

Amazon SageMaker Data Wrangler - Diabetic Patient Readmission Prediction

Patient readmission to hospital after prior visits for the same disease results in additional burden on healthcare providers and health system. Being able to understand, and predict readmission allows providers to create a better treatment plans and care. Reduction in cost is another benefit of such predictive modeling. In this example, we show how we prepare a machine learning dataset and build a predictive model using a diabetic patient readmission dataset that captures 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks in Amazon SageMaker Studio Data Wrangler.

[Amazon SageMaker Data Wrangler](#) in Amazon SageMaker Studio is a tool designed to allow data scientist quickly and iteratively explore and transform data for machine learning use cases. This example showcases how you can build a machine learning data transformation pipeline without writing sophisticated coding and create a model training, feature store or a ML pipeline with reproducibility for a diabetic patient readmission prediction use case.

In this example, you will be running machine learning workflow with Amazon SageMaker Data Wrangler and Amazon SageMaker features using a HCLS dataset.

Here are the high-level activities:

1. [Load UCI Source Dataset into your S3 bucket](#)
2. [Design your DataWrangler flow file](#)
3. [Processing & Training Jobs for Model building](#)
4. [Host trained Model for real-time inference](#)

1. Source Dataset

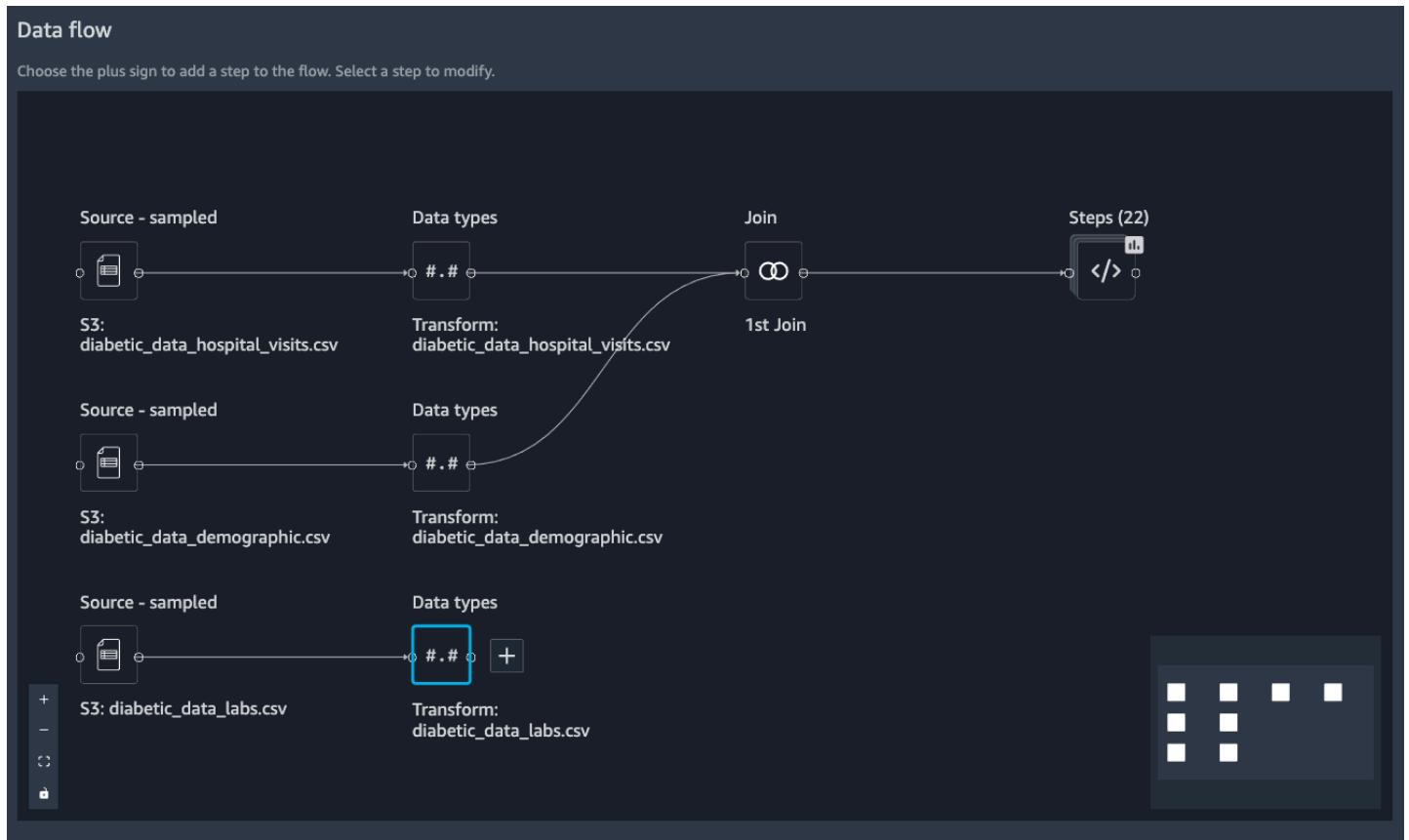
[UCI diabetic patient readmission dataset](#). The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes.

You will start by downloading the dataset and uploading it to a S3 bucket for you to run the example. Please review and execute the code in [datawrangler_workshop_pre_requisite.ipynb](#). The data will be available in `s3://sagemaker-$\{region\}-$\{account_number\}/sagemaker/demo-diabetic-datawrangler/` if you leave everything default.

2. Design your DataWrangler flow file

Data Wrangler flow overview and highlights

This project comes with a pre-built Data Wrangler flow file that can be customized with your `s3Uri` for reusability: [datawrangler_diabetes_readmission.flow](#).



It has multiple files from S3 loaded in: `diabetic_data_hospital_visits.csv`, `diabetic_data_demographic.csv` and `diabetic_data_labs.csv` for demonstration. It performs a inner join between the tables in `diabetic_data_hospital_visits.csv` and `diabetic_data_demographic.csv` by `encounter_id`. It has 28 transformation steps applied to process the data to meet the following requirements:

- no duplicate columns
- no duplicate entries
- no missing values (either fill the missing ones or remove columns that are largely missing)
- one hot encode the categorical features
- ordinally encode the age feature
- normalization (Standard scaler)
- Custom Transformation (Feature Store - EventTime needed)
- Analysis (Quick Model, Histogram)
- ready for ML training (Export notebook steps)

These are analyses created at different stage of the wrangling to serve as indication of the value these wrangling steps add. Most noticeably the Quick Model tells us that patient readmission prediction increases F1 score after performing transformation steps between 1 and 28 (in [datawrangler_diabetes_readmission.flow](#)). Data Scientists can use `Quick Model` analysis to perform iterative experimentation leading to efficient feature

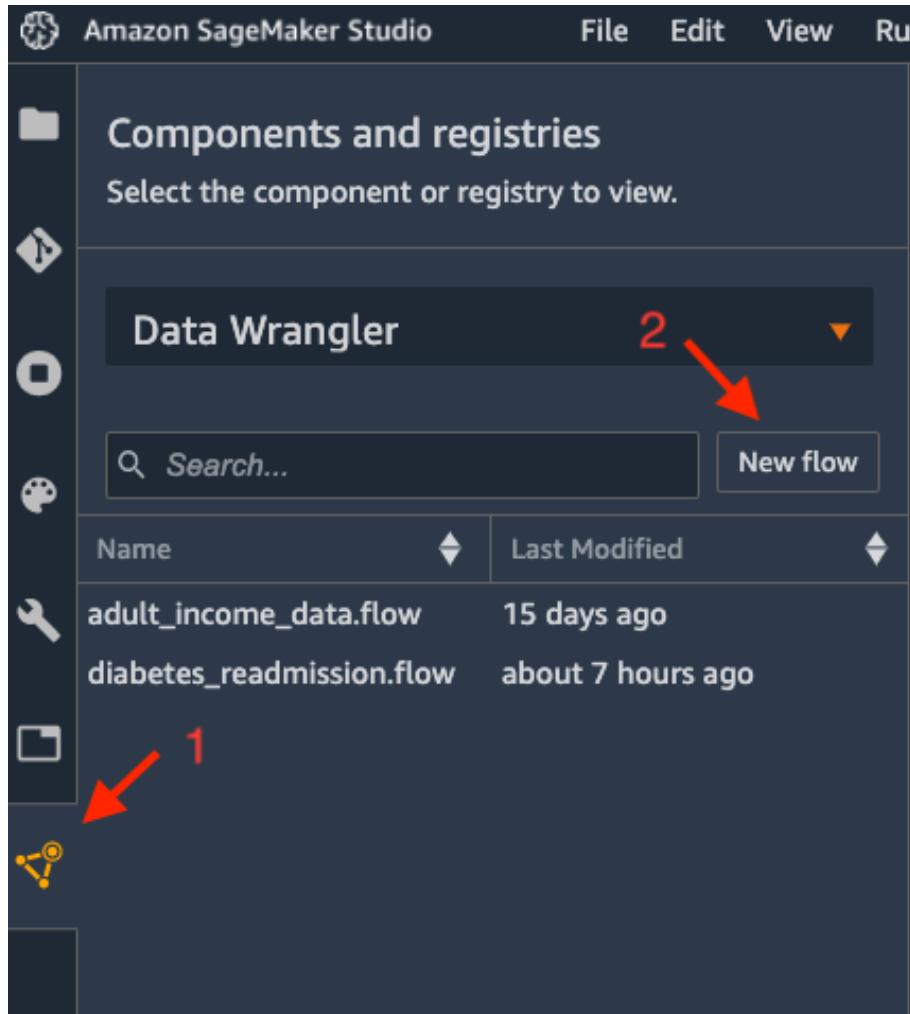
engineering for ML.

In this lab, we will perform data preprocessing using a combination of transformations described below to demonstrate the capability of Amazon SageMaker Data Wrangler. We will then train a XGBoost model to show you the process after data wrangling. We will then be hosting a trained model to SageMaker Hosted Endpoint for real-time inferencing.

Walk through

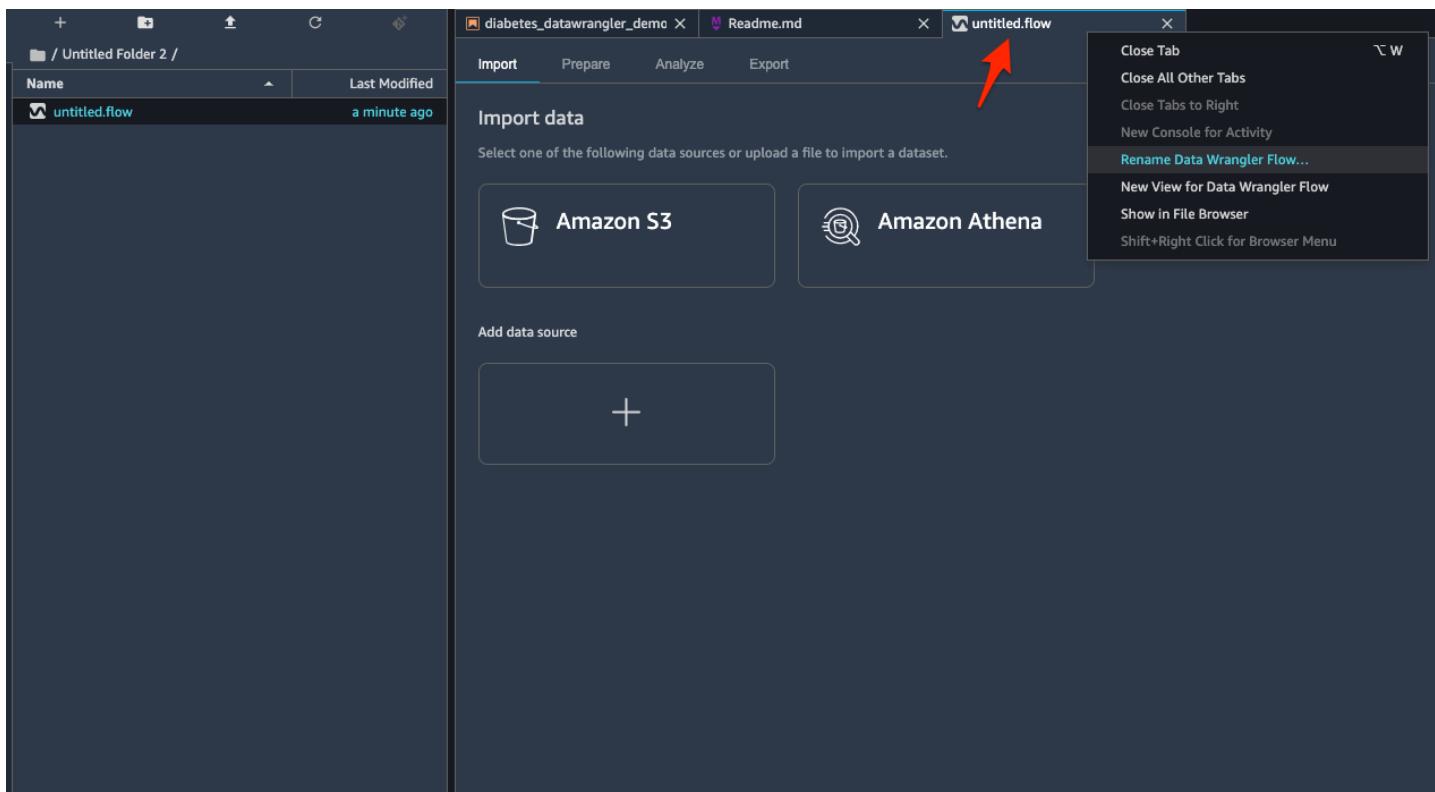
Create a new flow

Please click on the **SageMaker component and registry** tab and click **New flow**.



Rename your DW dataflow file

Right click on the **untitled.flow** file tab to reveal below options. Then choose Rename file to change the file name.



The screenshot shows the AWS Data Wrangler interface. At the top, there are three tabs: "diabetes_datawrangler_demo" (active), "Readme.md", and "dw-workshop.flow". Below the tabs, there are four navigation buttons: Import (underlined), Prepare, Analyze, and Export. The main area is titled "Import data" and contains the instruction "Select one of the following data sources or upload a file to import a dataset." Two options are visible: "Amazon S3" (with a bucket icon) and "Amazon Athena" (with a magnifying glass icon). Below these is a button labeled "Add data source" with a plus sign. A modal dialog box is overlaid on the interface, titled "Rename File". It displays the "File Path" as "Untitled Folder 2/dw-workshop.flow" and the "New Name" field containing "dw-workshop.flow". At the bottom of the dialog are two buttons: "Cancel" (gray) and "Rename" (blue).

Load the data from S3 into Data Wrangler

Select Amazon S3 as data source in **Import Data** view.

Import data

Select one of the following data sources or upload a file to import a dataset.

Amazon S3

Amazon Athena

Add data source

+

Note: You could also import data from Athena: [how databases and tables in Amazon Athena can be imported](#).

Select the csv files from the bucket: `s3://sagemaker-${region}-${account_number}/sagemaker/demo-diabetic-datawrangler/` one at a time.

Import Prepare Analyze Export

Data sources / S3 source / sagemaker-us-east-1-... / sagemaker / demo-diabetic-datawrangler / dataset_diabetes

Import a dataset from S3

Use the following table to browse S3. Select a file to see import options. The following file formats are supported: CSV and Parquet.

Object Name	Size	Last Modified
diabetic_data_demographic.csv	4.44MB	2021-03-23 17:12:38+00:00
diabetic_data_hospital_visits.csv	6.67MB	2021-03-23 17:12:39+00:00
diabetic_data_labs.csv	2.75MB	2021-03-23 17:12:39+00:00
diabetic_data_medication.csv	9.53MB	2021-03-23 17:12:40+00:00

1

2

DETAILS

Name: `diabetic_data_hospital_visits.csv`

Required

URI: `s3://sagemaker-us-east-1-.../sagemaker/dem...`

File type: `csv`

First row is header

Enable sampling

Import dataset

Previous Displaying 1 - 4 Next

Preview

encounter_id	patient_nbr	admission_type_id	discharge_dispositio...	admission_source_id	time_in_hospital	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?

Join the CSV files

Click the + sign on the Data-types icon for `diabetic_data_hospital_visits.csv` Select **Join** and new panel is presented for configuring input dataset join.

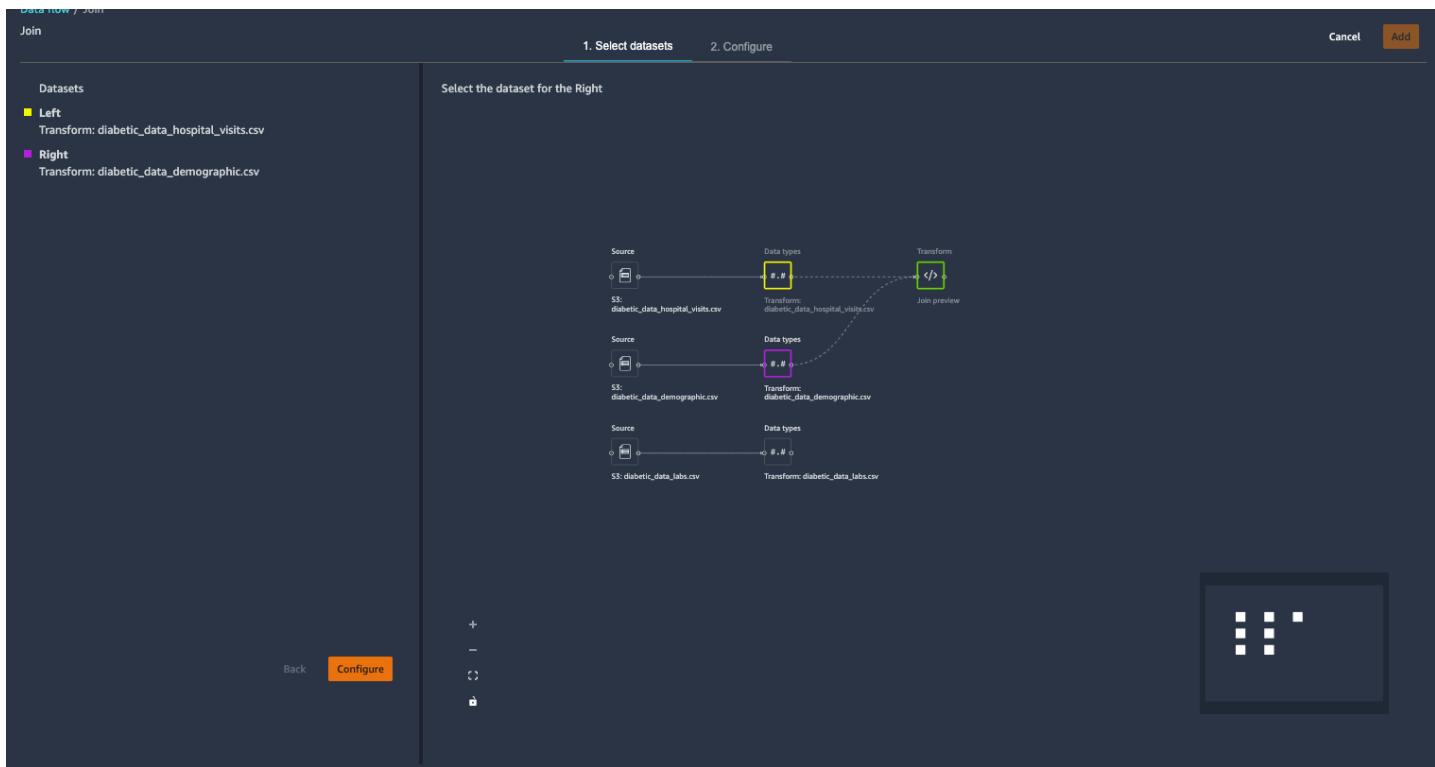
The screenshot shows the 'Data flow / Join' configuration interface. On the left, under 'Datasets', there is a section for 'Left' datasets, which includes 'diabetic_data_hospital_visits.csv'. In the center, under '1. Select datasets', there is a header 'Select the dataset for the Right'. Below this, three datasets are listed:

- Source: S3: diabetic_data_hospital_visits.csv, Data types: #, #, #, Transform: diabetic_data_hospital_visits.csv
- Source: S3: diabetic_data_demographic.csv, Data types: #, #, #, Transform: diabetic_data_demographic.csv
- Source: S3: diabetic_data_labs.csv, Data types: #, #, #, Transform: diabetic_data_labs.csv

On the right side of the interface, there is a large preview area with a grid icon. At the bottom left, there are buttons for 'Back', 'Configure' (which is highlighted in orange), and a set of small icons for operations like add, minus, copy, and paste.

