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

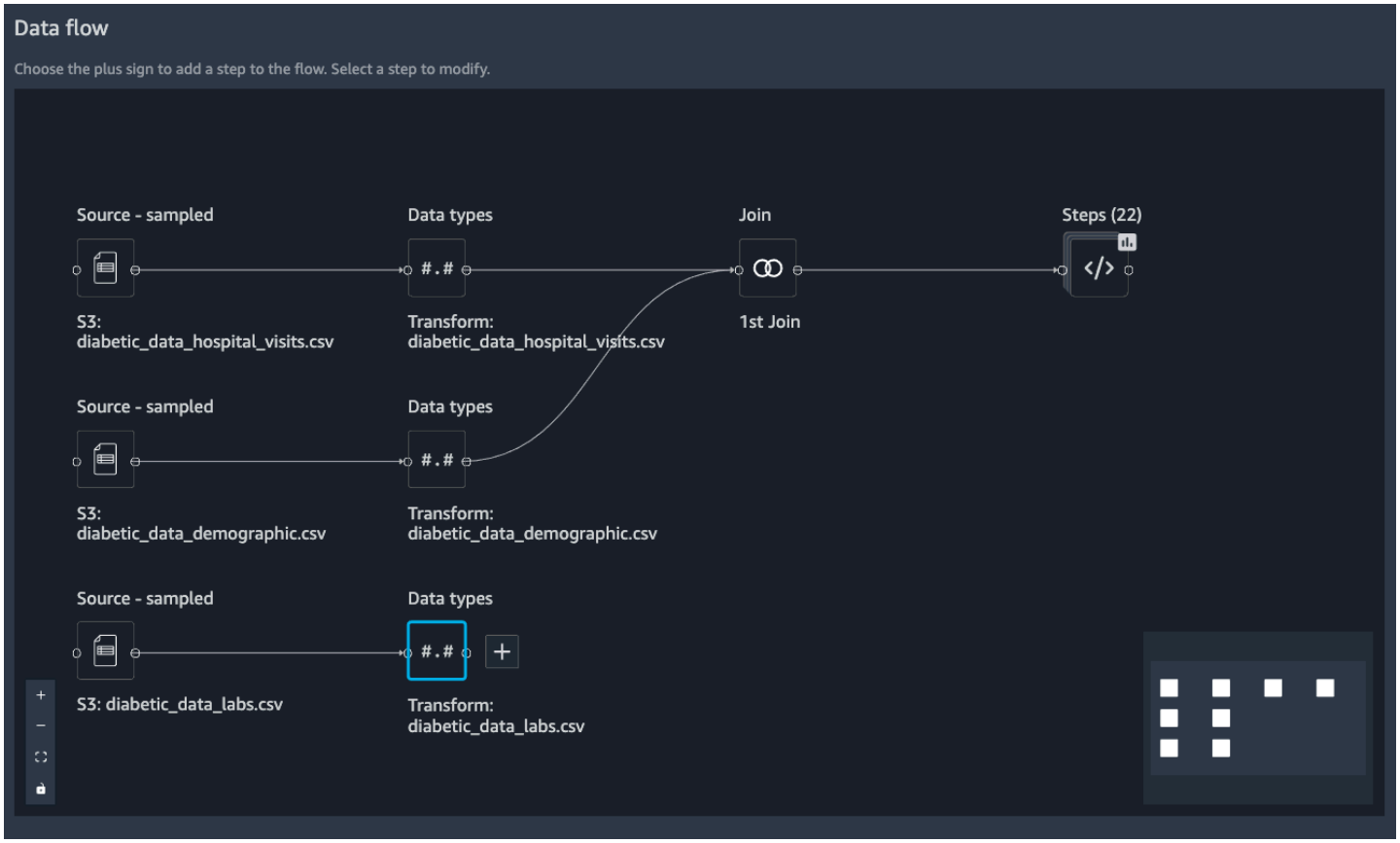
UCI diabetic patient readmission dataset. The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes.

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

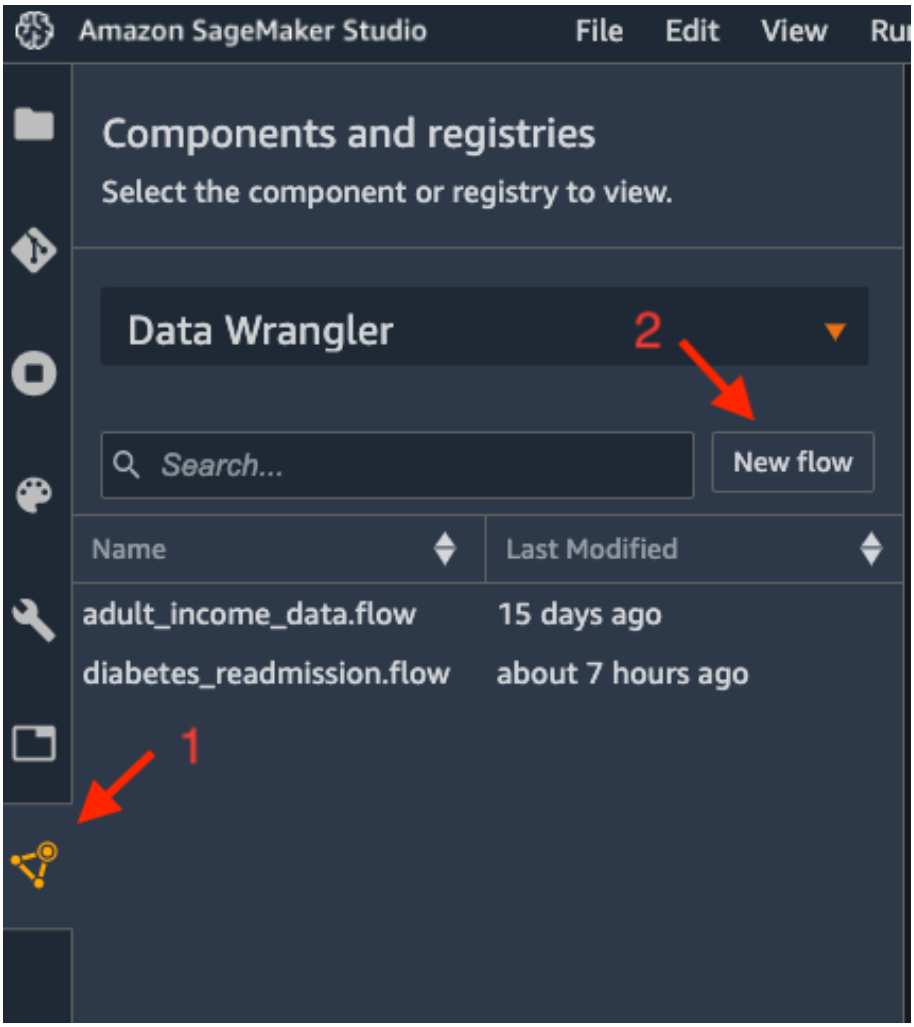
engineering for ML.

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

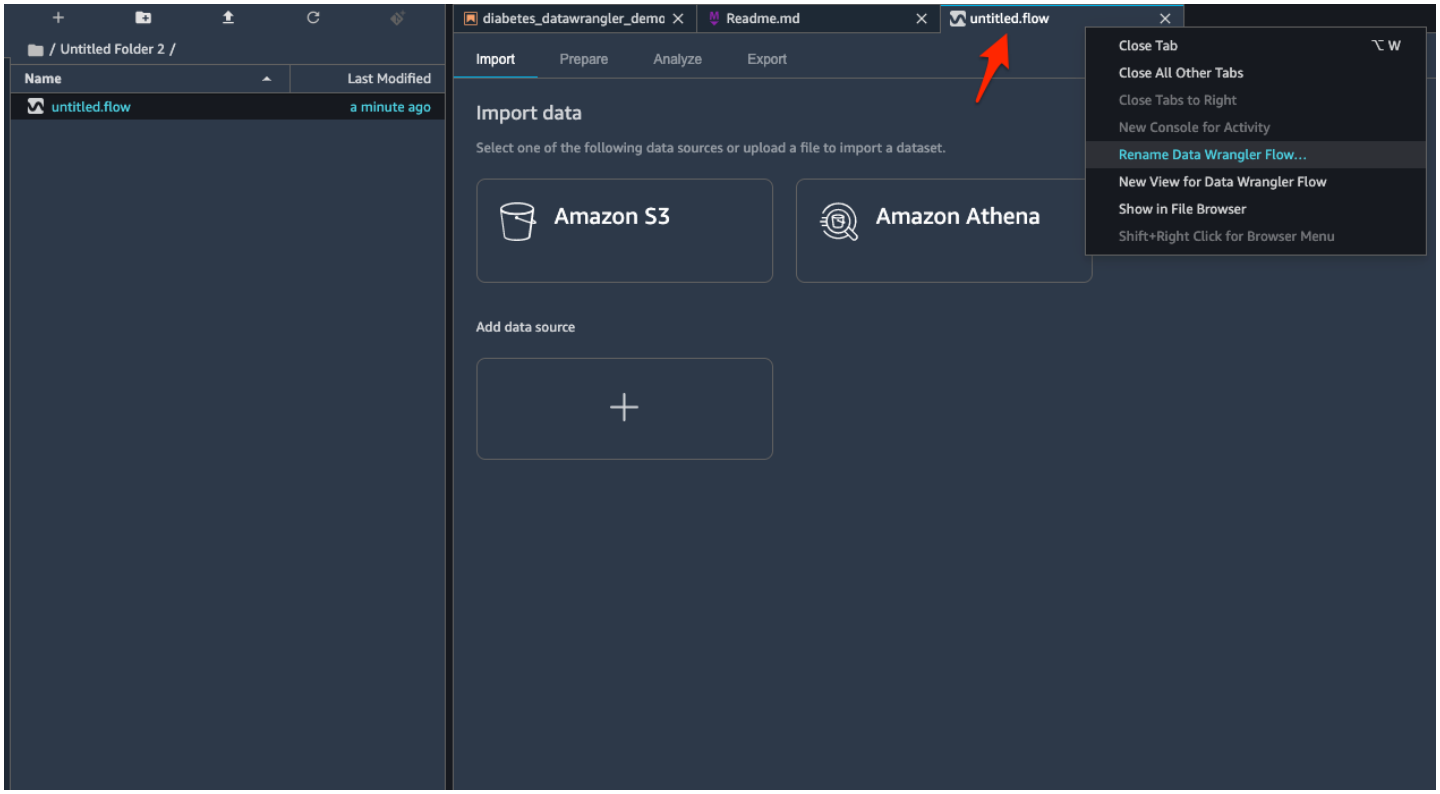
Create a new flow

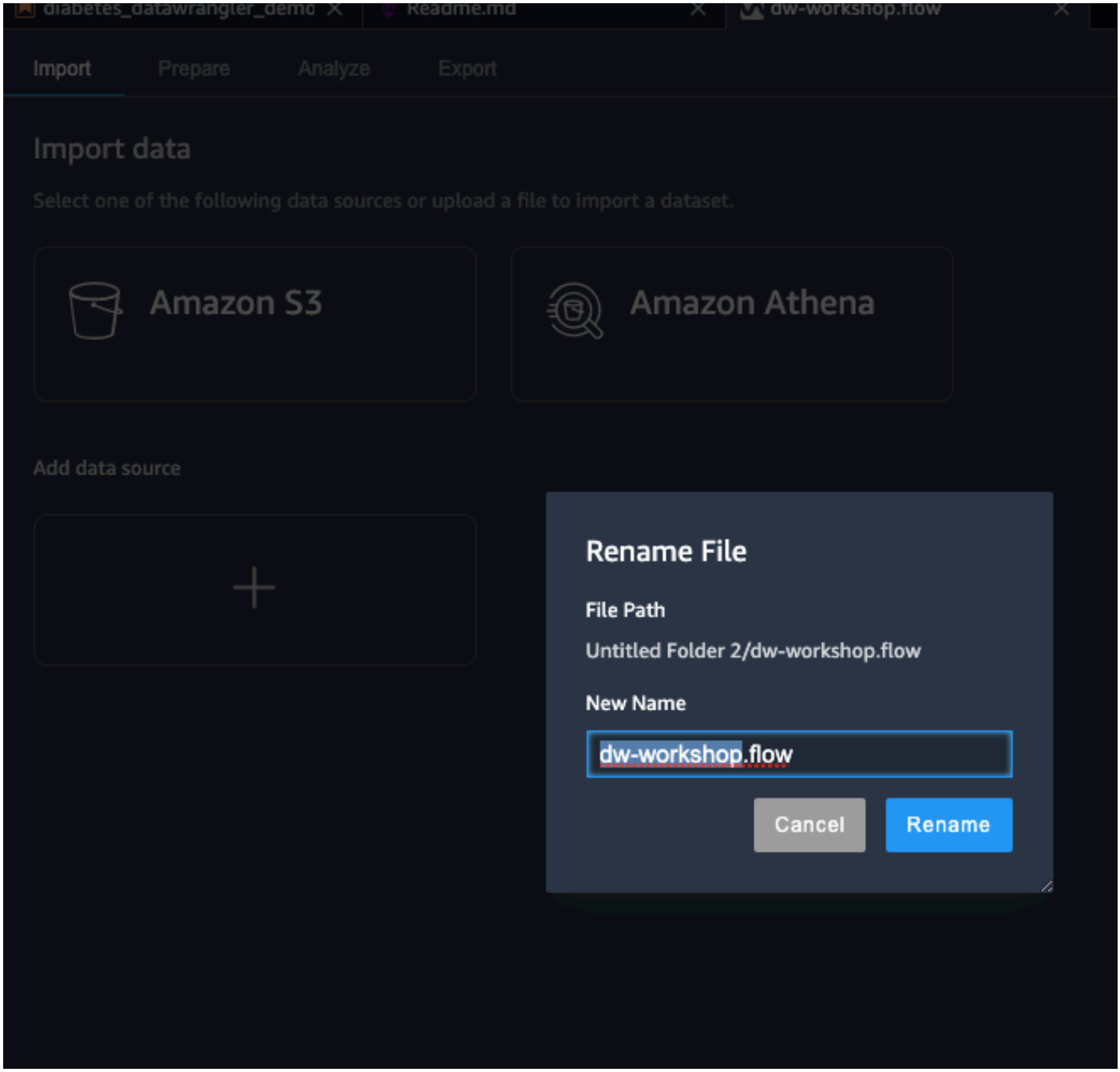
Please click on the **SageMaker component and registry** tab and click **New flow**.



Rename your DW dataflow file

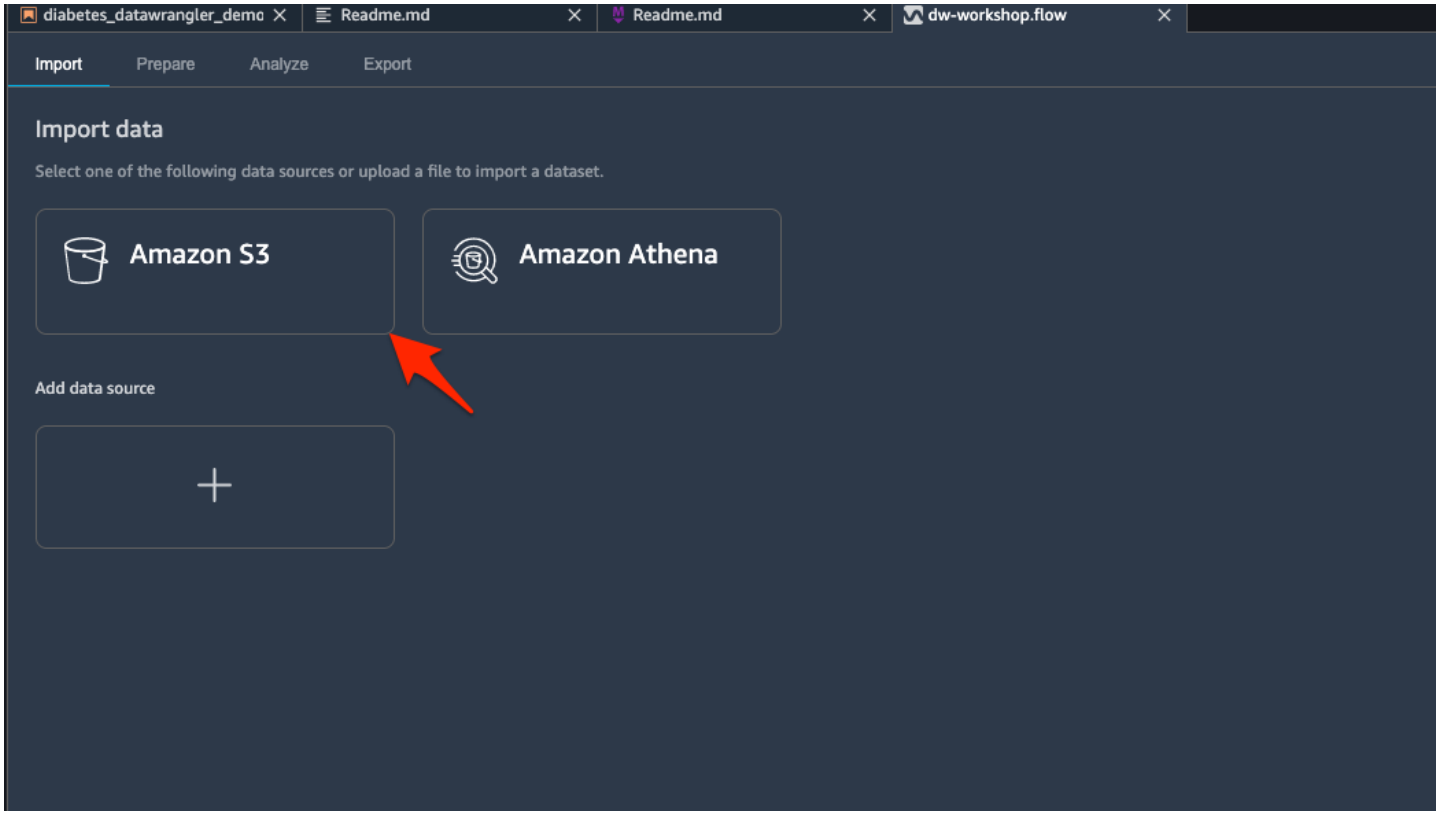
Right click on the **untitled.flow** file tab to reveal below options. Then choose Rename file to change the file name.


/ Untitled Folder 2 /
Name
Last Modified
untitled.flow
a minute ago
diabetes_datawrangler_demo
Readme.md
untitled.flow
Import
Prepare
Analyze
Export
Import data
Select one of the following data sources or upload a file to import a dataset.
Amazon S3
Amazon Athena
Add data source
Close Tab
⌥ W
Close All Other Tabs
Close Tabs to Right
New Console for Activity
Rename Data Wrangler Flow...
New View for Data Wrangler Flow
Show in File Browser
Shift+Right Click for Browser Menu



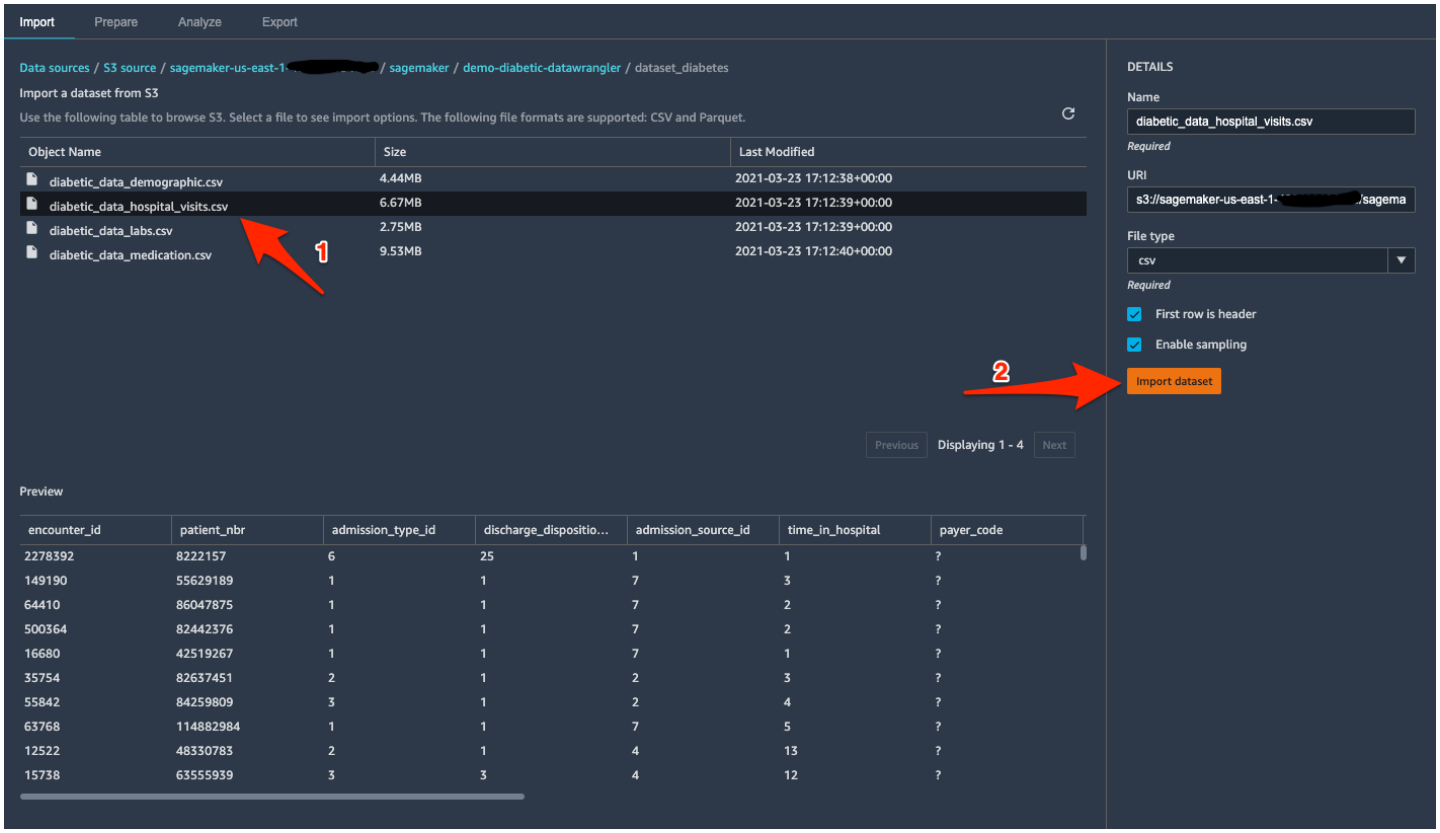
Load the data from S3 into Data Wrangler

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



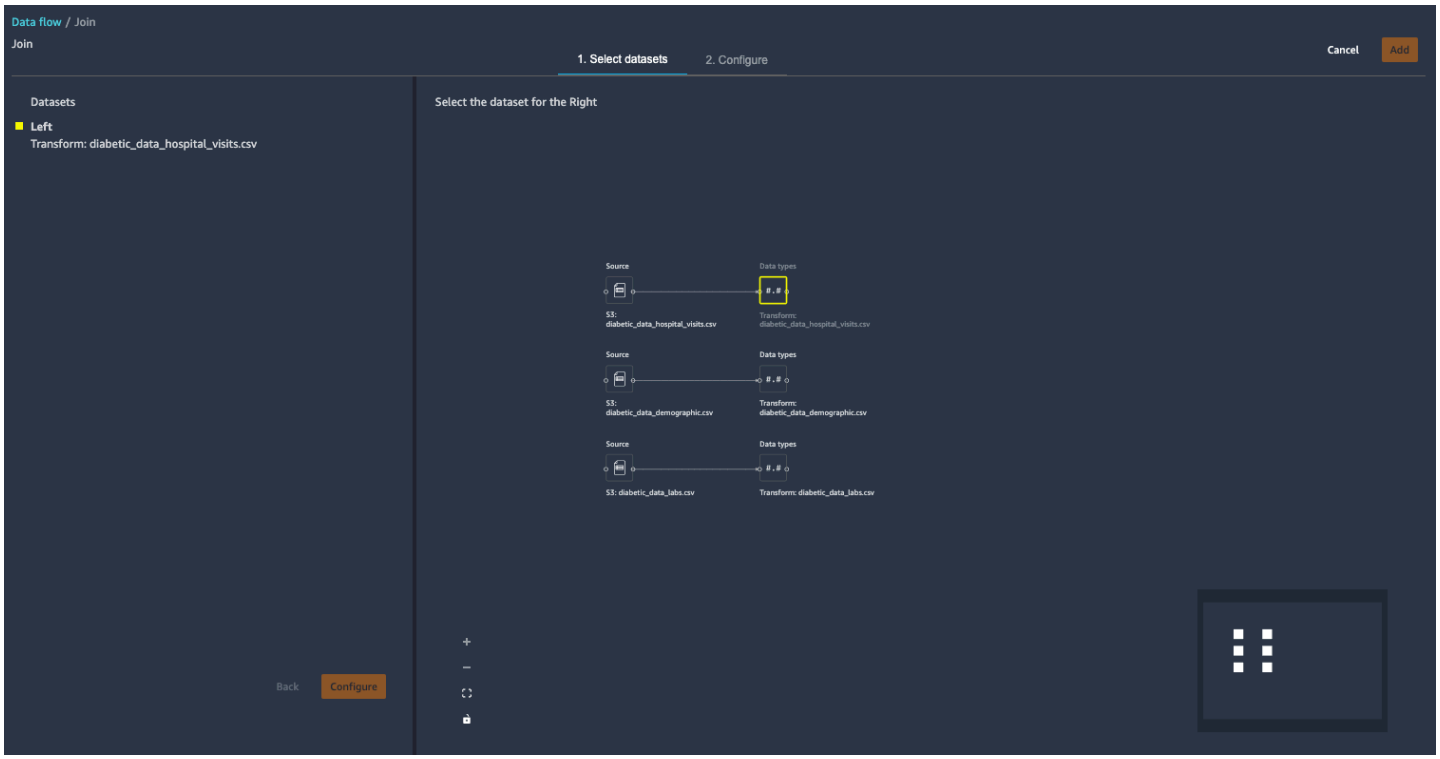
Note: You could also import data from Athena: [how databases and tables in Amazon Athena can be imported](#).

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



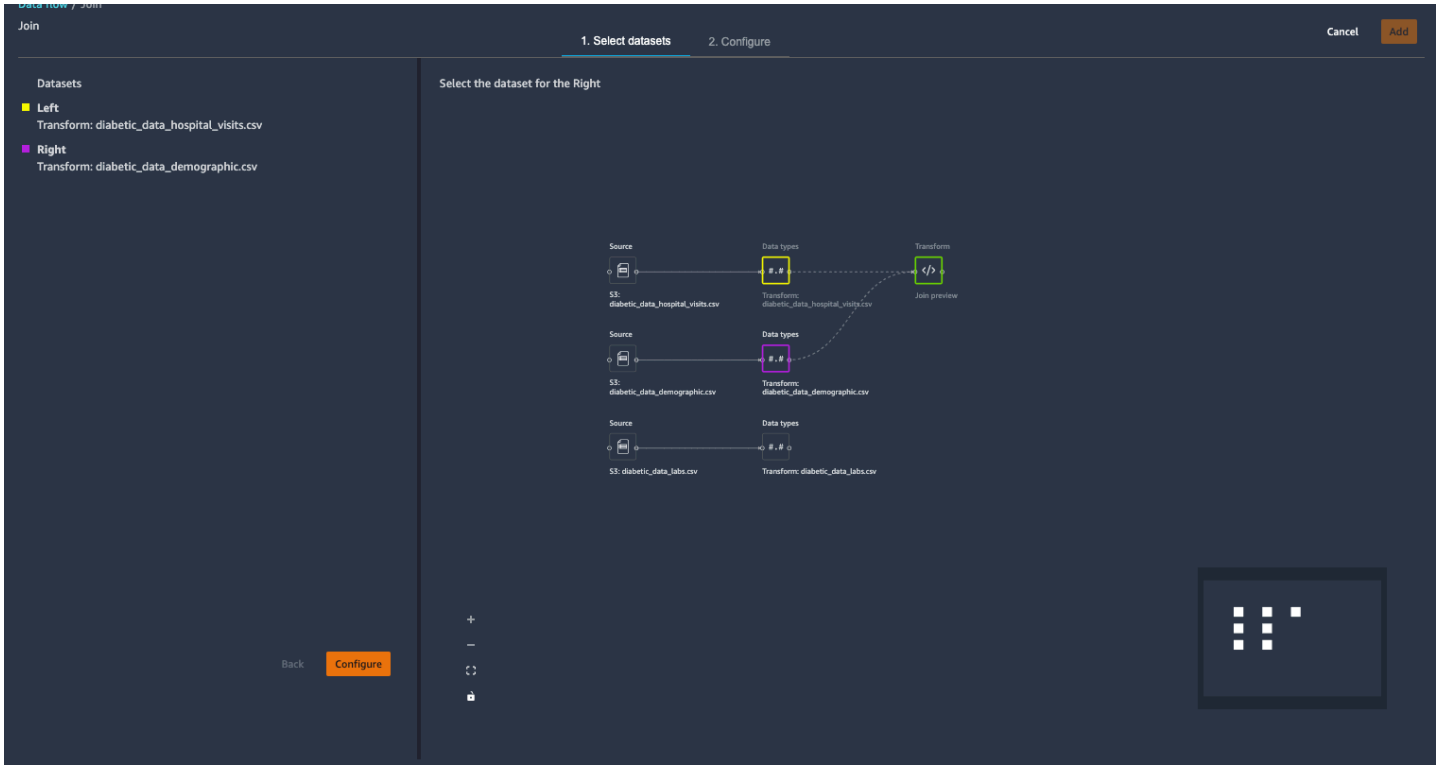
Join the CSV files

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.

