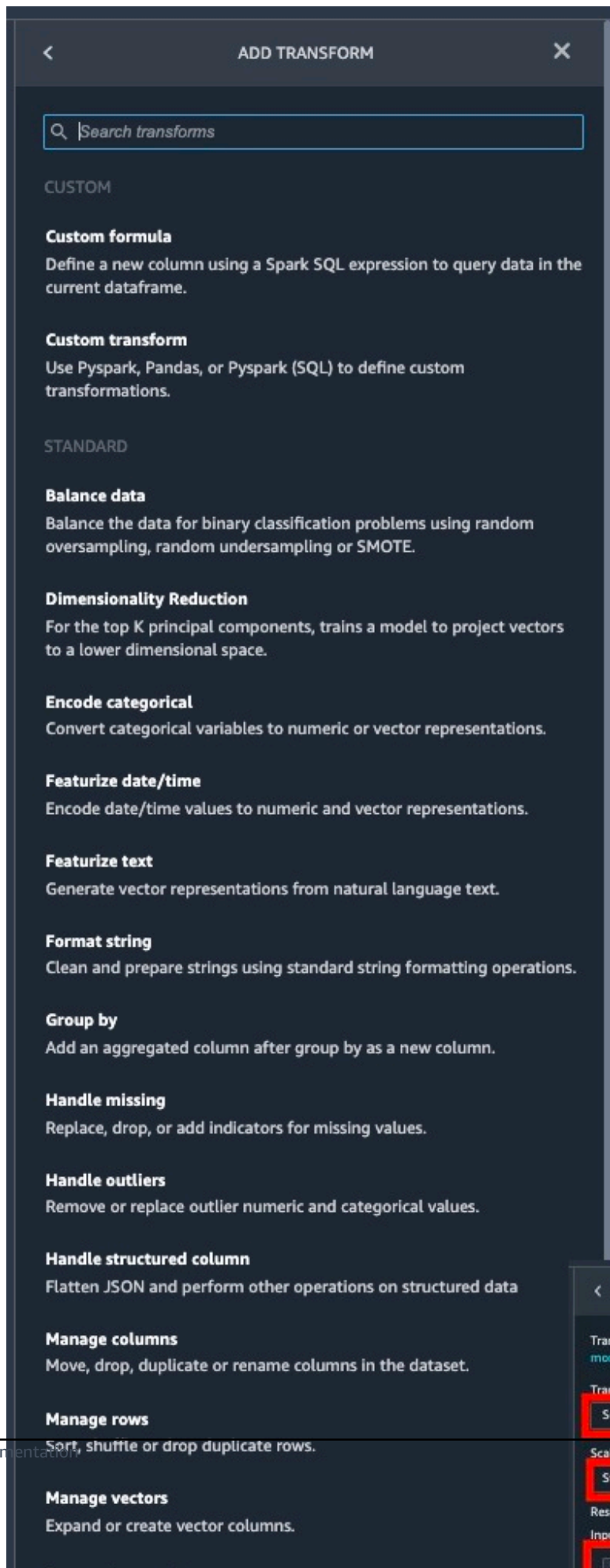
Replace with NaN

Clear Preview Add

## 2. Configure scaling

To scale the numerical columns `area` and `floor`, apply a scaler transformation to normalize the distribution of the data in these columns:

Under the **ALL STEPS** pane, Choose **+ Add step**. From the **ADD TRANSFORM** list, choose **Process numeric**. For **Scaler**, select the default option **Standard scaler**. For **Input** columns, select **area** and **floor**. Choose **Preview**, and then **Add**.