1) Select `diabetic_data_demographic.csv` dataset as Right dataset.

2) Click `Join` and new Preview panel is presented.



3) Give a name to the Join and choose join type and Left & Right columns for join condition

Preview

INPUT

encounter_id (long)	patient_nbr (long)	admission_type_id (long)	discharge_disposition_id (long)
2278392	8222157	6	2
149190	55629189	1	1
64410	86047875	1	1
500364	82442376	1	1
16680	42519267	1	1
35754	82637451	2	1
55842	84259809	3	1
63768	114882984	1	1
12522	48330783	2	1
15738	63555939	3	3

RIGHT

encounter_id (long)	patient_nbr (long)	race (string)	gender (string)
2278392	8222157	Caucasian	Female
149190	55629189	Caucasian	Female
64410	86047875	AfricanAmerican	Female
500364	82442376	Caucasian	M
16680	42519267	Caucasian	M
35754	82637451	Caucasian	M
55842	84259809	Caucasian	M
63768	114882984	Caucasian	M
12522	48330783	Caucasian	Female
15738	63555939	Caucasian	Female

OUTPUT

Joined dataset
hospital_demographic_join

4) Click Apply to preview the joined dataset

Join

1. Select datasets 2. Configure

Cancel Add

Datasets

- Left**
Transform: diabetic_data_hospital_visits.csv
- Right**
Transform: diabetic_data_demographic.csv
- Joined dataset**
Name: hospital_demographic_join

Optional
Join Type
Select the join type:
Inner

Columns
Select Left and Right to join
Left: encounter_id Right: encounter_id

Preview

INPUT

encounter_id (long)	patient_nbr (long)	admission_type_id (long)	discharge_dispositio...
2278392	8222157	6	25
149190	55629189	1	1
64410	86047875	1	1
500364	82442376	1	1
16680	42519267	1	1
35754	82637451	2	1
55842	84259809	3	1
63768	114882984	1	1
12522	48330783	2	1
15738	63555939	3	3

INPUT

encounter_id (long)	patient_nbr (long)	race (string)	ge
2278392	8222157	Caucasian	F
149190	55629189	Caucasian	F
64410	86047875	AfricanAmerican	F
500364	82442376	Caucasian	M
16680	42519267	Caucasian	M
35754	82637451	Caucasian	M
55842	84259809	Caucasian	M
63768	114882984	Caucasian	M
12522	48330783	Caucasian	F
15738	63555939	Caucasian	F

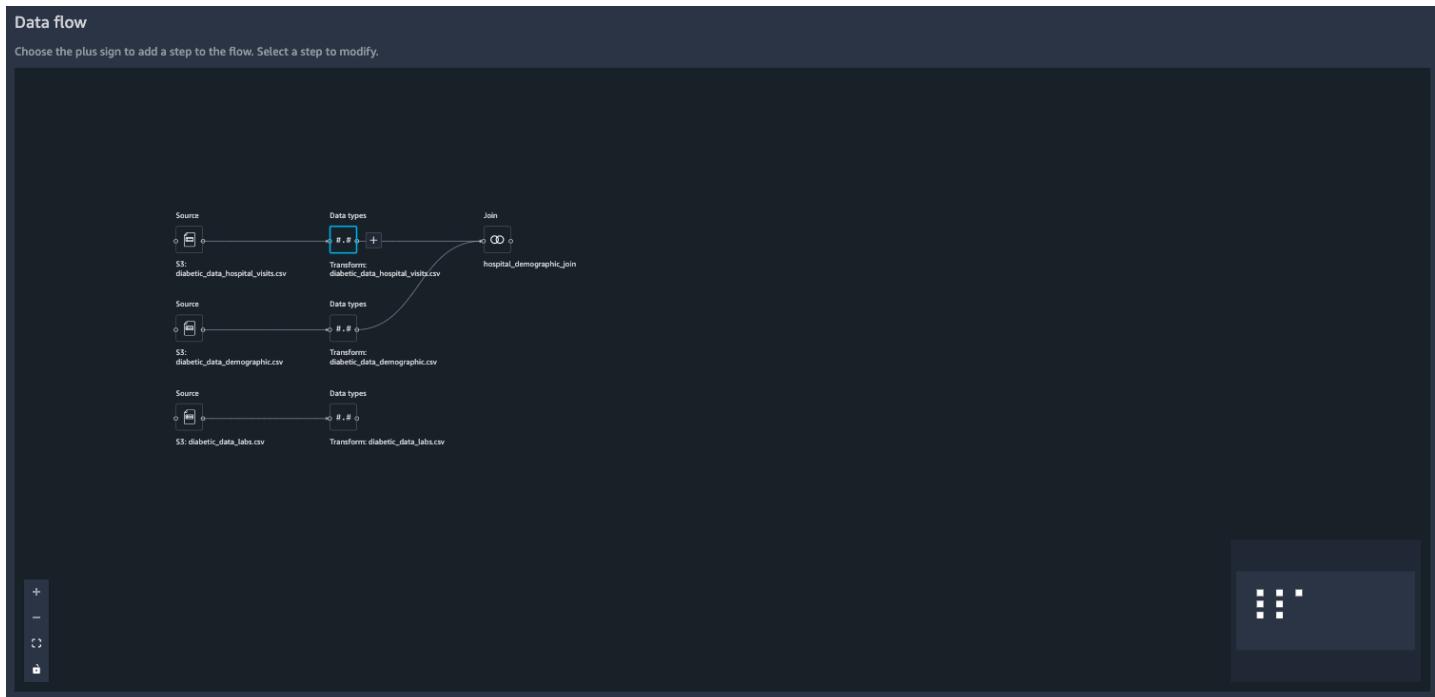
OUTPUT

Joined dataset: hospital_demographic_join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (long)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code (\$)
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?

Back Apply

5) Click Add for the join configuration to be added to the data-flow



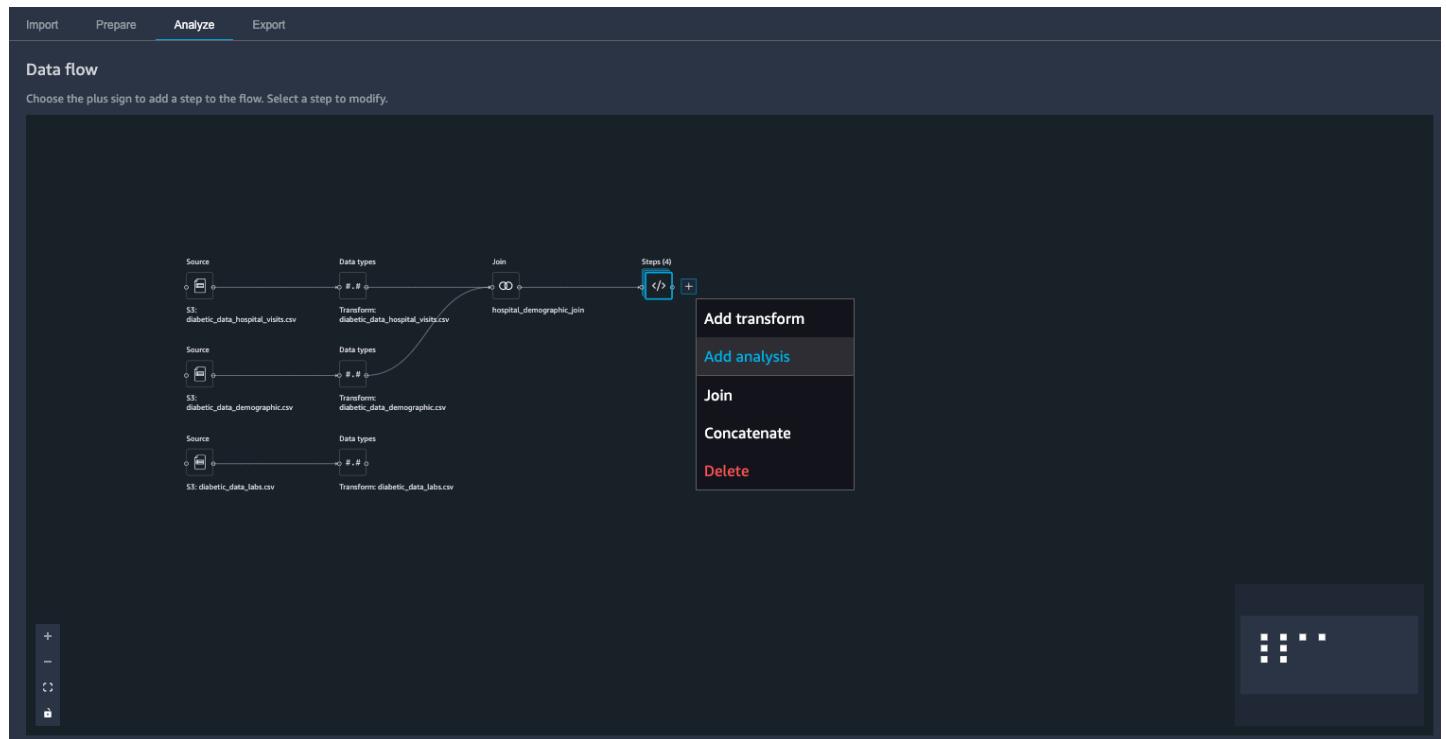
Analysis

Before we apply any transformations on the input source, let's perform a quick analysis of the dataset. SageMaker Data Wrangler provides built-in Analysis types like `Histogram` `Scatter Plot` `Target Leakage` `Bias Report` `Histogram & Quick Model`. You can find all types details [SageMaker Data Wrangler Analyses](#). Let's touch on 2 of these types to get feel of the potential.

Histogram

You can use histograms to see the counts of feature values for a specific feature. You can inspect the relationships between features using the Color by option. You can also use the Facet by feature to create histograms of one column, for each value in another column.

- 1) Click + sign next to Join flow icon and choose `Add analysis`



- 2) Select `Histogram` from the list of Analysis types on the right panel.



Import Prepare Analyze Export

Imported datasets / joined / Untitled

Create Analysis

Create an analysis of your data.

[Learn more](#)

Histogram: Untitled

No Preview available

Use Configure for built-in analyses
Use Code to create a custom analysis

Data table

encounter_id_0	patient_nbr_0	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	payer_code	medical_spl
2278392	8222157	6	25	1	1	?	Pediatrics
149190	55629189	1	1	7	3	?	?
64410	86047875	1	1	7	2	?	?
500364	82442376	1	1	7	2	?	?
16680	42519267	1	1	7	1	?	?
35754	82637451	2	1	2	3	?	?
55842	84259809	3	1	2	4	?	?
63768	114882984	1	1	7	5	?	?
12522	48330783	2	1	4	13	?	?
15738	63555939	3	3	4	12	?	InternalMe

Configure Code

Analysis type

- Histogram
- Bias Report
- Histogram
- Quick Model
- Scatter Plot
- Table Summary
- Target Leakage

Color by

Select... Optional

Facet by

Select... Optional

[Clear](#) [Preview](#) [Save](#)

3) Give a name to your analysis and select the `X axis` as `race`, `Color by` as `age` & `Facet by` as `gender`. Which means we want to plot histograms by `race` with `age` factor reflected by color legend and also faceted by `gender`.

Import Prepare Analyze Export

Imported datasets / joined / Demographics by Race, Age & Gender

Create Analysis

Create an analysis of your data.

[Learn more](#)

Histogram: Demographics by Race, Age & Gender

No Preview available

Use Configure for built-in analyses
Use Code to create a custom analysis

Data table

encounter_id_0	patient_nbr_0	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	payer_code	medical_spl
2278392	8222157	6	25	1	1	?	Pediatrics
149190	55629189	1	1	7	3	?	?
64410	86047875	1	1	7	2	?	?
500364	82442376	1	1	7	2	?	?
16680	42519267	1	1	7	1	?	?
35754	82637451	2	1	2	3	?	?
55842	84259809	3	1	2	4	?	?
63768	114882984	1	1	7	5	?	?
12522	48330783	2	1	4	13	?	?
15738	63555939	3	3	4	12	?	InternalMe

Configure Code

Analysis type

Histogram

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

Analysis name

Demographics by Race, Age & Gender

Optional

X axis

race

Color by

gender

Optional

Facet by

age

Optional

[Clear](#) [Preview](#) [Save](#)

4) Click `Preview` and wait for the model to be results to be displayed on the screen

Import Prepare Analyze Export



5) Click `create` button to add the histogram analysis to the data flow.

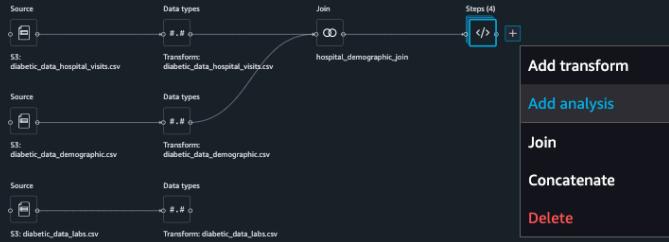
Quick Model

Let's explore `Quick Model` transformation. `Quick Model` visualization helps to quickly evaluate your data and produce importance scores for each feature. A feature importance score indicates how useful a feature is at predicting a target label. The feature importance score is between [0, 1] and a higher number indicates that the feature is more important to the whole dataset. On the top of the quick model chart, there is a model score. A classification problem shows an F1 score. A regression problem has a mean squared error (MSE) score.

1) Click + sign next to Join flow icon and choose `Add analysis`

Data flow

Choose the plus sign to add a step to the flow. Select a step to modify.



2) Select **Quick Model** from the list of Analysis types on the right panel.

Import Prepare Analyze Export

Imported datasets / Transform: 1st Join / Quick_Model_post_transform

Create Analysis

Create an analysis of your data. [Learn more.](#)

Quick Model: Quick_Model_post_transform

No Preview available

Use Configure for built-in analyses

Use Code to create a custom analysis

Analysis type

- Quick Model (selected)
- Bias Report
- Histogram
- Scatter Plot
- Table Summary
- Target Leakage

Cancel Preview Create

encounter_id_0	patient_nbr_0	admission_type_id	discharge_dispositio...	admission_source_id	time_in_hospital	payer_code	medical_specialty
2278392	8222157	6	25	1	1	?	Pediatrics-Endocrinology
149190	55629189	1	1	7	3	?	?
64410	86047875	1	1	7	2	?	?
500364	82442376	1	1	7	2	?	?
16680	42519267	1	1	7	1	?	?
35754	82637451	2	1	2	3	?	?
55842	84259809	3	1	2	4	?	?

3) Give a name to your analysis and select the target label in **Label** field.

Imported datasets / joined / Initial Quick Model

Create Analysis

Create an analysis of your data.

[Learn more](#)

Quick Model: Initial Quick Model

No Preview available

Use Configure for built-in analyses
Use Code to create a custom analysis

Configure Code

Analysis type: Quick Model

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

Analysis name: Initial Quick Model

Optional

Label: readmitted

Data table

encounter_id_0	patient_nbr_0	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	payer_code	medical_spl
2278392	8222157	6	25	1	1	?	Pediatrics
149190	55629189	1	1	7	3	?	?
64410	86047875	1	1	7	2	?	?
500364	82442376	1	1	7	2	?	?
16680	42519267	1	1	7	1	?	?
35754	82637451	2	1	2	3	?	?
55842	84259809	3	1	2	4	?	?
63768	114882984	1	1	7	5	?	?
12522	48330783	2	1	4	13	?	?
15738	63555939	3	3	4	12	?	InternalMe

Clear Preview Save

4) Click **Preview** and wait for the model to be results to be displayed on the screen

Import Prepare Analyze Export

Imported datasets / joined / Initial Quick Model

Create Analysis

Create an analysis of your data.

[Learn more](#)

Quick Model: Initial Quick Model

Model achieved a 0.527 F1 on a test set.

Configure Code

Analysis type: Quick Model

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

Analysis name: Initial Quick Model

Optional

Label: readmitted

Data table

encounter_id_0	patient_nbr_0	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	payer_code	medical_spl
2278392	8222157	6	25	1	1	?	Pediatrics
149190	55629189	1	1	7	3	?	?
64410	86047875	1	1	7	2	?	?
500364	82442376	1	1	7	2	?	?
16680	42519267	1	1	7	1	?	?
35754	82637451	2	1	2	3	?	?
55842	84259809	3	1	2	4	?	?
63768	114882984	1	1	7	5	?	?
12522	48330783	2	1	4	13	?	?
15738	63555939	3	3	4	12	?	InternalMe

Clear Preview Save

Here we see that the Quick Model has given 0.527 (your individual score may be slightly different) F1 score on

the current state of the dataset. Remember, we haven't applied any data transformations. We'll revisit [Quick Model](#) after we apply a few data transformations and assess whether or not those transformations improve F1 score.

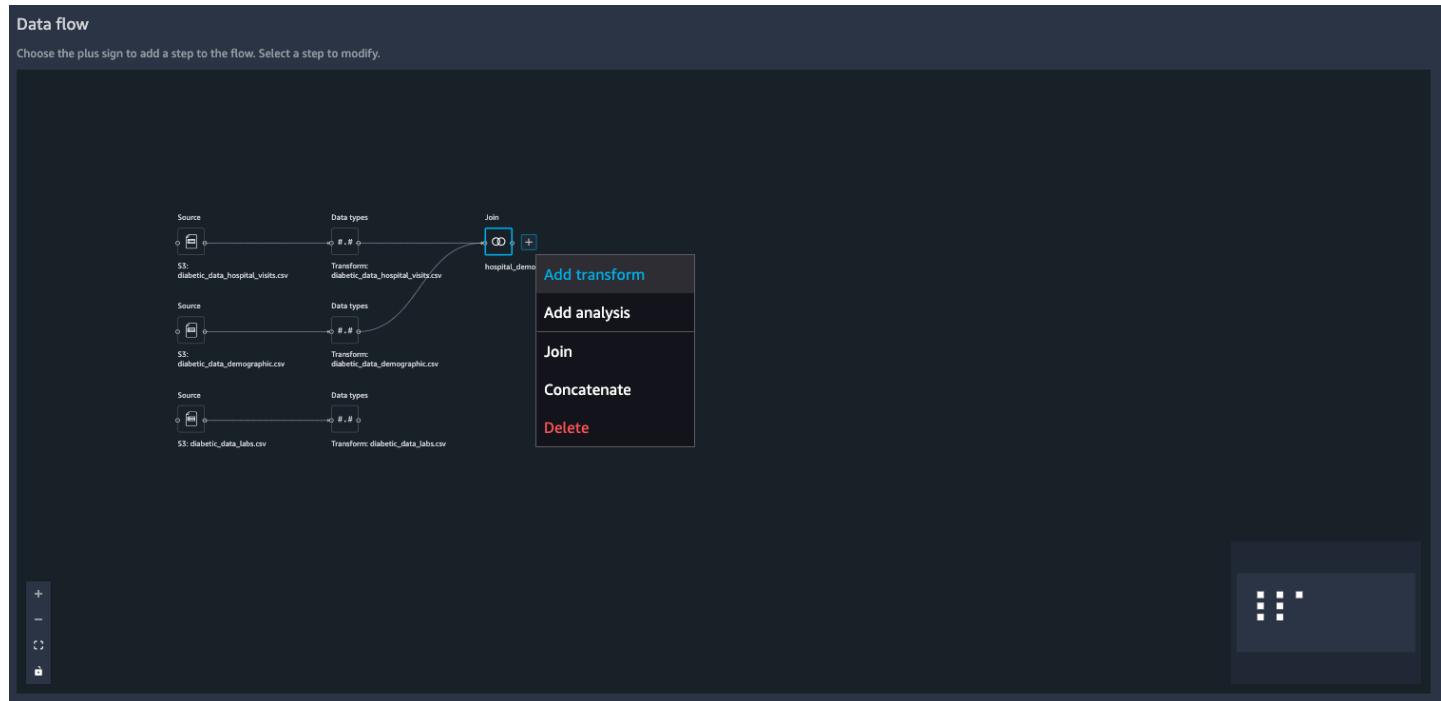
5) Click [create](#) button to add the quick model analysis to the data flow.