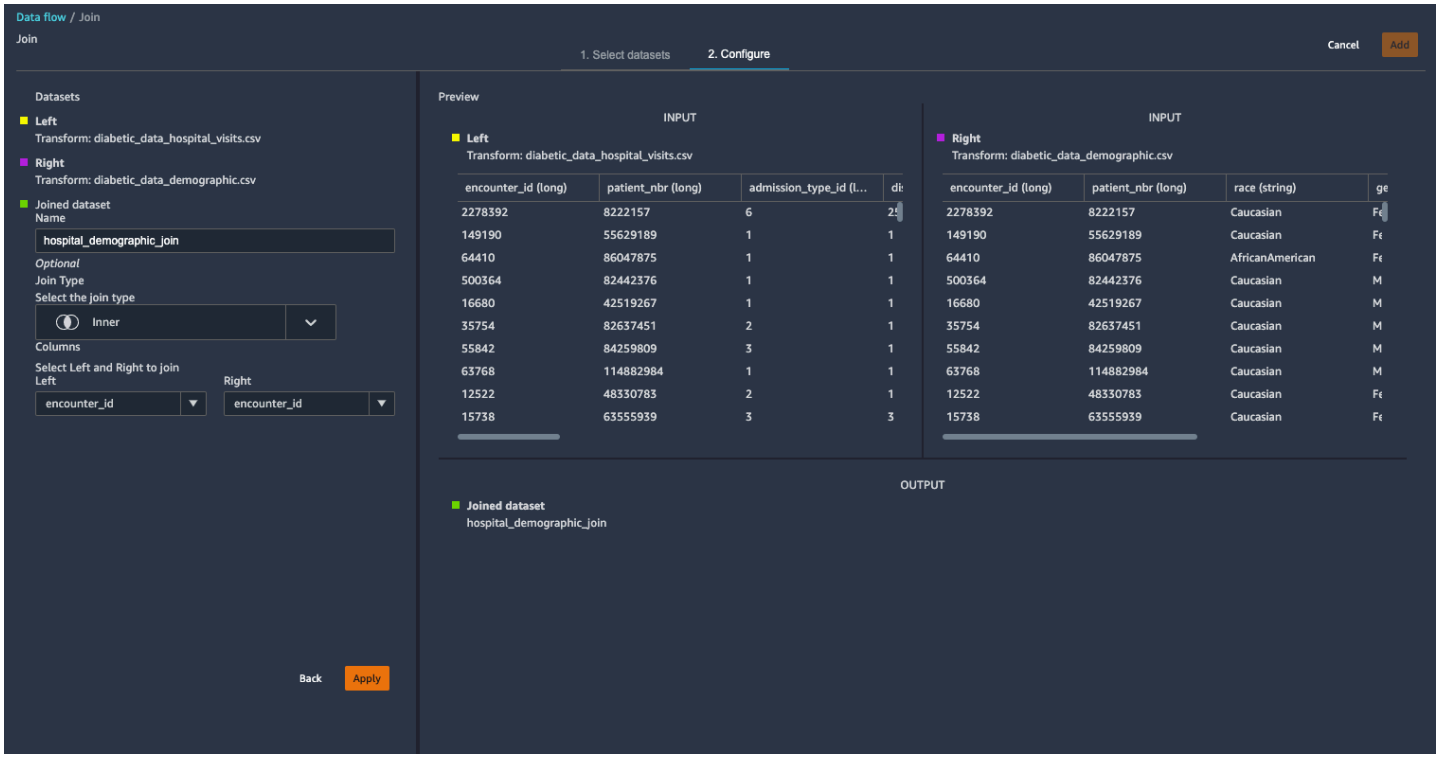


1) Select `diabetic_data_demographic.csv` dataset as Right dataset.

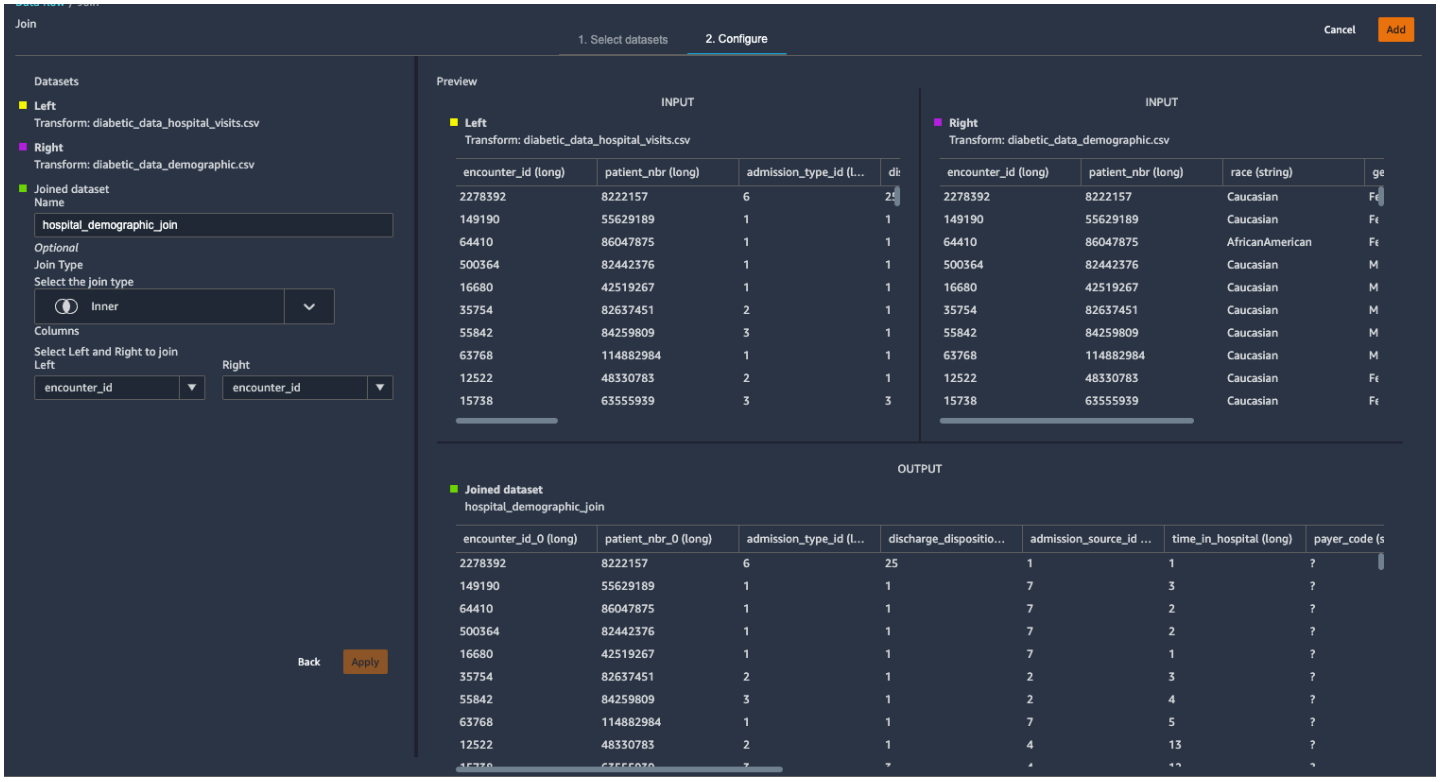
2) Click `Join` and new Preview panel is presented.



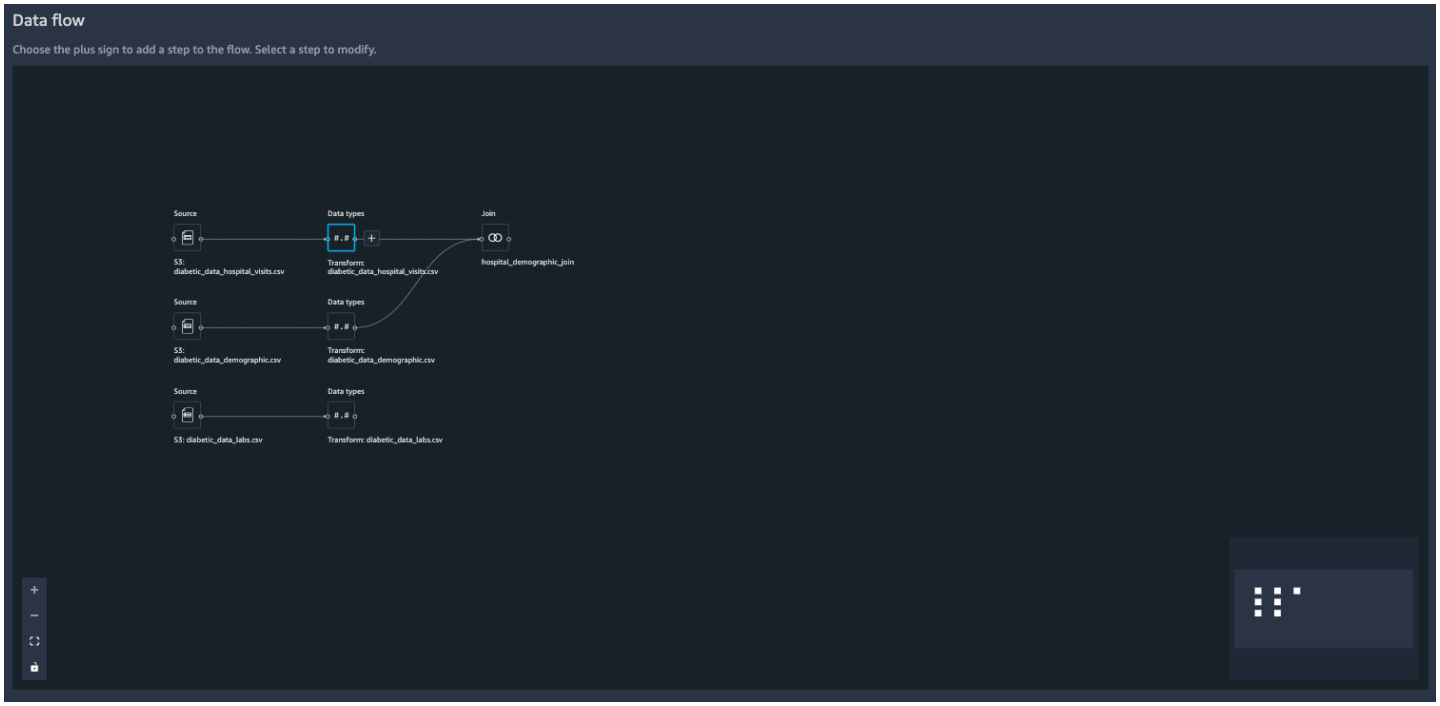
3) Give a name to the Join and choose join type and Left & Right columns for join condition



4) Click Apply to preview the joined dataset



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



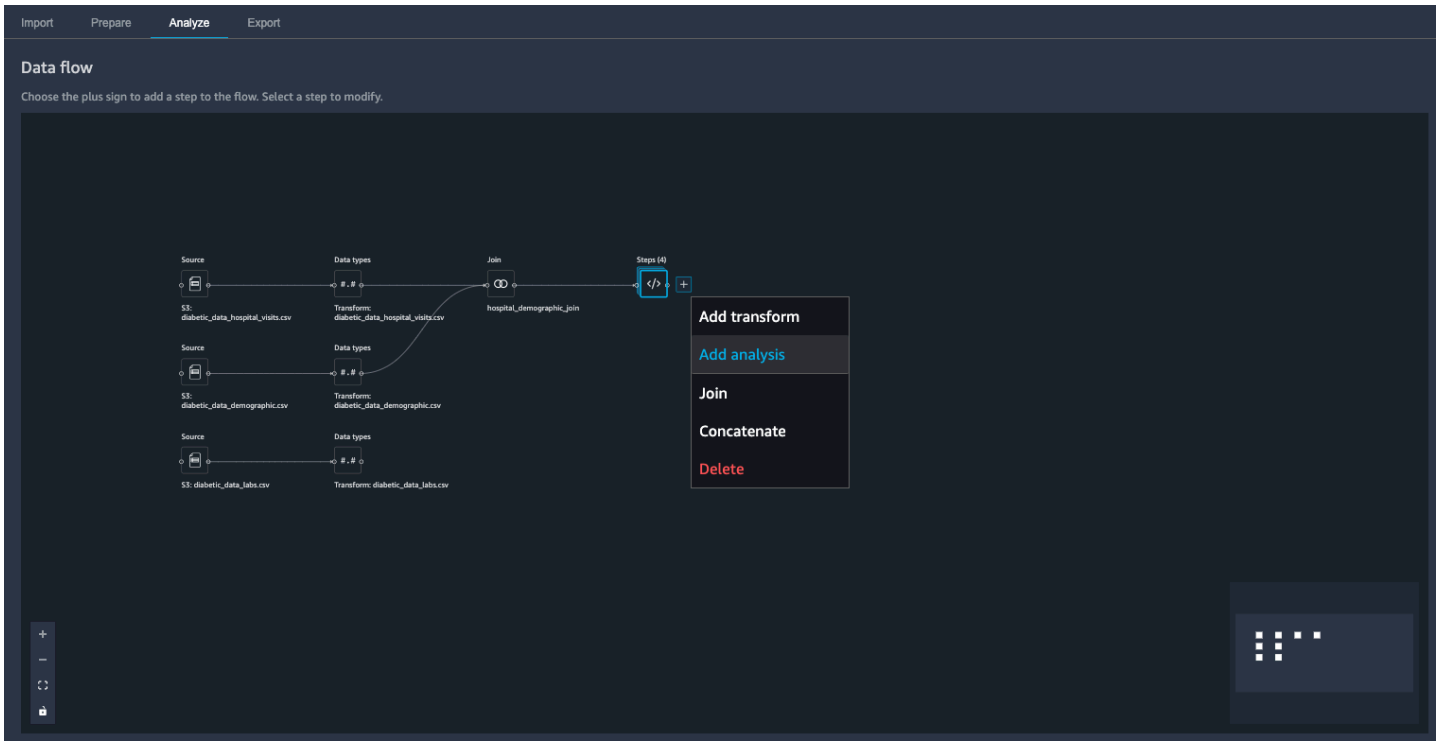
Analysis

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.

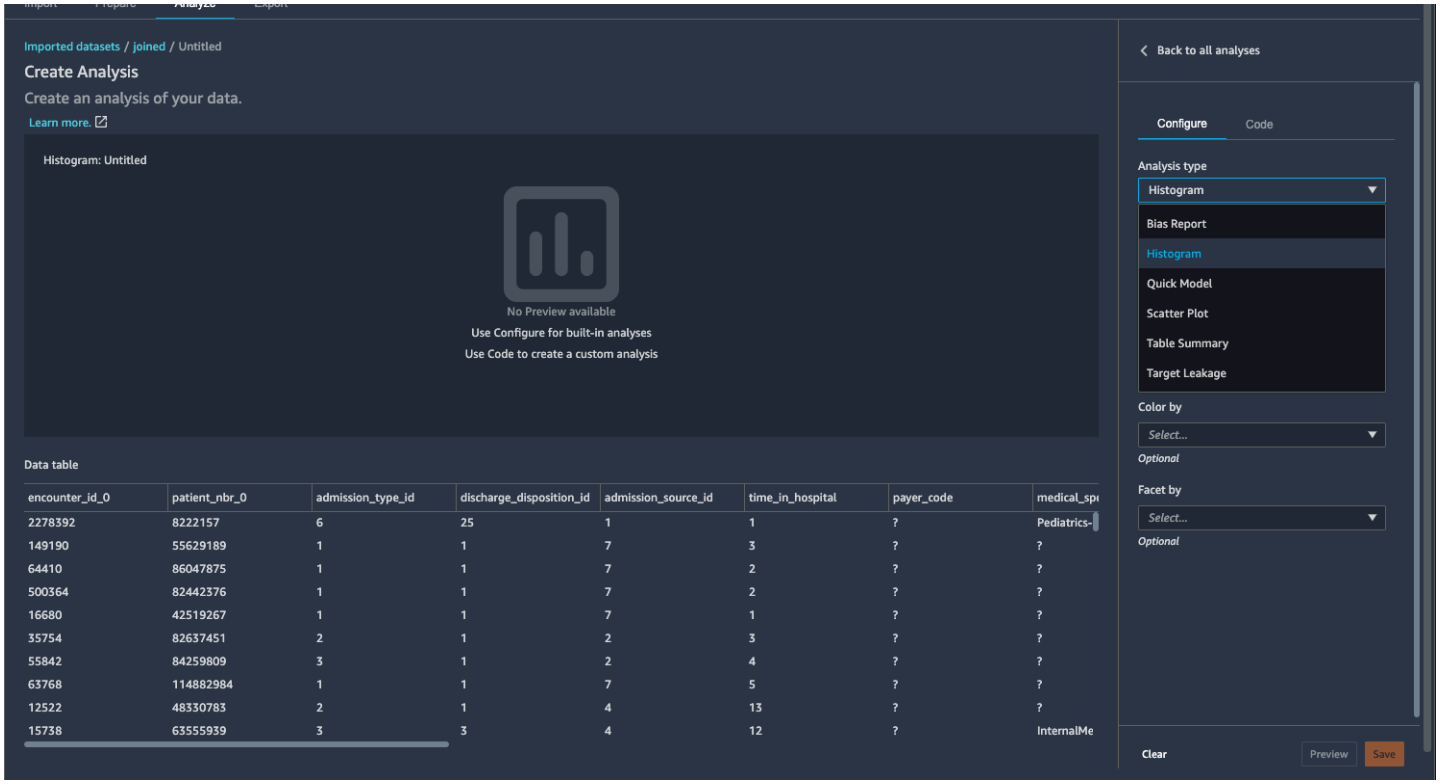
Histogram

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.

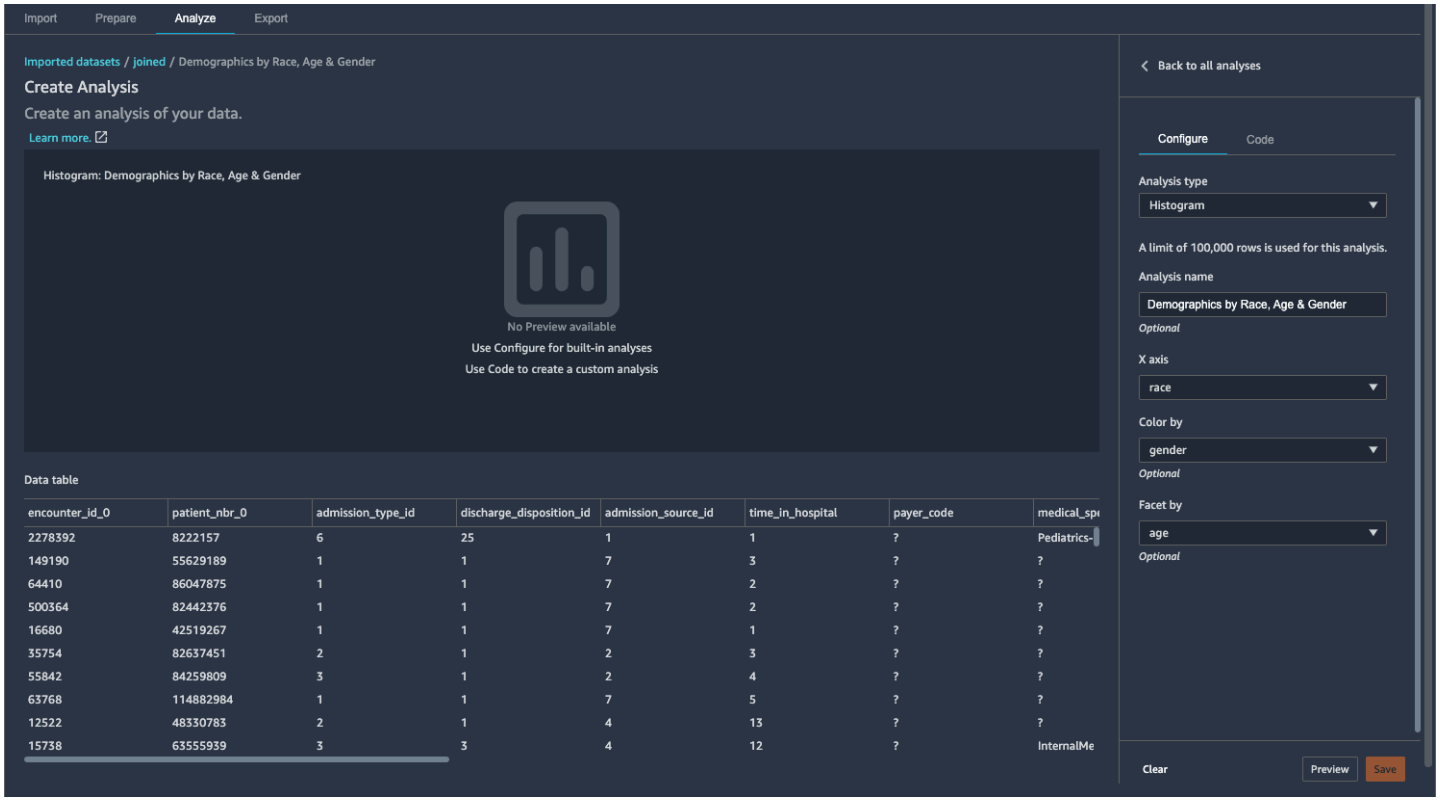
1) Click + sign next to Join flow icon and choose `Add analysis`



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



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`.



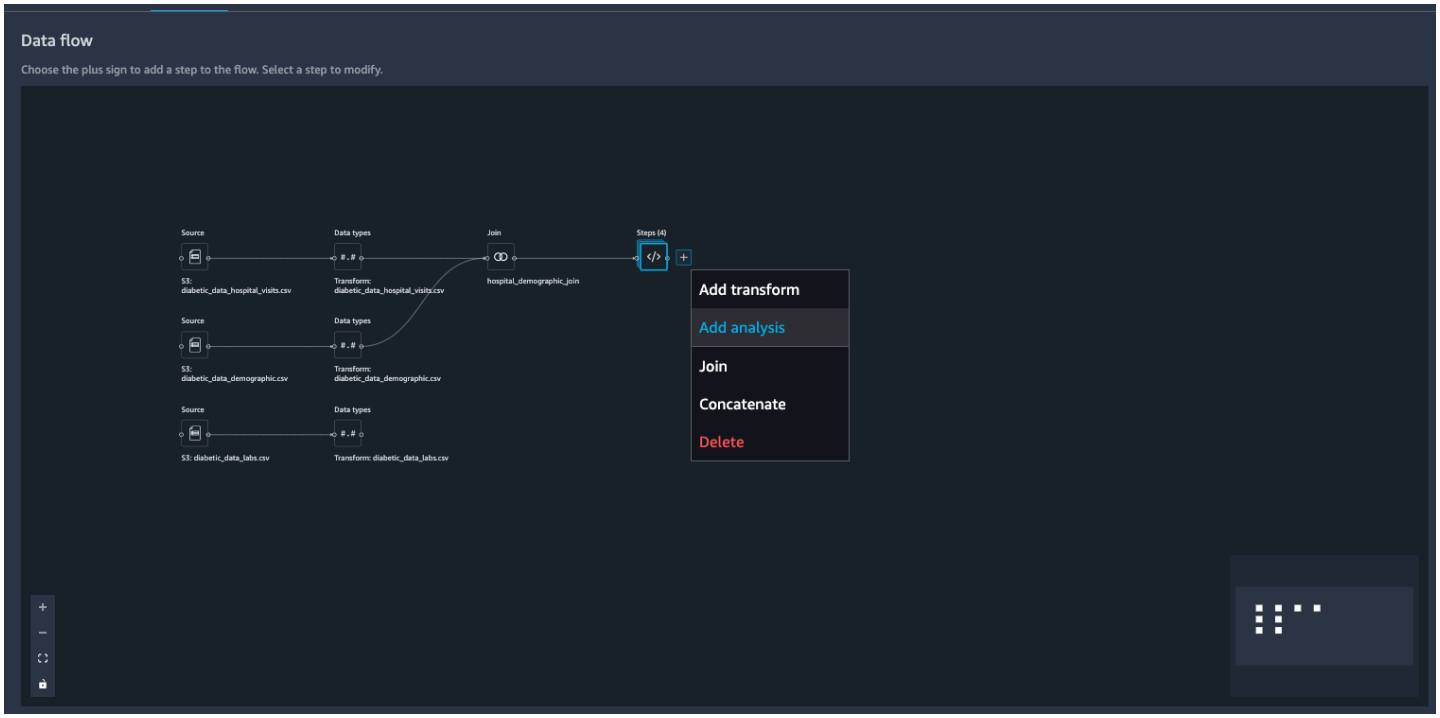
4) Click `Preview` and wait for the model to be results to be displayed on the screen

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

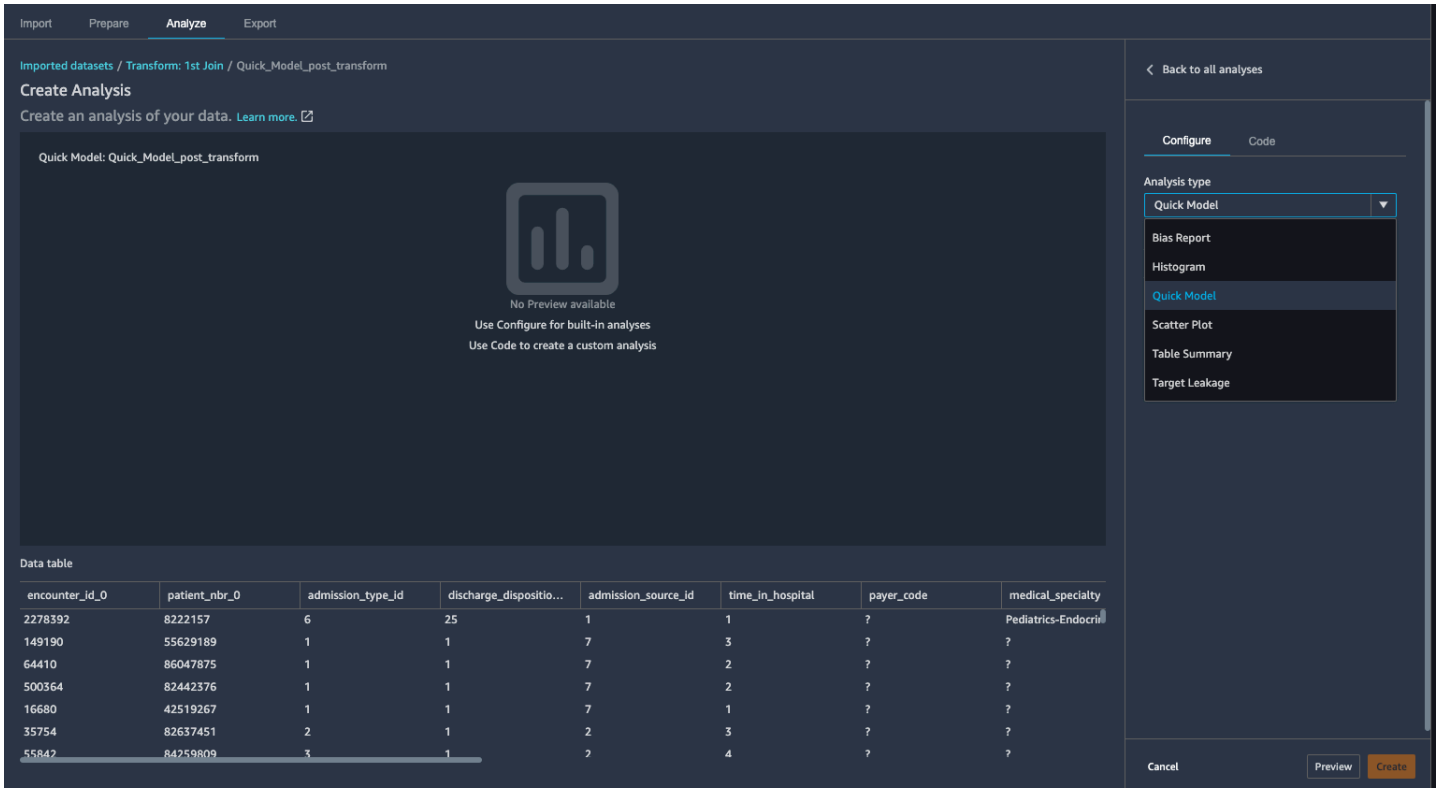
Quick Model

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.