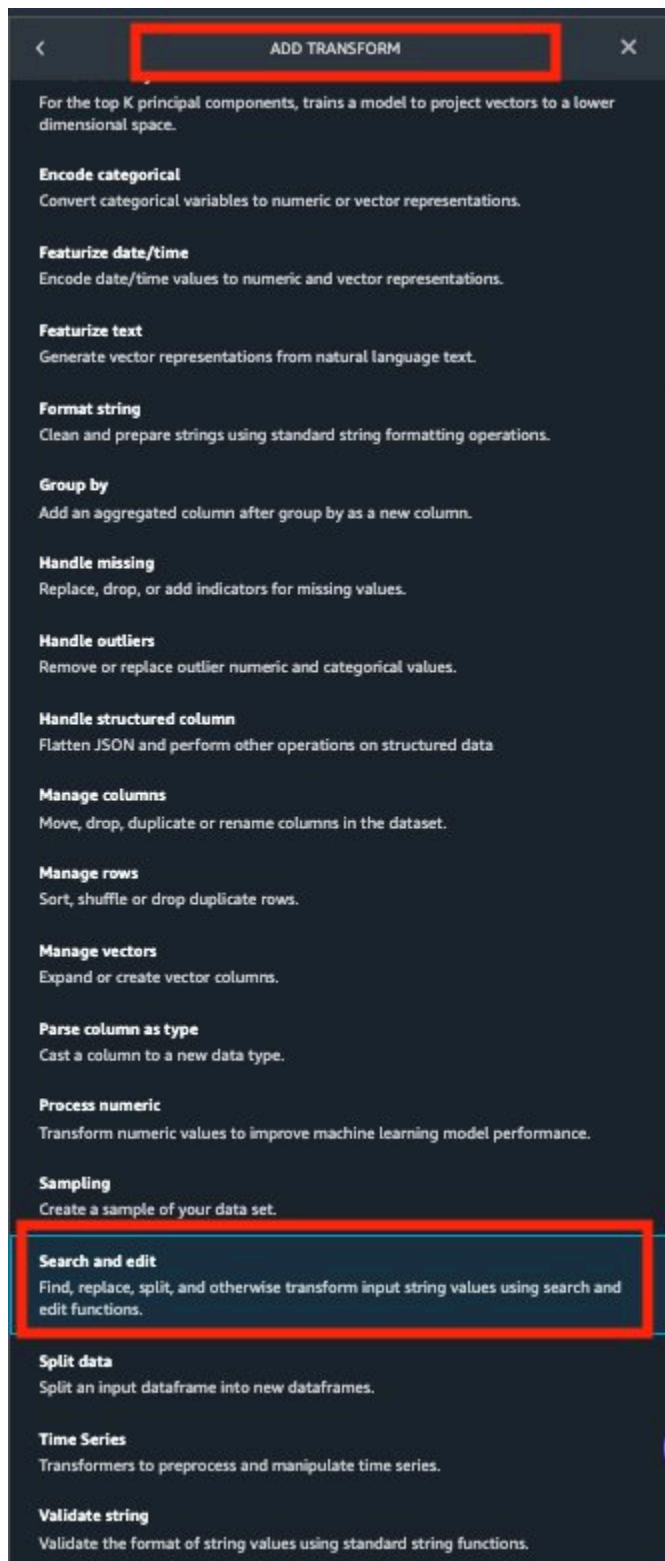




### 3. Choose transformation type

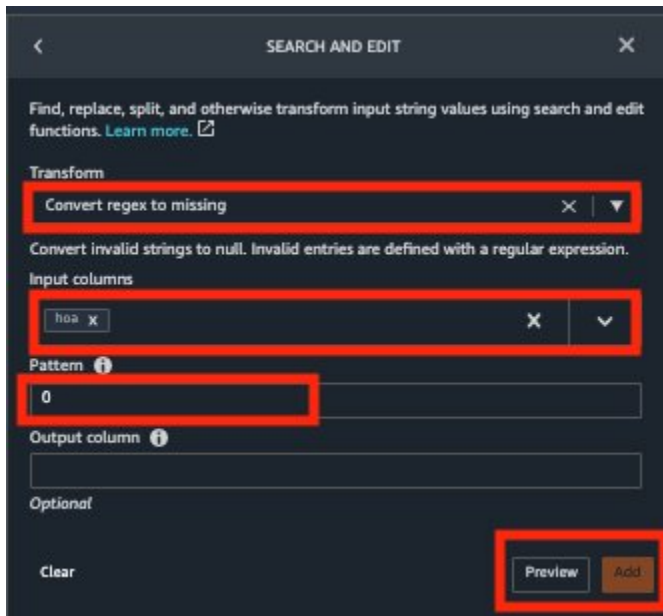
Finally, we will follow another recommendation from the Data Insight report and replace the 0s in the Home Owner Association (hoa) feature with **NaN** because they indicate missing data and should not be treated as valid inputs that might skew the model.

Under the **ALL STEPS** pane, choose **+ Add step**. From the **ADD TRANSFORM** list, choose **Search and edit**.



#### 4. Replace zero values

Choose **Convert regex to missing**. Choose **hoa** as the **Input** column, specify **0** as the **Pattern**. Click **Preview**, and then choose **Add**.



## Step 6: Check for data bias

In this step, check your data for bias using Amazon SageMaker AI Clarify, which provides you with greater visibility into your training data and models so you can identify and limit bias and better explain predictions.

### 1. Create a bias report

Choose **Data flow** in the upper left to return to the data flow diagram. Choose the **+ icon**, **Add analysis**.

In the **Create analysis** pane, for **Analysis type**, select **Bias Report**.

For **Analysis name**, enter **RentalDataBiasReport**.

For **Select the column your model predicts (target)**, select **rent**. Then select **Threshold** as the predicted column type since this is a regression problem.

Specify **3000** as the **predicted threshold** which corresponds to the average of the **rent** column in the dataset. Then select **city** as the column to analyze for bias because we are interested in whether the dataset is imbalanced and over-represents some cities instead of others.

Then for **Choose bias metrics**, keep the default selections. Then choose **Check for bias** and then **Save**.

< All analyses

## Create analysis

Analysis type

Bias Report

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

Analysis name

RentalDataBiasReport

Optional

Select the column your model predicts (target)

rent

Is your predicted column a value or threshold?

☐ Value ☒ Threshold

Predicted threshold

3000

Select the column to analyze for bias

city

Is your column a value or threshold?

☒ Value ☐ Threshold

Column value(s) to analyze for bias

Enter column value(s)

Optional

Choose bias metrics

☒ Class imbalance (CI)

☒ Difference in Positive Proportions in Labels (DPL)

☒ JS divergence (JS)

☐ Conditional Demographic Disparity in Labels (CDDL)

To measure CDDL, select a column in the dataset to be used as the group variable.

Select...

Optional

Would you like to analyze additional metrics?