Transformations

We will use Data Wrangler built-in transforms to apply the listed transformations to our dataset.

No Duplicate Columns

1) Click + sign next to Join flow icon and choose [Add Transform](#)



2) Pick [Manage Columns](#) from the list of transforms on the right panel

Data flow / hospital_demographic_join

hospital_demographic_join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (I...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389323	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?
236316	40523301	1	3	7	6	?
248916	115196778	1	1	1	2	?

TRANSFORM

- Add Previous steps
- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns
 - Move, drop, duplicate or rename columns in the dataset. [Learn more](#).
- Transform
 - Drop column
 - Drop column
 - Duplicate column
 - Rename column
 - Move column
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit
- > Validate string

3) Choose **Drop Column** transform and select `encounter_id_1` column to drop

Import Prepare Analyze Export

Data flow / hospital_demographic_join

hospital_demographic_join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (I...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389323	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?
236316	40523301	1	3	7	6	?
248916	115196778	1	1	1	2	?

TRANSFORM

- Add Previous steps
- > Custom Transform
- > Custom formula
- > Encode categorical
 - diag_1
 - diag_2
 - diag_3
 - readmitted
 - encounter_id_1
- > patient_nbr_1
- > race
- gender
- age
- weight
- |Select...
- Clear

- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit
- > Validate string

4) Click **Preview** to preview the changes to the data set

Import Prepare Analyze Export

Data flow / hospital_demographic_join

hospital_demographic_join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389323	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?
236316	40523301	1	3	7	6	?
248916	115196778	1	1	1	2	?

TRANSFORM

Add Previous steps

- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

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

Transform

Drop column

Column to drop

encounter_id_1
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit
- > Validate string

Preview Add

5) Click **Add** to add the change to the data flow

Data flow / hospital_demographic_join

hospital_demographic_join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389323	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?

TRANSFORM

Add Previous steps

- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

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

Transform

Drop column

Column to drop

encounter_id_1
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit
- > Validate string

Preview Add

6) Repeat above steps 1 through 5 for **patient_nbr_1** column.

Data flow / demographic_visits_join

demographic_visits_join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...	discharge_dispositio...	admission_source_id ...	time_in,
2278392	8222157	6	25	1	1
149190	55629189	1	1	7	3
64410	86047875	1	1	7	2
500364	82442376	1	1	7	2
16680	42519267	1	1	7	1
35754	82637451	2	1	2	3
55842	84259809	3	1	2	4
63768	114882984	1	1	7	5
12522	48330783	2	1	4	13
15738	63555939	3	3	4	12
28236	89869032	1	1	7	9
36900	77391171	2	1	4	7
40926	85504905	1	3	7	7
42570	77586282	1	6	7	10
62256	49726791	3	1	2	1
73578	86328819	1	3	7	12
77076	92519352	1	1	7	4
84222	108662661	1	1	7	3
89682	107389323	1	1	7	5
148530	69422211	3	6	2	6
150006	22864131	2	1	4	2
150048	21239181	2	1	4	2
182796	63000108	2	1	4	2
183930	107400762	2	6	1	11
216156	62718876	3	1	2	3
221634	21861756	1	1	7	1
236316	40523301	1	3	7	6

Back to data flow

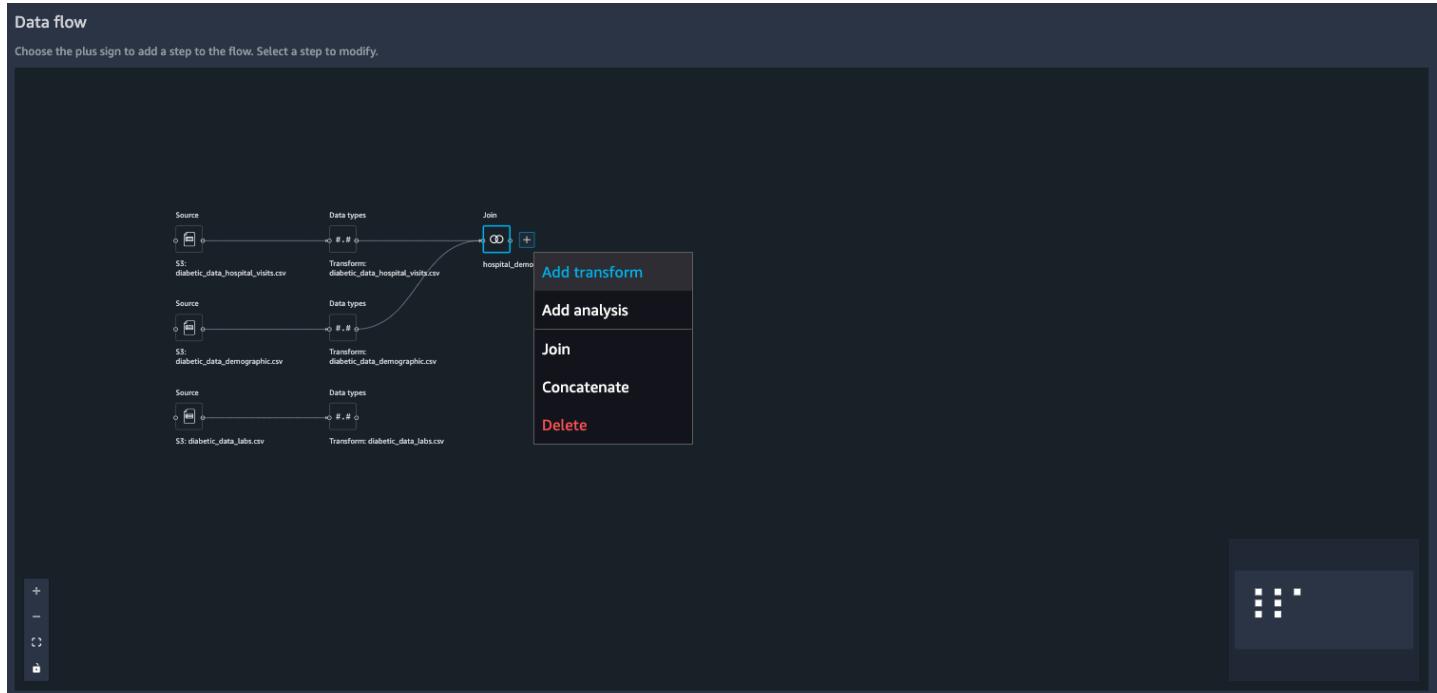
Add Previous steps

- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

Move, drop, duplicate or rename columns in the dataset. [Learn more](#).
- > Transform
 - Drop column
 - Column to drop
 - Clear
 - Preview Add
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit
- > Validate

No Duplicate Rows/Observations

1) Click + sign next to Join flow icon and choose **Add Transform**



2) Pick **Custom Transform** from the list of transforms on the right panel

Data flow / Transform: 1st Join

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389323	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?
236316	40523301	1	3	7	6	?
248916	115196778	1	1	1	2	?

TRANSFORM

Add Previous steps (3)

Custom Transform

- Python (PySpark)
- Python (PySpark)
- Python (Pandas)
- SQL (PySpark SQL)

Custom formula

Encode categorical

Featurize date/time

Featurize text

Format string

Handle missing

Handle outliers

Manage columns

Manage rows

Manage vectors

Parse column as type

Process numeric

Search and edit

Validate string

3) select **Python (Pandas)** and enter below line of code in the text box. Then click **Preview** to view the results.

```
df.drop_duplicates(subset=['encounter_id_0', 'patient_nbr_0'], keep='first', inplace=True)
```

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389323	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?
236316	40523301	1	3	7	6	?
248916	115196778	1	1	1	2	?

TRANSFORM

Add Previous steps (3)

Custom Transform

- Python (Pandas)

```
1 # Table is available as variable `df`
2 df.drop_duplicates(subset=['encounter_id_0', 'patient_nb
```

Clear Preview Add

Custom formula

Encode categorical

Featurize date/time

Featurize text

Format string

Handle missing

Handle outliers

Manage columns

Manage rows

Manage vectors

Parse column as type

Process numeric

Search and edit

Validate string

4) Click **Add** to add the change to the data flow

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Previewing Python (Pandas)

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389523	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?

TRANSFORM

Add Previous steps (3)

Custom Transform

Python (Pandas)

```
1 # Table is available as variable `df`
2 df.drop_duplicates(subset=['encounter_id_0', 'patient_nb'])
```

Clear Preview Add

- Custom formula
- Encode categorical
- Featurize date/time
- Featurize text
- Format string
- Handle missing
- Handle outliers
- Manage columns
- Manage rows
- Manage vectors
- Parse column as type
- Process numeric
- Search and edit
- Validate string

Custom Transformation - Add new features to your dataset

Feature Store would need Event Time feature to be present to be able to store inside Feature Groups

1) Click + sign next to Join flow icon and choose **Add Transform**

Data flow

Choose the plus sign to add a step to the flow. Select a step to modify.

Add transform

Add analysis

Join

Concatenate

Delete

2) Pick **Custom Transform** from the list of transforms on the right panel

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389323	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?
236316	40523301	1	3	7	6	?
248916	115196778	1	1	1	2	?

TRANSFORM

Add Previous steps (3)

Custom Transform

- Python (PySpark) (selected)
- Python (PySpark)
- Python (Pandas)
- SQL (PySpark SQL)

Custom formula

Encode categorical

Featurize date/time

Featurize text

Format string

Handle missing

Handle outliers

Manage columns

Manage rows

Manage vectors

Parse column as type

Process numeric

Search and edit

Validate string

3) select `Python (Pandas)` and enter below line of code in the text box. Then click `Preview` to view the results.

```
import time
df['eventTime'] = time.time()
```

Data flow / Transform: 1st Join

Transform: 1st Join

	readmitted (string)	race (string)	gender (string)	age (string)	weight (string)	payer_code_na_fill (st...)
NO	Caucasian	Female	[0-10)	?	?	
>30	Caucasian	Female	[10-20)	?	?	
NO	AfricanAmerican	Female	[20-30)	?	?	
NO	Caucasian	Male	[30-40)	?	?	
NO	Caucasian	Male	[40-50)	?	?	
>30	Caucasian	Male	[50-60)	?	?	
NO	Caucasian	Male	[60-70)	?	?	
>30	Caucasian	Male	[70-80)	?	?	
NO	Caucasian	Female	[80-90)	?	?	
NO	Caucasian	Female	[90-100)	?	?	
>30	AfricanAmerican	Female	[40-50)	?	?	
<30	AfricanAmerican	Male	[60-70)	?	?	
<30	Caucasian	Female	[40-50)	?	?	
NO	Caucasian	Male	[80-90)	?	?	
>30	AfricanAmerican	Female	[60-70)	?	?	
NO	AfricanAmerican	Male	[60-70)	?	?	
<30	AfricanAmerican	Male	[50-60)	?	?	
NO	Caucasian	Female	[50-60)	?	?	
>30	AfricanAmerican	Male	[70-80)	?	?	
NO	?	Male	[70-80)	?	?	
NO	?	Female	[50-60)	?	?	
NO	?	Male	[60-70)	?	?	
NO	AfricanAmerican	Female	[70-80)	?	?	
>30	Caucasian	Female	[80-90)	?	?	
NO	AfricanAmerican	Female	[70-80)	?	?	
NO	Other	Female	[50-60)	?	?	
NO	Caucasian	Male	[80-90)	?	?	
>30	Caucasian	Female	[50-60)	?	?	

TRANSFORM

Add Previous steps (5)

Custom Transform

Python (Pandas)

```

1 # Table is available as variable `df`
2 import time
3 df['eventTime'] = time.time()

```

Clear Preview Add

- Custom formula
- Encode categorical
- Featurize date/time
- Featurize text
- Format string
- Handle missing
- Handle outliers
- Manage columns
- Manage rows
- Manage vectors
- Parse column as type
- Process numeric
- Search and edit
- Validate string

4) Click **Add** to add the change to the data flow

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Previewing Python (Pandas)

Transform: 1st Join

	readmitted (string)	race (string)	gender (string)	age (string)	weight (string)	payer_code_na_fill (st...)
NO	Caucasian	Female	[0-10)	?	?	
>30	Caucasian	Female	[10-20)	?	?	
NO	AfricanAmerican	Female	[20-30)	?	?	
NO	Caucasian	Male	[30-40)	?	?	
NO	Caucasian	Male	[40-50)	?	?	
>30	Caucasian	Male	[50-60)	?	?	
NO	Caucasian	Male	[60-70)	?	?	
>30	Caucasian	Male	[70-80)	?	?	
NO	Caucasian	Female	[80-90)	?	?	
NO	Caucasian	Female	[90-100)	?	?	
>30	AfricanAmerican	Female	[40-50)	?	?	
<30	AfricanAmerican	Male	[60-70)	?	?	
<30	Caucasian	Female	[40-50)	?	?	
NO	Caucasian	Male	[80-90)	?	?	
>30	AfricanAmerican	Female	[60-70)	?	?	
NO	AfricanAmerican	Male	[60-70)	?	?	
<30	AfricanAmerican	Male	[50-60)	?	?	
NO	Caucasian	Female	[50-60)	?	?	
>30	AfricanAmerican	Male	[70-80)	?	?	
NO	?	Male	[70-80)	?	?	
NO	?	Female	[50-60)	?	?	
NO	?	Male	[60-70)	?	?	
NO	AfricanAmerican	Female	[70-80)	?	?	
>30	Caucasian	Female	[80-90)	?	?	
NO	AfricanAmerican	Female	[70-80)	?	?	
NO	Other	Female	[50-60)	?	?	

TRANSFORM

Add Previous steps (5)

Custom Transform

Python (Pandas)

```

1 # Table is available as variable `df`
2 import time
3 df['eventTime'] = time.time()

```

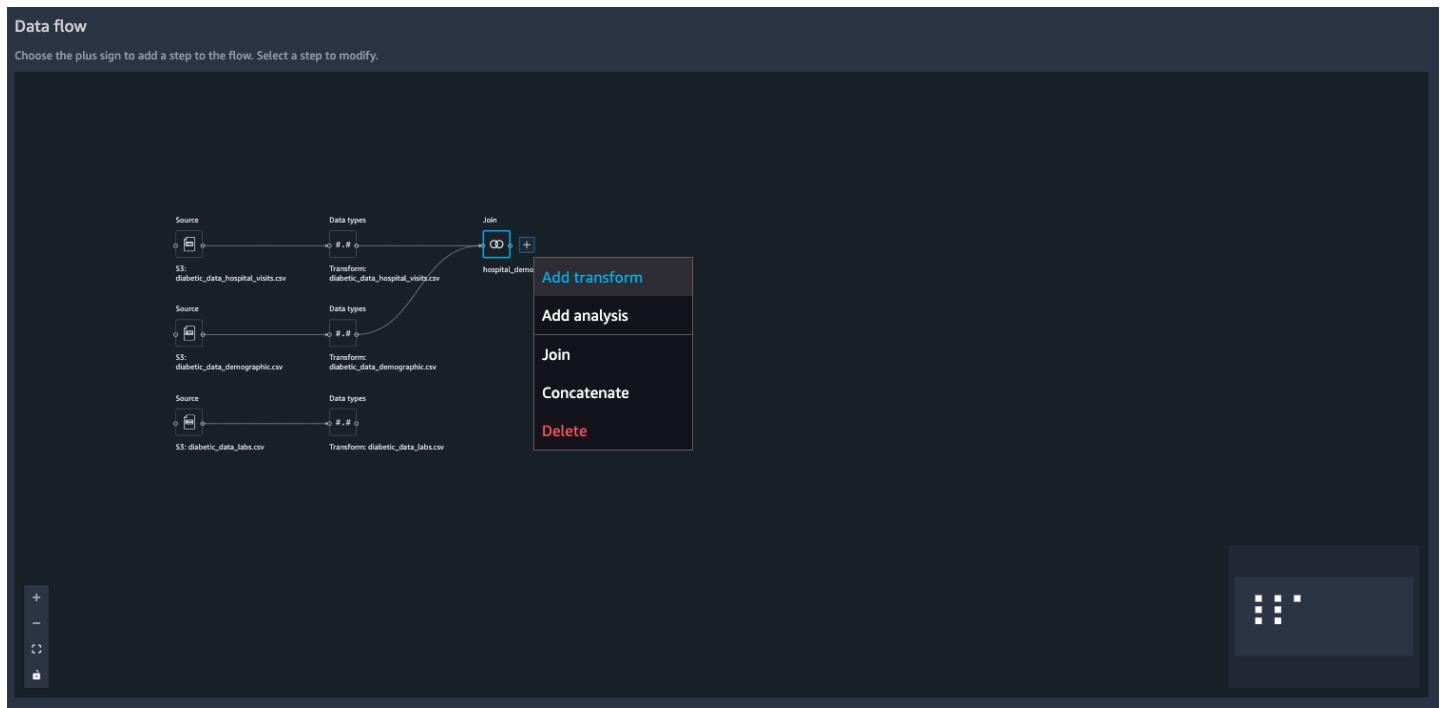
Clear Preview Add

- Custom formula
- Encode categorical
- Featurize date/time
- Featurize text
- Format string
- Handle missing
- Handle outliers
- Manage columns
- Manage rows
- Manage vectors
- Parse column as type
- Process numeric
- Search and edit
- Validate string

More Custom Transformations - impute fillers for undesirable

values

1) Click + sign next to Join flow icon and choose Add Transform



2) Pick Custom Transform from the list of transforms on the right panel

The screenshot shows the 'Prepare' tab of the data flow interface. On the left, there is a preview of the data for the 'Transform: 1st Join' step, showing columns like 'encounter_id_0 (long)', 'patient_nbr_0 (long)', 'admission_type_id (l...)', etc. On the right, the 'TRANSFORM' panel is open, showing a list of available transforms. 'Custom Transform' is selected, and 'Python (PySpark)' is chosen from the dropdown. Other options include 'Python (PySpark)', 'Python (Pandas)', and 'SQL (PySpark SQL)'. Below the dropdown, there is a list of other transform types: 'Custom formula', 'Encode categorical', 'Featurize date/time', 'Featurize text', 'Format string', 'Handle missing', 'Handle outliers', 'Manage columns', 'Manage rows', 'Manage vectors', 'Parse column as type', 'Process numeric', 'Search and edit', and 'Validate string'. There is also a link to 'Back to data flow'.

3) select Python (Pandas) and enter below line of code in the text box. Then click Preview to view the

results.

```
# Table is available as variable `df`  
df["race"]=df["race"].str.replace("?", "Other")  
df["weight"]=df["weight"].str.replace("?", "0")  
df["payer_code"]=df["payer_code"].str.replace("?", "0")  
df["medical_specialty"]=df["medical_specialty"].str.replace("?", "Other")
```

The screenshot shows a data preparation interface with a top navigation bar: Import, Prepare (selected), Analyze, Export. Below the navigation is a section titled 'Data flow / Transform: 1st Join' with a sub-section 'Transform: 1st Join'. A table is displayed with columns: encounter_id_0 (long), patient_nbr_0 (long), admission_type_id (l...), discharge_dispositio... (truncated), admission_source_id ... (truncated), and time_in. The table contains approximately 30 rows of data. To the right of the table is a 'TRANSFORM' panel. At the top of the panel is an 'Add' button and a 'Previous steps' link. Below this is a 'Custom Transform' section with a dropdown set to 'Python (Pandas)'. A code editor window shows the Python code provided at the top of the page. Below the code editor are 'Clear', 'Preview', and 'Add' buttons. The 'Add' button is highlighted with a red box. The 'TRANSFORM' panel also lists several other options: Custom formula, Encode categorical, Featurize date/time, Featurize text, Format string, Handle missing, Handle outliers, Manage columns, Manage rows, Manage vectors, Parse column as type, Process numeric, and Search and edit.