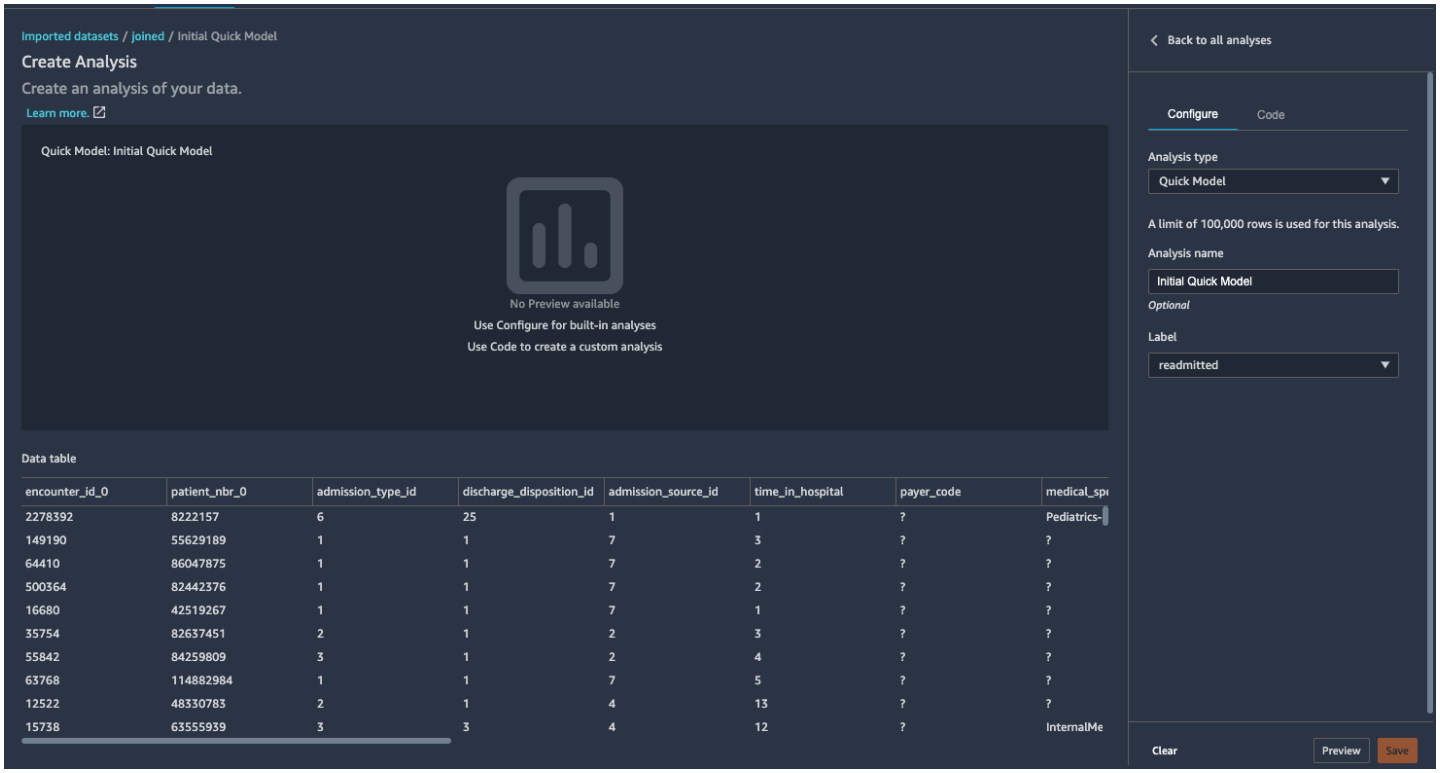
1) Click + sign next to Join flow icon and choose `Add analysis`



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



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



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

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.

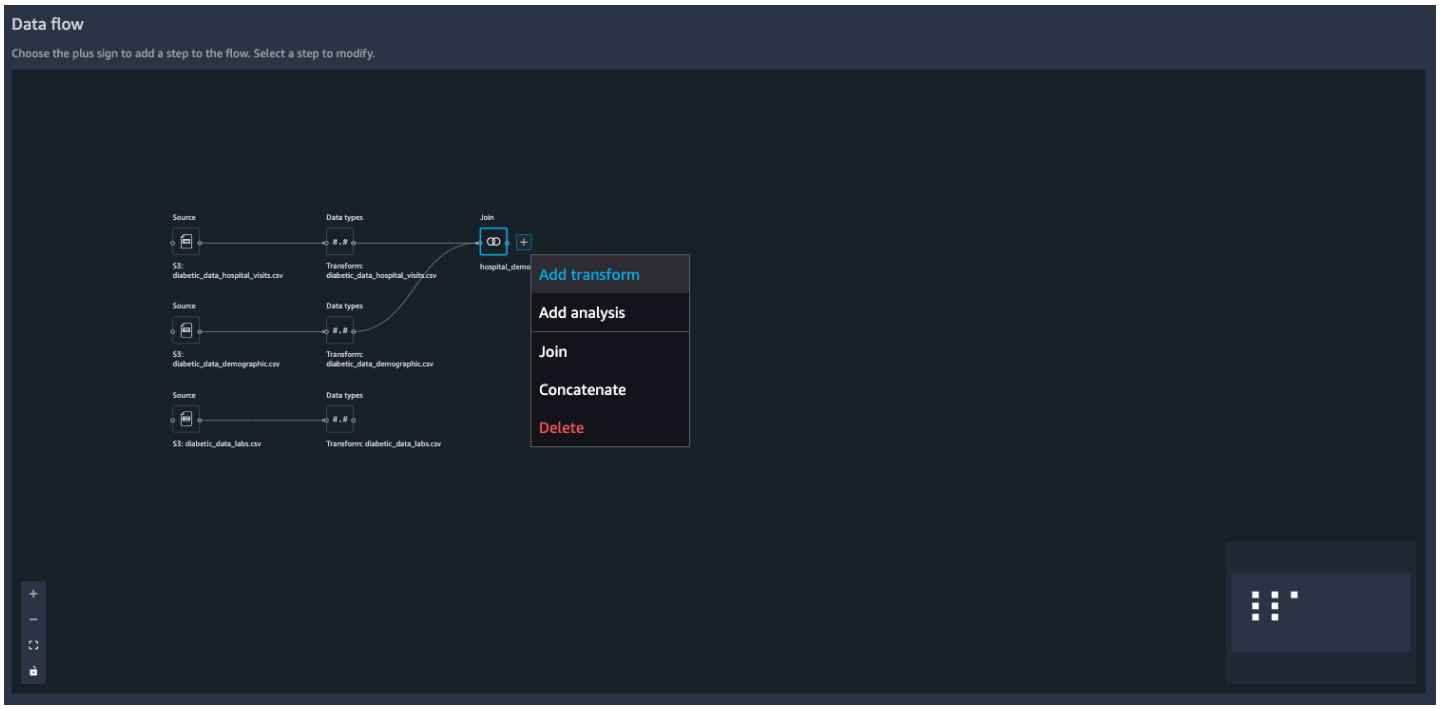
5) Click `Create` button to add the quick model analysis to the data flow.

Transformations

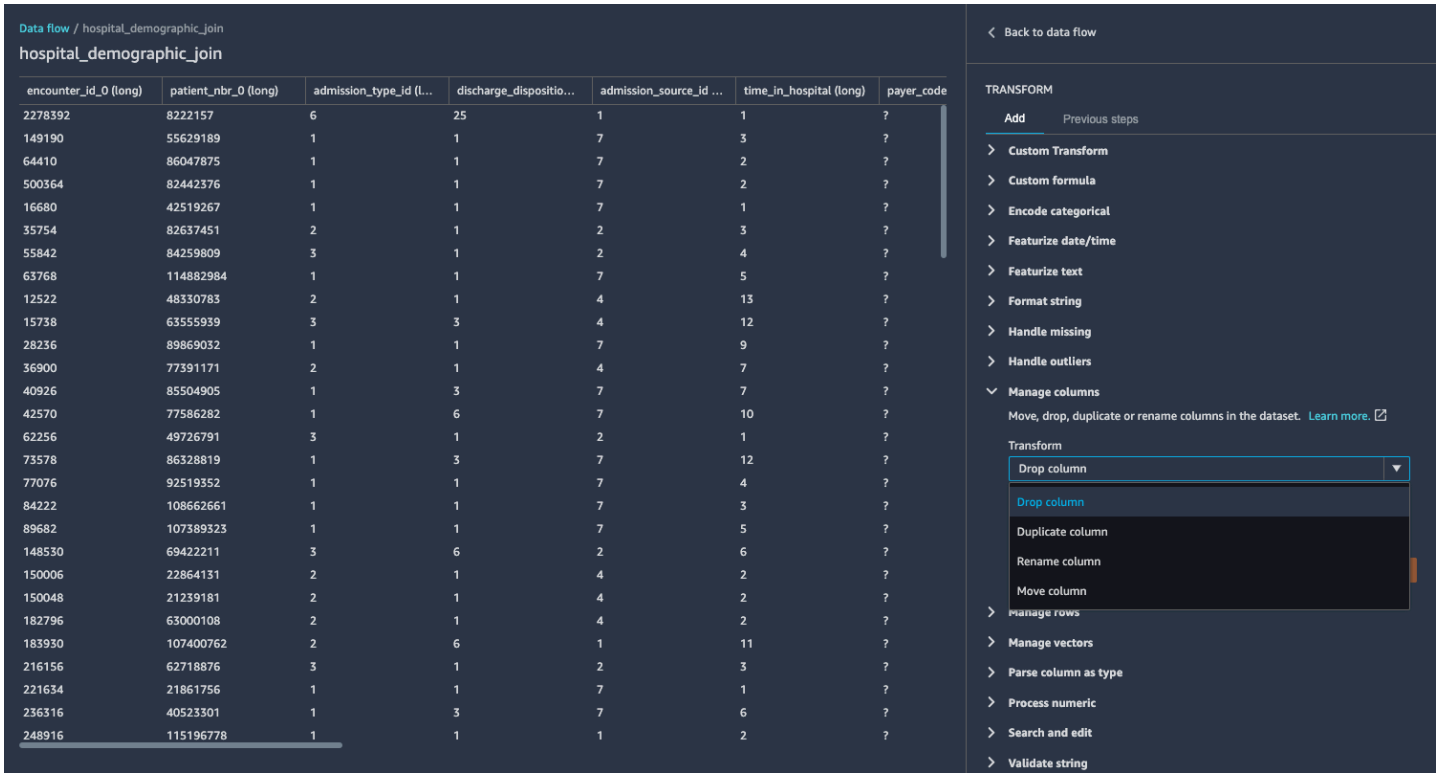
We will use Data Wrangler built-in transforms to apply the listed transformations to our dataset.

No Duplicate Columns

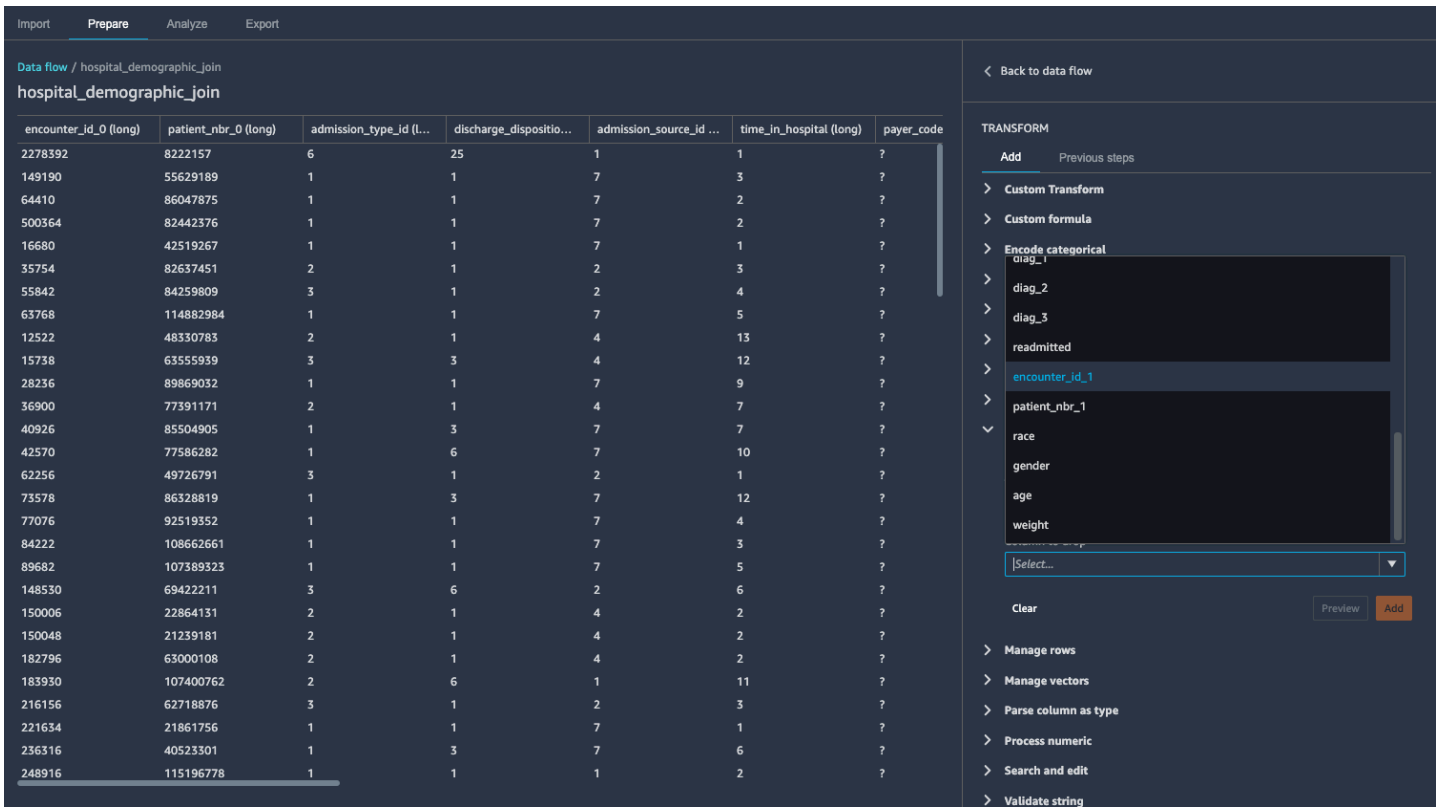
1) Click + sign next to Join flow icon and choose `Add Transform`



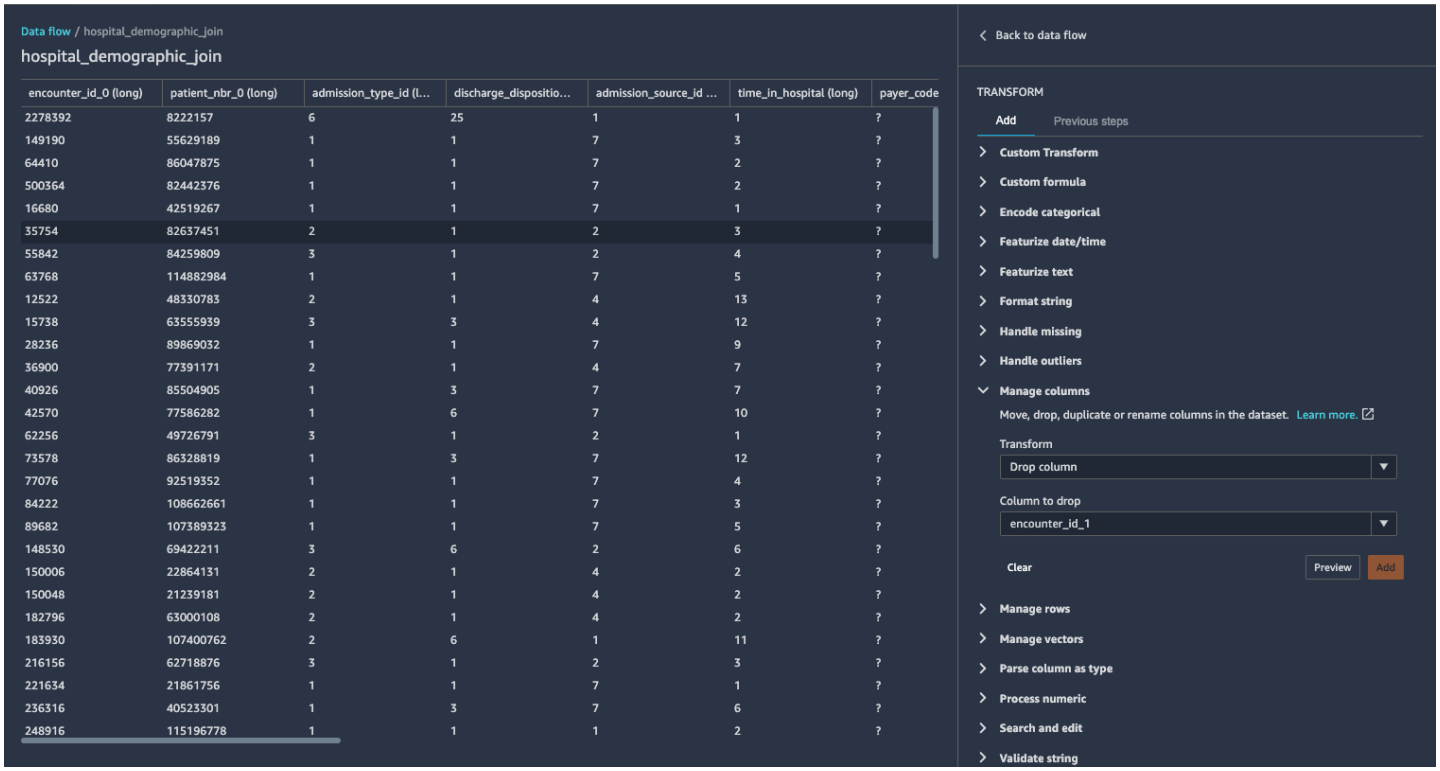
2) Pick `Manage Columns` from the list of transforms on the right panel



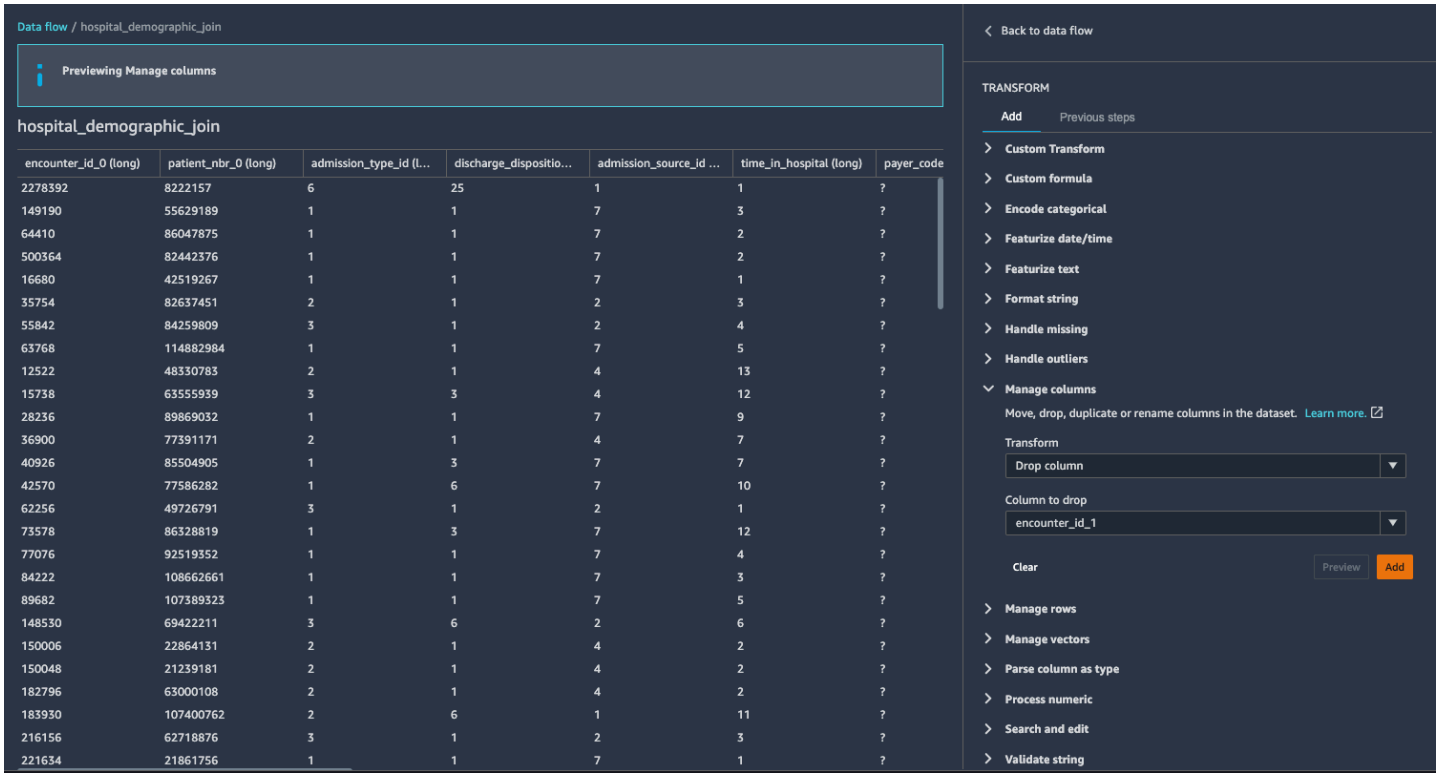
3) Choose `Drop Column` transform and select `encounter_id_1` column to drop



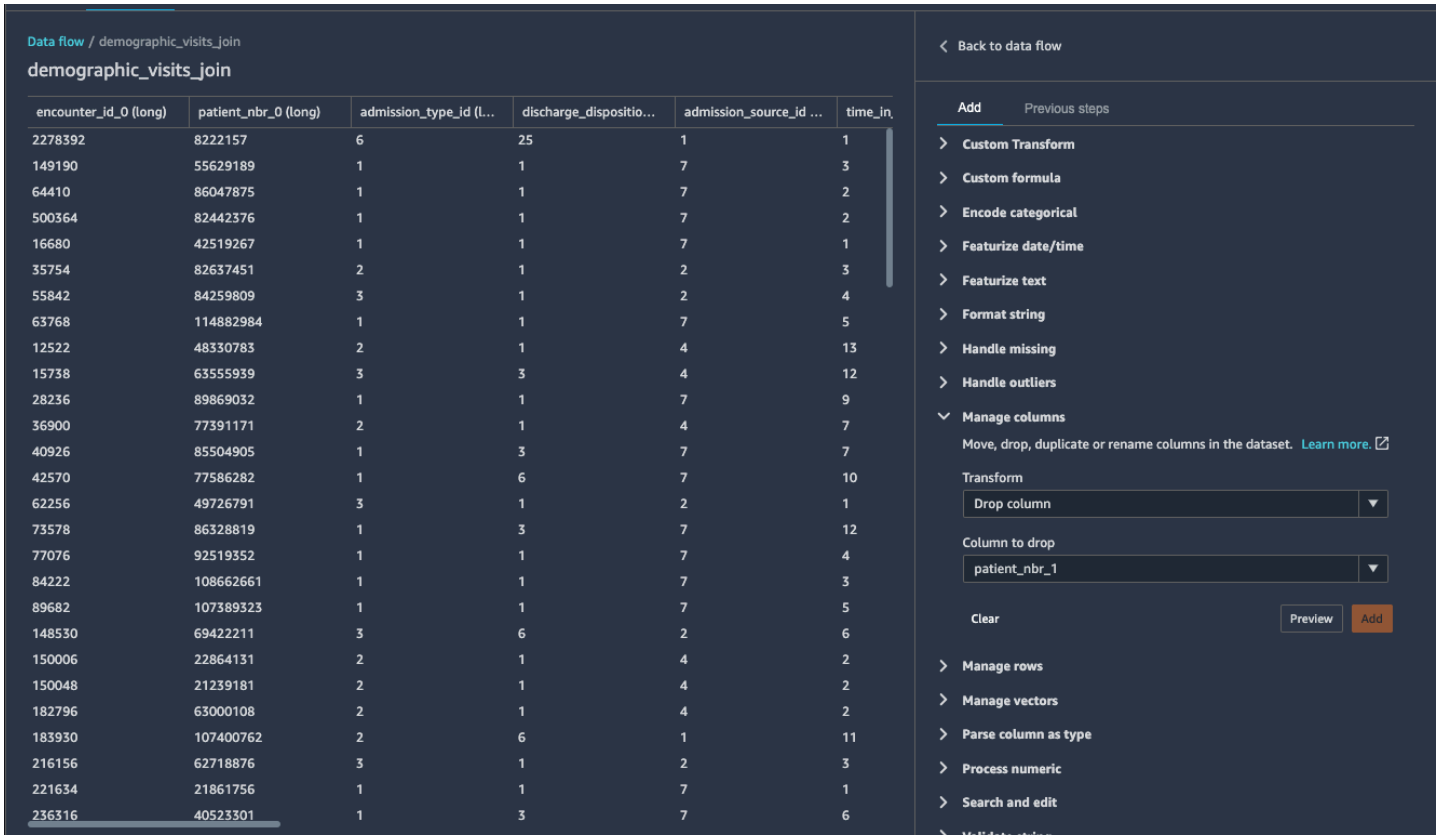
4) Click `Preview` to preview the changes to the data set



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

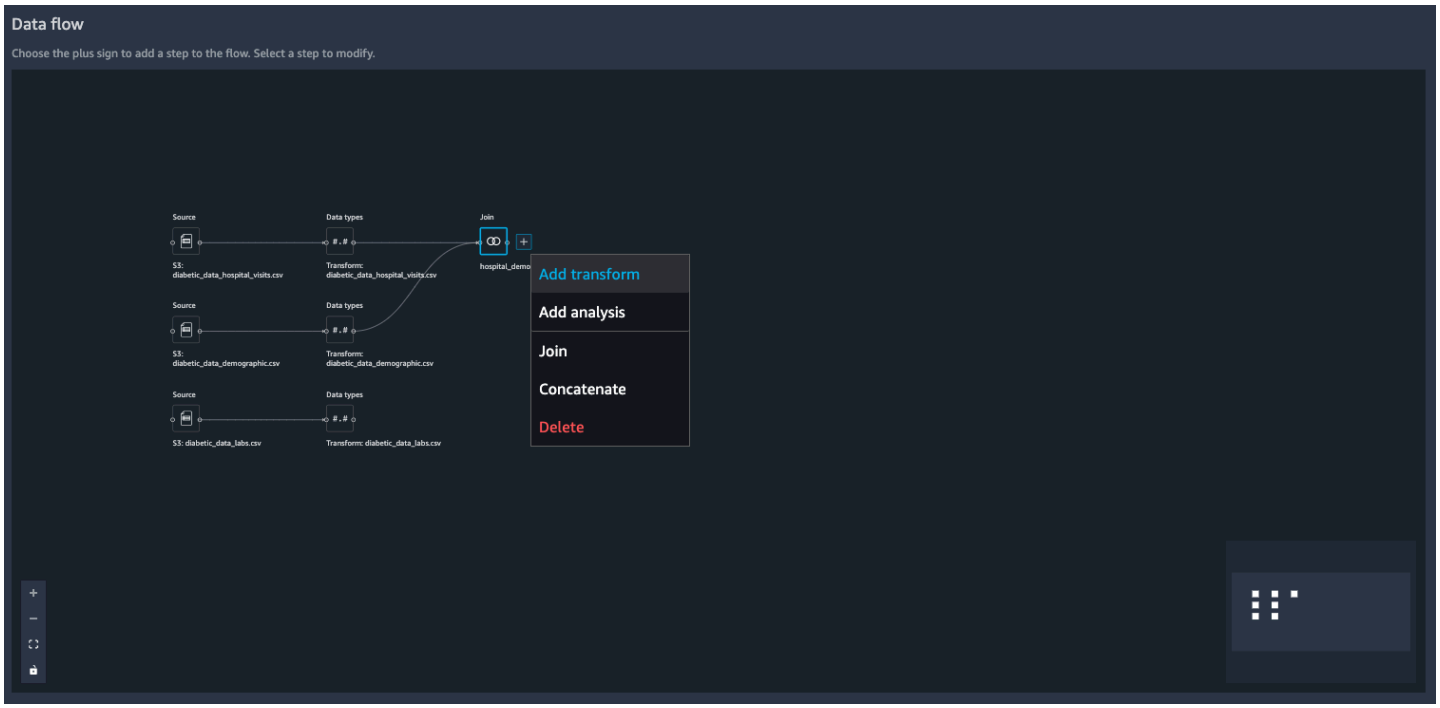


6) Repeat above steps 1 through 5 for `patient_nbr_1` column.