☐ Yes ☒ No

Clear

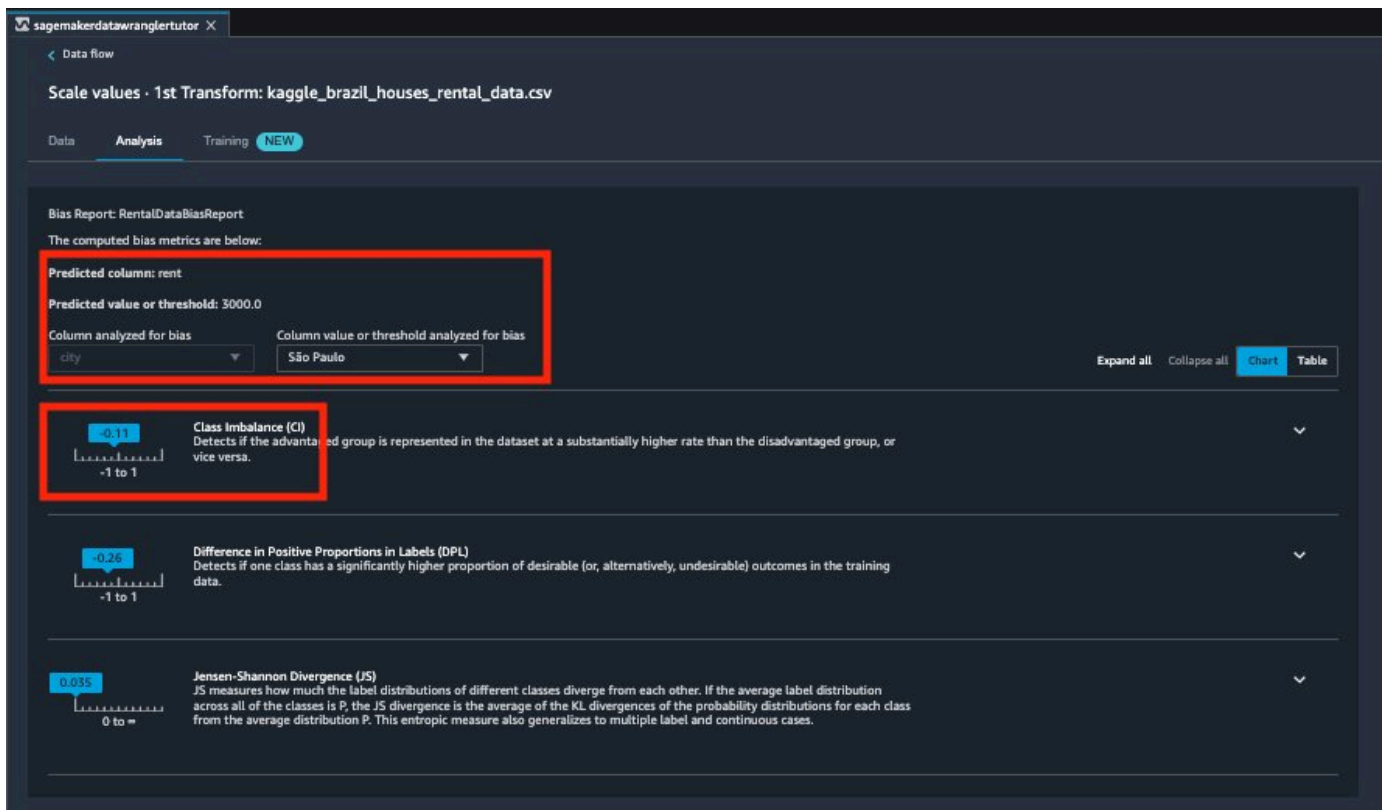
Check for bias Save

## 2. Review bias metrics

After several seconds, SageMaker AI Clarify generates a report, which shows how the target and feature columns score on a number of bias-related metrics including Class Imbalance (CI) and Difference in Positive Proportions in Labels (DPL).

In this case, the data is slightly biased with regards to rents in Sao Paulo (-0.11), and increasingly skewed for the cities of Rio de Janeiro (0.72), Belo Horizonte (0.77), and Porto Alegre (0.78).

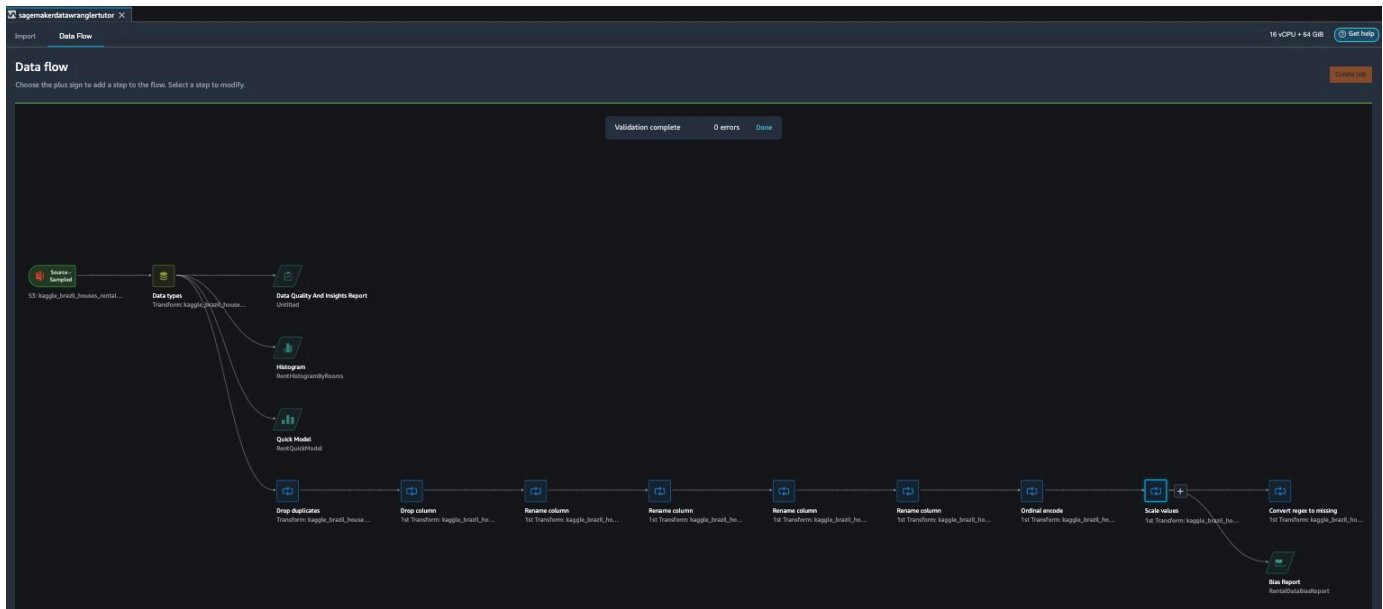
Based on this report, you might consider a bias remediation method, such as using SageMaker AI Data Wrangler's built-in SMOTE transformation. For the purpose of this tutorial, skip the remediation step. Choose **Save** to save the bias report to the data flow.