4) Click **Add** to add the change to the data flow

This screenshot shows the same data preparation interface as the previous one, but the 'Prepare' tab is now selected in the top navigation bar. The rest of the interface remains the same, including the table and the 'TRANSFORM' panel.

Data flow / Transform: 1st Join

Previewing Python (Pandas)

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (I...)	discharge_dispositio...	admission_source_id ...	time_in
2278392	8222157	6	25	1	1
149190	55629189	1	1	7	3
64410	86047875	1	1	7	2
500364	82442376	1	1	7	2
16680	42519267	1	1	7	1
35754	82637451	2	1	2	3
55842	84259809	3	1	2	4
63768	114882984	1	1	7	5
12522	48330783	2	1	4	13
15738	63555939	3	3	4	12
28236	89869032	1	1	7	9
36900	77391171	2	1	4	7
40926	85504905	1	3	7	7
42570	77586282	1	6	7	10
62256	49726791	3	1	2	1
73578	86328819	1	3	7	12
77076	92519352	1	1	7	4
84222	108662661	1	1	7	3
89682	107389323	1	1	7	5
148530	69422211	3	6	2	6
150006	22864131	2	1	4	2
150048	21239181	2	1	4	2
182796	63000108	2	1	4	2
183930	107400762	2	6	1	11

Transform: 1st Join

```

1 # Table is available as variable `df`
2 df['race']=df['race'].str.replace("?", "Other")
3 df['weight']=df['weight'].str.replace("?", "0")
4 df['payer_code']=df['payer_code'].str.replace("?", "")
5 df['medical_specialty']=df['medical_specialty'].str.
6

```

Custom Transform

Python (Pandas)

Custom formula

Encode categorical

Featurize date/time

Featurize text

Format string

Handle missing

Handle outliers

Manage columns

Manage rows

Manage vectors

Parse column as type

Process numeric

Search and edit

Handle missing values

1) Click + sign next to Join flow icon and choose Add Transform

Data flow

Choose the plus sign to add a step to the flow. Select a step to modify.

Add transform

Add analysis

Join

Concatenate

Delete

2) Pick Handle missing from the list of transforms on the right panel and choose Impute for Transform

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442576	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389323	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?
236316	40523301	1	3	7	6	?
248916	115196778	1	1	1	2	?

TRANSFORM

Add Previous steps (4)

- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- < Handle missing

Replace, drop, or add indicators for missing values. [Learn more](#)

Transform (1)

Impute

impute

Fill missing

Add indicator for missing

Drop missing

Select...

Imputing strategy (1)

Approximate Median

Output column (1)

Optional

Clear

Preview Add

3) Choose Column type as Numeric and select Input column as diag_1. Let's use Mean for Imputing strategy. You can also provide optional Output column name.

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in
2278392	8222157	6	25	1	1
149190	55629189	1	1	7	3
64410	86047875	1	1	7	2
500364	82442376	1	1	7	2
16680	42519267	1	1	7	1
35754	82637451	2	1	2	3
55842	84259809	3	1	2	4
63768	114882984	1	1	7	5
12522	48330783	2	1	4	13
15738	63555939	3	3	4	12
28236	89869032	1	1	7	9
36900	77391171	2	1	4	7
40926	85504905	1	3	7	7
42570	77586282	1	6	7	10
62256	49726791	3	1	2	1
73578	86328819	1	3	7	12
77076	92519352	1	1	7	4
84222	108662661	1	1	7	3
89682	107389323	1	1	7	5
148530	69422211	3	6	2	6
150006	22864131	2	1	4	2
150048	21239181	2	1	4	2
182796	63000108	2	1	4	2
183930	107400762	2	6	1	11
216156	62718876	3	1	2	3
221634	21861756	1	1	7	1
236316	40523301	1	3	7	6

TRANSFORM

Add Previous steps

- Custom Transform
- Custom formula
- Encode categorical
- Featurize date/time
- Featurize text
- Format string
- Handle missing

Replace, drop, or add indicators for missing values. [Learn more](#).

Transform Impute

Column type Numeric

The name of the column that will be created to contain the transformed data. If not set it will override the input column.

Input column diag_1

Imputing strategy Mean

Output column diag_1_imputed

Optional

Clear Preview Add

4) Click Add to add the change to the data flow

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Previewing Handle missing

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in
2278392	8222157	6	25	1	1
149190	55629189	1	1	7	3
64410	86047875	1	1	7	2
500364	82442376	1	1	7	2
16680	42519267	1	1	7	1
35754	82637451	2	1	2	3
55842	84259809	3	1	2	4
63768	114882984	1	1	7	5
12522	48330783	2	1	4	13
15738	63555939	3	3	4	12
28236	89869032	1	1	7	9
36900	77391171	2	1	4	7
40926	85504905	1	3	7	7
42570	77586282	1	6	7	10
62256	49726791	3	1	2	1
73578	86328819	1	3	7	12
77076	92519352	1	1	7	4
84222	108662661	1	1	7	3
89682	107389323	1	1	7	5
148530	69422211	3	6	2	6
150006	22864131	2	1	4	2
150048	21239181	2	1	4	2
182796	63000108	2	1	4	2
183930	107400762	2	6	1	11

TRANSFORM

Add Previous steps

- Custom Transform
- Custom formula
- Encode categorical
- Featurize date/time
- Featurize text
- Format string
- Handle missing

Replace, drop, or add indicators for missing values. [Learn more](#).

Transform Impute

Column type Numeric

The name of the column that will be created to contain the transformed data. If not set it will override the input column.

Input column diag_1

Imputing strategy Mean

Output column diag_1_imputed

Optional

Clear Preview Add

5) Repeat above steps 1 through 4 for diag_2 feature as shown below

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in,
2278392	8222157	6	25	1	1
149190	55629189	1	1	7	3
64410	86047875	1	1	7	2
500364	82442376	1	1	7	2
16680	42519267	1	1	7	1
35754	82637451	2	1	2	3
55842	84259809	3	1	2	4
63768	114882984	1	1	7	5
12522	48330783	2	1	4	13
15738	63555939	3	3	4	12
28236	89869032	1	1	7	9
36900	77391171	2	1	4	7
40926	85504905	1	3	7	7
42570	77586282	1	6	7	10
62256	49726791	3	1	2	1
73578	86328819	1	3	7	12
77076	92519352	1	1	7	4
84222	108662661	1	1	7	3
89682	107389323	1	1	7	5
148530	69422211	3	6	2	6
150006	22864131	2	1	4	2
150048	21239181	2	1	4	2
182796	63000108	2	1	4	2
183930	107400762	2	6	1	11
216156	62718876	3	1	2	3
221634	21861756	1	1	7	1
236316	40523301	1	3	7	6

TRANSFORM

Add Previous steps

- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing

Replace, drop, or add indicators for missing values. [Learn more](#).

Transform ?

Impute

Column type ?

Numeric

The name of the column that will be created to contain the transformed data. If not set it will override the input column.

Input column

diag_2

Imputing strategy ?

Mean

Output column ?

diag_2_imputed

Optional

Clear

Preview Add

6) Repeat above steps 1 through 4 for `diag_3` feature as shown below

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

patient (lo...)	diag_1 (float)	diag_2 (float)	diag_3 (float)	race (string)	gender (string)	age
250				Caucasian	Female	[0-10]
276	250	255		Caucasian	Female	[10-20]
648	250			AfricanAmerican	Female	[20-30]
8	250	403		Caucasian	Male	[30-40]
197	157	250		Caucasian	Male	[40-50]
414	411	250		Caucasian	Male	[50-60]
414	411			Caucasian	Male	[60-70]
428	492	250		Caucasian	Male	[70-80]
398	427	38		Caucasian	Female	[80-90]
434	198	486		Caucasian	Female	[90-100]
250	403	996		AfricanAmerican	Female	[100-110]
157	288	197		AfricanAmerican	Male	[110-120]
428	250	250		Caucasian	Female	[120-130]
428	411	427		Caucasian	Male	[130-140]
518	998	627		AfricanAmerican	Female	[140-150]
999	507	996		AfricanAmerican	Male	[150-160]
410	411	414		AfricanAmerican	Male	[160-170]
682	174	250		Caucasian	Female	[170-180]
402	425	416		AfricanAmerican	Male	[180-190]
737	427	714	?		Male	[190-200]
410	427	428	?		Female	[200-210]
572	456	427	?		Male	[210-220]
410	401	582		AfricanAmerican	Female	[220-230]
	715			Caucasian	Female	[230-240]
189	496	427		AfricanAmerican	Female	[240-250]
786	401	250		Other	Female	[250-260]
427	428	414		Caucasian	Male	[260-270]
996	585	250		Caucasian	Female	[270-280]

Custom formula

- Encode categorical
- Featurize date/time
- Featurize text
- Format string
- Handle missing

Replace, drop, or add indicators for missing values. [Learn more.](#)

Transform [?](#)

Impute

Column type [?](#)

Numeric

The name of the column that will be created to contain the transformed data. If not set it will override the input column.

Input column

diag_3

Imputing strategy [?](#)

Mean

Output column [?](#)

diag_3_imputed

Optional

Clear

Preview Add

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

string	gender (string)	age (string)	weight (string)	payer_code_na_fill (s...)	diag_3_imputed (float)
an	Female	[0-10)	?	?	418.5872022124472
an	Female	[10-20)	?	?	255
American	Female	[20-30)	?	?	418.5872022124472
an	Male	[30-40)	?	?	403
an	Male	[40-50)	?	?	250
an	Male	[50-60)	?	?	250
an	Male	[60-70)	?	?	418.5872022124472
an	Male	[70-80)	?	?	250
an	Female	[80-90)	?	?	38
an	Female	[90-100)	?	?	486
American	Female	[40-50)	?	?	996
American	Male	[60-70)	?	?	197
an	Female	[40-50)	?	?	250
an	Male	[80-90)	?	?	427
American	Female	[60-70)	?	?	627
American	Male	[60-70)	?	?	996
American	Male	[50-60)	?	?	414
an	Female	[50-60)	?	?	250
American	Male	[70-80)	?	?	416
	Male	[70-80)	?	?	714
	Female	[50-60)	?	?	428
	Male	[60-70)	?	?	427
American	Female	[70-80)	?	?	582
an	Female	[80-90)	?	?	418.5872022124472
American	Female	[70-80)	?	?	427
	Female	[50-60)	?	?	250

Custom formula

- Encode categorical
- Featurize date/time
- Featurize text
- Format string
- Handle missing

Replace, drop, or add indicators for missing values. [Learn more.](#)

Transform [?](#)

Impute

Column type [?](#)

Numeric

The name of the column that will be created to contain the transformed data. If not set it will override the input column.

Input column

diag_3

Imputing strategy [?](#)

Mean

Output column [?](#)

diag_3_imputed

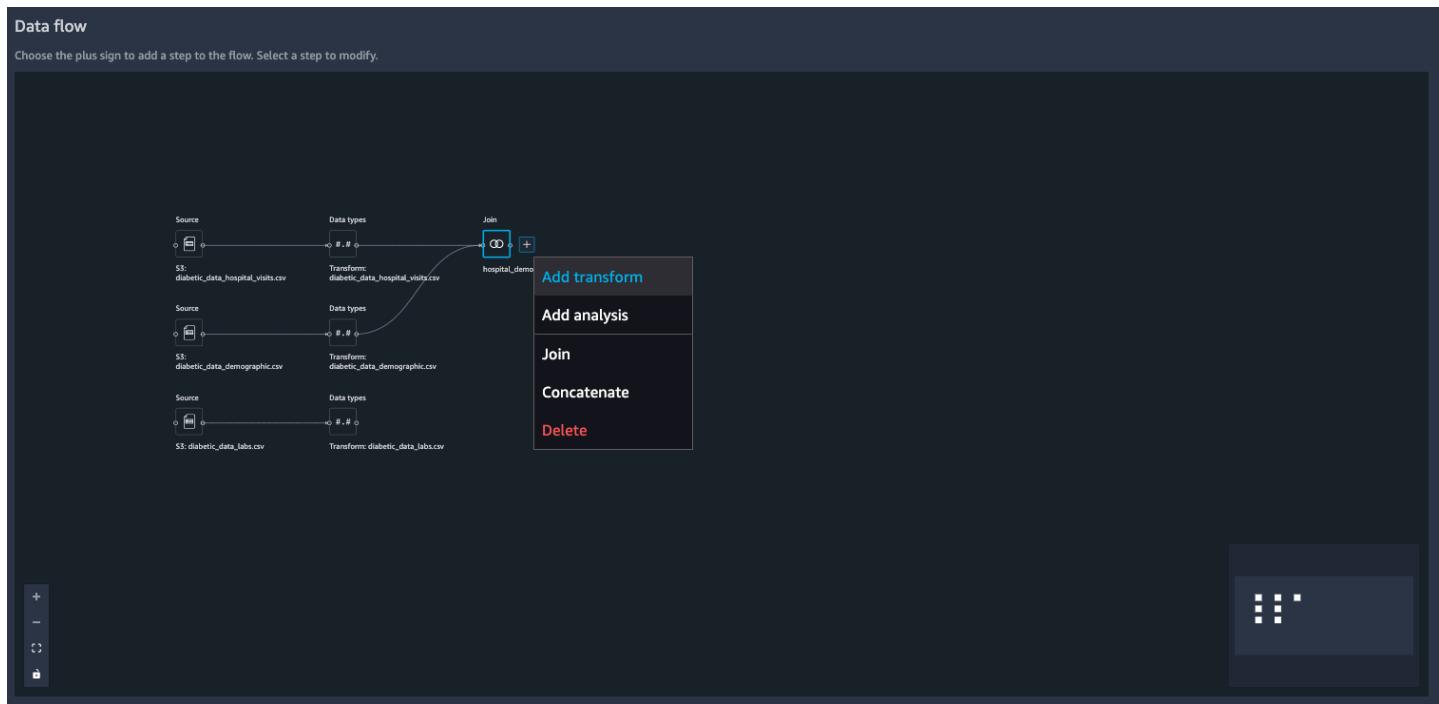
Optional

Clear

Preview Add

One-hot Encoding for categorical features

1) Click + sign next to Join flow icon and choose Add Transform



2) Pick Encode categorical from the list of transforms on the right panel. Select One-hot encode and gender for input column. For output style, choose Columns. After filling the fields click Preview

The screenshot shows the 'Data flow / Transform: 1st Join' interface. On the left, there's a preview of the data with columns: encounter_id_0, patient_nbr_0, admission_type_id, discharge_dispositio..., admission_source_id, time_in_hospital, and payer_code. On the right, the 'Encode categorical' transform configuration is shown. Under 'Transform', 'One-hot encode' is selected for 'Input column' 'gender'. Under 'Output style', 'Columns' is selected for 'Output column' 'gender_encoded'. A 'Preview' button is visible at the bottom right.

3) After review the transformation results, Click Add to add the change to the data flow

The screenshot shows the 'Data flow / Transform: 1st Join' interface after the transformation. The 'gender_encoded' column has been added to the preview. The 'Back to data flow' button is visible at the top right.

Previewing Encode categorical

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389323	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?

TRANSFORM

Add Previous steps (4)

- > Custom Transform
- > Custom formula
- > Encode categorical

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

Transform

One-hot encode

Input column gender

Input already ordinal encoded

Invalid handling strategy

Keep

Drop last

Output style

Columns

Output column gender_encoded

Optional
- > Feature date/time
- > Feature text
- > Format string
- > Handle missing

Preview Add

4) Repeat above steps 1 through 3 to encode `race` feature as shown below

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in...
2278392	8222157	6	25	1	1
149190	55629189	1	1	7	3
64410	86047875	1	1	7	2
500364	82442376	1	1	7	2
16680	42519267	1	1	7	1
35754	82637451	2	1	2	3
55842	84259809	3	1	2	4
63768	114882984	1	1	7	5
12522	48330783	2	1	4	13
15738	63555939	3	3	4	12
28236	89869032	1	1	7	9
36900	77391171	2	1	4	7
40926	85504905	1	3	7	7
42570	77586282	1	6	7	10
62256	49726791	3	1	2	1
73578	86328819	1	3	7	12
77076	92519352	1	1	7	4
84222	108662661	1	1	7	3
89682	107389323	1	1	7	5
148530	69422211	3	6	2	6
150006	22864131	2	1	4	2
150048	21239181	2	1	4	2
182796	63000108	2	1	4	2
183930	107400762	2	6	1	11
216156	62718876	3	1	2	3
221634	21861756	1	1	7	1
236316	40523301	1	3	7	6

TRANSFORM

Add Previous steps

- > Custom Transform
- > Custom formula
- > Encode categorical

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

Transform

One-hot encode

Input column race

Input already ordinal encoded

Invalid handling strategy

Skip

Drop last

Output style

Columns

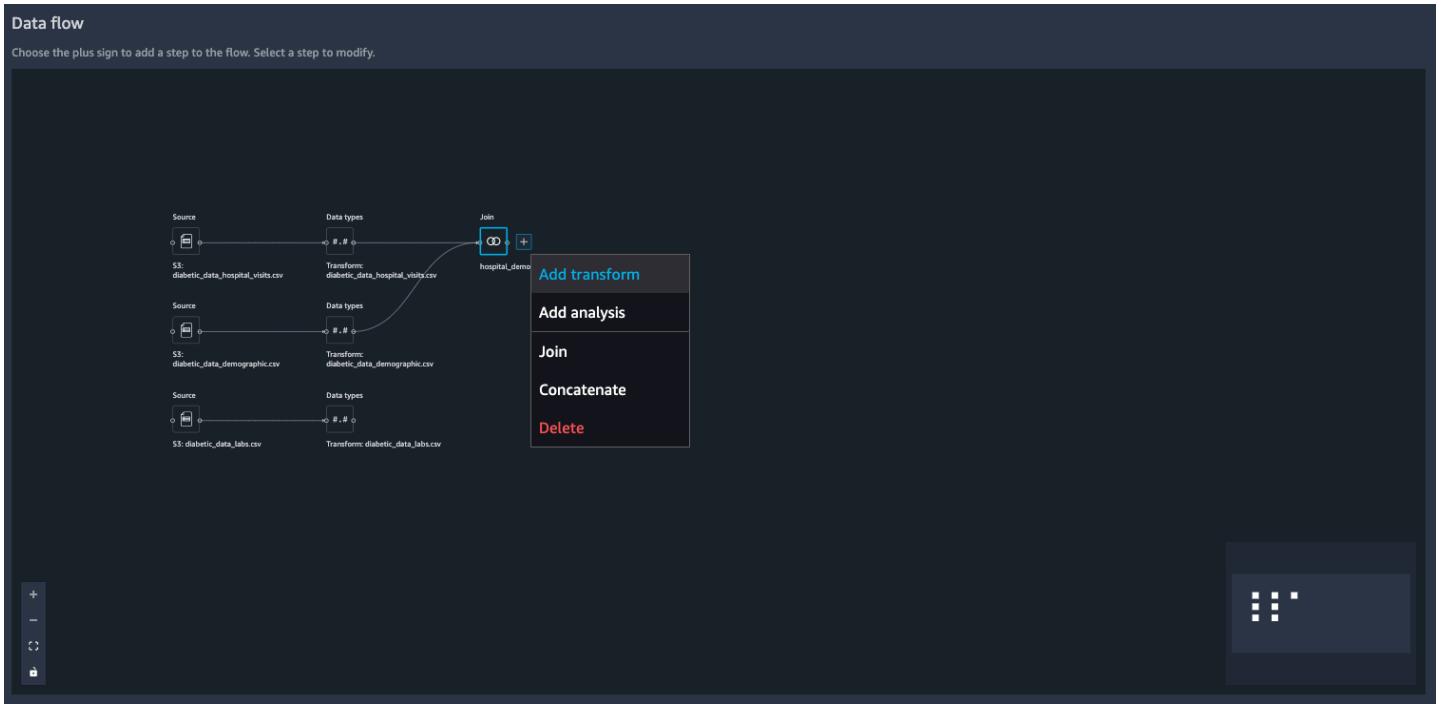
Output column race_1hot

Optional
- > Feature date/time
- > Feature text
- > Format string

Preview Add

Ordinal Encoding for categorical features

1) Click + sign next to Join flow icon and choose Add Transform



2) Pick Encode categorical from the list of transforms on the right panel. Select Ordinal encode and age for input column. For Invalid handling strategy select skip. After filling the fields click Preview

The screenshot shows the 'Prepare' tab with the 'Transform: 1st Join' step selected. The preview pane displays a portion of the joined dataset. The right panel shows the 'Encode categorical' transform configuration, which includes:

- TRANSFORM**: Add, Previous steps (4)
- Custom Transform**
- Custom formula**
- Encode categorical** (selected)
 - Convert categorical variables to numeric or vector representations. [Learn more.](#)
 - Transform**: Ordinal encode
 - Input column**: age
 - Output column**: age_ordinal_encoded
 - Optional**
 - Invalid handling strategy**: Skip
- Featurize date/time**
- Featurize text**
- Format string**
- Handle missing**
- Handle outliers**
- Manage columns**
- Manage rows**

3) After review the transformation results, Click Add to add the change to the data flow

The screenshot shows the 'Prepare' tab with the 'Transform: 1st Join' step selected. The preview pane displays the transformed dataset. The right panel shows the 'Add' button highlighted in orange.