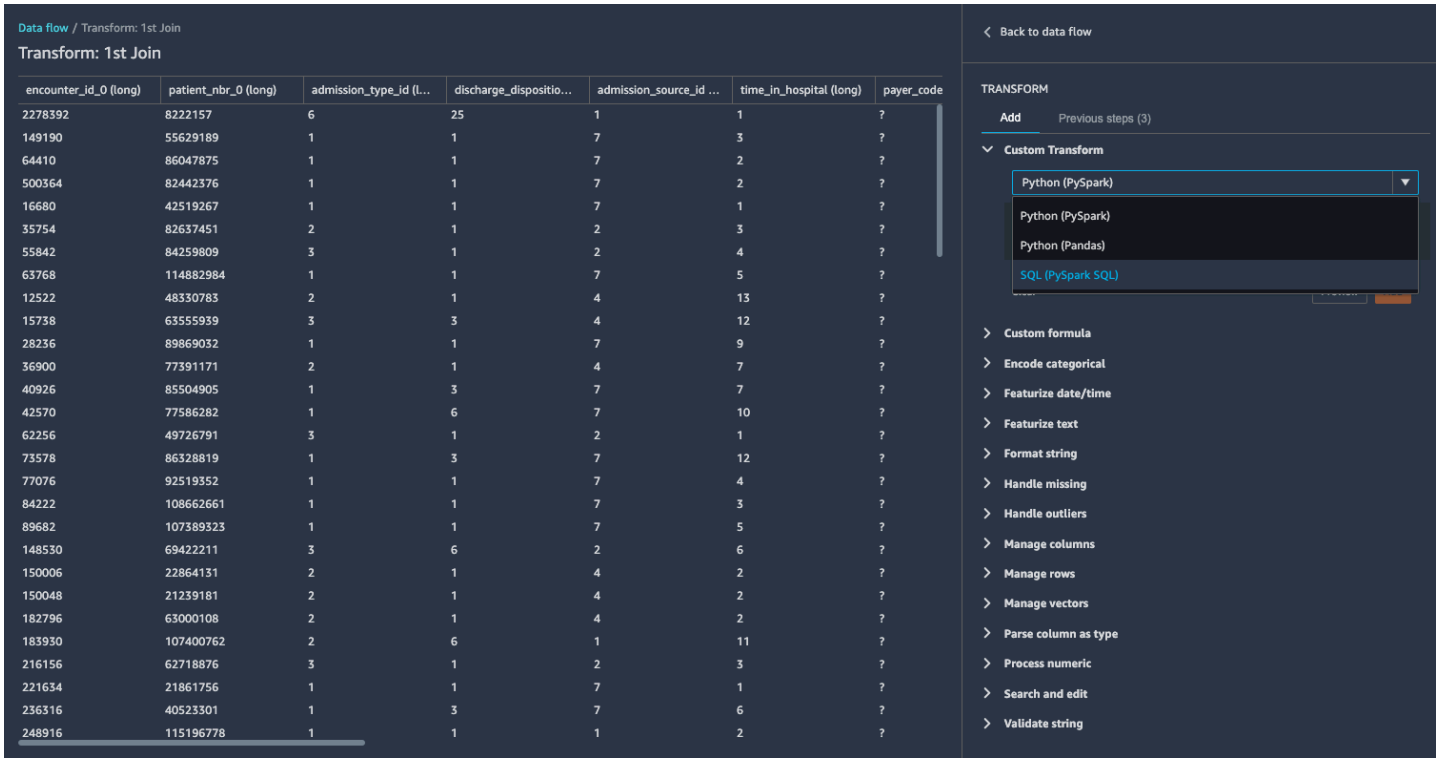


No Duplicate Rows/Observations

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

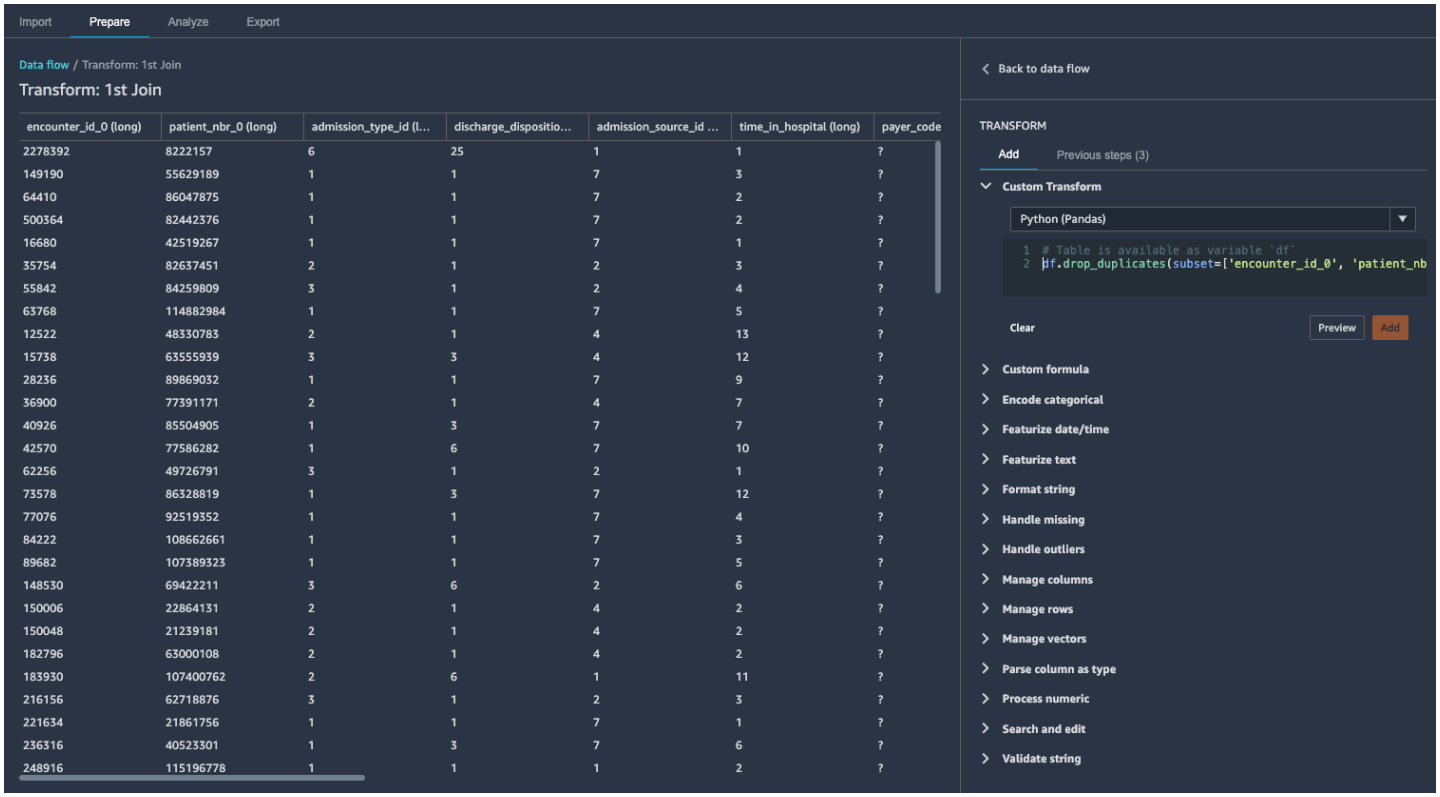


2) Pick `Custom Transform` from the list of transforms on the right panel

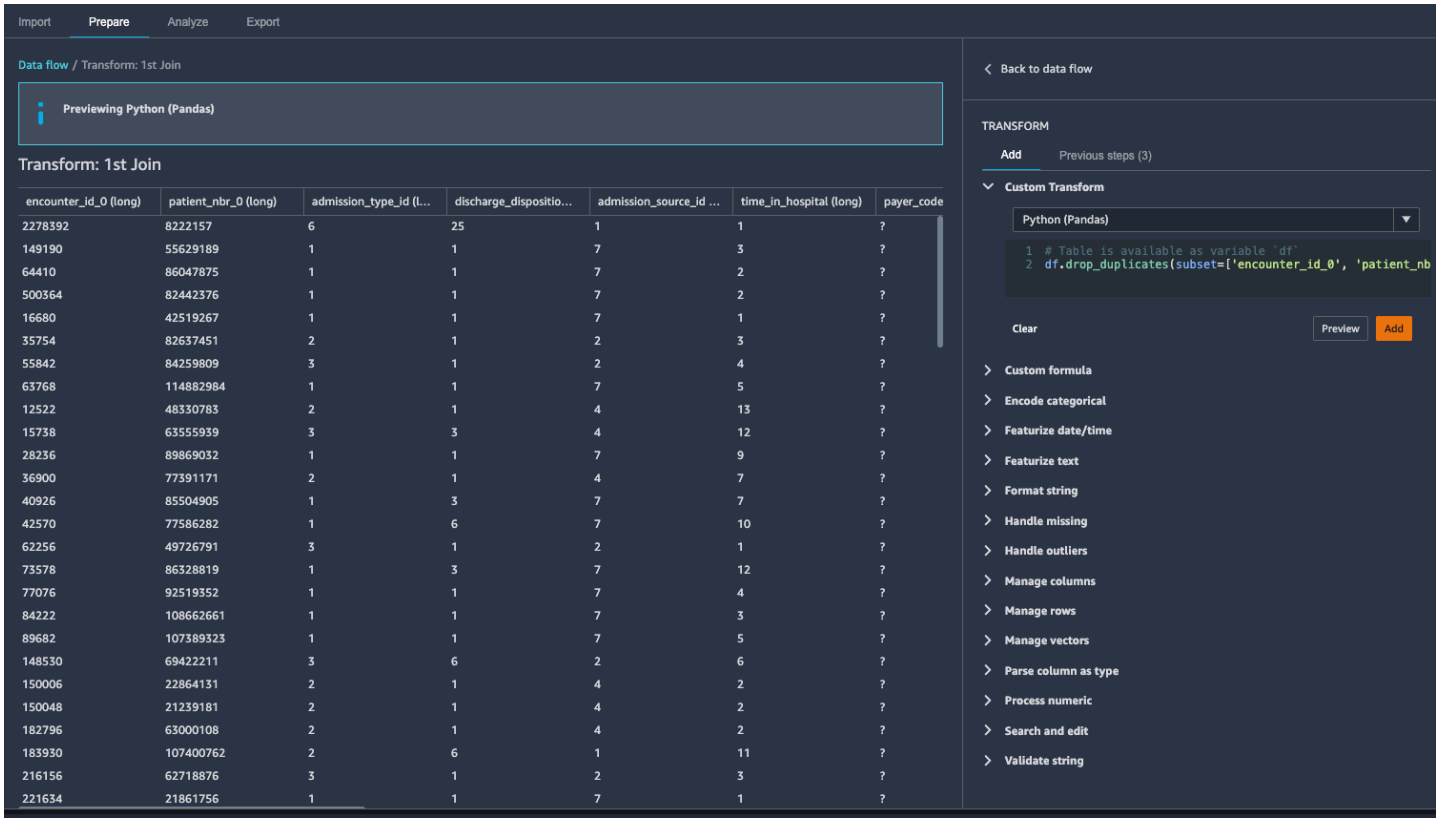


3) select `Python (Pandas)` and enter below line of code in the text box. Then click `Preview` to view the results.

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



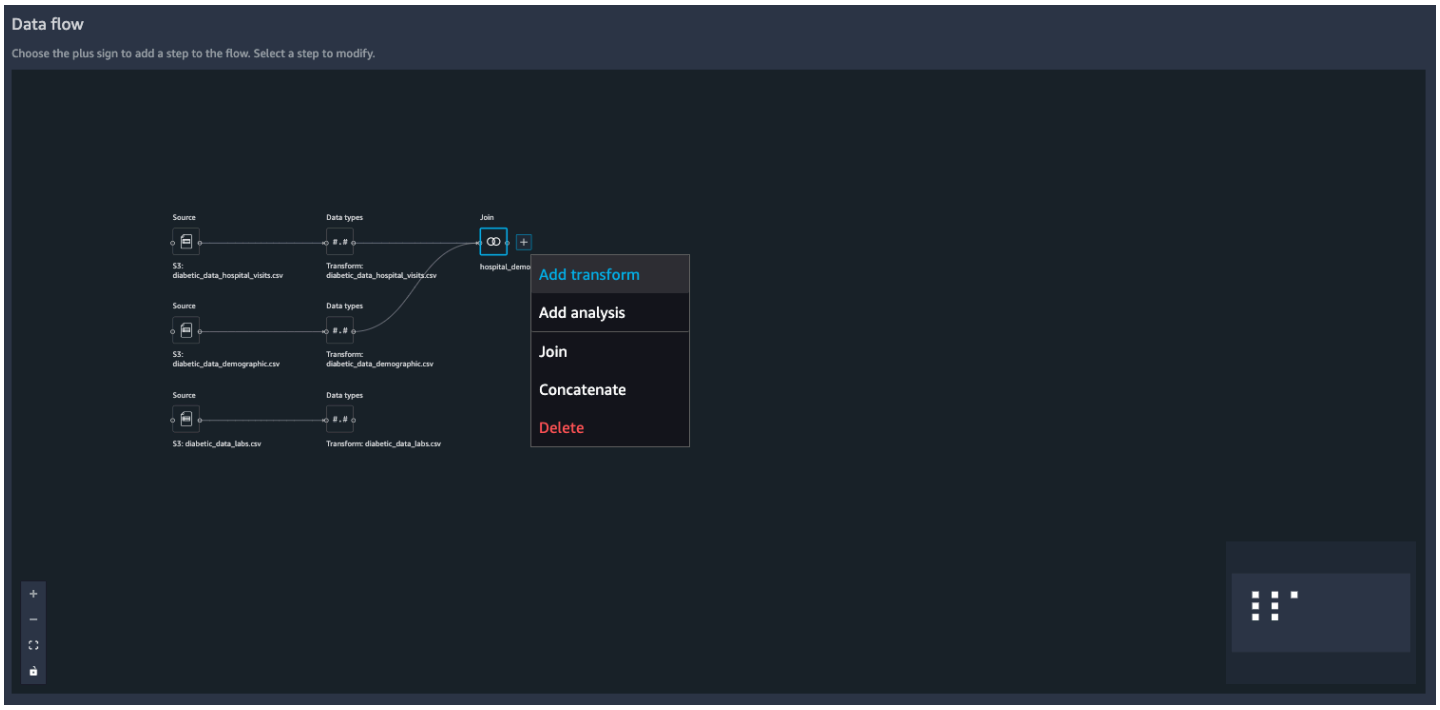
4) Click `Add` to add the change to the data flow



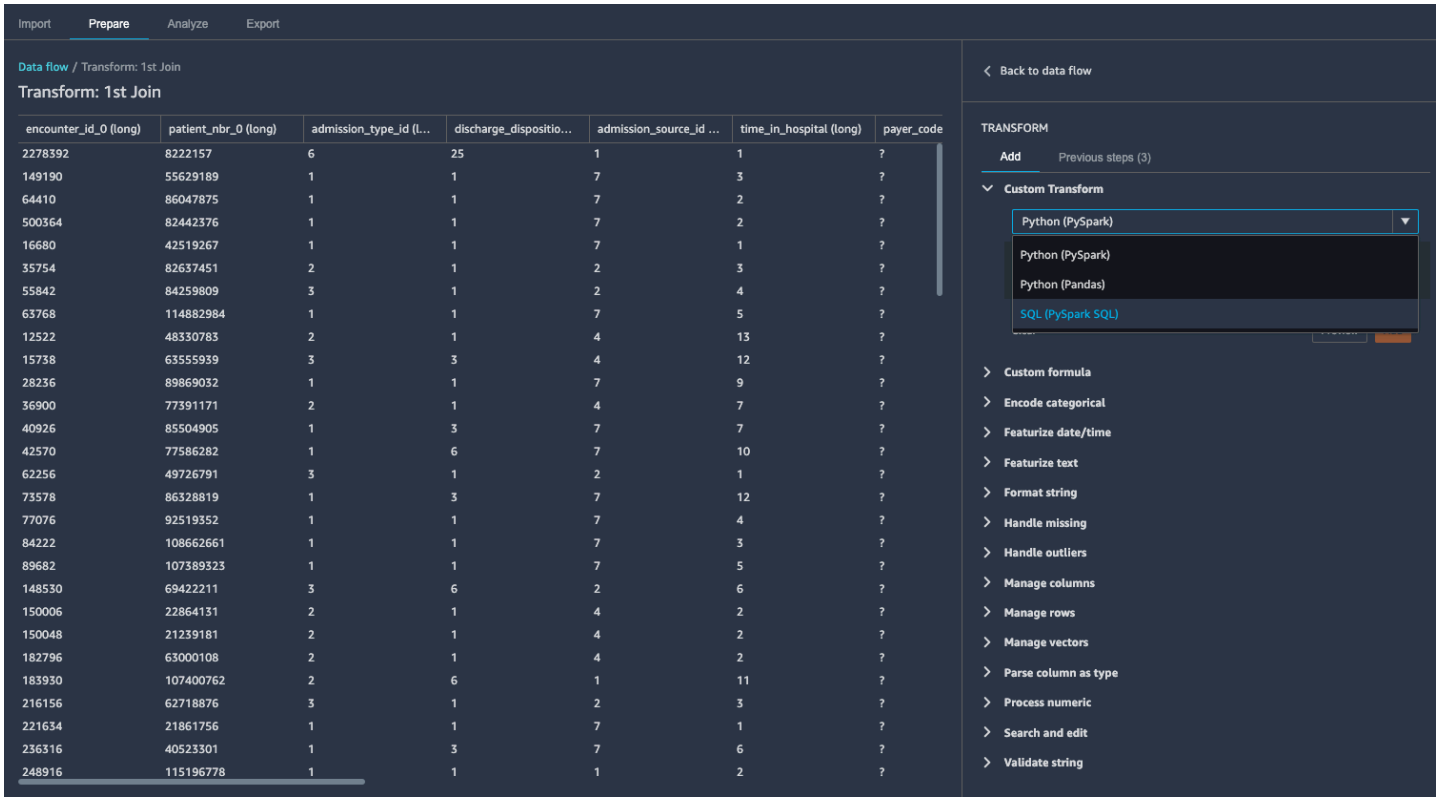
Custom Transformation - Add new features to your dataset

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`

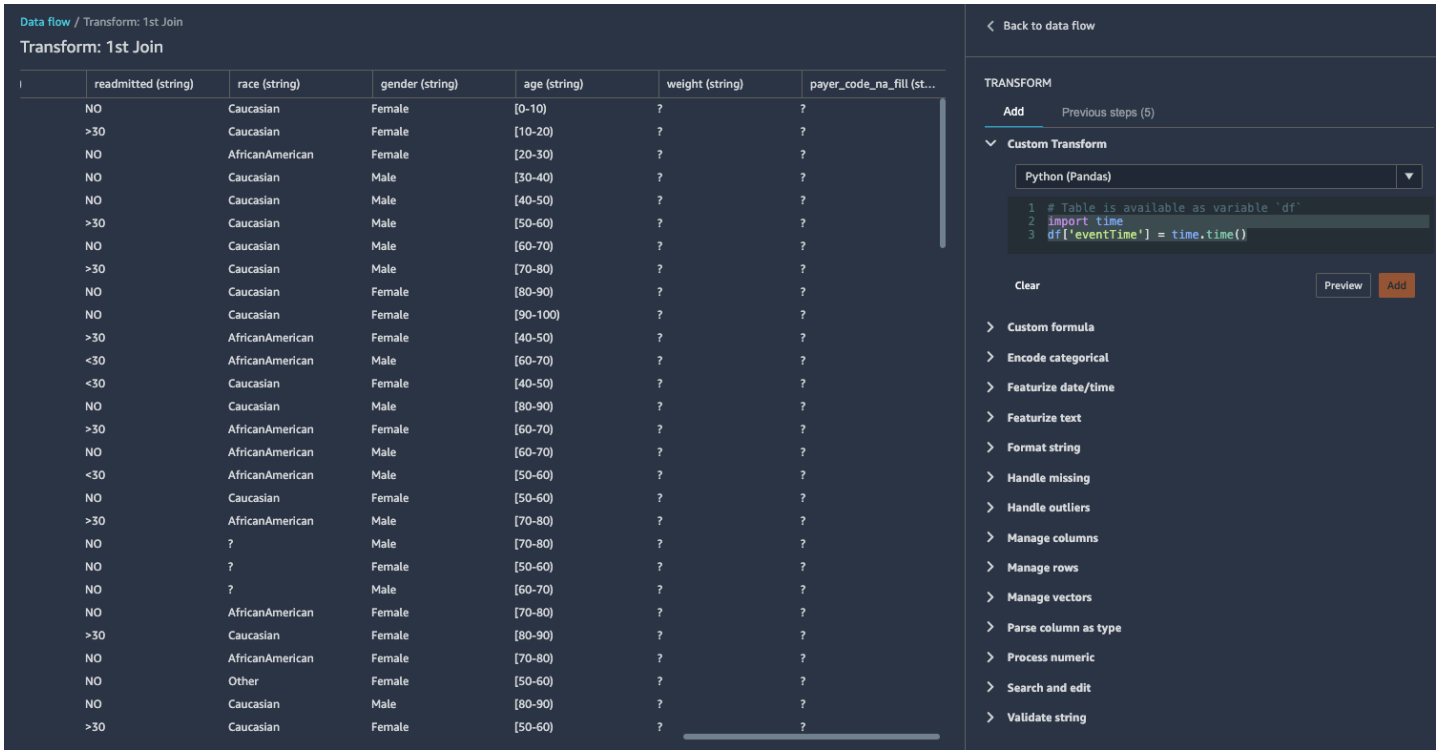


2) Pick `Custom Transform` from the list of transforms on the right panel

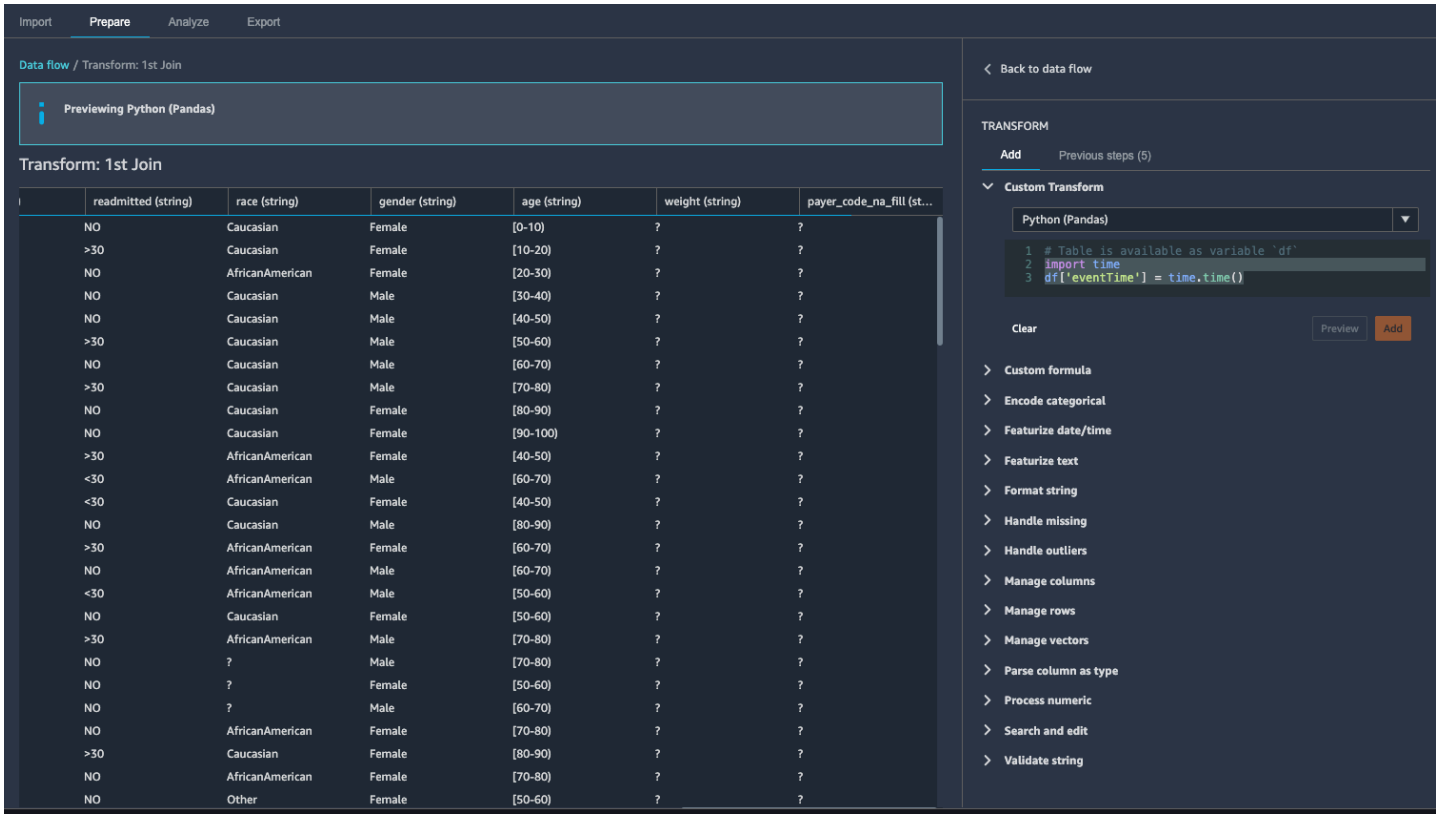


3) select `Python (Pandas)` and enter below line of code in the text box. Then click `Preview` to view the results.