## Step 7: Review, integrate, and export your data flow

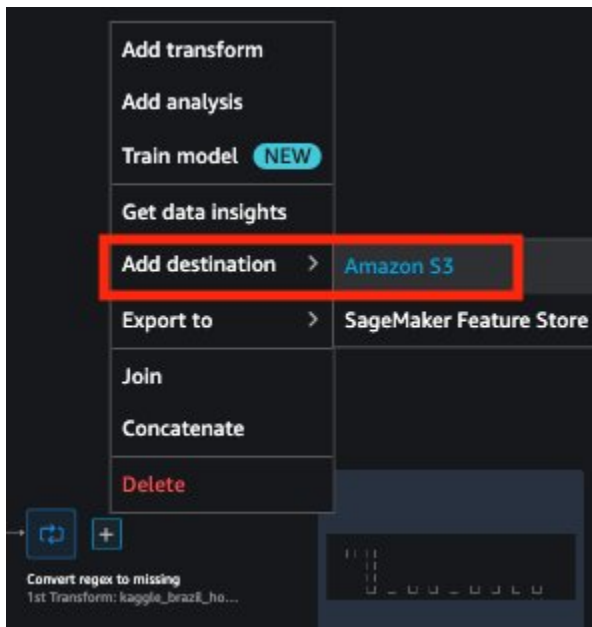
### 1. View your data flow

From the **Data Flow** tab, review your end-to-end data flow graph including the data source, analytical artifacts, and data transformations. You can easily navigate, view, modify, and delete data flow steps iteratively.



## 2. Export to Amazon S3

Data Wrangler further streamlines the automation process of exporting the output of the data flow to a persistent destination and can orchestrate the schedule of the flow's execution. First, set the storage destination to Amazon S3.



## 3. Specify output settings

Then specify the output dataset name (**kaggle\_brazil\_houses\_rental\_data\_dw\_processed.csv**) and the Amazon S3 location as your preferred S3 bucket. Then choose **Add destination**.



**Add a destination** ✕

**Amazon S3**

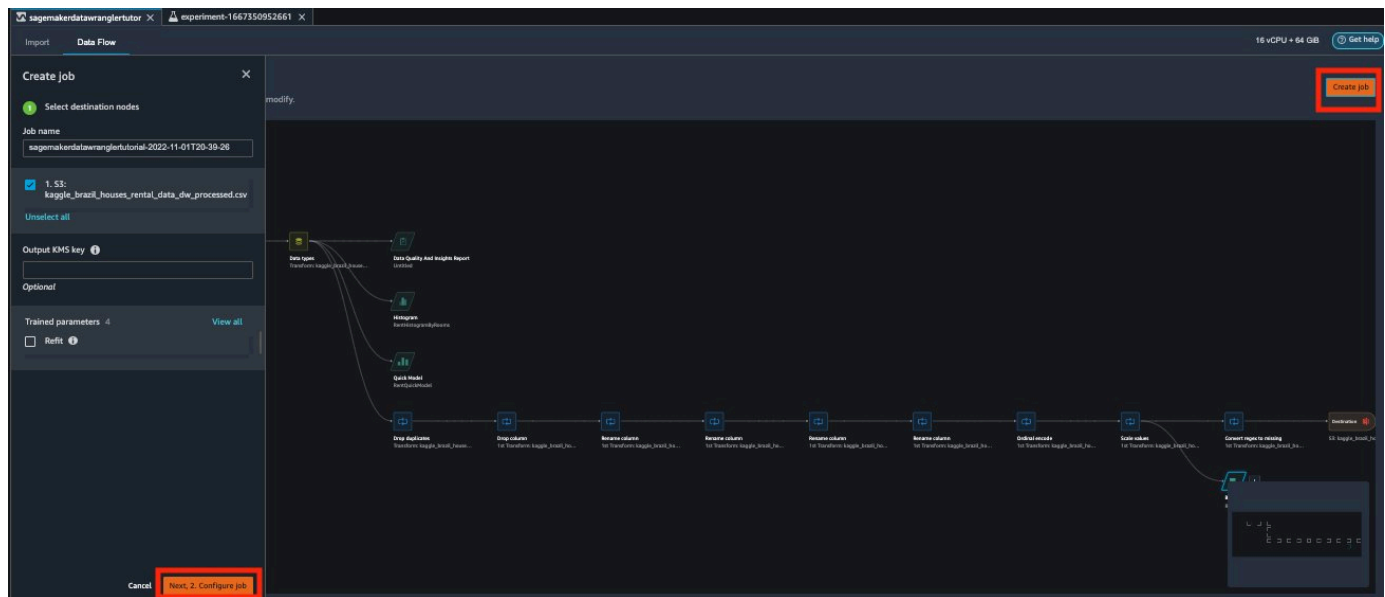
Dataset name

File type:  Delimiter:  Compression:

Amazon S3 location

#### 4. Create job

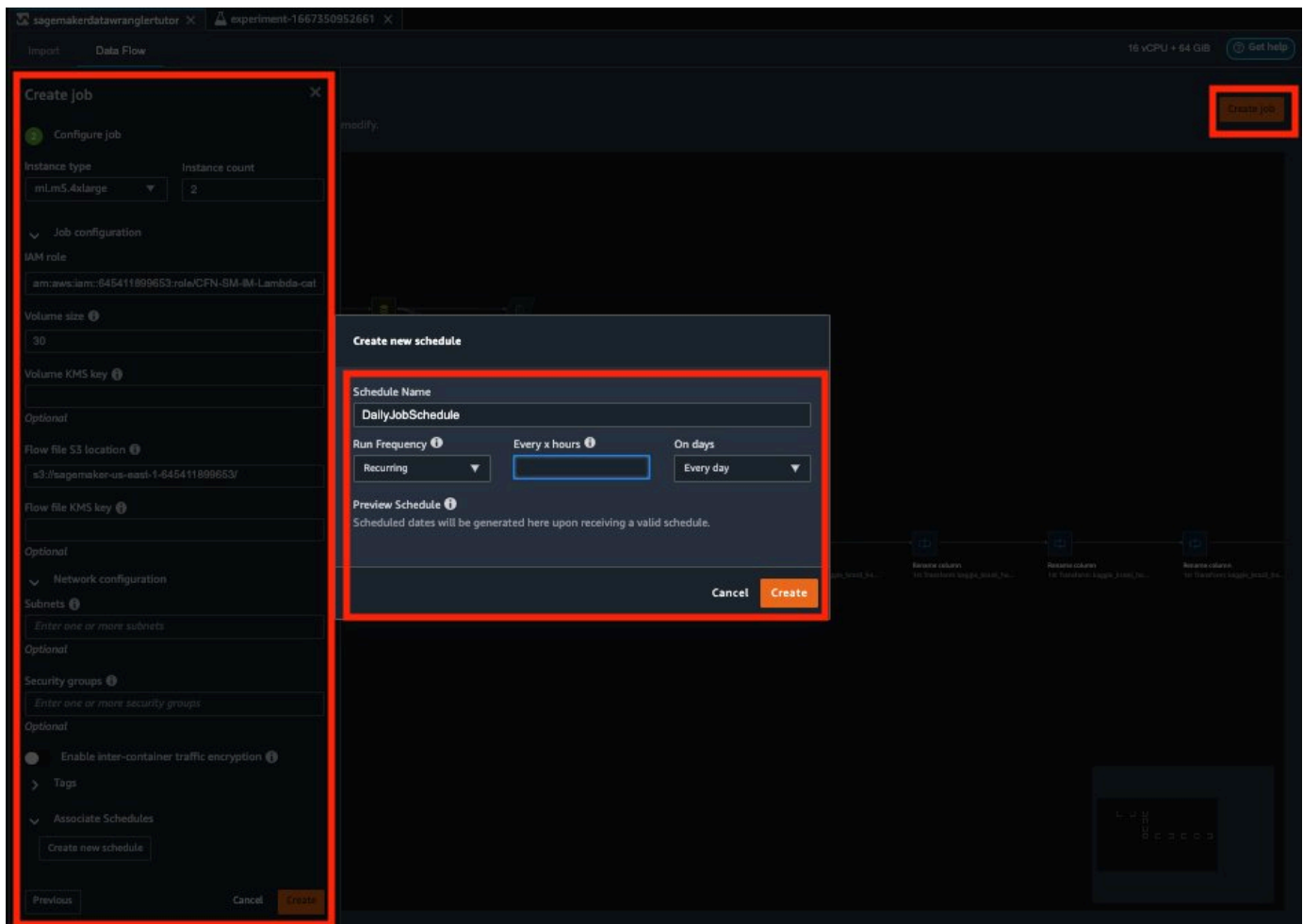
Lastly, create the scheduled job that will export the data flow output to Amazon S3 by choosing the **Create job** button from the **Data Flow** diagram pane, and then choosing **Configure job**.



## 5. Configure job

Then you can decide on the job instance type, instance count, the job's IAM security role, and the job schedule.



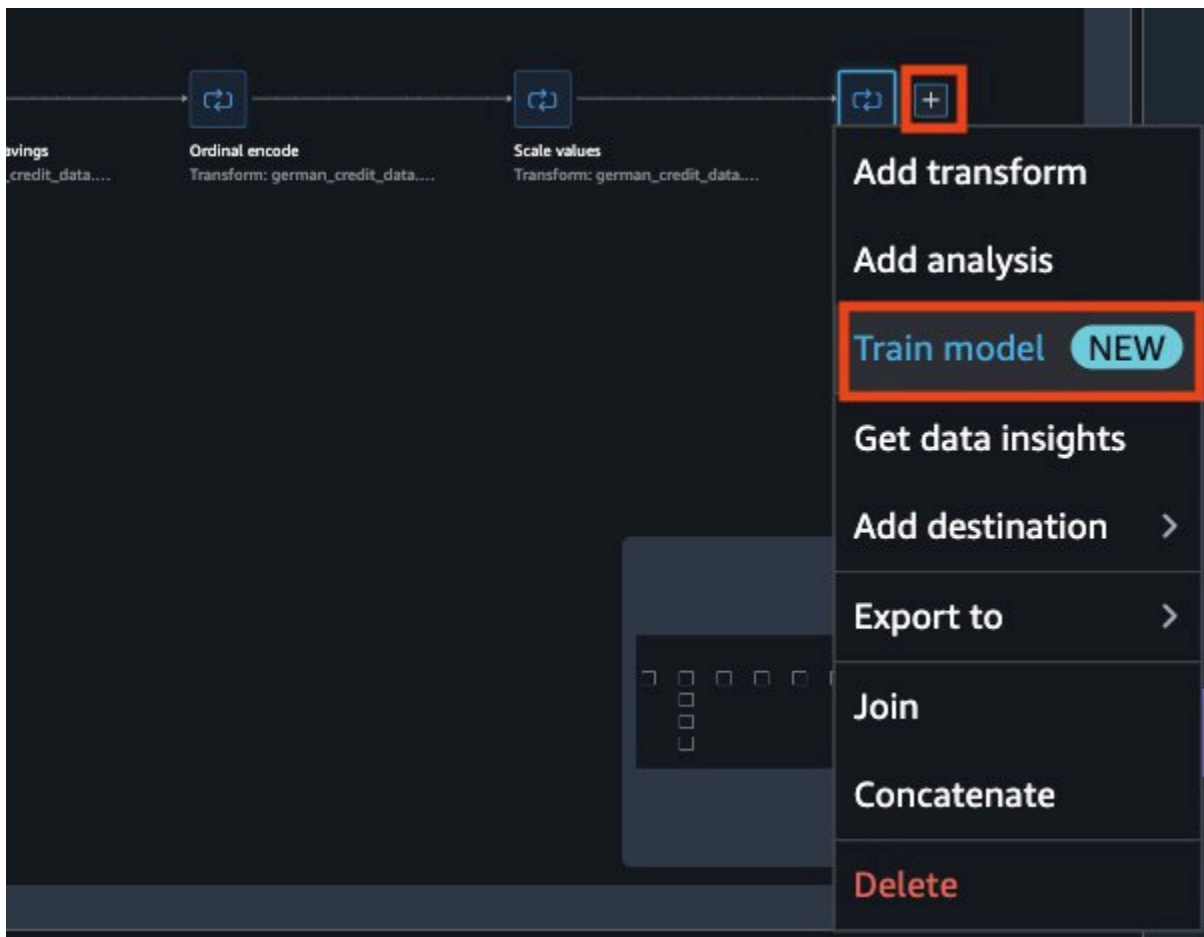


## Step 8: SageMaker AI Autopilot integration

You can also integrate your data flow with [SageMaker AI Autopilot](#) which automates key tasks of training and deploying a machine learning model.