Previewing Encode categorical

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389323	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?

TRANSFORM

Add Previous steps (4)

- > Custom Transform
- > Custom formula
- > Encode categorical

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

Ordinal encode

Input column [age](#)

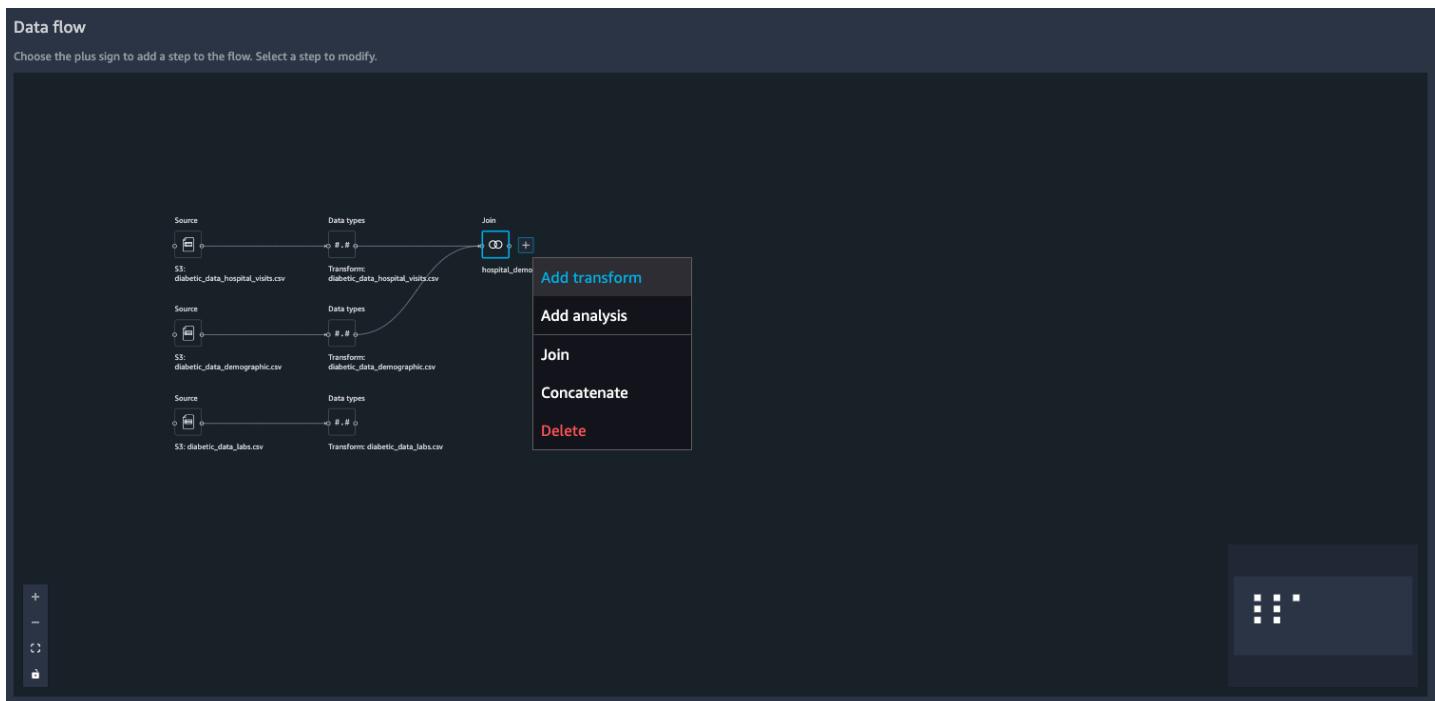
Output column [age_ordinal_encoded](#)

Optional

Invalid handling strategy [Skip](#)
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns
- > Manage rows

Normalization with standard scaler using Process numeric Transform

1) Click + sign next to Join flow icon and choose Add Transform



2) Pick Process numeric from the list of transforms on the right panel. Select Scale values for Transform and Standard scaler for Scaler. Provide diag_1_imputed as Input column. After filling the fields click

and Standard Scaler) for Scaler. Provide diag_1_imputed as Input column, after hitting the Next button.

Preview

Import **Prepare** Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

id	race (string)	gender (string)	age (string)	weight (string)	diag_1_imputed (float)
1	Caucasian	Female	[0-10)	0	250
2	Caucasian	Female	[10-20)	0	276
3	AfricanAmerican	Female	[20-30)	0	648
4	Caucasian	Male	[30-40)	0	8
5	Caucasian	Male	[40-50)	0	197
6	Caucasian	Male	[50-60)	0	414
7	Caucasian	Male	[60-70)	0	414
8	Caucasian	Male	[70-80)	0	428
9	Caucasian	Female	[80-90)	0	398
10	Caucasian	Female	[90-100)	0	434
11	AfricanAmerican	Female	[40-50)	0	250
12	AfricanAmerican	Male	[60-70)	0	157
13	Caucasian	Female	[40-50)	0	428
14	Caucasian	Male	[80-90)	0	428
15	AfricanAmerican	Female	[60-70)	0	518
16	AfricanAmerican	Male	[60-70)	0	999
17	AfricanAmerican	Male	[50-60)	0	410
18	Caucasian	Female	[50-60)	0	682
19	AfricanAmerican	Male	[70-80)	0	402
20	Other	Male	[70-80)	0	737
21	Other	Female	[50-60)	0	410
22	Other	Male	[60-70)	0	572
23	AfricanAmerican	Female	[70-80)	0	410
24	Caucasian	Female	[80-90)	0	489.94228256609034
25	AfricanAmerican	Female	[70-80)	0	189
26	Other	Female	[50-60)	0	786
27	Caucasian	Male	[80-90)	0	427

< Back to data flow

Handle missing

Handle outliers

Manage columns

Manage rows

Manage vectors

Parse column as type

Process numeric

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

Transform

Scale values

Scaler

Standard scaler

Rescale the column to have unit standard deviation.

Input column

diag_1_imputed

Center

Scale

Output column

diag_1_scaler

Optional

Clear

Preview Add

Search and edit

Validate string

3) After review the transformation results, Click **Add** to add the change to the data flow

Import **Prepare** Analyze Export

Data flow / Transform: 1st Join

Previewing Process numeric

Transform: 1st Join

gender (string)	age (string)	weight (string)	diag_1_imputed (float)	diag_1_scaler (float)
Female	[0-10)	0	250	1.2191437876792175
Female	[10-20)	0	276	1.345934741597856
Female	[20-30)	0	648	3.1600206976645318
Male	[30-40)	0	8	0.03901260120573496
Male	[40-50)	0	197	0.9606853046912234
Male	[50-60)	0	414	2.018902112396784
Male	[60-70)	0	414	2.018902112396784
Male	[70-80)	0	428	2.0871741645068203
Female	[80-90)	0	398	1.9408769099853143
Female	[90-100)	0	434	2.1164336154111214
Female	[40-50)	0	250	1.2191437876792175
Male	[60-70)	0	157	0.7656222986625485
Female	[40-50)	0	428	2.0871741645068203
Male	[80-90)	0	428	2.0871741645068203
Female	[60-70)	0	518	2.5260659280713385
Male	[60-70)	0	999	4.871698575566153
Male	[50-60)	0	410	1.9993958117939166
Female	[50-60)	0	682	3.3258242527889053
Male	[70-80)	0	402	1.9603832105881815
Male	[70-80)	0	737	3.594035886078353
Female	[50-60)	0	410	1.9993958117939166
Male	[60-70)	0	572	2.7894009862100493
Female	[70-80)	0	410	1.9993958117939166
Female	[80-90)	0	489.94228256609034	2.389240360447299
Female	[70-80)	0	100	0.02167370748516024

Handle missing

Handle outliers

Manage columns

Manage rows

Manage vectors

Parse column as type

Process numeric

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

Scale values

Standard scaler

Rescale the column to have unit standard deviation.

Input column: diag_1_imputed

Center:

Scale:

Output column: diag_1_scaler

Optional

Clear

Preview Add

Search and edit

Validate string

4) Repeat above steps 1 through 3 to apply scaler for `diag_2_imputed` & `diag_3_imputed` feature as shown below

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (long)	discharge_disposition....	admission_source_id (l...	time_in_hospital (long)	payer
2278392	8222157	6	25	1	1	0
149190	55629189	1	1	7	3	0
64410	86047875	1	1	7	2	0
500364	82442376	1	1	7	2	0
16680	42519267	1	1	7	1	0
35754	82637451	2	1	2	3	0
55842	84259809	3	1	2	4	0
63768	114862984	1	1	7	5	0
12522	48330783	2	1	4	13	0
15738	63555939	3	3	4	12	0
28236	89869032	1	1	7	9	0
36900	77391171	2	1	4	7	0
40926	85504905	1	3	7	7	0
42570	77586282	1	6	7	10	0
62256	49726791	3	1	2	1	0
73578	86328819	1	3	7	12	0
77076	92519352	1	1	7	4	0
84222	108662661	1	1	7	3	0
89682	107389323	1	1	7	5	0
148530	69422211	3	6	2	6	0
150006	22864131	2	1	4	2	0
150048	21239181	2	1	4	2	0
182796	63000108	2	1	4	2	0
183930	107400762	2	6	1	11	0
216156	62718876	3	1	2	3	0
221634	21861756	1	1	7	1	0
236316	40523301	1	3	7	6	0

Handle missing

Handle outliers

Manage columns

Manage rows

Manage vectors

Parse column as type

Process numeric

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

Scale values

Standard scaler

Rescale the column to have unit standard deviation.

Input column: diag_2_imputed

Center:

Scale:

Output column: diag_2_imputed_scaler

Optional

Clear

Preview Add

Search and edit

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Back to data flow

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (lo...)	discharge_disposition_...	admission_source_id (l...	time_in_hospital (long)	payer
2278392	8222157	6	25	1	1	0
149190	55629189	1	1	7	3	0
64410	86047875	1	1	7	2	0
500364	82442376	1	1	7	2	0
16680	42519267	1	1	7	1	0
35754	82637451	2	1	2	3	0
55842	84259809	3	1	2	4	0
63768	114882984	1	1	7	5	0
12522	48330783	2	1	4	13	0
15738	63555939	3	3	4	12	0
28236	89869032	1	1	7	9	0
36900	77391171	2	1	4	7	0
40926	85504905	1	3	7	7	0
42570	77586282	1	6	7	10	0
62256	49726791	3	1	2	1	0
73578	86328819	1	3	7	12	0
77076	92519352	1	1	7	4	0
84222	108662661	1	1	7	3	0
89682	107389523	1	1	7	5	0
148530	69422211	3	6	2	6	0
150006	22864131	2	1	4	2	0
150048	21239181	2	1	4	2	0
182796	63000108	2	1	4	2	0
183930	107400762	2	6	1	11	0
216156	62718876	3	1	2	3	0
221634	21861756	1	1	7	1	0
236316	40523301	1	3	7	6	0

Export Data

Format string

Handle missing

Handle outliers

Manage columns

Manage rows

Manage vectors

Parse column as type

Process numeric

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

Scale values

Scaler

Standard scaler

Rescale the column to have unit standard deviation.

Input column

diag_3_imputed

Center

Scale

Output column

diag_3_imputed_scaler

Optional

Clear

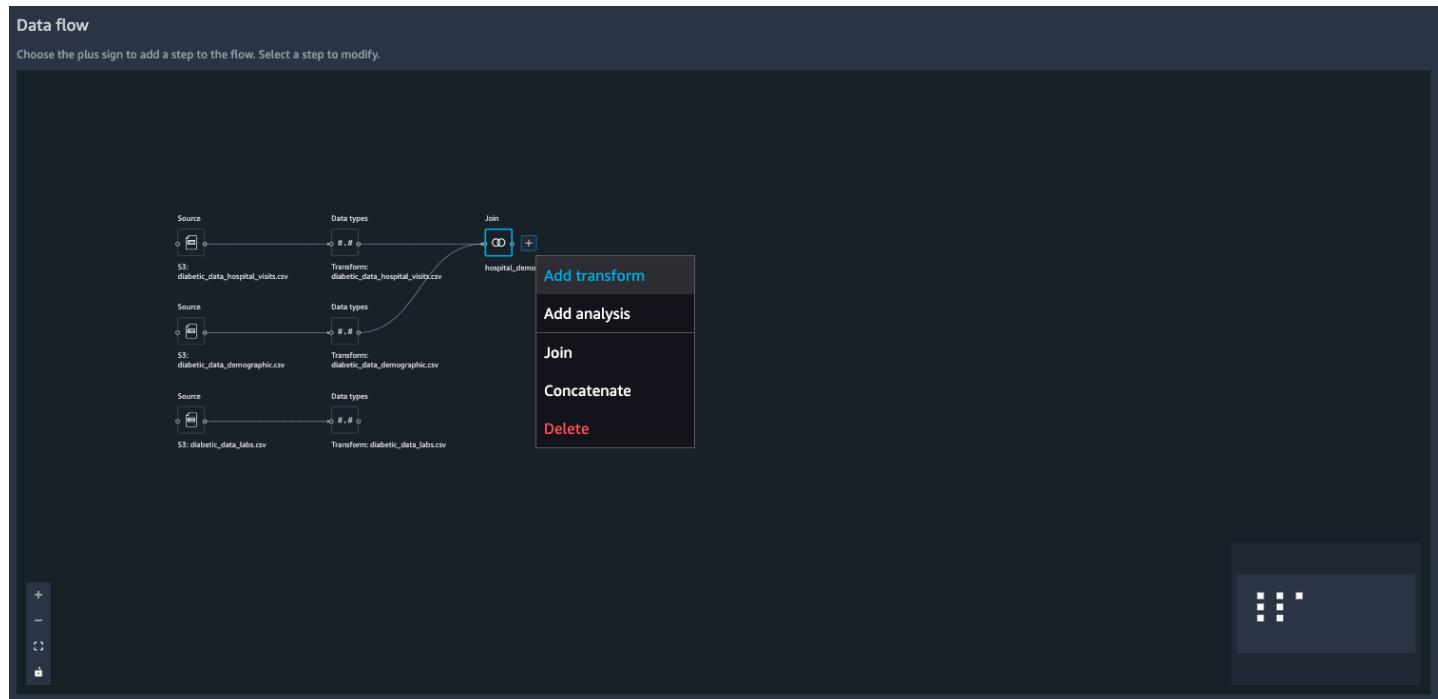
Preview Add

Search and edit

Transform the target Label

Our Target Label needs a couple transformations to be effective. Let's use **Search and Edit** Transform to convert string values to binary values.

1) Click + sign next to Join flow icon and choose **Add Transform**



2) Pick **Search and edit** from the list of transforms on the right panel. Select **Find and replace substring** as shown below

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

nber_inpatient (long)	diag_1 (float)	diag_2 (float)	diag_3 (float)	readmitted (string)	race (string)	gender (st)
250				NO	Caucasian	Female
276	250	255		>30	Caucasian	Female
648	250			NO	AfricanAmerican	Female
8	250	403		NO	Caucasian	Male
197	157	250		NO	Caucasian	Male
414	411	250		>30	Caucasian	Male
414	411			NO	Caucasian	Male
428	492	250		>30	Caucasian	Male
398	427	38		NO	Caucasian	Female
434	198	486		NO	Caucasian	Female
250	403	996		>30	AfricanAmerican	Female
157	288	197		<30	AfricanAmerican	Male
428	250	250		<30	Caucasian	Female
428	411	427		NO	Caucasian	Male
518	998	627		>30	AfricanAmerican	Female
999	507	996		NO	AfricanAmerican	Male
410	411	414		<30	AfricanAmerican	Male
682	174	250		NO	Caucasian	Female
402	425	416		>30	AfricanAmerican	Male
737	427	714		NO	Other	Male
410	427	428		NO	Other	Female
572	456	427		NO	Other	Male
410	401	582		NO	AfricanAmerican	Female
		715		>30	Caucasian	Female
189	496	427		NO	AfricanAmerican	Female
786	401	250		NO	Other	Female
427	428	414		NO	Caucasian	Male
...

Export Data

Back to data flow

- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > 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 column

Pattern

Replacement string

Output column

Optional
- Clear
- Preview Add
- > Validate string

3) Select the target column `readmitted` for `Input column` and use `>30|<30` regex for `Pattern`. For the `Replacement String` use `1`. Also, give a name to your new `output column` as `readmitted_parser_1`. So, here we are converting all the values that have either `>30` or `<30` values to `1`. After making your config selections, hit `Preview` to review the converted column

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Back to data flow

Previewing Search and edit

Transform: 1st Join

(f...)	race_1hot_Asian (float)	age_ordinal (float)	diag_1_imputed_scale...	diag_2_imputed_scale...	diag_3_imputed_scale...	readmitted_parser_1 (s...
0	9	1.2320095759161842	2.461510423177554	2.435538630906403	NO	
0	8	1.3601385718114674	1.4361949029038146	1.5192558154451987	1	
0	7	3.1933688207747495	1.4361949029038146	2.435538630906403	NO	
0	5	0.03942430642931789	1.4361949029038146	2.401019974997706	NO	
0	4	0.9708235458219531	0.9019303990235956	1.4894664857305868	NO	
0	2	2.040207857717201	2.3611044203738714	1.4894664857305868	1	
0	1	2.040207857717201	2.3611044203738714	2.435538630906403	NO	
0	0	2.109200393968507	2.8264315689147073	1.4894664857305868	1	
0	3	1.9613592448585653	2.4530208941597156	0.2263989058310492	NO	
0	6	2.1387686237904955	1.1374663630998212	2.895522848260261	NO	
0	4	1.2320095759161842	2.315146183480949	5.934034479150658	1	
0	1	0.7737020136753636	1.6544965281451944	1.1736995907557024	1	
0	4	2.109200393968507	1.4361949029038146	1.4894664857305868	1	
0	3	2.109200393968507	2.3611044203738714	2.5440087576278425	NO	
0	1	2.5527238412983335	5.733290052392028	3.735581946212312	1	
0	1	4.923110265361072	2.9126032630889362	5.934034479150658	NO	
0	2	2.020495704502542	2.3611044203738714	2.466556500369852	1	
0	2	3.3609221230993502	0.999591652421055	1.4894664857305868	NO	
0	0	1.9810713980732242	2.441531334936485	2.4784722322556965	1	
0	0	3.6319642298009107	2.4530208941597156	4.253916283246556	NO	
0	2	2.020495704502542	2.4530208941597156	2.5499666235707648	NO	
0	1	2.8188379096962293	2.619619502896558	2.5440087576278425	NO	
0	0	2.020495704502542	2.3036566242577186	3.4674779787808063	NO	

Export Data

- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > 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 column: readmitted

Pattern: >30|<30

Replacement string: 1

Output column: readmitted_parser_1

Optional

Clear Preview Add

4) Once reviewed, click Add to add the transform to your data-flow.