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



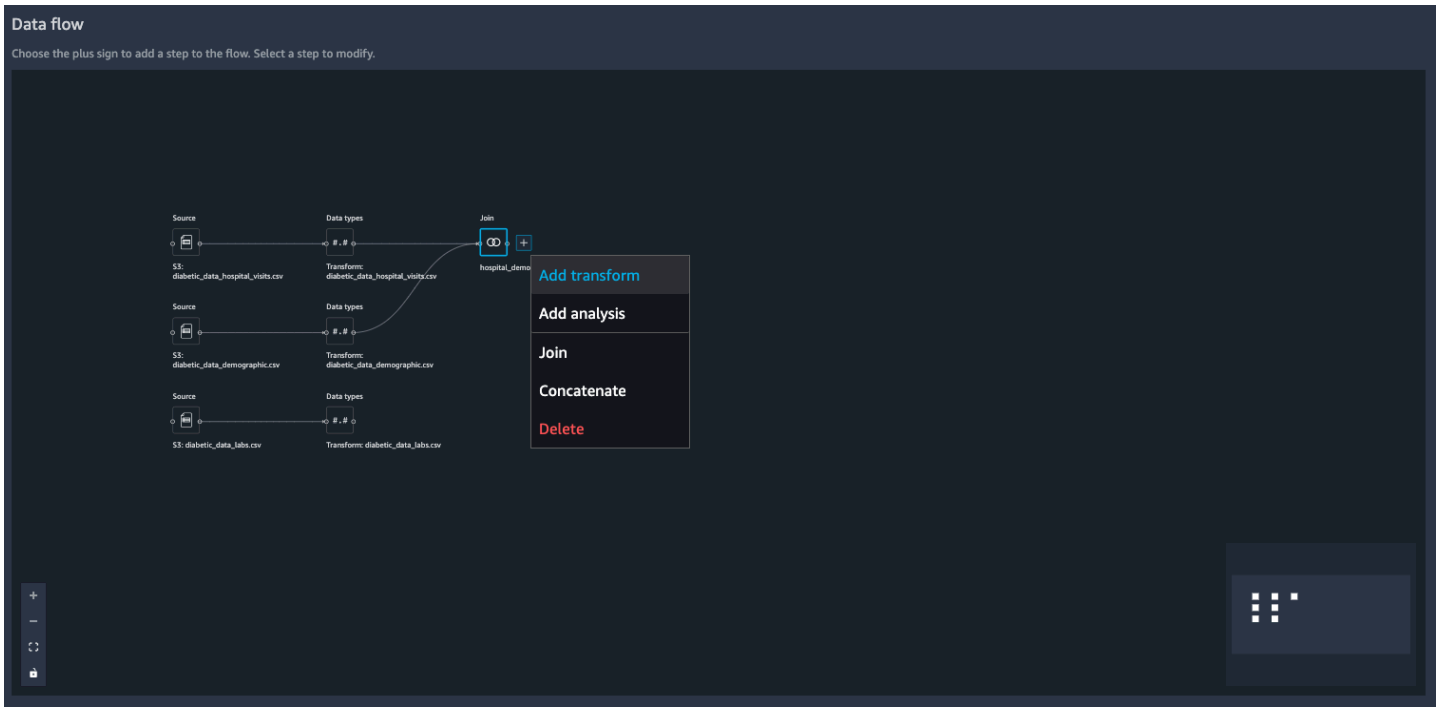
4) Click `Add` to add the change to the data flow



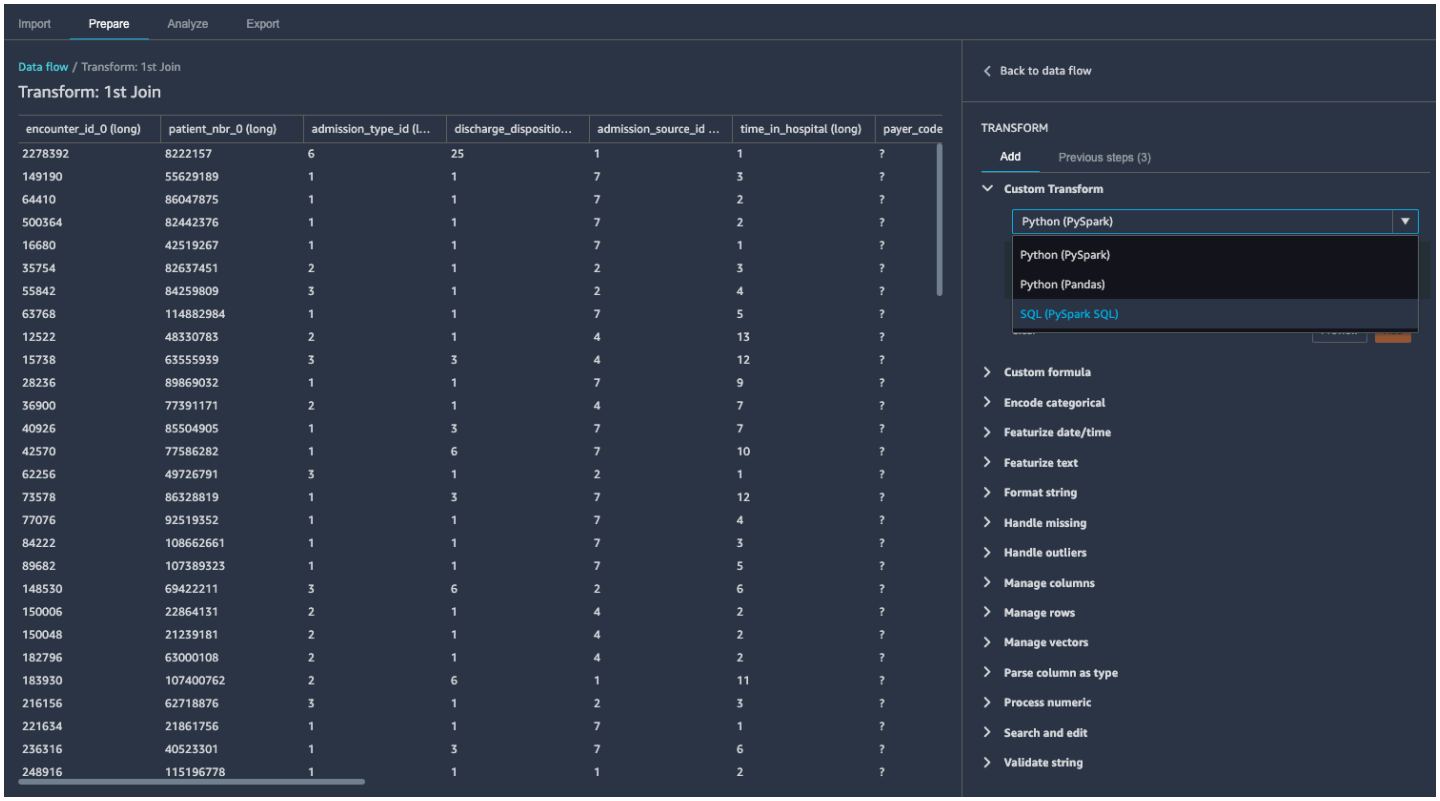
More Custom Transformations - impute fillers for undesirable

values

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



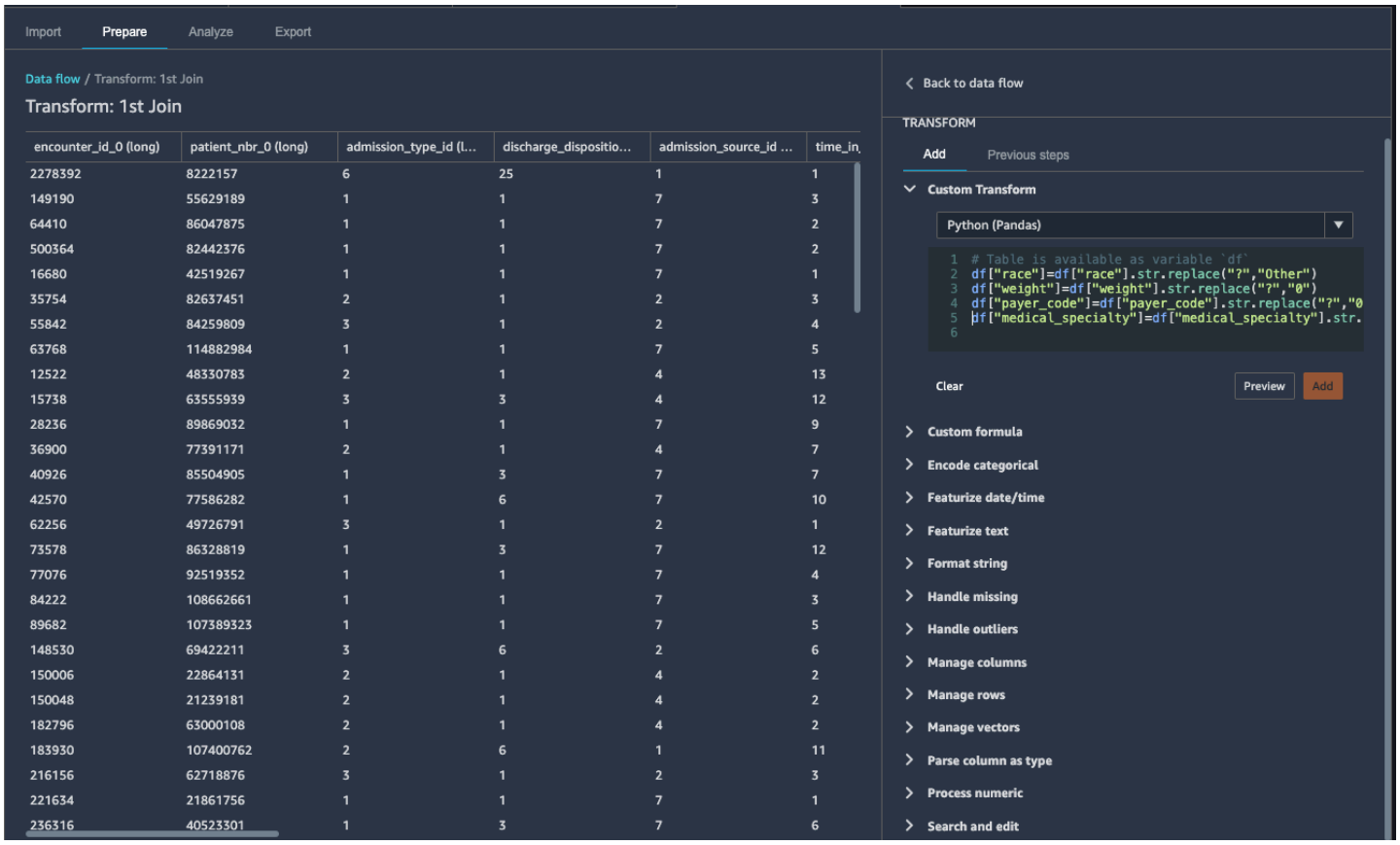
2) Pick `Custom Transform` from the list of transforms on the right panel



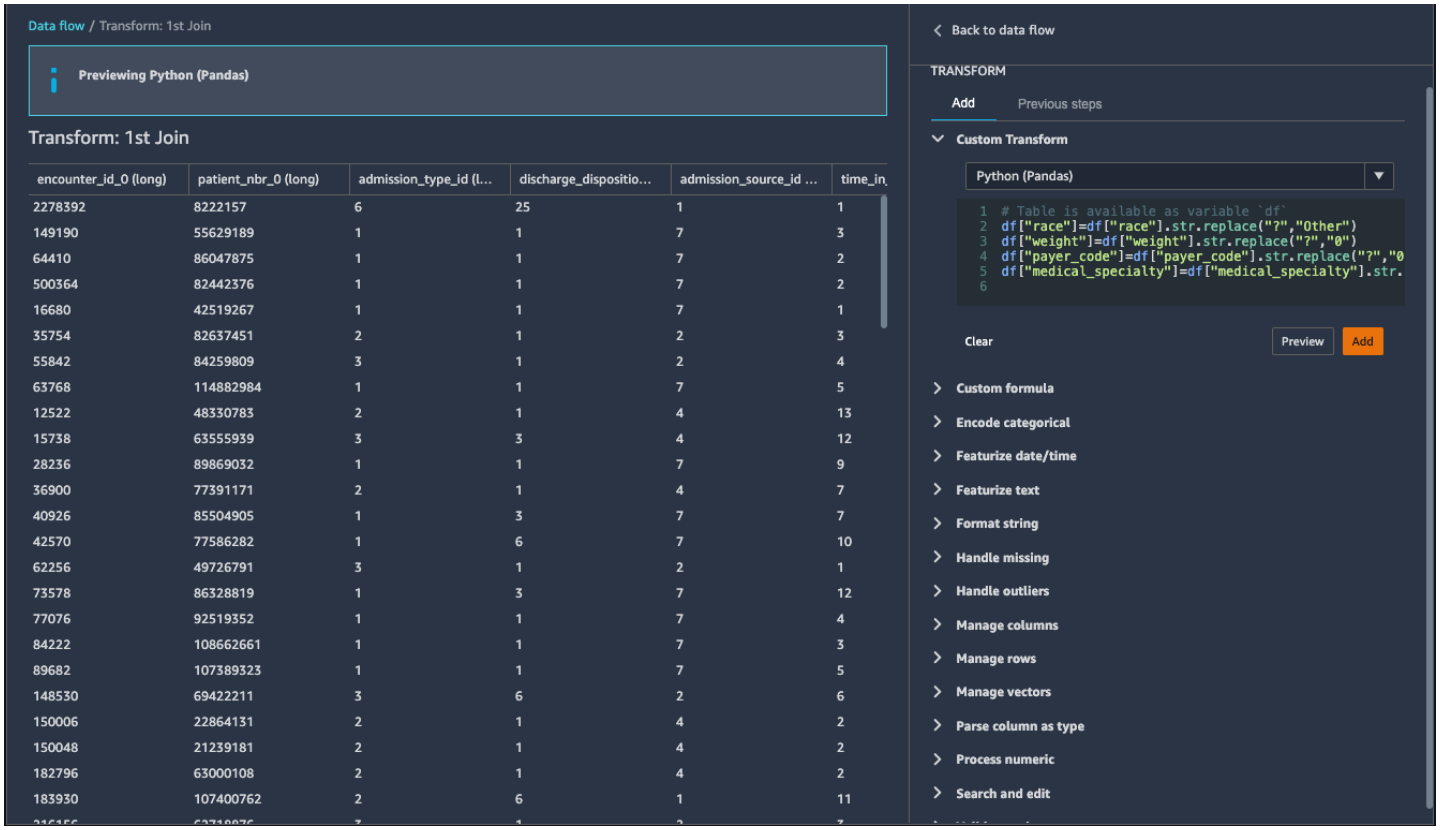
3) select `Python (Pandas)` and enter below line of code in the text box. Then click `Preview` to view the