### 1. Open model training

From the **Data Flow** tab, choose the **+ icon** and then choose **Train model**.



## 2. Configure and start Autopilot experiment

Choose **Export and Train** to export the Data Wrangler flow and associate its output with the Autopilot Experiment input.

Choose the **S3 location** where the Data Wrangler flow saved the processed input dataset and specify the **target** column as **rent** for the Autopilot model.

Specify the Autopilot **Training method**. You can choose Ensembling, Hyperparameter Optimization, or Auto. For the purposes of this tutorial, choose **Auto**.

For **Deployment**, select the machine learning problem type as **Regression** with the object metric as **MSE**.

Confirm the Autopilot Experiment deployment settings and then choose **Create experiment**.

This action launches a SageMaker AI Autopilot job that inspects the input data, generates and evaluates multiple ML models, and then selects the best model for subsequent deployment according to the desired performance metric (such as MSE in this tutorial).

The Autopilot job may take several minutes to run and complete. Autopilot provides full visibility into how the models were selected, trained, and tuned through a visual leaderboard and programmatic APIs. Finally, Autopilot explains how models make predictions using feature attribution and explainability statistics using SageMaker AI Clarify.

The screenshot shows the SageMaker AI Autopilot job interface for Experiment-1667350952661. The problem type is Regression. The best model endpoint is Experiment-1667350952661-7c9ada956c654fc3a47131e2621a59c9. The interface displays a table of models with columns for Model name, Objective: Mse, MAE, RMSE, R2, and Algorithm. The best model is Experiment-16673509526619b15sg8l-010-d83aa984 with an MSE of 328196.75. A 'Deploy model' button is visible on the right.

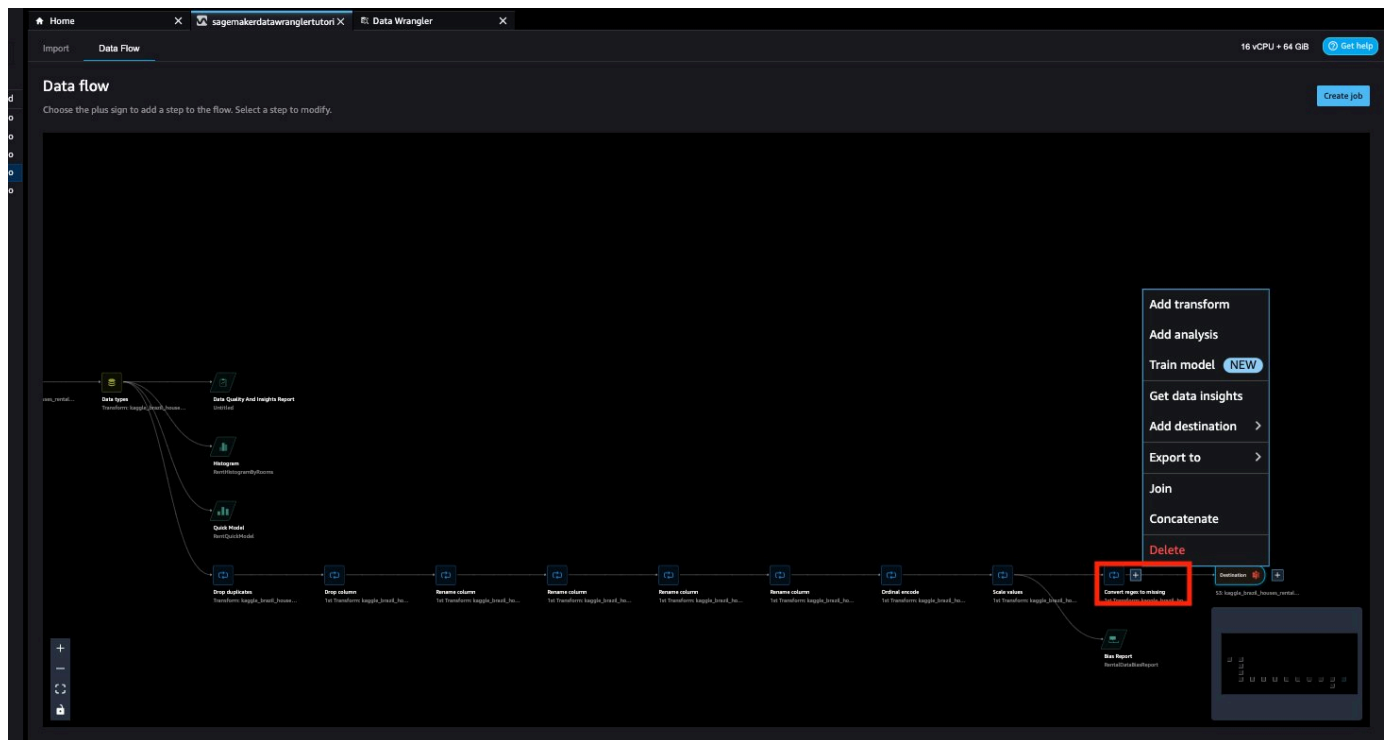
Model name	Objective: Mse	MAE	RMSE	R2	Algorithm
Experiment-16673509526619b15sg8l-010-d83aa984	328196.75	184.006	559.49	0.973	XGBoost
Experiment-16673509526619b15sg8l-022-92...	366718048				
Experiment-16673509526619b15sg8l-044-3f...	340098080				
Experiment-16673509526619b15sg8l-031-85...	325408864				
Experiment-16673509526619b15sg8l-026-f5...	320702976				
Experiment-16673509526619b15sg8l-024-97...	310899936				
Experiment-16673509526619b15sg8l-037-a8...	164719440				

## Step 9: SageMaker AI Pipeline integration

Data Wrangler can also be integrated with SageMaker AI Inference Pipelines to process data at the time of inference, thereby streamlining the steps between data processing and model inference. When you export one or more steps from the data flow to an inference endpoint, Data Wrangler creates a Jupyter notebook that you can use to define, instantiate, customize, run, and manage the inference pipeline.