5) Let's repeat the same to convert `NO` in the new target column to `0`. Select the target column

`readmitted_parser_1` for `Input column` and use `NO` regex for Pattern. For the `Replacement String` use `0`. Also, give a name to your new `output column` as `readmitted_parser_2`. After making your config selections, hit `Preview` to review the converted column

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Back to data flow

Transform: 1st Join

{fl...	race_1hot_Asian (float)	age_ordinal (float)	diag_1_imputed_scale...	diag_2_imputed_scale...	diag_3_imputed_scale...	readmitted_parser_1 (s...
0	9	1.2320095759161842	2.461510423177554	2.435538630906403	NO	
0	8	1.3601385718114674	1.4361949029038146	1.5192558154451987	1	
0	7	3.1933688207747495	1.4361949029038146	2.435538630906403	NO	
0	5	0.03942430642931789	1.4361949029038146	2.401019974997706	NO	
0	4	0.9708235458219531	0.9019303990235956	1.4894664857305868	NO	
0	2	2.040207857717201	2.3611044203738714	1.4894664857305868	1	
0	1	2.040207857717201	2.3611044203738714	2.435538630906403	NO	
0	0	2.109200393968507	2.8264315689147073	1.4894664857305868	1	
0	3	1.9613592448585653	2.4530208941597156	0.2263989058310492	NO	
0	6	2.1387686237904955	1.1374663630998212	2.895522848260261	NO	
0	4	1.2320095759161842	2.315146183480949	5.934034479150658	1	
0	1	0.7737020136753636	1.6544965281451944	1.1736995907557024	1	
0	4	2.109200393968507	1.4361949029038146	1.4894664857305868	1	
0	3	2.109200393968507	2.3611044203738714	2.5440087576278425	NO	
0	1	2.5527238412983335	5.733290052392028	3.735581946212312	1	
0	1	4.923110265361072	2.9126032630889362	5.934034479150658	NO	
0	2	2.020495704502542	2.3611044203738714	2.466556500369852	1	
0	2	3.3609221230993502	0.999591652421055	1.4894664857305868	NO	
0	0	1.9810713980732242	2.441531334936485	2.478472232556965	1	
0	0	3.6319642298009107	2.4530208941597156	4.253916283246556	NO	
0	2	2.020495704502542	2.4530208941597156	2.5499666235707648	NO	
0	1	2.81883790596962293	2.619619502896558	2.5440087576278425	NO	
0	0	2.020495704502542	2.3036566242577186	3.4674779787808063	NO	
0	3	2.4119467087055853	4.10751742230491	2.435538630906403	1	
0	0	0.9313992393926352	2.849410687361168	2.5440087576278425	NO	
0	2	3.873438106680483	2.3036566242577186	1.4894664857305868	NO	
0	3	2.1042723556648424	2.458765673771331	2.466556500369852	NO	

Format string

Handle missing

Handle outliers

Manage columns

Manage rows

Manage vectors

Parse column as type

Process numeric

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 column

readmitted_parser_1

Pattern

NO

Replacement string

0

Output column

readmitted_parser_2

Optional

Clear

Preview

Add

6) Once reviewed, click Add to add the transform to your data-flow. Now our target label is ready for ML.

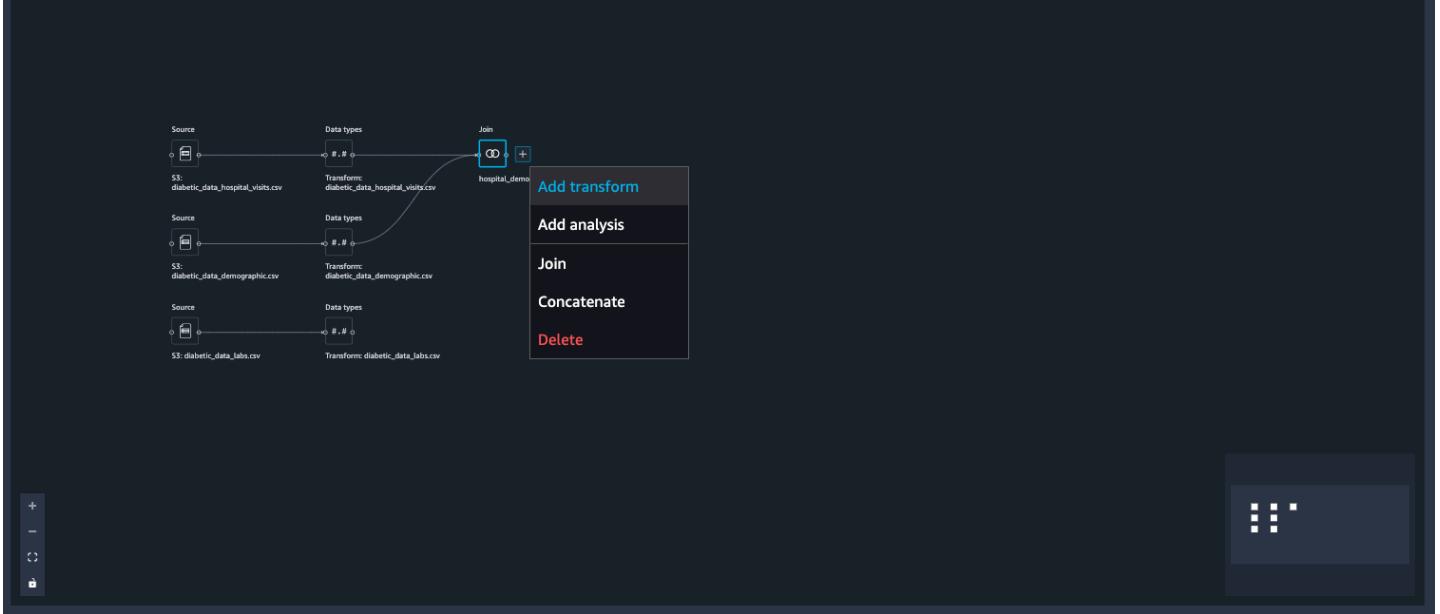
Position the target label as first column to utilize XGBoost algorithm

We will be using XGBoost built-in SageMaker algorithm to train the model using our dataset. For CSV training, the algorithm assumes that the target label is in the first column. Let's do that.

1) Click + sign next to Join flow icon and choose

Data flow

Choose the plus sign to add a step to the flow. Select a step to modify.



2) Pick **Manage Column** from the list of transforms on the right panel. Select **Move Column** for **Transform** and select **Move to start** for **Move type**. Provide a name to new column `readmitted_1hot_NO`. After filling the fields click **Preview**

ikno...	age_ordinal (float)	diag_1_scaled (float)	readmitted_1hot_N...	readmitted_1hot_>3...	readmitted_1hot_<30...
9	1.219143787692175	1	0	0	0
8	1.345934741597856	0	1	0	0
7	3.1600206976645318	1	0	0	0
5	0.03901260120573496	1	0	0	0
4	0.9606853046912234	1	0	0	0
2	2.018902112396784	0	1	0	0
1	2.018902112396784	1	0	0	0
0	2.0871741645068203	0	1	0	0
3	1.9408769099853143	1	0	0	0
6	2.1164336154111214	1	0	0	0
4	1.219143787692175	0	1	0	0
1	0.7656222986625485	0	0	1	0
4	2.0871741645068203	0	0	1	0
3	2.0871741645068203	1	0	0	0
1	2.5260659280713385	0	1	0	0
1	4.87169857556153	1	0	0	0
2	1.9993958117939166	0	0	1	0
2	3.3258242527889053	1	0	0	0
0	1.9603852105881815	0	1	0	0
0	3.594035886078333	1	0	0	0
2	1.9993958117939166	1	0	0	0
1	2.7894009862100493	1	0	0	0
0	1.9993958117939166	1	0	0	0
3	2.389240360447299	0	1	0	0
0	0.9216727034854884	1	0	0	0
2	3.8329880084634596	1	0	0	0
3	2.0822975893561035	1	0	0	0

3) After review the transformation results, Click **Add** to add the change to the data flow

Previewing Manage columns

Transform: 1st Join

id (f...)	gender_1hot_Unc... 9	age_ordinal (float) 8	diag_1_scaled (float) 1.345934741597856	readmitted_1hot_>3... 0	readmitted_1hot_<30... 0
0	0	0	3.1600206976645318	0	0
0	0	5	0.03901260120573496	0	0
0	0	4	0.9606853046912234	0	0
0	0	2	2.018902112396784	1	0
0	0	1	2.018902112396784	0	0
0	0	0	2.0871741645068203	1	0
0	0	3	1.9408769099853143	0	0
0	0	6	2.1164336154111214	0	0
0	0	4	1.2191437876792175	1	0
0	0	1	0.765622986625485	0	1
0	0	4	2.0871741645068203	0	1
0	0	3	2.0871741645068203	0	0
0	0	1	2.5260659280713385	1	0
0	0	1	4.871698575566153	0	0
0	0	2	1.9993958117939166	0	1
0	0	2	3.3258242527889053	0	0
0	0	0	1.9603832105881815	1	0
0	0	0	3.594035886078333	0	0
0	0	2	1.9993958117939166	0	0
0	0	1	2.7894009862100493	0	0
0	0	0	1.9993958117939166	0	0
0	0	3	2.389240360447299	1	0
0	0	0	0.9216727034854884	0	0

Add Previous steps

- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

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

Transform

Move column

Move type

Move to start

Moves the column so it is the first column in the dataset.

Column to move

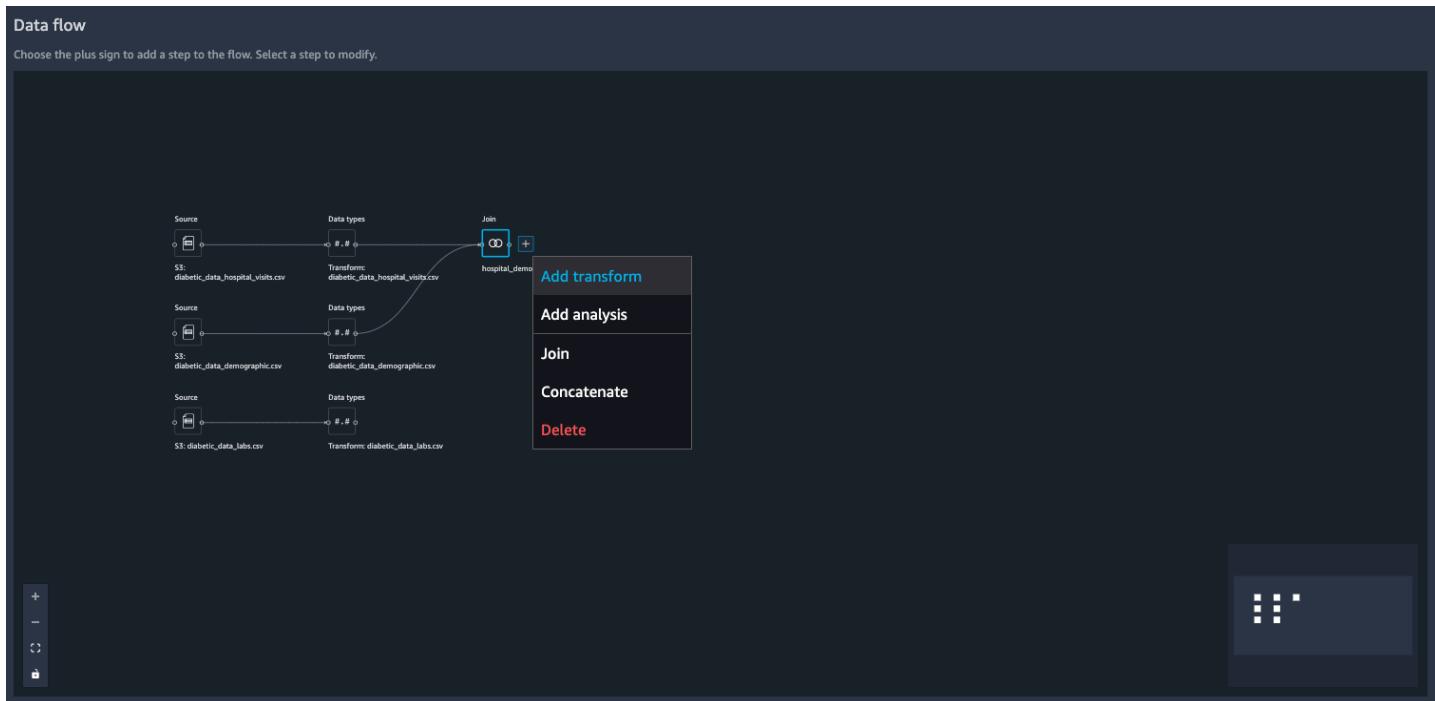
readmitted_1hot_NO

Clear Preview Add

- > Manage rows
- > Manage vectors
- > Parse column as type

Drop redundant columns

1) Click + sign next to Join flow icon and choose **Add Transform**



2) Pick **Manage Columns** from the list of transforms on the right panel

Import Prepare Analyze Export

Data flow / hospital_demographic_join
hospital_demographic_join

Back to data flow

The screenshot shows a data preparation interface with a table of patient records on the left and a sidebar with a 'Transform' menu on the right.

Table Data:

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in_hospital (long)	payer_code
2278392	8222157	6	25	1	1	?
149190	55629189	1	1	7	3	?
64410	86047875	1	1	7	2	?
500364	82442376	1	1	7	2	?
16680	42519267	1	1	7	1	?
35754	82637451	2	1	2	3	?
55842	84259809	3	1	2	4	?
63768	114882984	1	1	7	5	?
12522	48330783	2	1	4	13	?
15738	63555939	3	3	4	12	?
28236	89869032	1	1	7	9	?
36900	77391171	2	1	4	7	?
40926	85504905	1	3	7	7	?
42570	77586282	1	6	7	10	?
62256	49726791	3	1	2	1	?
73578	86328819	1	3	7	12	?
77076	92519352	1	1	7	4	?
84222	108662661	1	1	7	3	?
89682	107389323	1	1	7	5	?
148530	69422211	3	6	2	6	?
150006	22864131	2	1	4	2	?
150048	21239181	2	1	4	2	?
182796	63000108	2	1	4	2	?
183930	107400762	2	6	1	11	?
216156	62718876	3	1	2	3	?
221634	21861756	1	1	7	1	?
236316	40523301	1	3	7	6	?
248916	115196778	1	1	1	2	?

Transform Sidebar:

- TRANSFORM
 - Add Previous steps
 - Custom Transform
 - Custom formula
 - Encode categorical
 - Featurize date/time
 - Featurize text
 - Format string
 - Handle missing
 - Handle outliers
 - Manage columns
 - Move, drop, duplicate or rename columns in the dataset. [Learn more](#).
 - Transform
 - Drop column
 - Duplicate column
 - Rename column
 - Move column
 - Manage rows
 - Manage vectors
 - Parse column as type
 - Process numeric
 - Search and edit
 - Validate string

3) Choose **Drop Column** transform and select **age** column to drop

The screenshot shows a data preparation interface with a table of patient records on the left and a sidebar with a 'Transform' menu on the right.

Table Data:

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in...
2278392	8222157	6	25	1	1
149190	55629189	1	1	7	3
64410	86047875	1	1	7	2
500364	82442376	1	1	7	2
16680	42519267	1	1	7	1
35754	82637451	2	1	2	3
55842	84259809	3	1	2	4
63768	114882984	1	1	7	5
12522	48330783	2	1	4	13
15738	63555939	3	3	4	12
28236	89869032	1	1	7	9
36900	77391171	2	1	4	7
40926	85504905	1	3	7	7
42570	77586282	1	6	7	10
62256	49726791	3	1	2	1
73578	86328819	1	3	7	12
77076	92519352	1	1	7	4
84222	108662661	1	1	7	3
89682	107389323	1	1	7	5
148530	69422211	3	6	2	6
150006	22864131	2	1	4	2
150048	21239181	2	1	4	2
182796	63000108	2	1	4	2
183930	107400762	2	6	1	11
216156	62718876	3	1	2	3
221634	21861756	1	1	7	1
236316	40523301	1	3	7	6

Transform Sidebar:

- Add Previous steps
- Custom Transform
- Custom formula
- Encode categorical
- Featurize date/time
- Featurize text
- Format string
- Handle missing
- Handle outliers
- Manage columns
 - Move, drop, duplicate or rename columns in the dataset. [Learn more](#).
- Transform
 - Drop column
 - Column to drop
 - age
 - Preview Add
- Clear
- Manage rows
- Manage vectors
- Parse column as type
- Process numeric
- Search and edit
- Validate string

4) Click **Preview** to preview the changes to the data set. Then **Add** to add the changes to flow file.

5) Repeat above steps 1 through 4 for other redundant columns **race** **gender** **diag_1** **diag_2** **diag_3** **diag_1_imputed** **diag_2_imputed** **diag_3_imputed** **readmitted** **readmitted_parser_1** as shown below

Import **Prepare** Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

encounter_id_0 (long)	patient_nbr_0 (long)	admission_type_id (l...)	discharge_dispositio...	admission_source_id ...	time_in,
2278392	8222157	6	25	1	1
149190	55629189	1	1	7	3
64410	86047875	1	1	7	2
500364	82442376	1	1	7	2
16680	42519267	1	1	7	1
35754	82637451	2	1	2	3
55842	84259809	3	1	2	4
63768	114882984	1	1	7	5
12522	48330783	2	1	4	13
15738	63555939	3	3	4	12
28236	89869032	1	1	7	9
36900	77391171	2	1	4	7
40926	85504905	1	3	7	7
42570	77586282	1	6	7	10
62256	49726791	3	1	2	1
73578	86328819	1	3	7	12
77076	92519352	1	1	7	4
84222	108662661	1	1	7	3
89682	107389323	1	1	7	5
148530	69422211	3	6	2	6
150006	22864131	2	1	4	2
150048	21239181	2	1	4	2
182796	63000108	2	1	4	2
183930	107400762	2	6	1	11
216156	62718876	3	1	2	3
221634	21861756	1	1	7	1
236316	40523301	1	3	7	6

Back to data flow

Add Previous steps

- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

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

Drop column

Column to drop
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit
- > Validate string

Import **Prepare** Analyze Export

Data flow / Transform: 1st Join

Back to data flow

Transform: 1st Join

diag_3 (float)	readmitted (string)	gender (string)	weight (string)	eventTime (float)	diag_1.
255	NO	Female	0	1625601807.674491	250
	>30	Female	0	1625601807.674491	276
403	NO	Male	0	1625601807.674491	648
250	NO	Male	0	1625601807.674491	197
250	>30	Male	0	1625601807.674491	414
	NO	Male	0	1625601807.674491	414
250	>30	Male	0	1625601807.674491	428
38	NO	Female	0	1625601807.674491	398
486	NO	Female	0	1625601807.674491	434
996	>30	Female	0	1625601807.674491	250
197	<30	Male	0	1625601807.674491	157
250	<30	Female	0	1625601807.674491	428
427	NO	Male	0	1625601807.674491	428
627	>30	Female	0	1625601807.674491	518
996	NO	Male	0	1625601807.674491	999
414	<30	Male	0	1625601807.674491	410
250	NO	Female	0	1625601807.674491	682
416	>30	Male	0	1625601807.674491	402
714	NO	Male	0	1625601807.674491	737
428	NO	Female	0	1625601807.674491	410
427	NO	Male	0	1625601807.674491	572
582	NO	Female	0	1625601807.674491	410
	>30	Female	0	1625601807.674491	489.4
427	NO	Female	0	1625601807.674491	189
250	NO	Female	0	1625601807.674491	786
414	NO	Male	0	1625601807.674491	427
...