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")
```

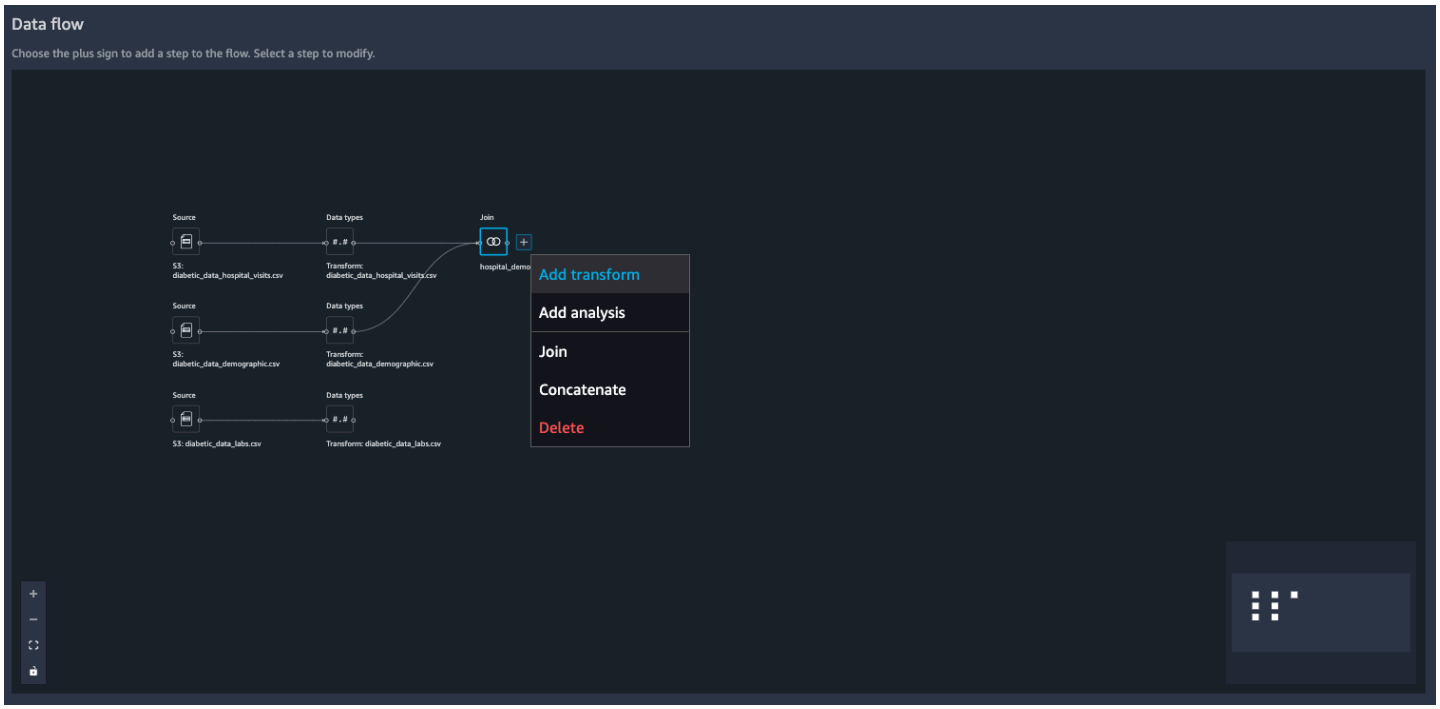


4) Click `Add` to add the change to the data flow

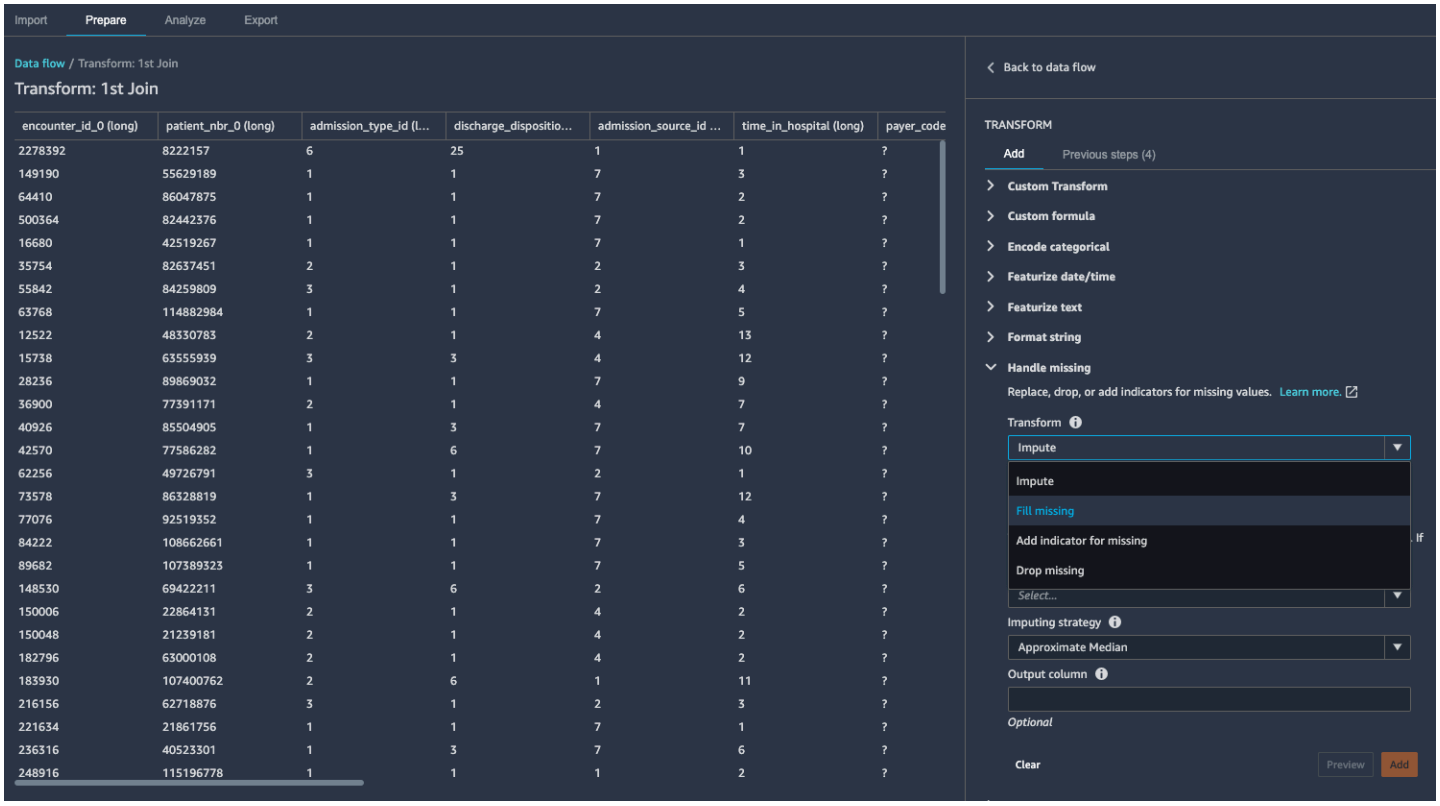


Handle missing values

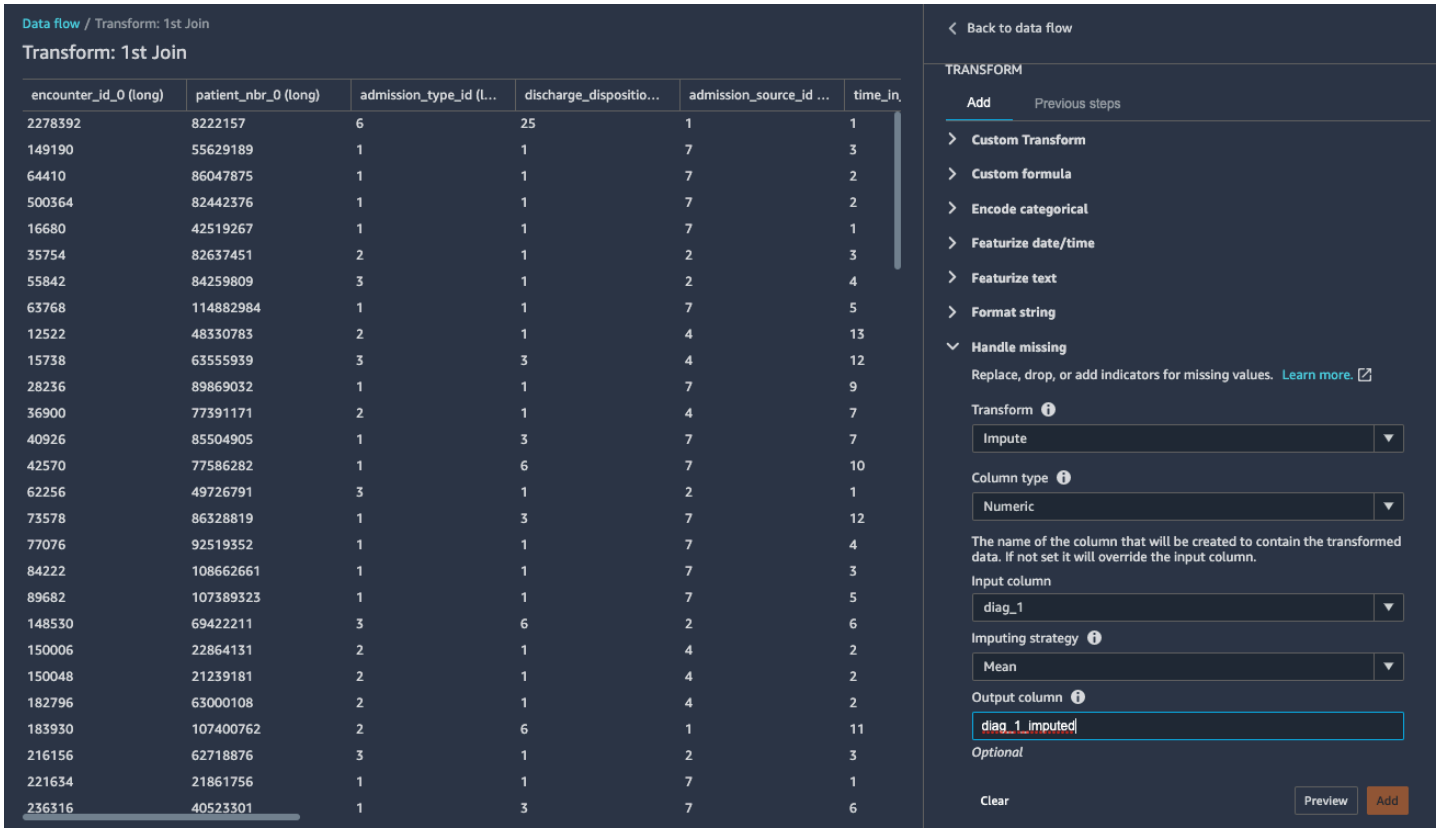
1) Click + sign next to Join flow icon and choose `Add Transform`



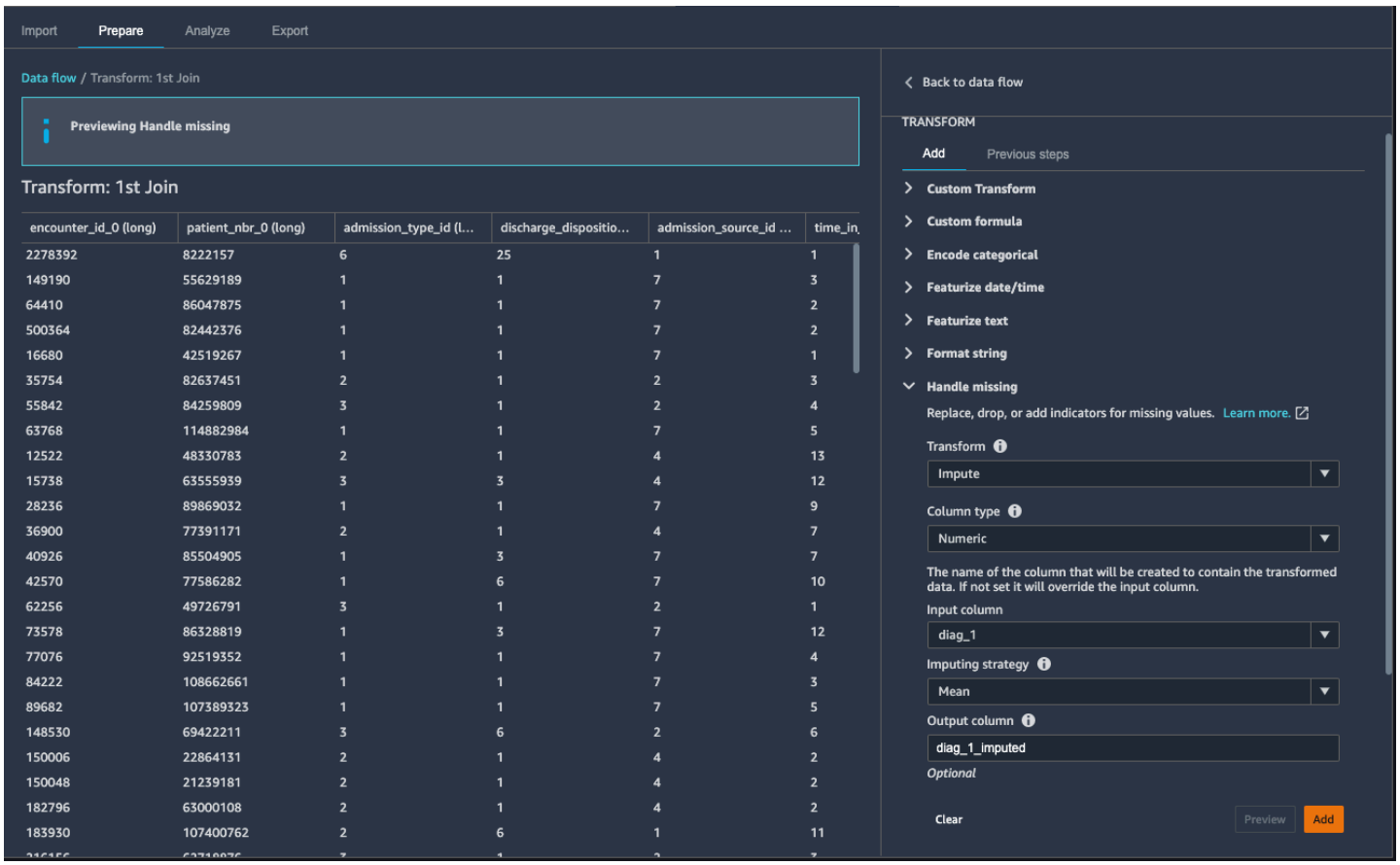
2) Pick `Handle missing` from the list of transforms on the right panel and choose `Impute` for Transform



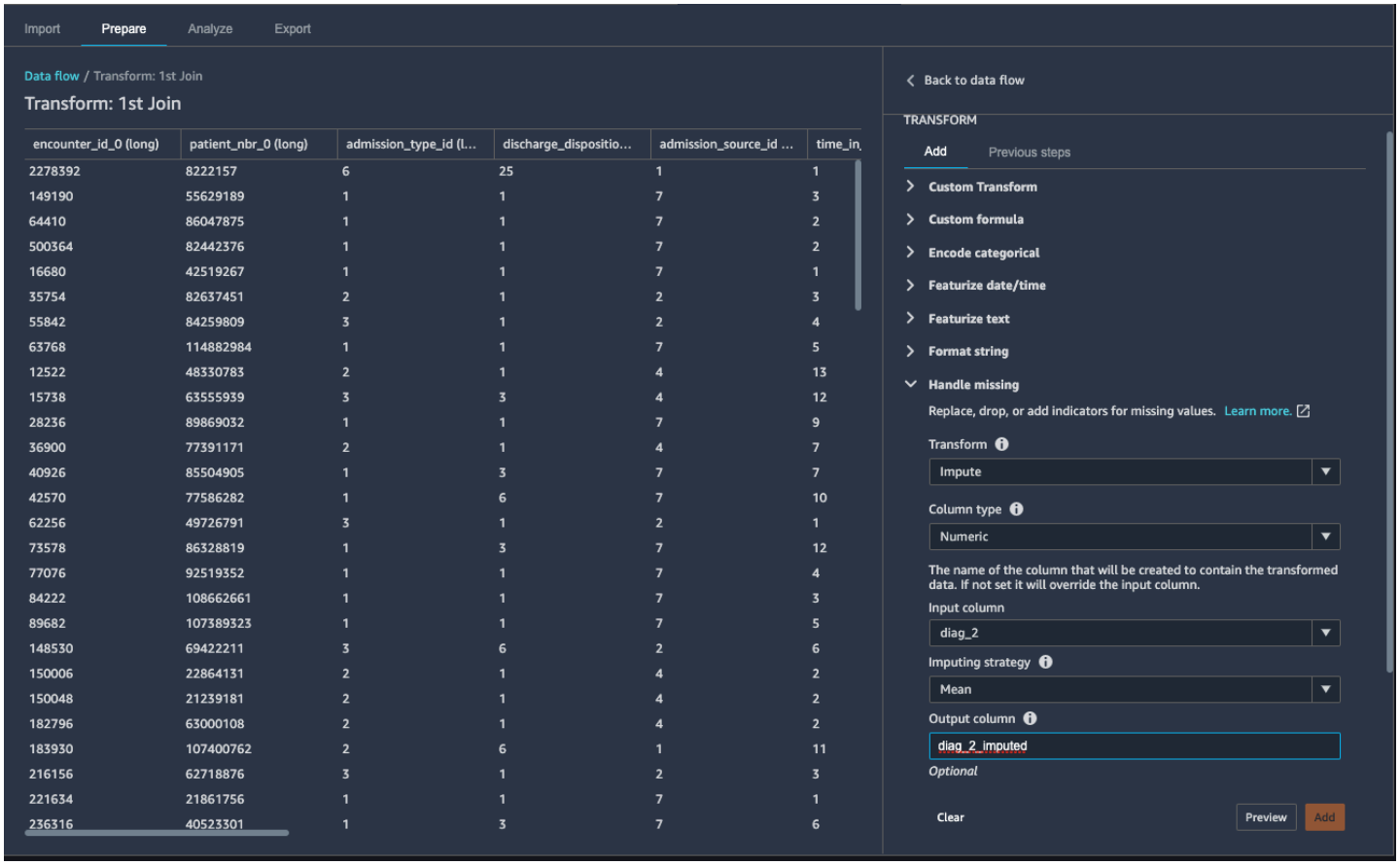
3) Choose `Column type` as `Numeric` and select `Input column` as `diag_1`. Let's use `Mean` for `Imputing strategy`. You can also provide optional Output column name.



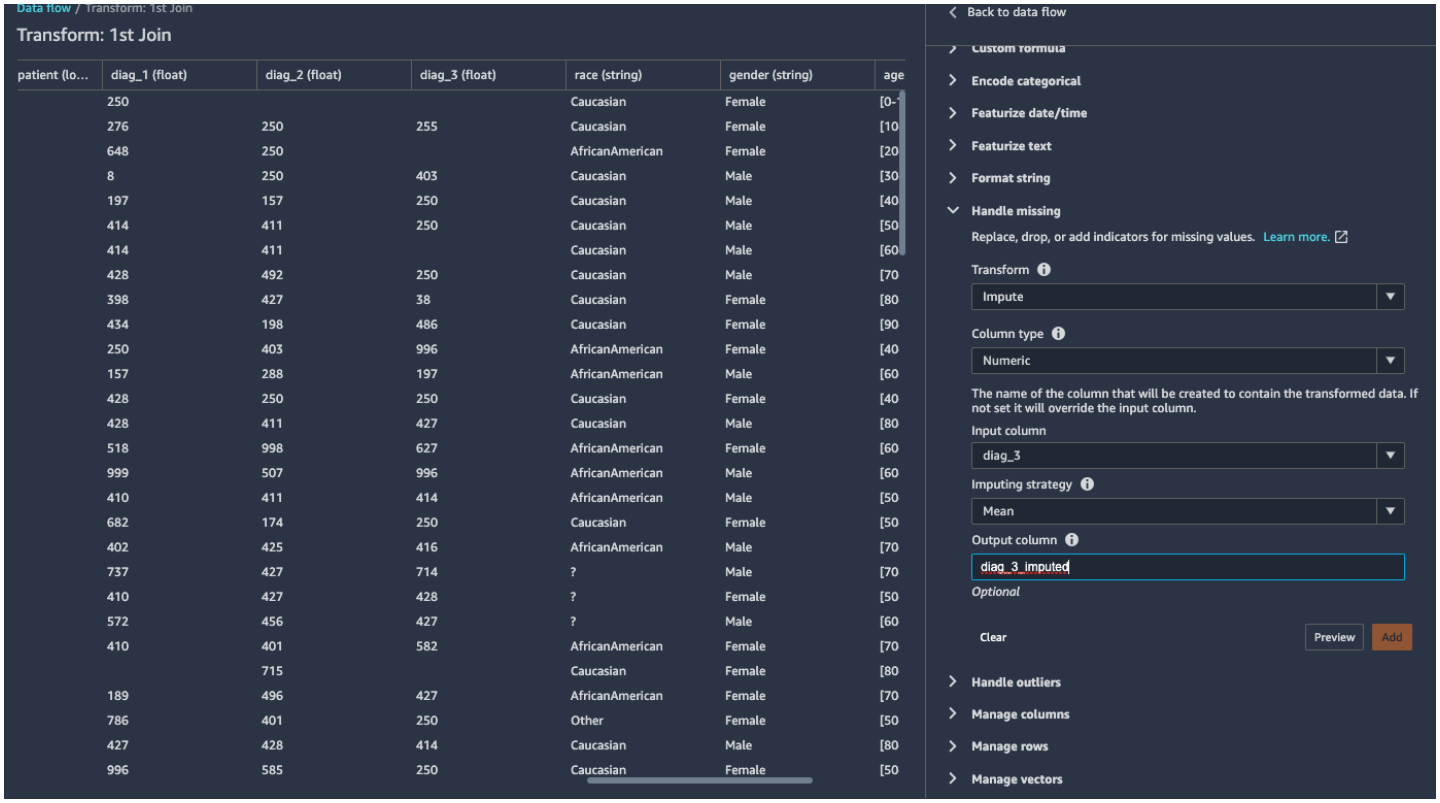
4) Click `Add` to add the change to the data flow

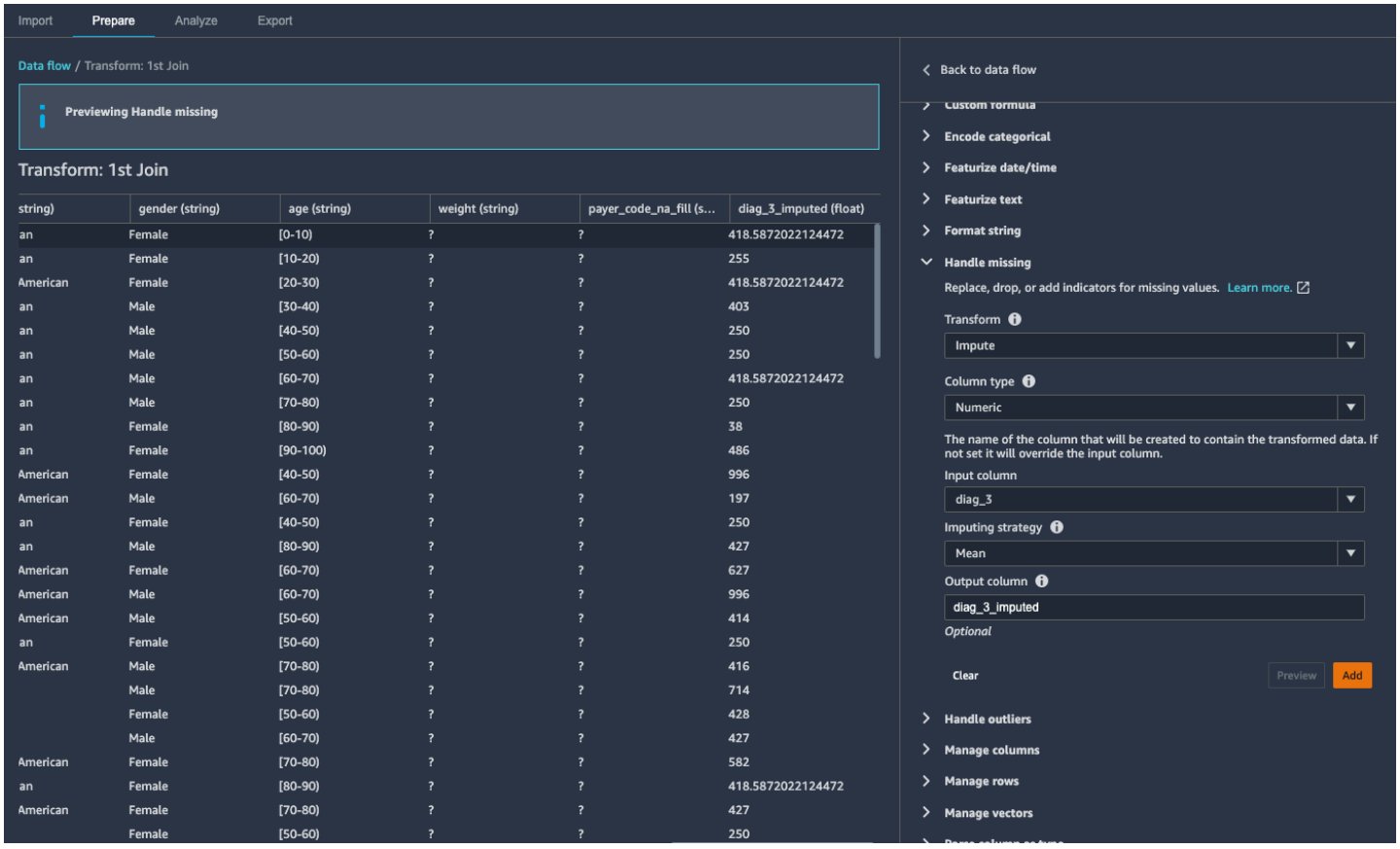


5) Repeat above steps 1 through 4 for `diag_2` feature as shown below



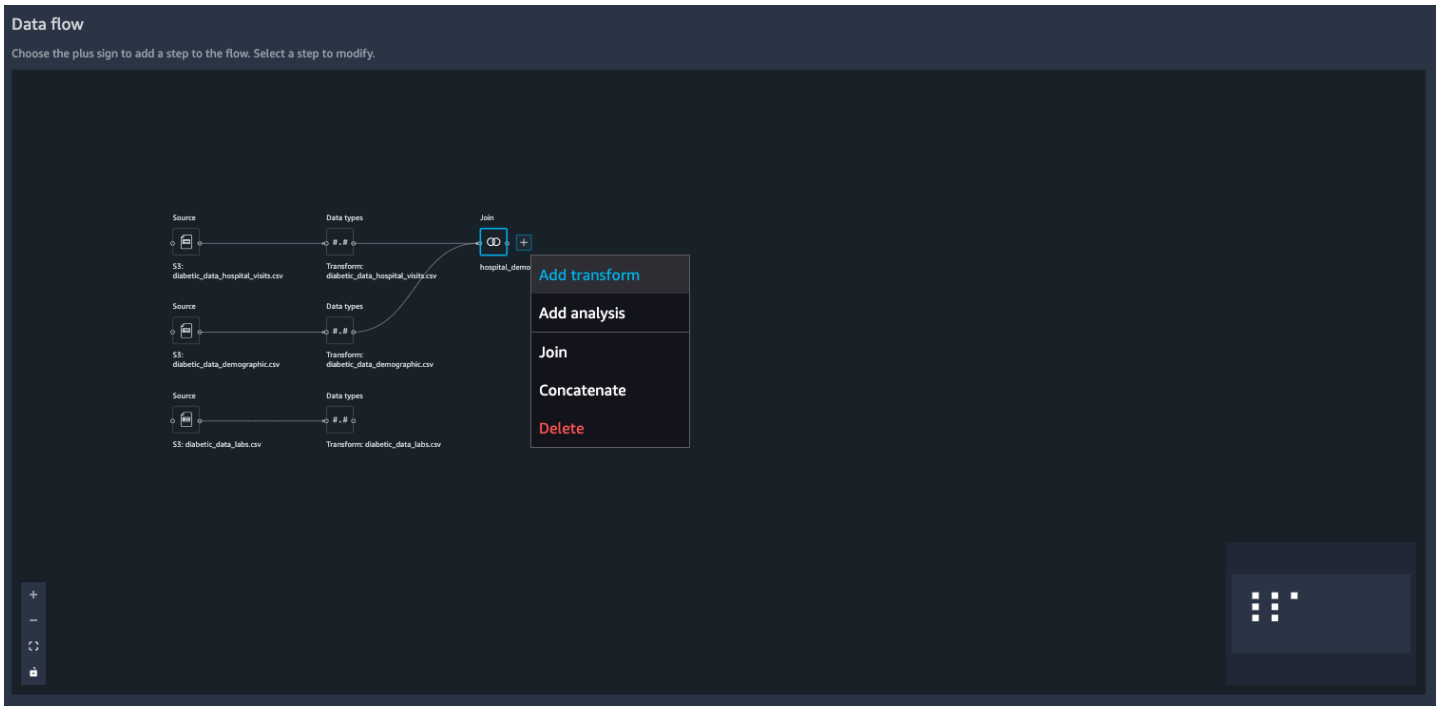
6) Repeat above steps 1 through 4 for `diag_3` feature as shown below



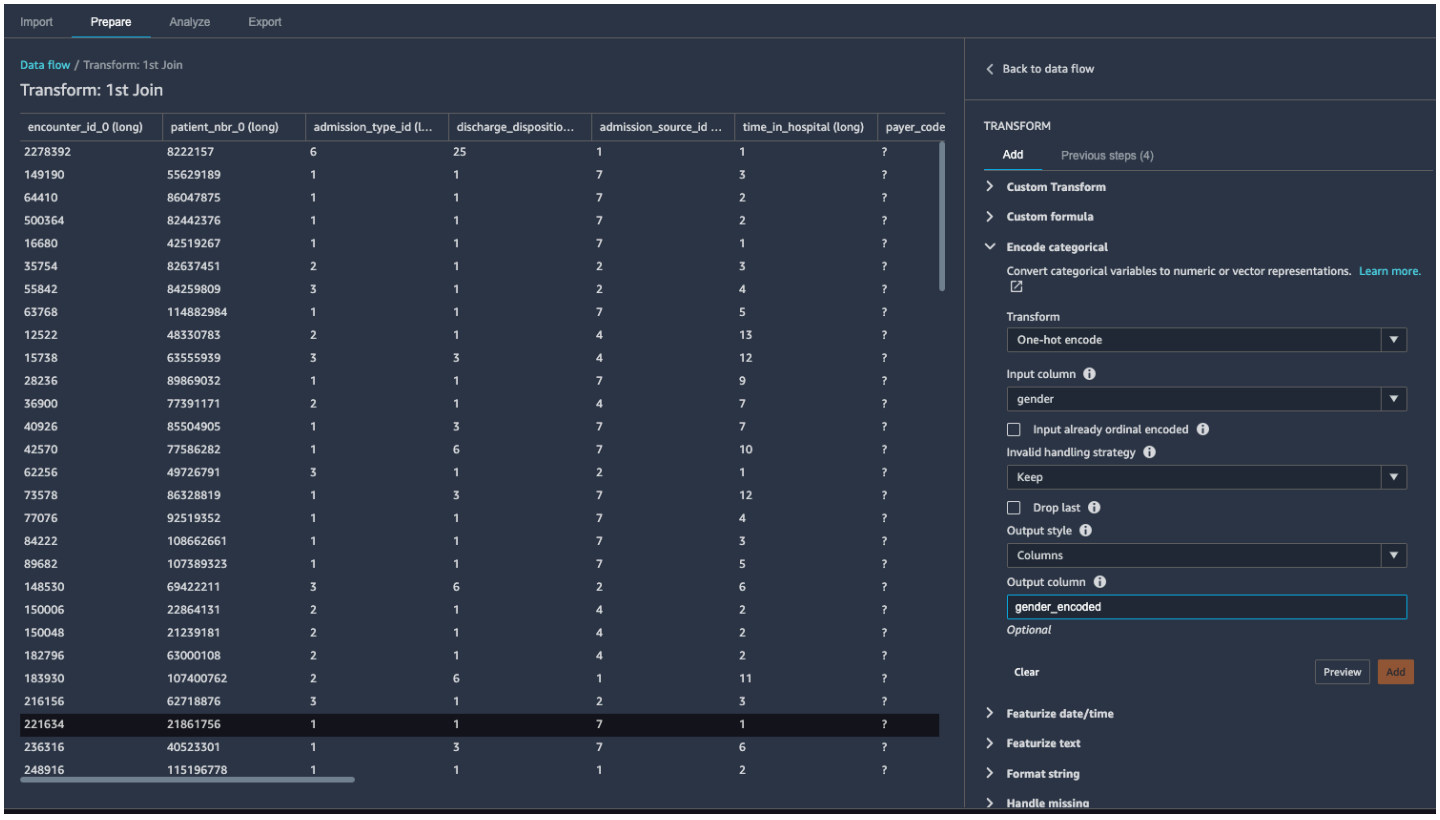


One-hot Encoding for categorical features

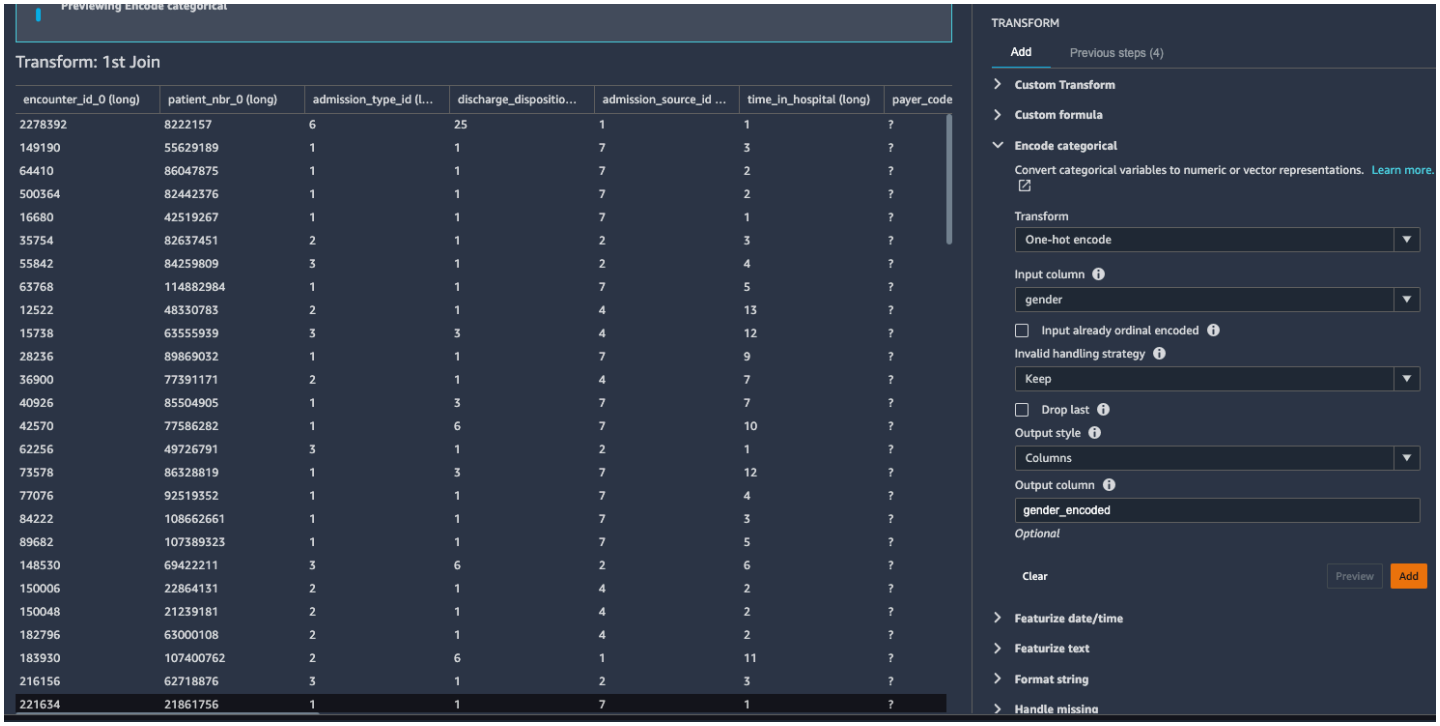
1) Click + sign next to Join flow icon and choose `Add Transform`



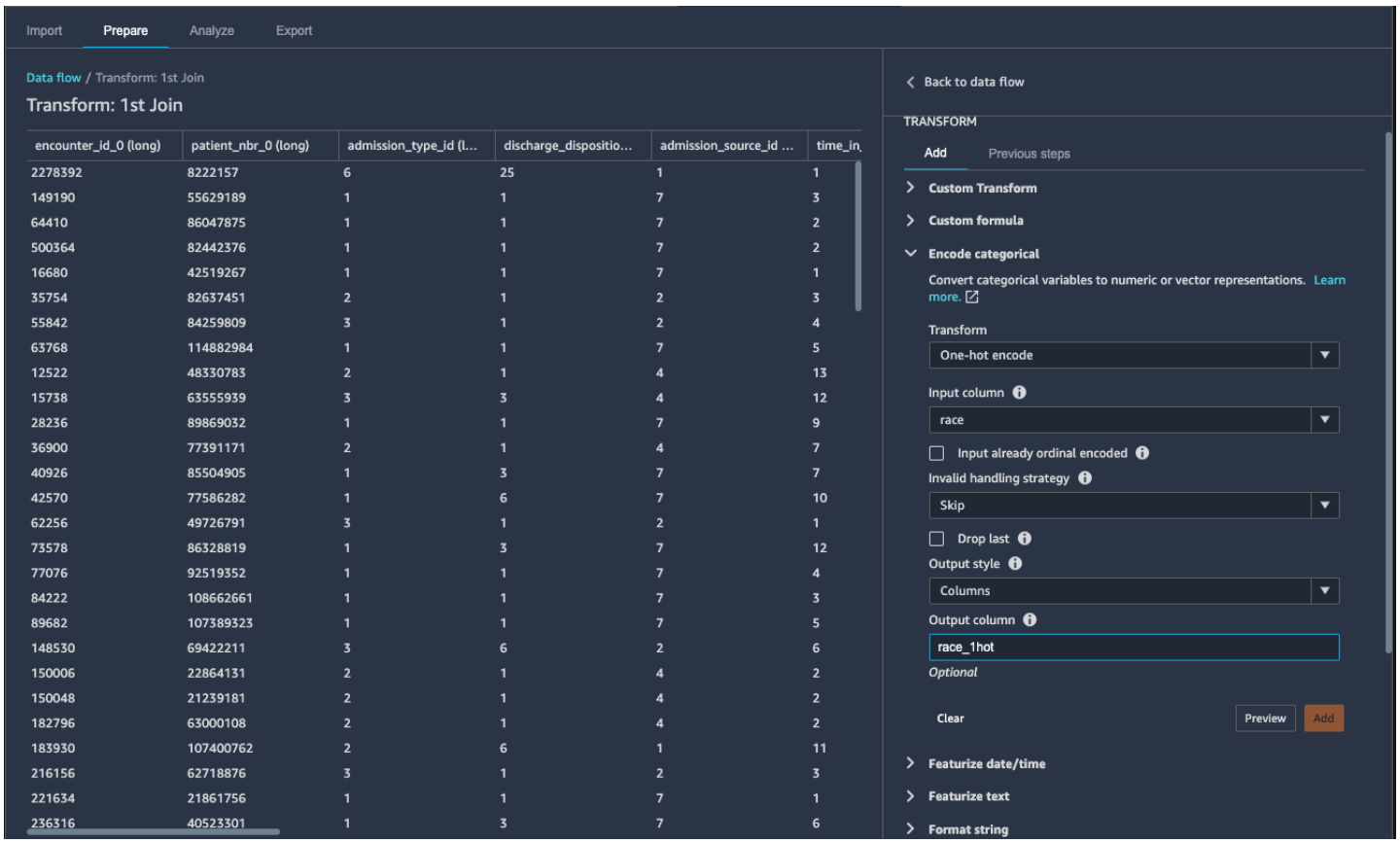
2) Pick `Encode categorical` from the list of transforms on the right panel. Select `One-hot encode` and `gender` for input column. For `Output style`, choose `Columns`. After filling the fields click `Preview`



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

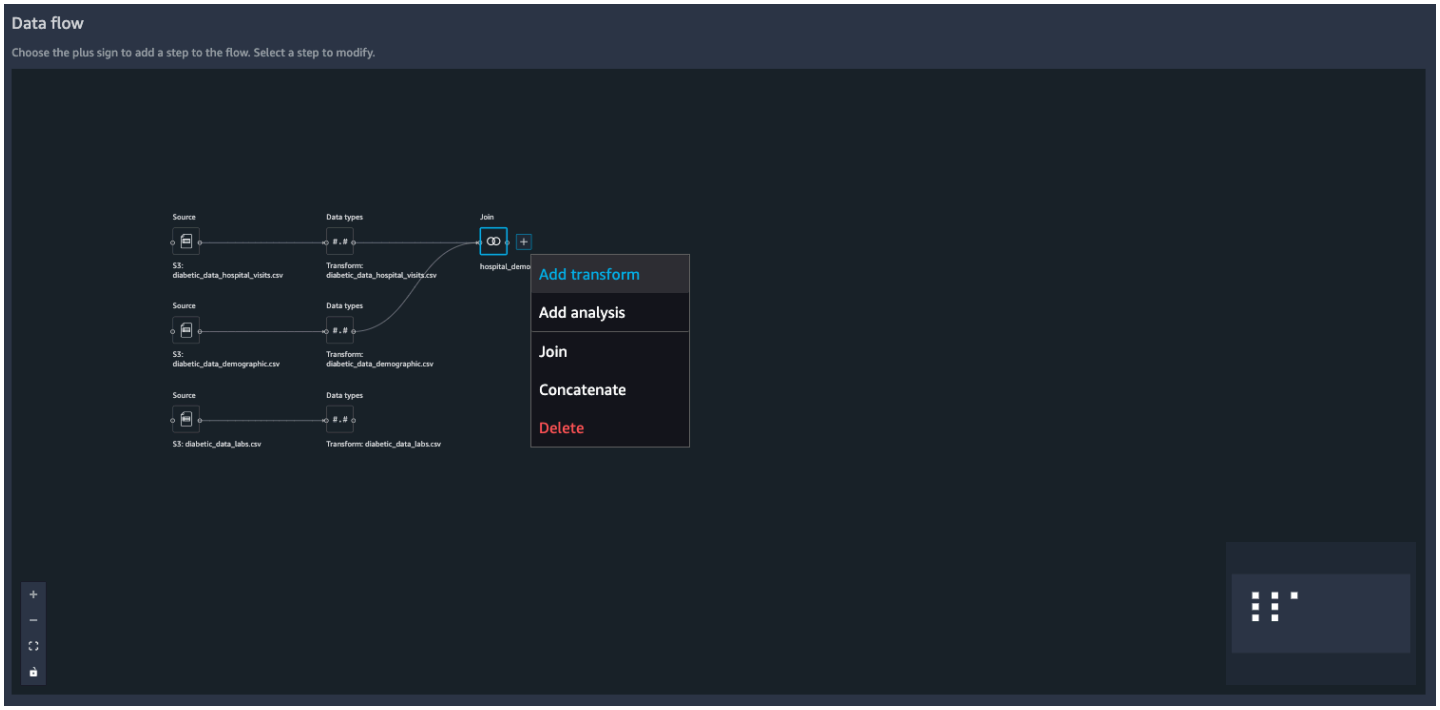


4) Repeat above steps 1 through 3 to encode `race` feature as shown below