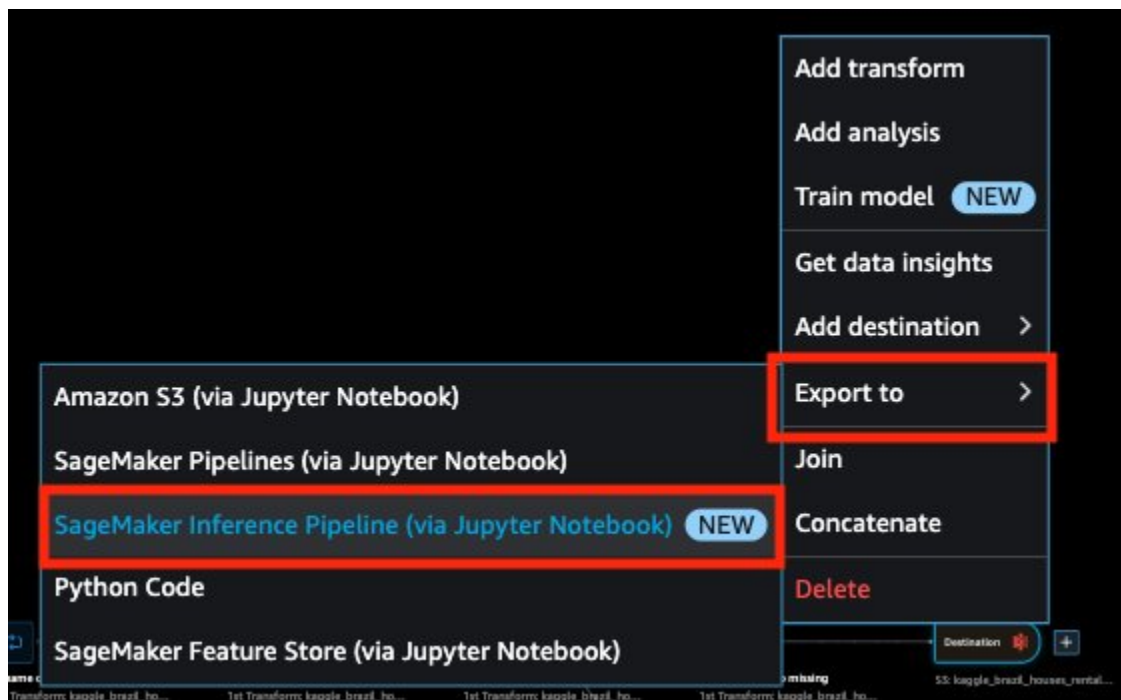
### 1. Create an inference endpoint

To create the inference endpoint, choose the **+** next to the final transformation step (Convert regex to missing) and choose **Export to**, and then choose **SageMaker AI Inference Pipeline (via Jupyter Notebook)**. Then inspect and run that Jupyter notebook.



## 2. (Optional) Export your data flow

You can optionally export your Data Wrangler data flow to a Jupyter notebook to run the flow steps as a SageMaker AI Processing job.



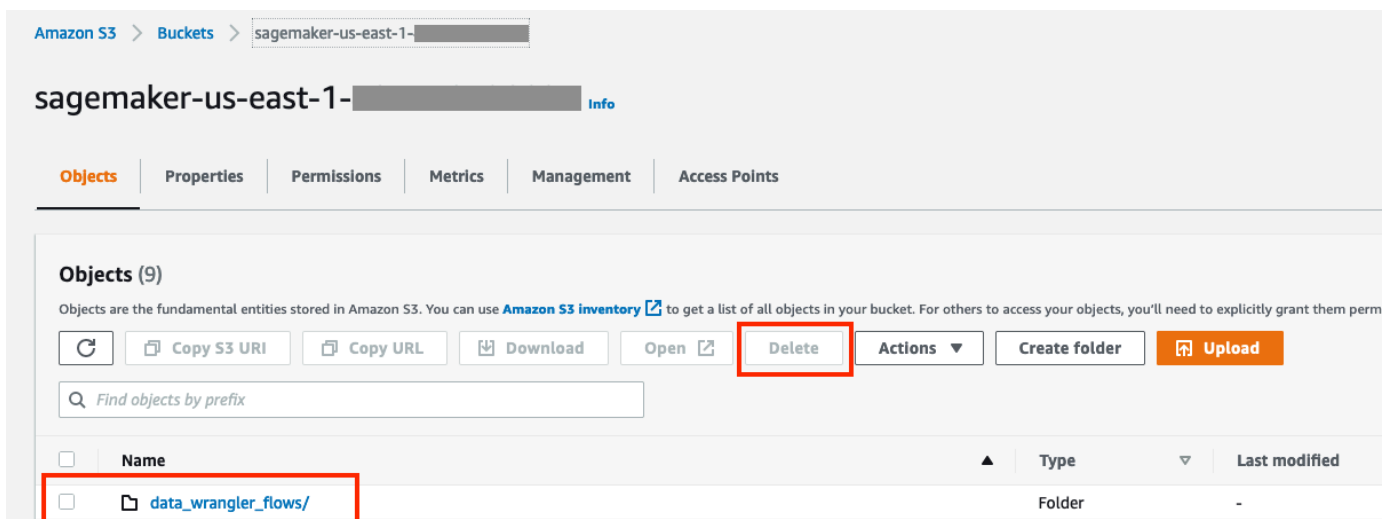
## Clean up resources

It is a best practice to delete resources that you are no longer using so that you don't incur unintended charges.