Export Data

TRANSFORM

- Add Previous steps (18)
- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

Move, drop, duplicate or rename columns in the dataset. [Learn more](#)
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit

Transform

Drop column

Column to drop

gender

Clear

Preview Add

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

diag_3 (float)	readmitted (string)	weight (string)	eventTime (float)	diag_1_imputed (float)	diag_2.
255	NO	0	1625601807.674491	250	428.4
	>30	0	1625601807.674491	276	250
403	NO	0	1625601807.674491	8	250
250	NO	0	1625601807.674491	197	157
250	>30	0	1625601807.674491	414	411
	NO	0	1625601807.674491	414	411
250	>30	0	1625601807.674491	428	492
38	NO	0	1625601807.674491	398	427
486	NO	0	1625601807.674491	434	198
996	>30	0	1625601807.674491	250	403
197	<30	0	1625601807.674491	157	288
250	<30	0	1625601807.674491	428	250
427	NO	0	1625601807.674491	428	411
627	>30	0	1625601807.674491	518	998
996	NO	0	1625601807.674491	999	507
414	<30	0	1625601807.674491	410	411
250	NO	0	1625601807.674491	682	174
416	>30	0	1625601807.674491	402	425
714	NO	0	1625601807.674491	737	427
428	NO	0	1625601807.674491	410	427
427	NO	0	1625601807.674491	572	456
582	NO	0	1625601807.674491	410	401
	>30	0	1625601807.674491	489.4334337685526	715
427	NO	0	1625601807.674491	189	496
250	NO	0	1625601807.674491	786	401
414	NO	0	1625601807.674491	427	428
...

Export Data

TRANSFORM

- Add Previous steps (18)
- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

Move, drop, duplicate or rename columns in the dataset. [Learn more](#)
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit
- > Validate string

Transform

Drop column

Column to drop

diag_1

Clear

Preview Add

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Back to data flow

Transform: 1st Join

readmitted (string)	weight (string)	eventTime (float)	diag_1_imputed (float)	diag_2_imputed (float)	diag_3 (float)
NO	0	1625601807.674491	250	428.4777814976007	408.7
>30	0	1625601807.674491	276	250	255
NO	0	1625601807.674491	648	250	408.7
NO	0	1625601807.674491	8	250	403
NO	0	1625601807.674491	197	157	250
>30	0	1625601807.674491	414	411	250
NO	0	1625601807.674491	414	411	408.7
>30	0	1625601807.674491	428	492	250
NO	0	1625601807.674491	398	427	38
NO	0	1625601807.674491	434	198	486
>30	0	1625601807.674491	250	403	996
<30	0	1625601807.674491	157	288	197
<30	0	1625601807.674491	428	250	250
NO	0	1625601807.674491	428	411	427
>30	0	1625601807.674491	518	998	627
NO	0	1625601807.674491	999	507	996
<30	0	1625601807.674491	410	411	414
NO	0	1625601807.674491	682	174	250
>30	0	1625601807.674491	402	425	416
NO	0	1625601807.674491	737	427	714
NO	0	1625601807.674491	410	427	428
NO	0	1625601807.674491	572	456	427
NO	0	1625601807.674491	410	401	582
>30	0	1625601807.674491	489.4334337685526	715	408.7
NO	0	1625601807.674491	189	496	427
NO	0	1625601807.674491	786	401	250
NO	0	1625601807.674491	427	428	414

TRANSFORM

- Add Previous steps (18)
- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

Move, drop, duplicate or rename columns in the dataset. [Learn more.](#)
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

gency (...	number_inpatient (lo...	diag_3 (float)	readmitted (string)	race (string)	gender (string)
0			NO	Caucasian	Female
0	255	>30	Caucasian	Female	
1		NO	AfricanAmerican	Female	
0	403	NO	Caucasian	Male	
0	250	NO	Caucasian	Male	
0	250	>30	Caucasian	Male	
0		NO	Caucasian	Male	
0	250	>30	Caucasian	Male	
0	38	NO	Caucasian	Female	
0	486	NO	Caucasian	Female	
0	996	>30	AfricanAmerican	Female	
0	197	<30	AfricanAmerican	Male	
0	250	<30	Caucasian	Female	
0	427	NO	Caucasian	Male	
0	627	>30	AfricanAmerican	Female	
0	996	NO	AfricanAmerican	Male	
0	414	<30	AfricanAmerican	Male	
0	250	NO	Caucasian	Female	
0	416	>30	AfricanAmerican	Male	
0	714	NO	Other	Male	
0	428	NO	Other	Female	
0	427	NO	Other	Male	
0	582	NO	AfricanAmerican	Female	
0		>30	Caucasian	Female	
0	427	NO	AfricanAmerican	Female	
0	250	NO	Other	Female	
0	414	NO	Caucasian	Male	

TRANSFORM

- Add Previous steps (18)
- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

Move, drop, duplicate or rename columns in the dataset. [Learn more.](#)
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit
- > Validate string

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

gency (...	number_inpatient (lo...	diag_3 (float)	readmitted (string)	race (string)	gender (string)
0			NO	Caucasian	Female
0	255	>30	Caucasian	Female	
1		NO	AfricanAmerican	Female	
0	403	NO	Caucasian	Male	
0	250	NO	Caucasian	Male	
0	250	>30	Caucasian	Male	
0		NO	Caucasian	Male	
0	250	>30	Caucasian	Male	
0	38	NO	Caucasian	Female	
0	486	NO	Caucasian	Female	
0	996	>30	AfricanAmerican	Female	
0	197	<30	AfricanAmerican	Male	
0	250	<30	Caucasian	Female	
0	427	NO	Caucasian	Male	
0	627	>30	AfricanAmerican	Female	
0	996	NO	AfricanAmerican	Male	
0	414	<30	AfricanAmerican	Male	
0	250	NO	Caucasian	Female	
0	416	>30	AfricanAmerican	Male	
0	714	NO	Other	Male	
0	428	NO	Other	Female	
0	427	NO	Other	Male	
0	582	NO	AfricanAmerican	Female	
0		>30	Caucasian	Female	
0	427	NO	AfricanAmerican	Female	
0	250	NO	Other	Female	
0	414	NO	Caucasian	Male	

TRANSFORM

- Add Previous steps (18)
- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

Move, drop, duplicate or rename columns in the dataset. [Learn more.](#)
- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit
- > Validate string

Transform: 1st Join

kno...	age_ordinal (float)	diag_1_scaled (float)	readmitted_1hot_N...	readmitted_1hot_>3...	readmitted_1hot_<30...
9	1.2191437876792175	1	0	0	
8	1.345934741597856	0	1	0	
7	3.1600206976645318	1	0	0	
5	0.03901260120573496	1	0	0	
4	0.9606853046912234	1	0	0	
2	2.018902112396784	0	1	0	
1	2.018902112396784	1	0	0	
0	2.0871741645068203	0	1	0	
3	1.9408769099853143	1	0	0	
6	2.1164336154111214	1	0	0	
4	1.2191437876792175	0	1	0	
1	0.7656222986625485	0	0	1	
4	2.0871741645068203	0	0	1	
3	2.0871741645068203	1	0	0	
1	2.5260659280713385	0	1	0	
1	4.871698575566153	1	0	0	
2	1.9993958117939166	0	0	1	
2	3.3258242527889053	1	0	0	
0	1.9603832105881815	0	1	0	
0	3.594035886078333	1	0	0	
2	1.9993958117939166	1	0	0	
1	2.7894009862100493	1	0	0	
0	1.9993958117939166	1	0	0	
3	2.389240360447299	0	1	0	
0	0.9216727034854884	1	0	0	
2	3.8329880684634596	1	0	0	
3	2.0822975893561035	1	0	0	

Add Previous steps

- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

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

Transform

Drop column

Column to drop

Preview Add

Clear

- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit
- > Validate string

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

diag_1_imputed (float)	diag_2_imputed (float)	diag_3_imputed (float)	gender_1hot_Female (...)	gender_1hot_Male (fl...)	genda...
250	428.4777814976007	408.79379533534205	1	0	0
276	250	255	1	0	0
648	250	408.79379533534205	1	0	0
8	250	403	0	1	0
197	157	250	0	1	0
414	411	250	0	1	0
414	411	408.79379533534205	0	1	0
428	492	250	0	1	0
398	427	38	1	0	0
434	198	486	1	0	0
250	403	996	1	0	0
157	288	197	0	1	0
428	250	250	1	0	0
428	411	427	0	1	0
518	998	627	1	0	0
999	507	996	0	1	0
410	411	414	0	1	0
682	174	250	1	0	0
402	425	416	0	1	0
737	427	714	0	1	0
410	427	428	1	0	0
572	456	427	0	1	0
410	401	582	1	0	0
489.4334337685526	715	408.79379533534205	1	0	0
189	496	427	1	0	0
786	401	250	1	0	0
427	428	414	0	1	0
---	---	---	-	-	-

[Back to data flow](#)

TRANSFORM

Add Previous steps (18)

- > Custom Transform
- > Custom formula
- > Encode categorical
- > Featurize date/time
- > Featurize text
- > Format string
- > Handle missing
- > Handle outliers
- > Manage columns

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

Transform

Drop column

Column to drop

Preview Add

Clear

- > Manage rows
- > Manage vectors
- > Parse column as type
- > Process numeric
- > Search and edit
- > Validate string

Data flow / Transform: 1st Join

Transform: 1st Join

Export Data

[Back to data flow](#)

TRANSFORM

	diag_2_imputed (float)	diag_3_imputed (float)	gender_1hot_Female (...)	gender_1hot_Male (fl...)	gender_1hot_Unknown...	race_1	Add	Previous steps (18)
428.4777814976007	408.79379533534205	1	0	0	1	1		
250	255	1	0	0	0	1		
250	408.79379533534205	1	0	0	0	0		
250	403	0	1	0	0	1		
157	250	0	1	0	0	1		
411	250	0	1	0	0	1		
411	408.79379533534205	0	1	0	0	1		
492	250	0	1	0	0	1		
427	38	1	0	0	0	1		
198	486	1	0	0	0	1		
403	996	1	0	0	0	0		
288	197	0	1	0	0	0		
250	250	1	0	0	0	1		
411	427	0	1	0	0	1		
998	627	1	0	0	0	0		
507	996	0	1	0	0	0		
411	414	0	1	0	0	0		
174	250	1	0	0	0	1		
425	416	0	1	0	0	0		
427	714	0	1	0	0	0		
427	428	1	0	0	0	0		
456	427	0	1	0	0	0		
401	582	1	0	0	0	0		
715	408.79379533534205	1	0	0	0	1		
496	427	1	0	0	0	0		
401	250	1	0	0	0	0		
428	414	0	1	0	0	1		
---	---	-	-	-	-	-		

Data flow / Transform: 1st Join

Transform: 1st Join

Export Data

	diag_3_imputed (float)	gender_1hot_Female (...)	gender_1hot_Male (fl...)	gender_1hot_Unknown...	race_1hot_Caucasian (...)	race_1	Add	Previous steps (18)
408.79379533534205	1	0	0	1	0			
255	1	0	0	1	0			
408.79379533534205	1	0	0	0	1			
403	0	1	0	1	0			
250	0	1	0	1	0			
250	0	1	0	1	0			
408.79379533534205	0	1	0	1	0			
250	0	1	0	1	0			
38	1	0	0	1	0			
486	1	0	0	1	0			
996	1	0	0	0	1			
197	0	1	0	0	1			
250	1	0	0	1	0			
427	0	1	0	1	0			
627	1	0	0	0	1			
996	0	1	0	0	1			
414	0	1	0	0	1			
250	1	0	0	1	0			
416	0	1	0	0	1			
714	0	1	0	0	0			
428	1	0	0	0	0			
427	0	1	0	0	0			
582	1	0	0	0	1			
408.79379533534205	1	0	0	1	0			
427	1	0	0	0	1			
250	1	0	0	0	0			
414	0	1	0	1	0			
---	---	-	-	-	-	-		

Import Prepare Analyze Export

Data flow / Transform: 1st Join

Transform: 1st Join

Back to data flow

Transform: TSL JOIN

float	age_ordinal (float)	diag_1_imputed_scale...	diag_2_imputed_scale...	diag_3_imputed_scale...	readmitted_parser_1 (s...
9	1.2320095759161842	2.461510423177554	2.435538630906403	NO	
8	1.3601385718114674	1.4361949029038146	1.5192558154451987	1	
7	3.1933688207747495	1.4361949029038146	2.435538630906403	NO	
5	0.03942430642931789	1.4361949029038146	2.401019974997706	NO	
4	0.9708235458219531	0.9019303990235956	1.4894664857305868	NO	
2	2.040207857717201	2.3611044203738714	1.4894664857305868	1	
1	2.040207857717201	2.3611044203738714	2.435538630906403	NO	
0	2.109200393968507	2.8264315689147073	1.4894664857305868	1	
3	1.9613592448585653	2.4530208941597156	0.2263989058310492	NO	
6	2.1587686237904955	1.1374663630998212	2.895522848260261	NO	
4	1.2320095759161842	2.315146183480949	5.934034479150658	1	
1	0.7737020136753636	1.6544965281451944	1.1736995907557024	1	
4	2.109200393968507	1.4361949029038146	1.4894664857305868	1	
3	2.109200393968507	2.3611044203738714	2.5440087576278425	NO	
1	2.5527238412983335	5.733290052392028	3.735581946212312	1	
1	4.923110265361072	2.9126032630889362	5.934034479150658	NO	
2	2.020495704502542	2.3611044203738714	2.466556500369852	1	
2	3.3609221230993502	0.999591652421055	1.4894664857305868	NO	
0	1.9810713980732242	2.441531334956485	2.4784722322556965	1	
0	3.6319642298009107	2.4530208941597156	4.253916283246556	NO	
2	2.020495704502542	2.4530208941597156	2.5499666235707648	NO	
1	2.8188379096962293	2.619619502896558	2.5440087576278425	NO	
0	2.020495704502542	2.3036566242577186	3.4674779787808063	NO	
3	2.4119467087055853	4.10751742230491	2.435538630906403	1	
0	0.9313992393926352	2.849410687361168	2.5440087576278425	NO	
2	3.873438106680483	2.3036566242577186	1.4894664857305868	NO	
3	2.1042723556648424	2.458765673771331	2.466556500369852	NO	

TRANSFORM

- Add Previous steps (18)
 - > Custom Transform
 - > Custom formula
 - > Encode categorical
 - > Featurize date/time
 - > Featurize text
 - > Format string
 - > Handle missing
 - > Handle outliers
 - > Manage columns

Move, drop, duplicate or rename columns in the dataset. [Learn more.](#)
 - > Manage rows
 - > Manage vectors
 - > Parse column as type
 - > Process numeric
 - > Search and edit
 - > Validate string

Quick Model

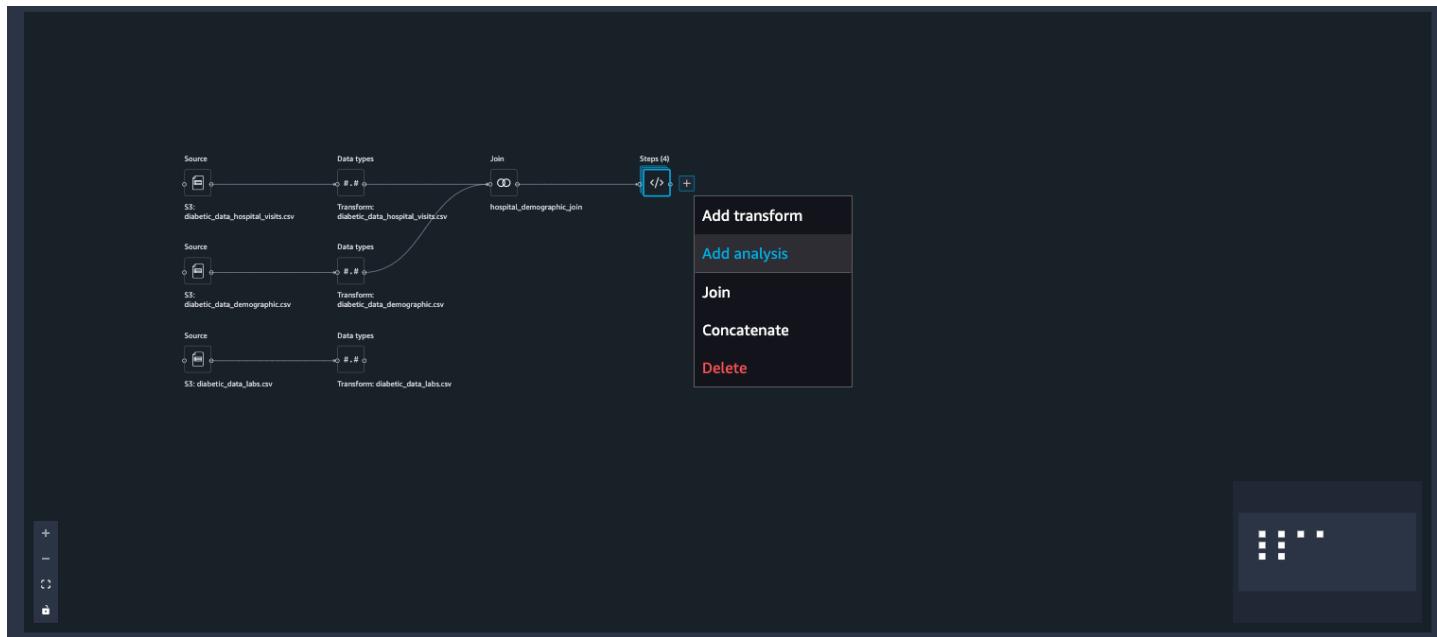
Now that we have transformed our initial dataset, Let's re-explore **Quick Model** transformation. This transform option allows you to visualize the target model accuracy by dynamically running the model on the fly with the transformed data. This allows you to quickly see the feature relationship weightage on target prediction.

1) Click + sign next to Join flow icon and choose **Add analysis**

Import **Prepare** **Analyze** **Export**

Data flow

Choose the plus sign to add a step to the flow. Select a step to modify.



2) Select **Quick Model** from the list of Analysis types on the right panel.

Import Prepare Analyze Export

Imported datasets / Transform: 1st Join / Quick_Model_post_transform

Create Analysis

Create an analysis of your data. [Learn more](#)

Quick Model: Quick_Model_post_transform

No Preview available

Use Configure for built-in analyses
Use Code to create a custom analysis

Data table

encounter_id_0	patient_nbr_0	admission_type_id	discharge_dispositio...	admission_source_id	time_in_hospital	payer_code	medical_specialty
2278392	8222157	6	25	1	1	?	Pediatrics-Endocrinology
149190	55629189	1	1	7	3	?	?
64410	8604765	1	1	7	2	?	?
500364	82442376	1	1	7	2	?	?
16680	42519267	1	1	7	1	?	?
35754	82637451	2	1	2	3	?	?
55842	84259809	3	1	2	4	?	?

Analysis type

- Quick Model (selected)
- Bias Report
- Histogram
- Scatter Plot
- Table Summary
- Target Leakage

Configure Code Back to all analyses Cancel Preview Create

3) Give a name to your analysis and select the target label in **Label** field.

Import Prepare Analyze Export

Imported datasets / Transform: 1st Join / Transformed Quick Model

Create Analysis

Label

Analysis type

- Quick Model (selected)
- Bias Report
- Histogram
- Scatter Plot
- Table Summary
- Target Leakage

Back to all analyses Create

Create an analysis of your data.

[Learn more](#)

Quick Model: Transformed Quick Model

No Preview available

Use Configure for built-in analyses
Use Code to create a custom analysis

Configure Code

Analysis type: Quick Model

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

Analysis name: Transformed Quick Model

Optional

Label: readmitted_parser_2

Data table

readmitted_parser_2	encounter_id_0	patient_nbr_0	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospit
0	2278392	8222157	6	25	1	1
1	149190	55629189	1	1	7	3
0	64410	86047875	1	1	7	2
0	500364	82442376	1	1	7	2
0	16680	42519267	1	1	7	1
1	35754	82637451	2	1	2	3
0	55842	84259809	3	1	2	4
1	63768	114882984	1	1	7	5
0	12522	48330783	2	1	4	13
0	15738	63555939	3	3	4	12

Clear Preview Save

4) Click **Preview** and wait for the model to be results to be displayed on the screen

Import Prepare Analyze Export

Imported datasets / Transform: 1st Join / Transformed Quick Model

Create Analysis

Create an analysis of your data.