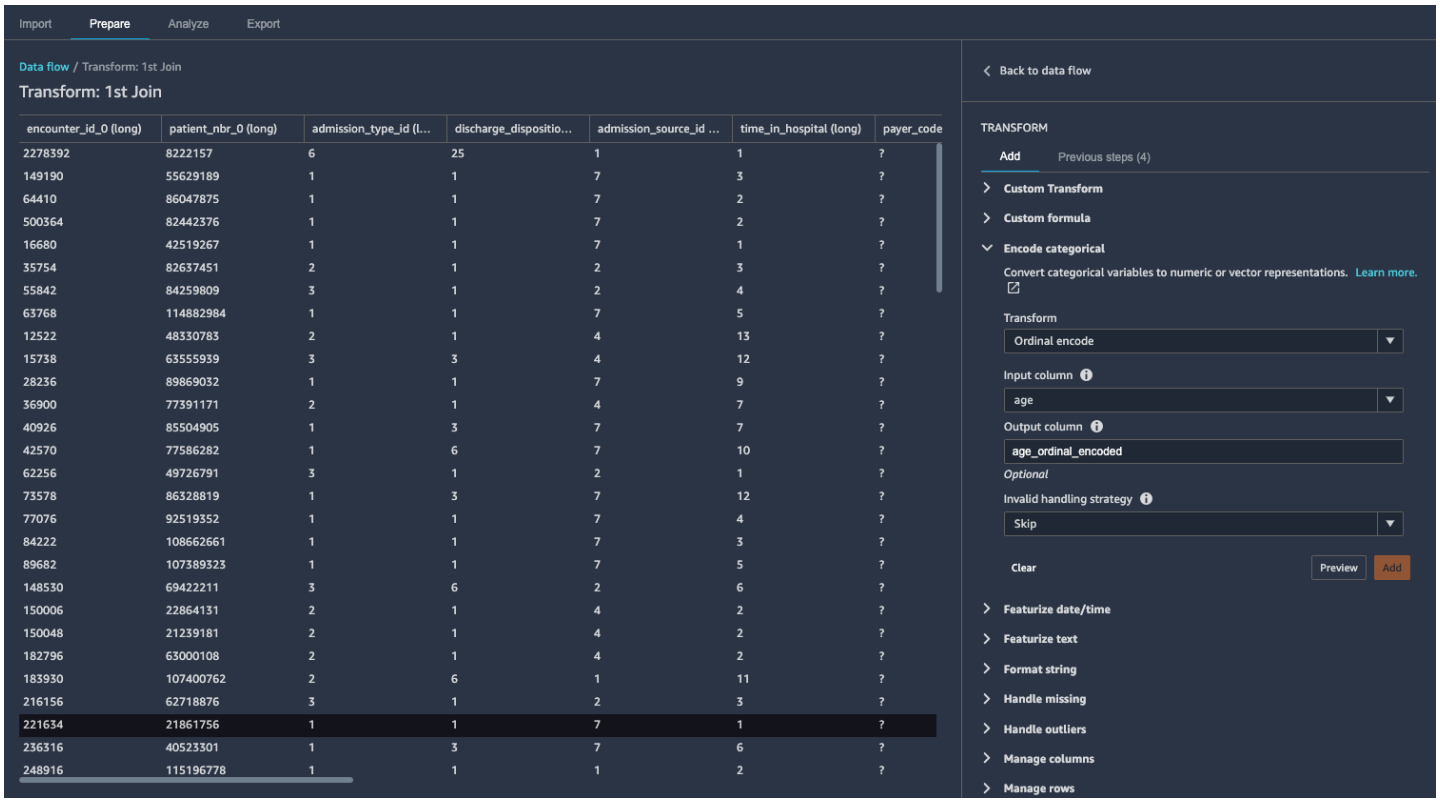


Ordinal Encoding for categorical features

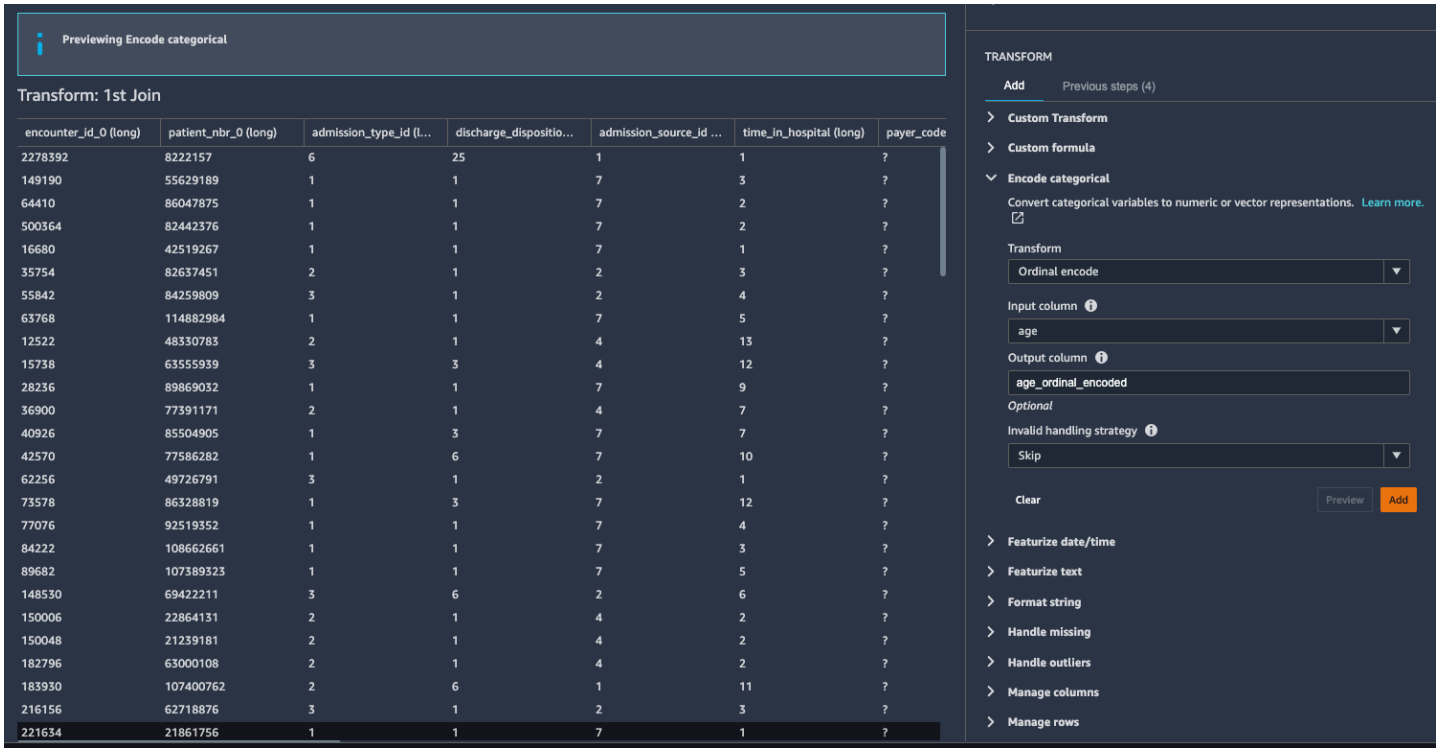
1) Click + sign next to Join flow icon and choose `Add Transform`



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`

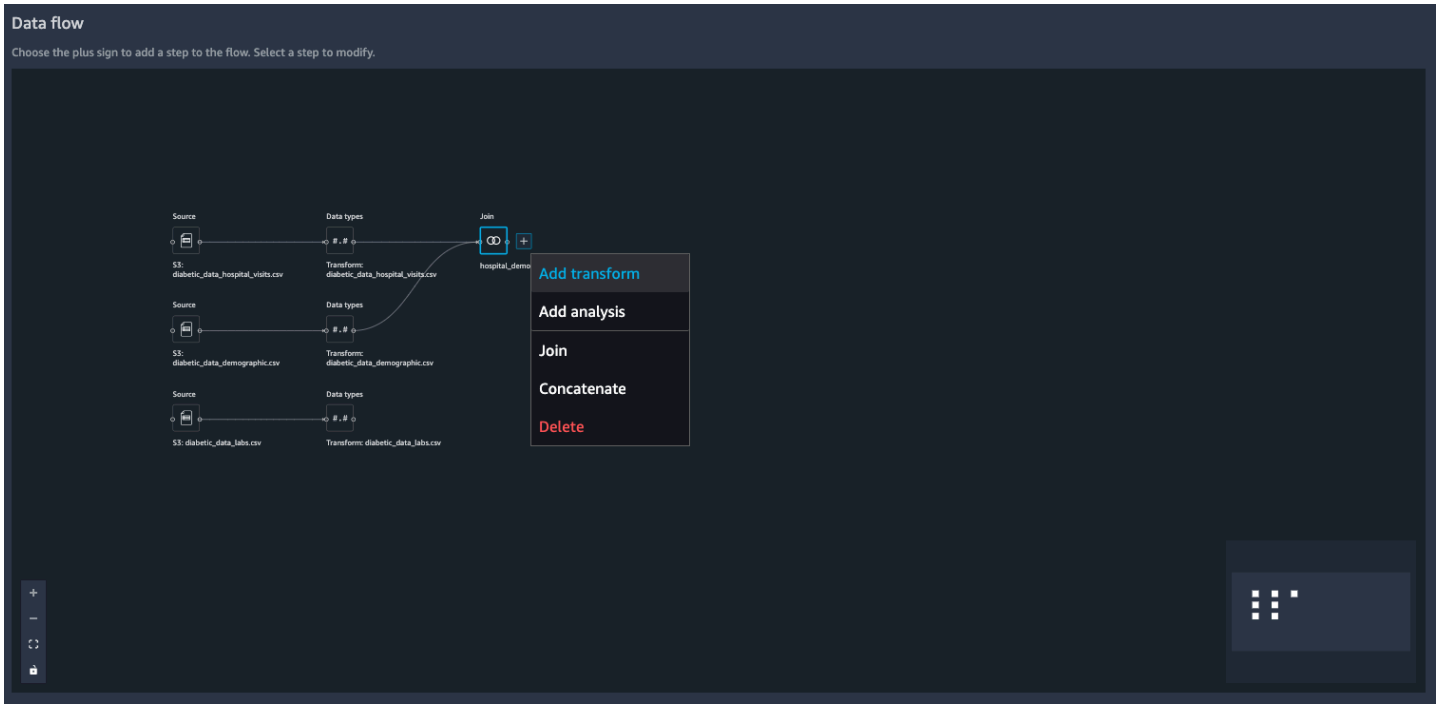


3) After review the transformation results, Click `Add` to add the change to the data flow



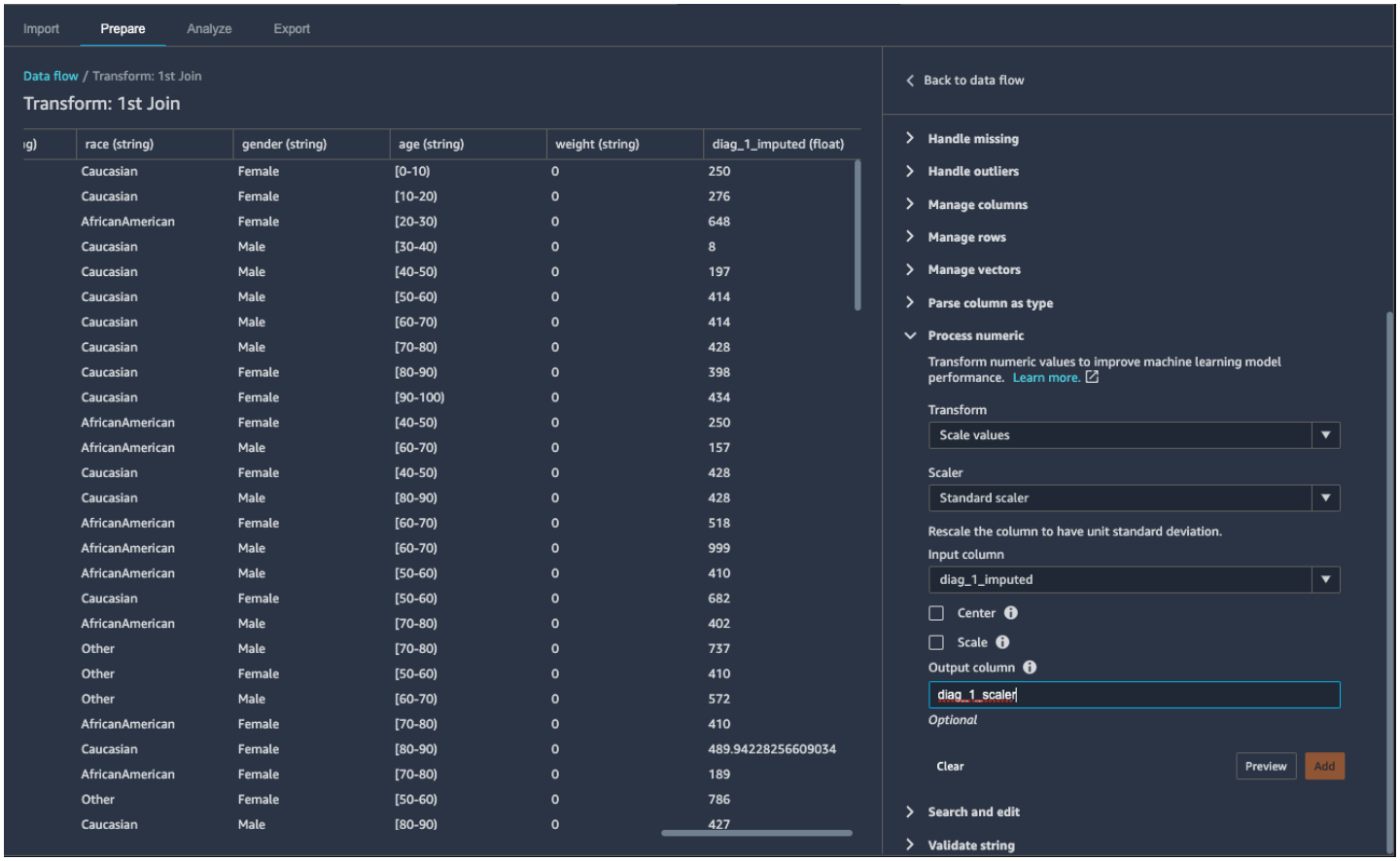
Normalization with `standard scaler` using `Process numeric` Transform

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

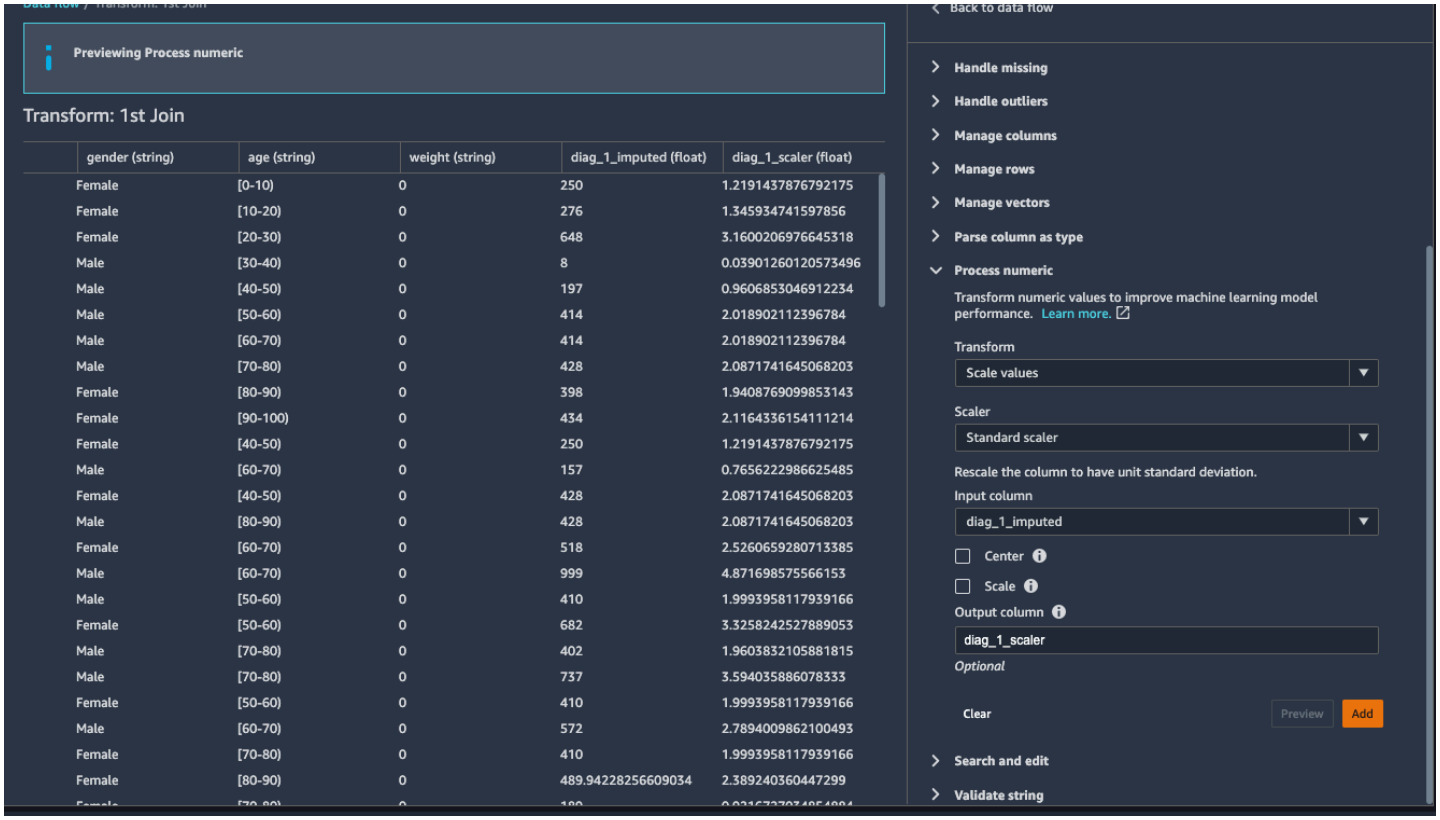


2) Pick `Process numeric` from the list of transforms on the right panel. Select `Scale values` for Transform and `Standard scaler` for Scaler. Provide `diag_1_imputed` as `Input column`. After filling the fields click

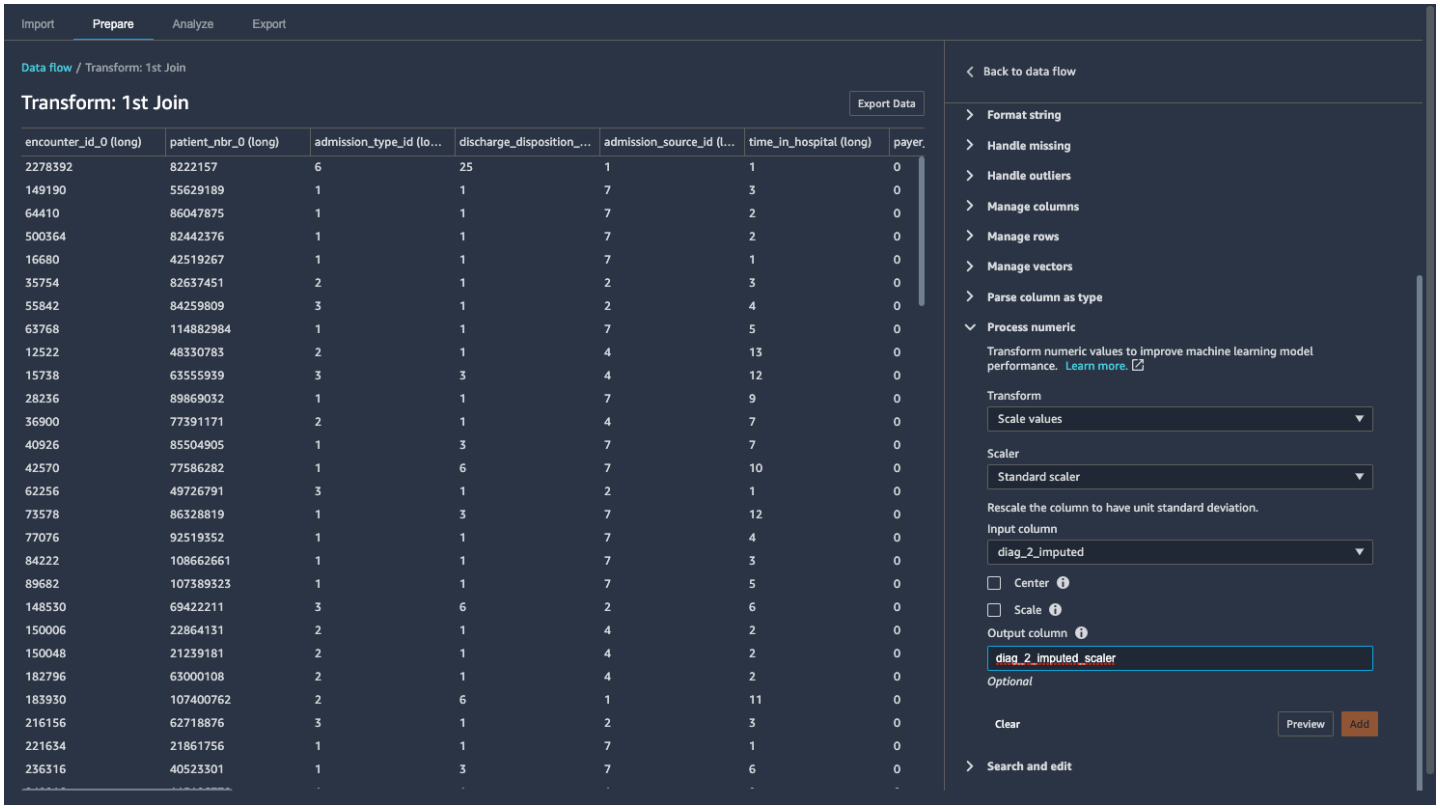
and `Standard scaler` for Scaler. Provide `diag_1_imputed` as `Input column`. After filling the fields click `Preview`

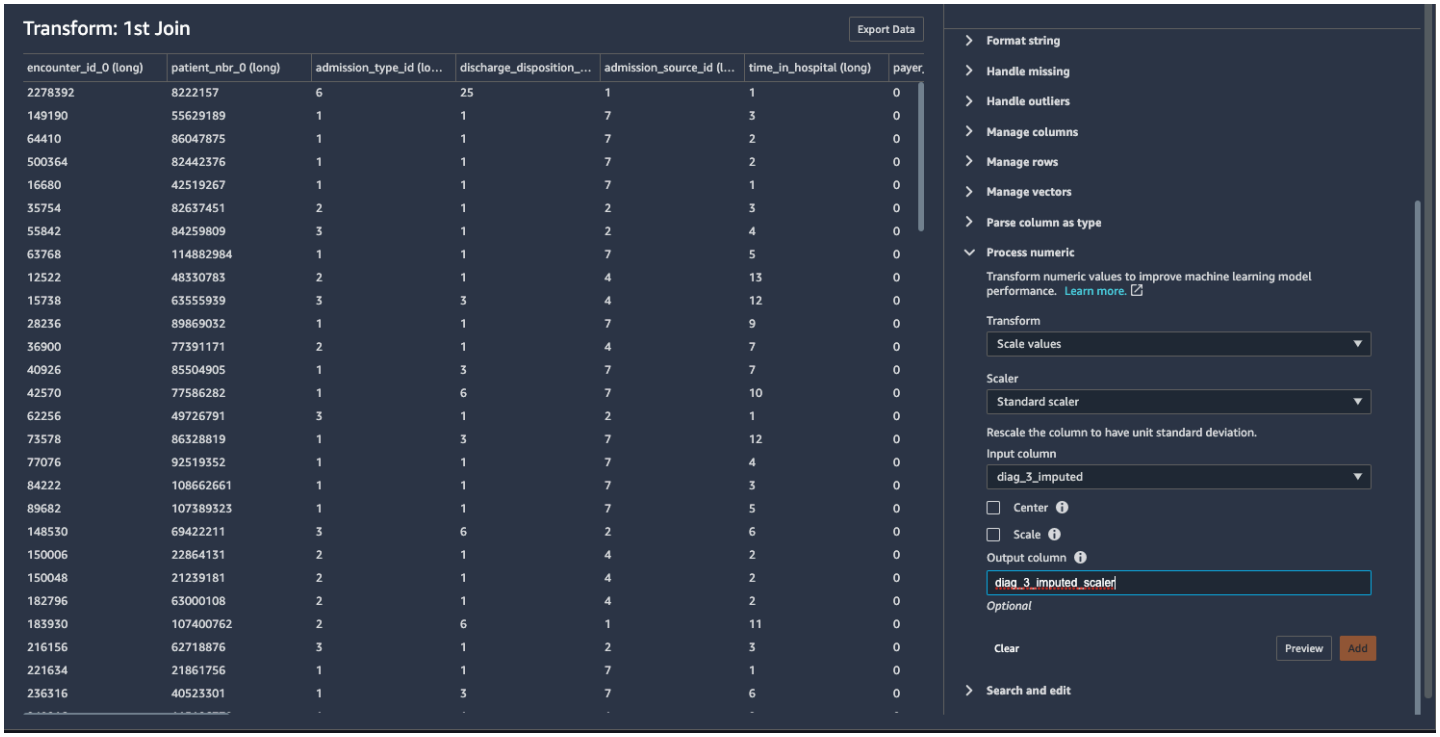


3) After review the transformation results, Click `Add` to add the change to the data flow



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

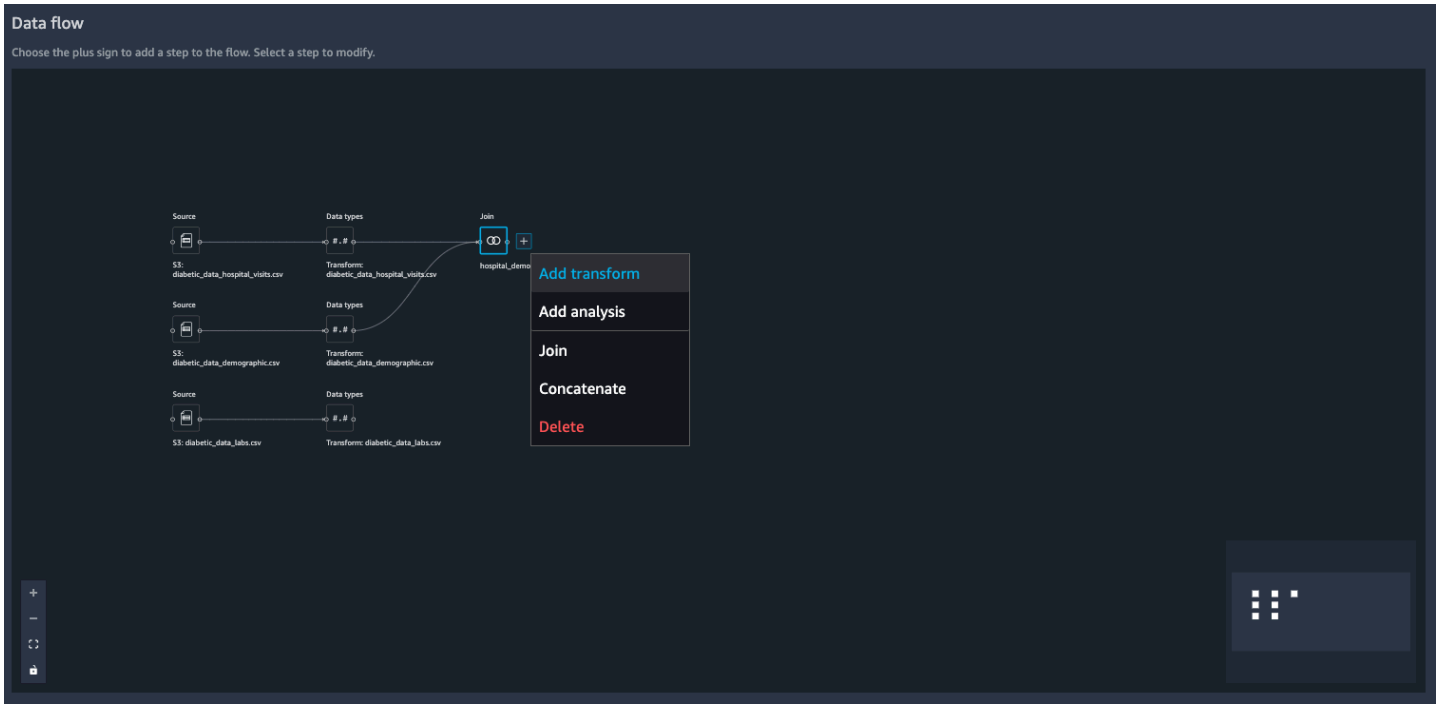




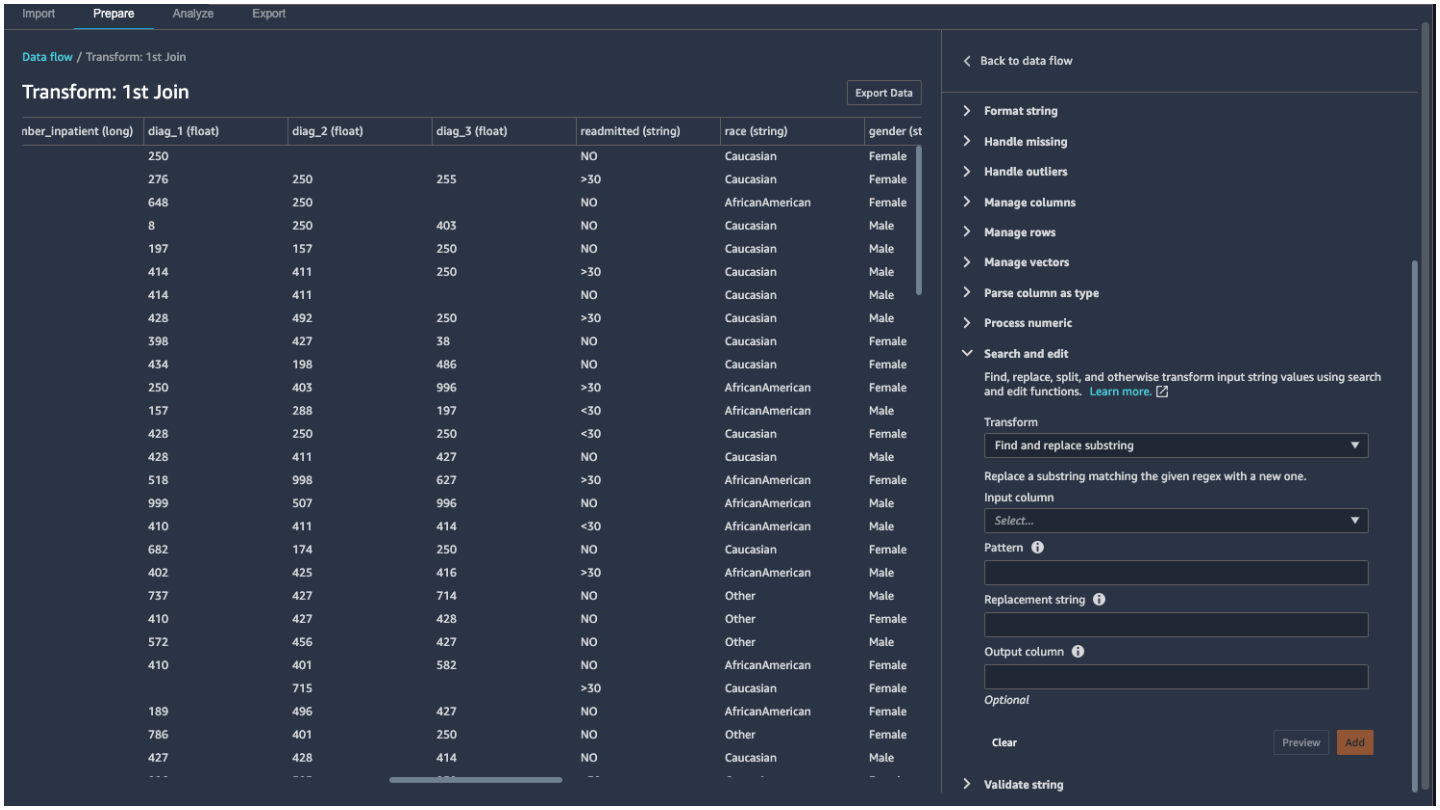
Transform the target Label

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

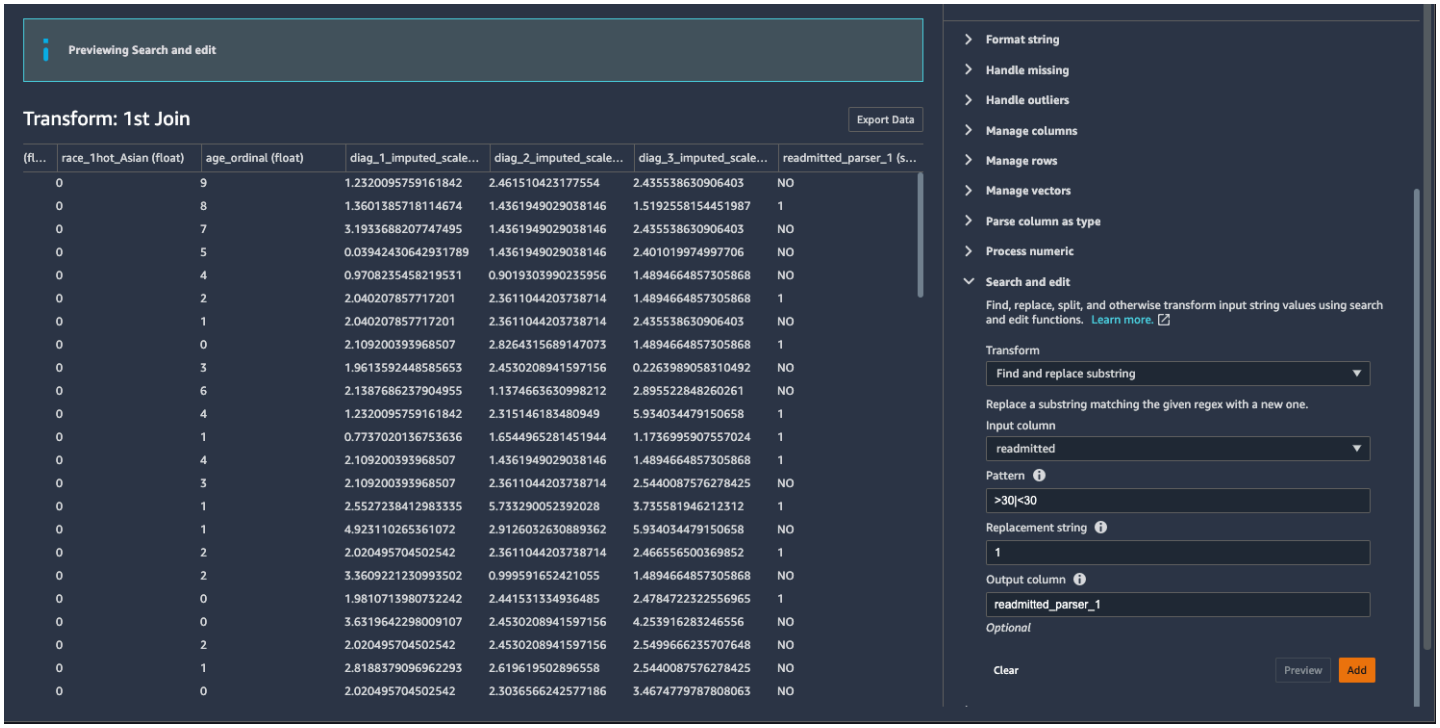
1) Click + sign next to Join flow icon and choose `Add Transform`



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

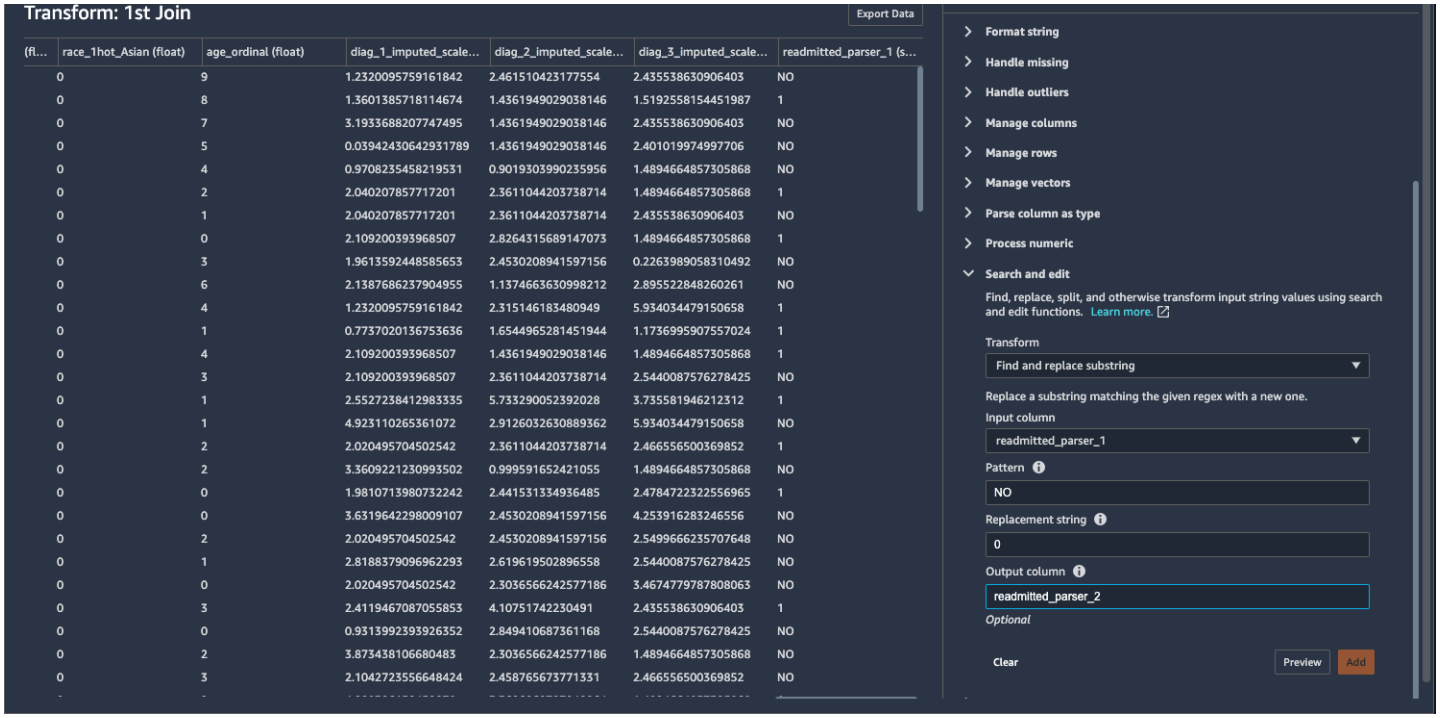