### 1. Empty and delete S3 bucket

To delete the S3 bucket, do the following:

- Open the Amazon S3 console. On the navigation bar, choose **Buckets**, **sagemaker-<your-Region>-<your-account-id>**, and then select the checkbox next to **data\_wrangler\_flows**. Then, choose **Delete**.
- In the **Delete objects** dialog box, verify that you have selected the proper object to delete and enter **permanently delete** into the **Permanently delete objects** confirmation box.
- Once this is complete and the bucket is empty, you can delete the **sagemaker-<your-Region>-<your-account-id>** bucket by following the same procedure again.



### 2. Delete Studio apps

The Data Science kernel used for running the notebook image in this tutorial will accumulate charges until you either stop the kernel or perform the following steps to delete the apps. For more information, see [Shut Down Resources](#) in the **Amazon SageMaker AI Developer Guide**.

To delete the SageMaker AI Studio apps, do the following: On the SageMaker AI Studio console, choose **studio-user**, and then delete all the apps listed under **Apps** by choosing **Delete app**. Wait until the **Status** changes to **Deleted**.

Amazon SageMaker > Control Panel

## User Details

General details about this user profile.

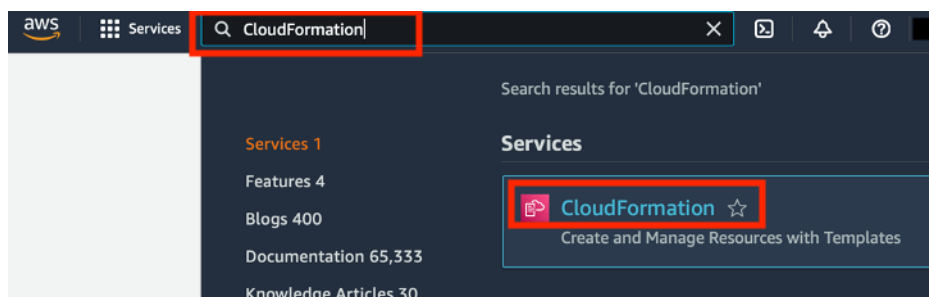
Apps				
App name	Status	App type	Created	Action
datascience-1-0-ml-t3-medium-1abf3407f667f989be9d86559395	✓ Ready	KernelGateway	Sat Apr 09 2022 15:25:16 GMT-0400 (Eastern Daylight Time)	<a href="#">Delete app</a>
default	✓ Ready	JupyterServer	Sat Apr 09 2022 15:22:55 GMT-0400 (Eastern Daylight Time)	<a href="#">Delete app</a>

## Delete the Studio domain

- If you used an existing SageMaker AI Studio domain, proceed directly to the conclusion section.
- If you ran the CloudFormation template to create a new SageMaker AI Studio domain, continue with the following steps to delete the domain, user, and the resources created by the CloudFormation template.

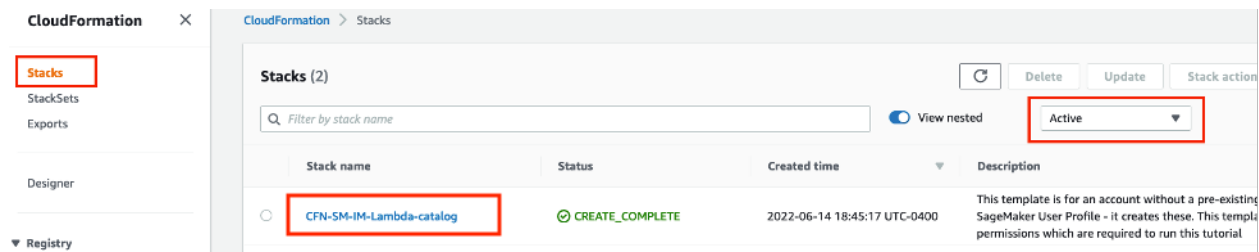
### 1. Open CloudFormation

To open the CloudFormation console, enter **CloudFormation** into the AWS console search bar, and choose **CloudFormation** from the search results.



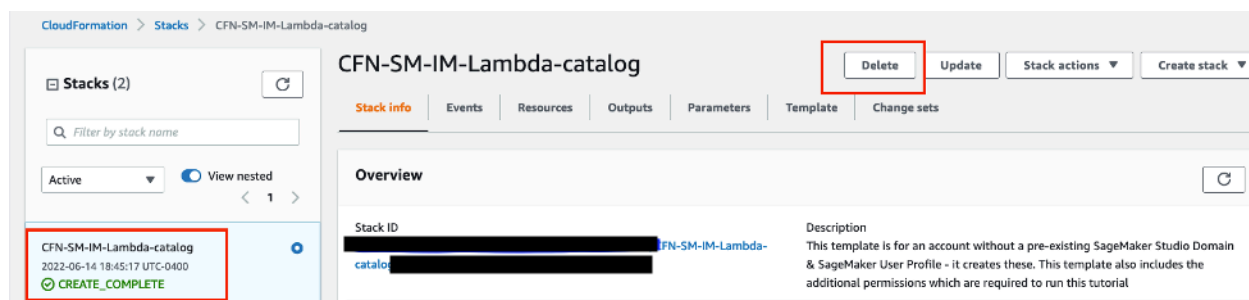
### 2. Choose the stack

In the **CloudFormation** pane, choose **Stacks**. From the status dropdown list, select **Active**. Under **Stack name**, choose **CFN-SM-IM-Lambda-catalog** to open the stack details page.



### 3. Delete the stack

On the **CFN-SM-IM-Lambda-catalog** stack details page, choose **Delete** to delete the stack along with the resources it created.



## Conclusion

Congratulations! You have completed the **Prepare Training Data for Machine Learning with Minimal Code** tutorial.

You have successfully used Amazon SageMaker AI Data Wrangler to prepare data for training a machine learning model. SageMaker AI Data Wrangler offers 300+ preconfigured data transformations, such as convert column type, one-hot encoding, impute missing data with mean or median, re-scale columns, and date/time embeddings, so you can transform your data into formats that can be effectively used for models without writing a single line of code.