[Learn more](#)

Quick Model: Transformed Quick Model

Model achieved a 0.643 f1 on a test set.

Configure Code

Analysis type: Quick Model

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

Analysis name: Transformed Quick Model

Optional

Label: readmitted_parser_2

Data table

readmitted_parser_2	encounter_id_0	patient_nbr_0	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospit
0	2278392	8222157	6	25	1	1
1	149190	55629189	1	1	7	3
0	64410	86047875	1	1	7	2
0	500364	82442376	1	1	7	2
0	16680	42519267	1	1	7	1
1	35754	82637451	2	1	2	3
0	55842	84259809	3	1	2	4
1	63768	114882984	1	1	7	5
0	12522	48330783	2	1	4	13
0	15738	63555939	3	3	4	12

Clear Preview Save

The resulting **Quick Model** F1 score shows **0.643** with the transformed dataset. This is an improvement from the original F1 score of **0.527**.

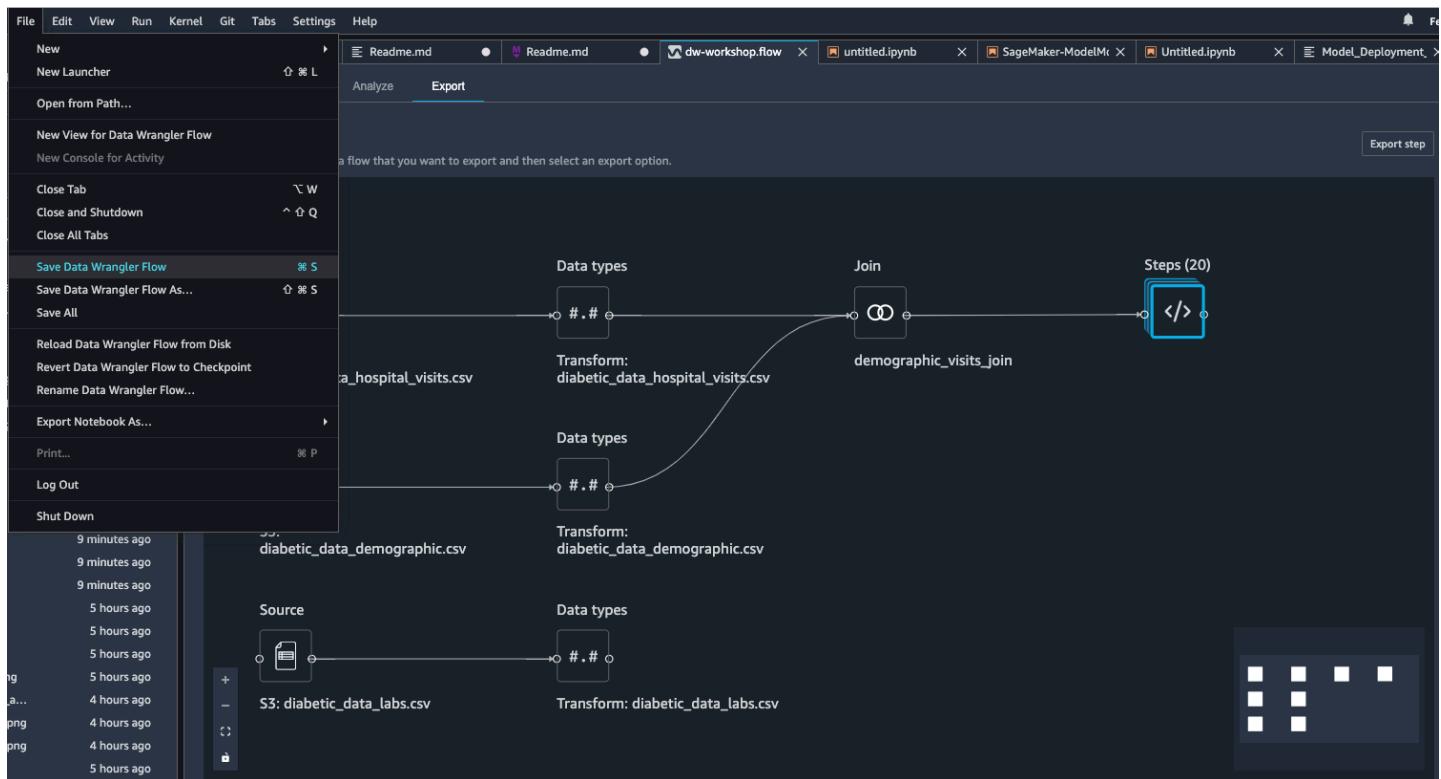
As you can see, Data Wrangler helped us to quickly experiment the transformations of input features and improve our model accuracy (without us building any ML models). Using this feature, Data Scientists can iterate through applicable transformations until they see desired transformed dataset that would potentially lead to business expectations.

5) Click **Create** button to add the quick model analysis to the data flow.

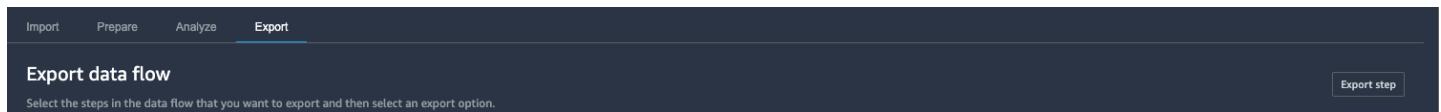
Export Options

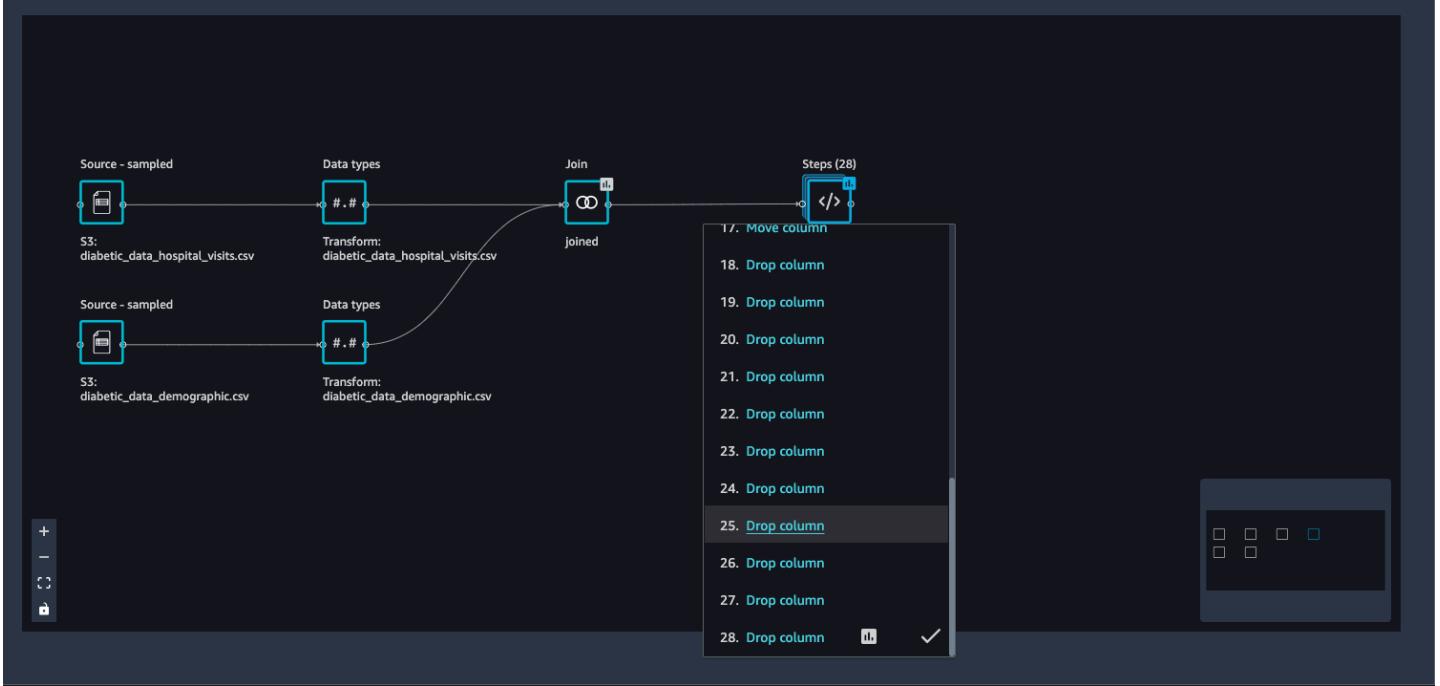
We are now ready to export dataflow for further processing.

1) Save the DW flow file as shown below



1) Click **Export** tab and select **Steps** icon to reveal all the DW flow steps. Click the last step to mark it as check (as shown in figure below)

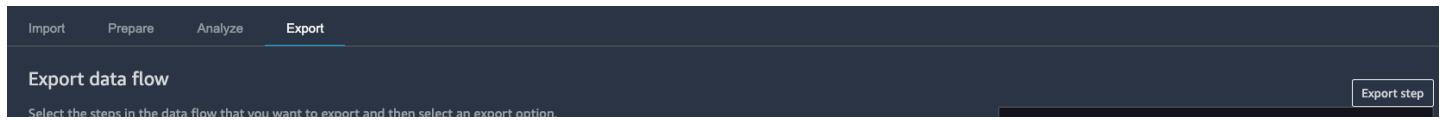


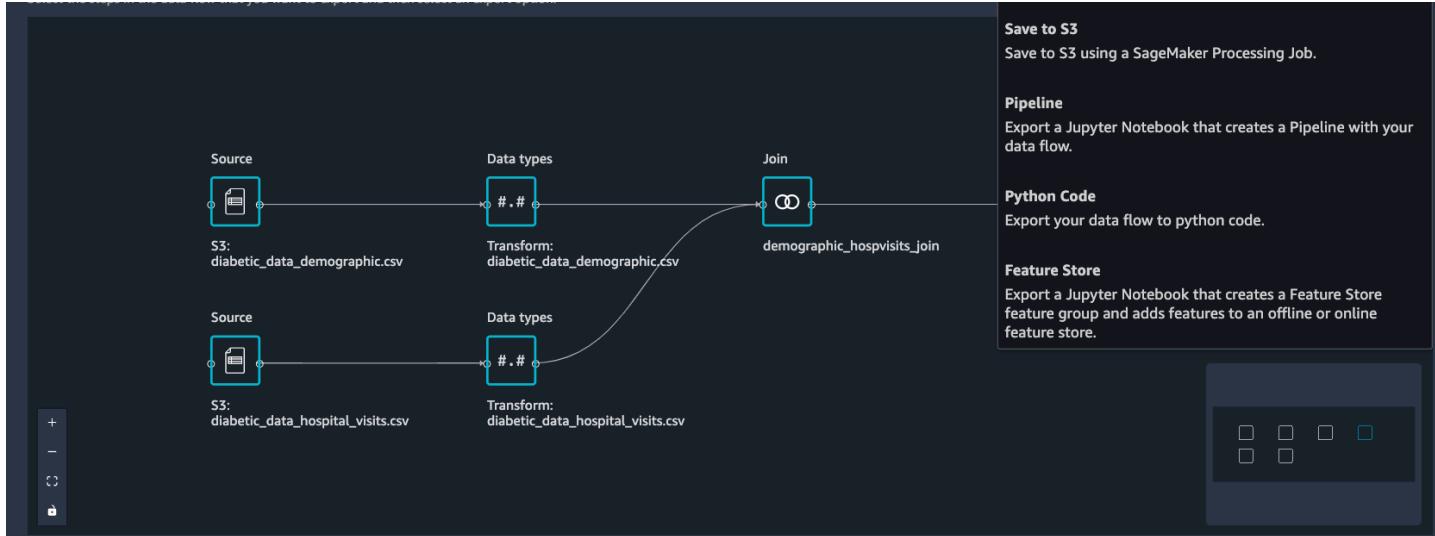


2) Click `Export step` to reveal the export options. You currently have 4 export options

- `Save to S3` Save the data to an S3 bucket using a [Amazon SageMaker Processing Job](#).
- `Pipeline` exports a Jupyter Notebook that creates an [Amazon SageMaker Pipeline](#) with your data flow.
- `Python Code` exports your data flow to python code.
- `Feature Store` exports a Jupyter Notebook that creates an [Amazon SageMaker Feature Store](#) feature group and adds features to an offline or online feature store.

You can find more information for each export option in this [page](#).





3) Select **Pipeline** to generate Jupyter Notebook that creates a Pre-processing Job using your data flow file.

Save to S3 with a SageMaker Processing Job

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

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

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

Contents

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

Inputs and Outputs

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

Configurable Settings

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

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

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

3. Processing and Training Jobs for Model building

Processing Job submission

1) We are now ready to submit a SageMaker Processing Job using the data flow file. Run all the cells upto

If we are now ready to submit a SageMaker Processing job using the data flow file. Run all the cells upto

Create Processing Job. This cell Create Processing Job will trigger a new SageMaker processing job by provisioning managed infrastructure and running the required DataWrangler docker container on that infrastructure.

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

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

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

2) You can check the status of the submitted processing job by running next cell Job Status & S3 Output Location

Job Status & S3 Output Location

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

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

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

3) You can also check the status of the submitted processing job from Amazon SageMaker Console as shown below

Job Status & S3 Output Location

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

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

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

Train a model with Amazon SageMaker

1) Now that the data has been processed, you may want to train a model using the data. The same notebook has sample steps to train a model using [Amazon SageMaker built-in XGBoost algorithm](#). Since our use case is binary classification, we need to change the objective to "binary:logistic" inside the sample training steps as shown below.

Configure the algorithm and training job

The Training Job hyperparameters are set. For more information on XGBoost Hyperparameters, see <https://xgboost.readthedocs.io/en/latest/parameter.html>.

```
[ ]: region = boto3.Session().region_name
container = sagemaker.image_uris.retrieve("xgboost", region, "1.2-1")
```

```
hyperparameters = {  
    "max_depth": "5",  
    "objective": "binary:logistic",  
    "num_round": "10",  
}  
train_content_type = (  
    "application/x-parquet" if output_content_type.upper() == "PARQUET"  
    else "text/csv"  
)  
train_input = sagemaker.inputs.TrainingInput(  
    s3_data=s3_training_input_path,  
    content_type=train_content_type,  
)
```



2) All set. Now we are ready to fire our training job using SageMaker managed infrastructure. Run the cell below.

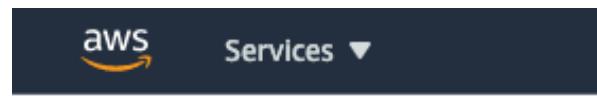
Start the Training Job

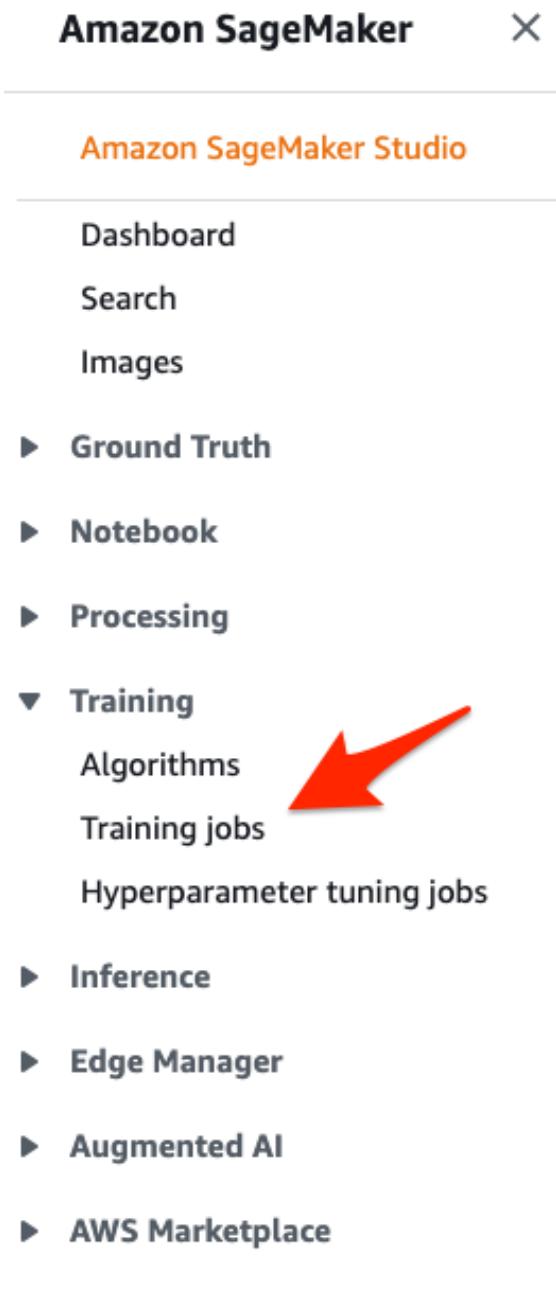
The TrainingJob configurations are set using the SageMaker Python SDK Estimator, and which is fit using the training data from the Processing Job that was run earlier.

```
[ ]: estimator = sagemaker.estimator.Estimator(  
    container,  
    iam_role,  
    hyperparameters=hyperparameters,  
    instance_count=1,  
    instance_type="ml.m5.2xlarge",  
)  
estimator.fit({"train": train_input})
```

Now that you have a trained model there are a number of different things you can do. For more details on training with SageMaker, please see
https://sagemaker.readthedocs.io/en/stable/frameworks/xgboost/using_xgboost.html.

3) You can monitor the status of submitted training job in SageMaker Console under `Training / Training jobs` tab on the left.





4. Host trained Model for real time inference

Deploy model for real-time inference

1) We will now use another notebook provided under project folder

`hosting/Model_deployment_Steps.ipynb`.

This is a simple notebook with 2 cells - First cell has code for deploying your model to persistent endpoint. Here you need to update `model_url` with your training job output `s3 model artifact`. Here are image for reference.



S3 model artifact
 s3://sagemaker-us-east-1-40010771400/sagemaker-xgboost-2021-06-10-20-40-57-927/output/model.tar.gz

- ▶ Inference
- ▶ Edge Manager
- ▶ Augmented AI

Amazon SageMaker Model Deployment using persistent endpoint for Real-Time inference

In this notebook we will deploy the Model that was trained using the preprocessed training_input from Data Wrangler preprocessing Job.

Perform model deployment using Amazon SageMaker

```
[ ]: import boto3
import sagemaker
import time
from sagemaker import get_execution_role, session
from time import gmtime, strftime
from sagemaker.model import Model
from sagemaker.image_uris import retrieve
from sagemaker.predictor import Predictor
from sagemaker.serializers import CSVSerializer

# Setup defaults
region= boto3.Session().region_name
role = get_execution_role()

# Update this model_url with the trained model generated from training job that we ran as part of Data Wrangler
model_url = 's3://sagemaker-us-east-1-40010771400/sagemaker-xgboost-2021-06-10-20-40-57-927/output/model.tar.gz' 

```

2) The second cell in the notebook will run inference on sample test file `test_data_UCI_sample.csv`.

Perform inference against the deployed model

```
[ ]: # Create SageMaker Predictor object to pass in the test data for inference
predictor = Predictor(endpoint_name=endpoint_name, serializer=CSVSerializer())

print("Sending test traffic to the endpoint {}".format(endpoint_name))

with open("test_data/test_data_UCI_sample.csv", 'r') as f:
    for row in f:
        payload = row.rstrip('\n')
        print('Incoming Payload => ', payload)
        response = predictor.predict(data=payload).decode('utf-8')
        print('Inference Output => ', response)
        time.sleep(1)

print("Done!")
```

Conclusion

This concludes the example. In this example you have learnt how to use SageMaker Data Wrangler capability to create data preprocessing, feature engineering steps using simple to use Data Wrangler GUI. We then used the generated notebook to submit a SageMaker managed processing job to perform the data preparation using our data flow file. Later we saw how to train a simple XGBoost algorithm using our processed dataset. In the end we hosted our trained model and ran inferences against synthetic test data.