3) Select the target column `readmitted` for `Input column` and use `>30|<30` regex for Pattern. For the `Replacement String` use `1`. Also, give a name to your new `output column` as `readmitted_parser_1`. So, here we are converting all the values that have either `>30` or `<30` values to `1`. After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



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

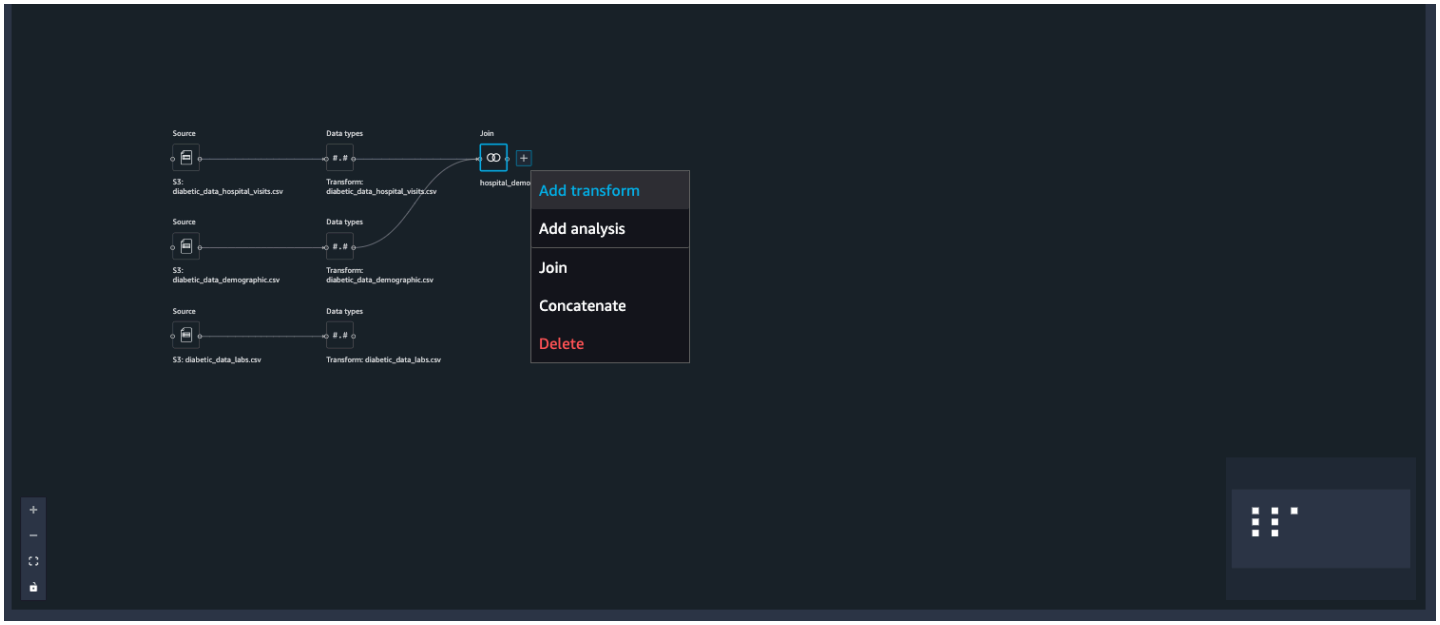


6) Once reviewed, click Add to add the transform to your data-flow. Now our target label is ready for ML.

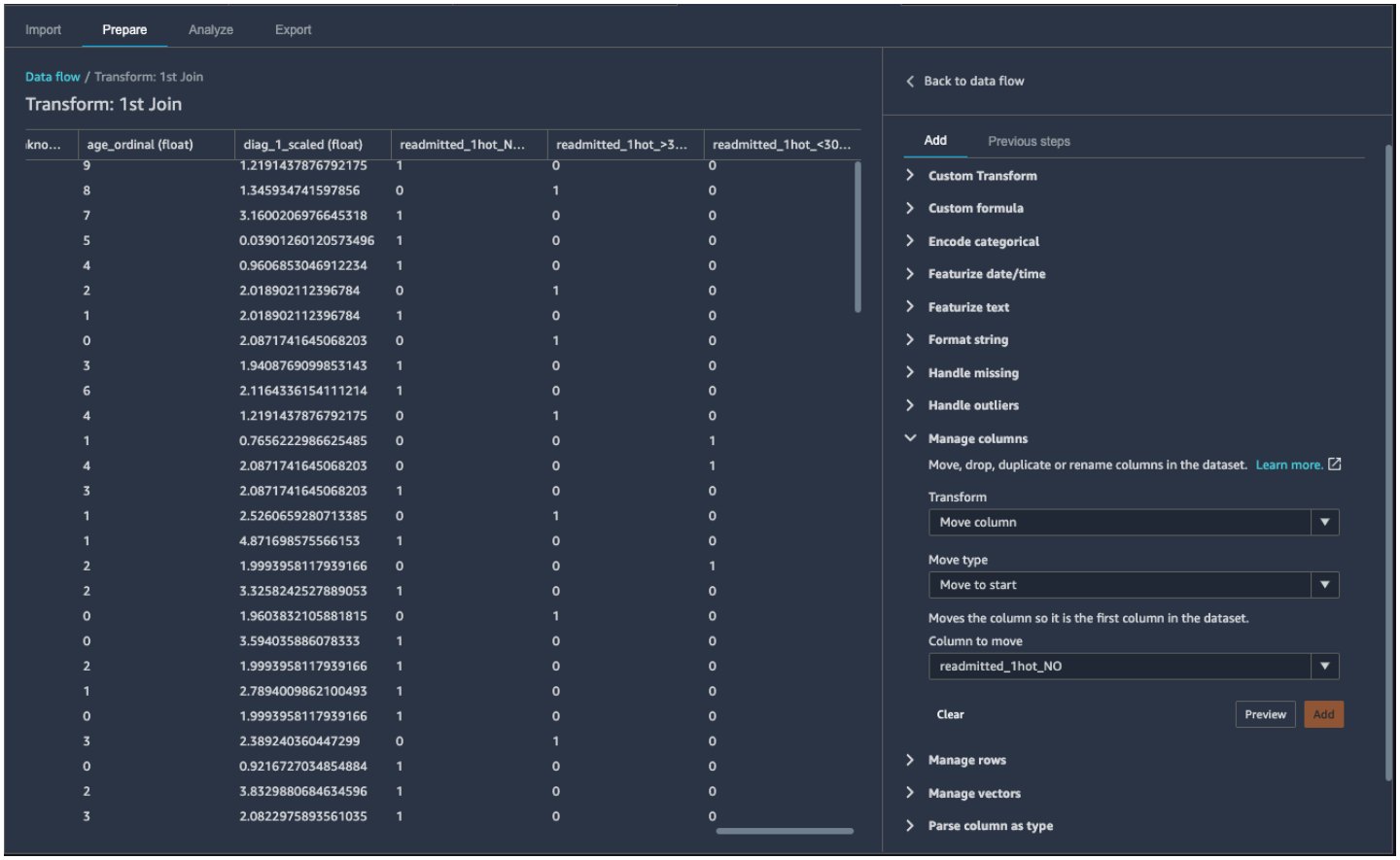
Position the target label as first column to utilize XGBoost algorithm

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 `Add Transform`

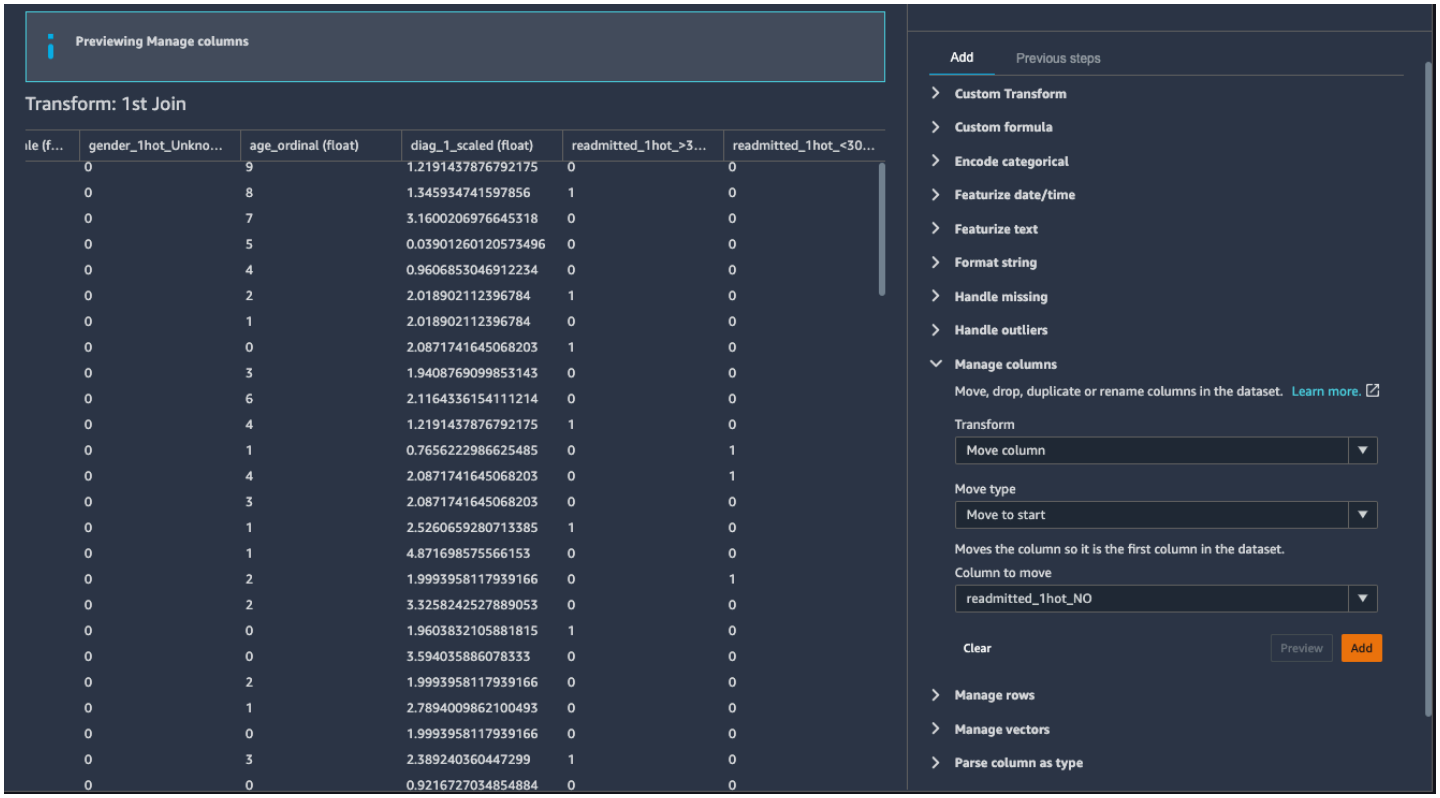


2) Pick `Manage Column` from the list of transforms on the right panel. Select `Move Column` for `Transform` and select `Move to start` for `Move type`. Provide a name to new column `readmitted_1hot_NO`. After filling the fields click `Preview`



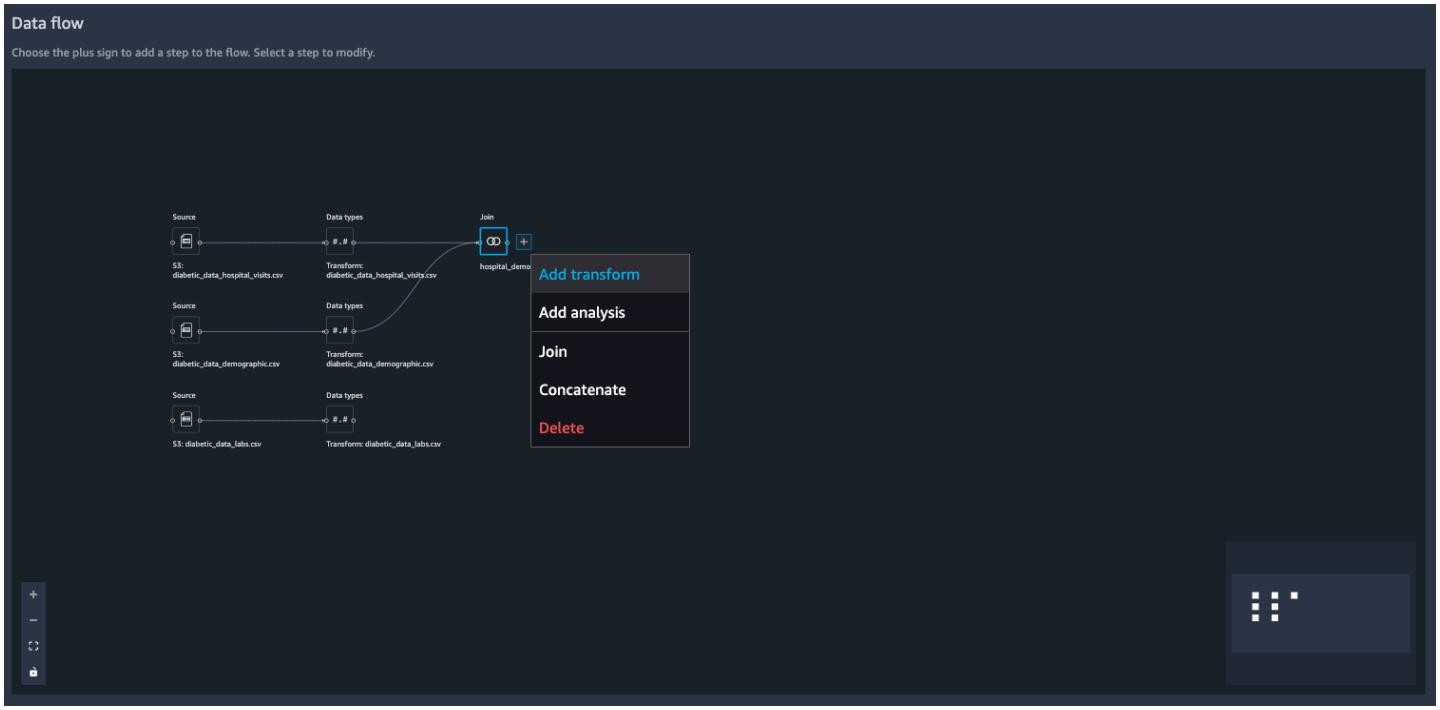
...kno...	age_ordinal (float)	diag_1_scaled (float)	readmitted_1hot_N...	readmitted_1hot_>3...	readmitted_1hot_<30...
	9	1.2191437876792175	1	0	0
	8	1.345934741597856	0	1	0
	7	3.1600206976645318	1	0	0
	5	0.03901260120573496	1	0	0
	4	0.9606853046912234	1	0	0
	2	2.018902112396784	0	1	0
	1	2.018902112396784	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

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

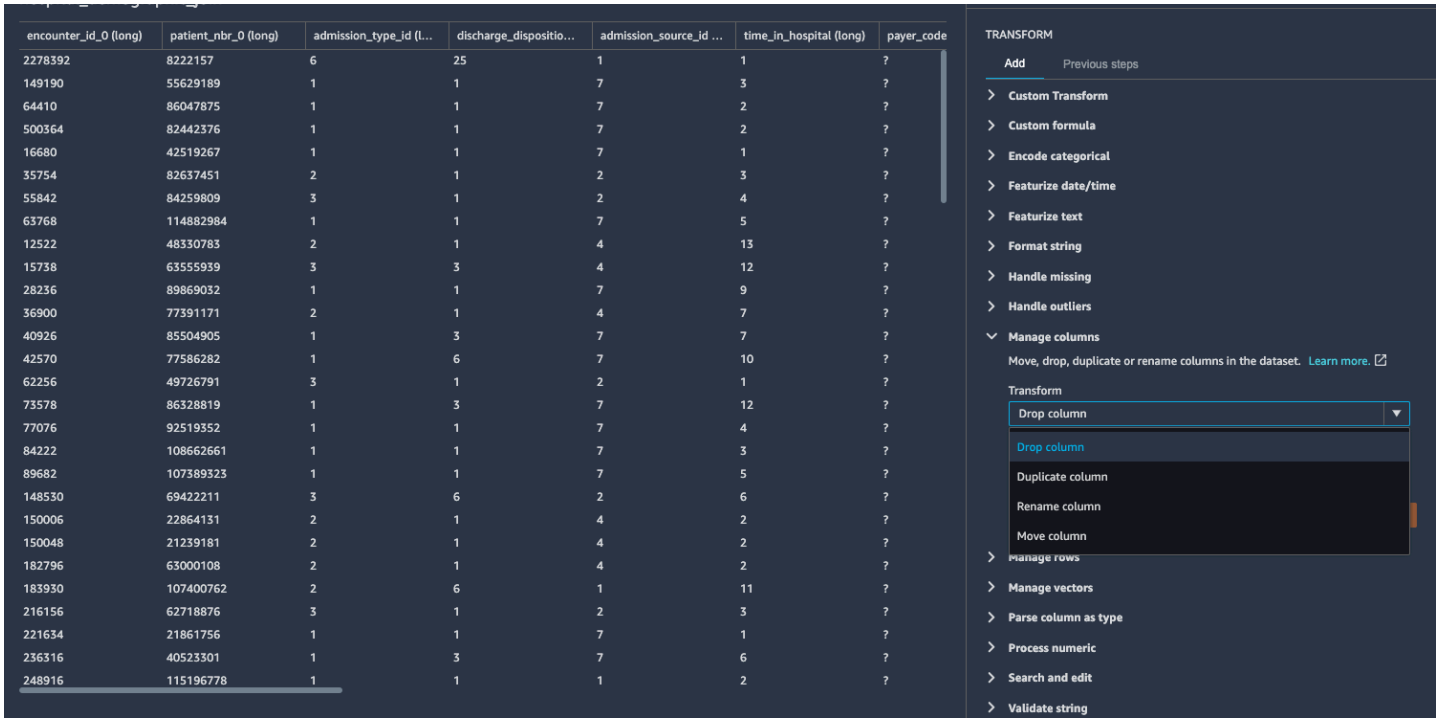


Drop redundant columns

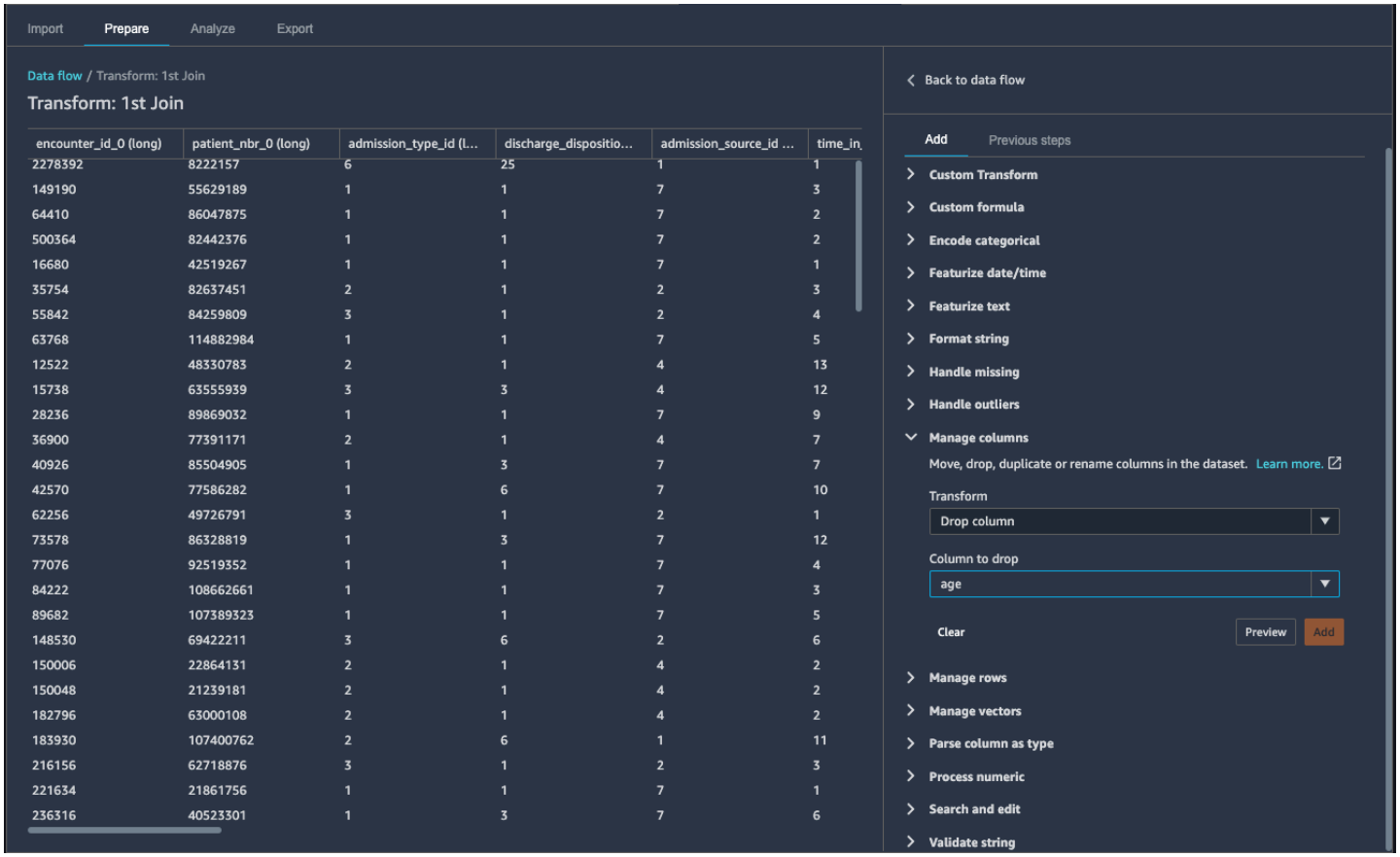
1) Click + sign next to Join flow icon and choose `Add Transform`



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

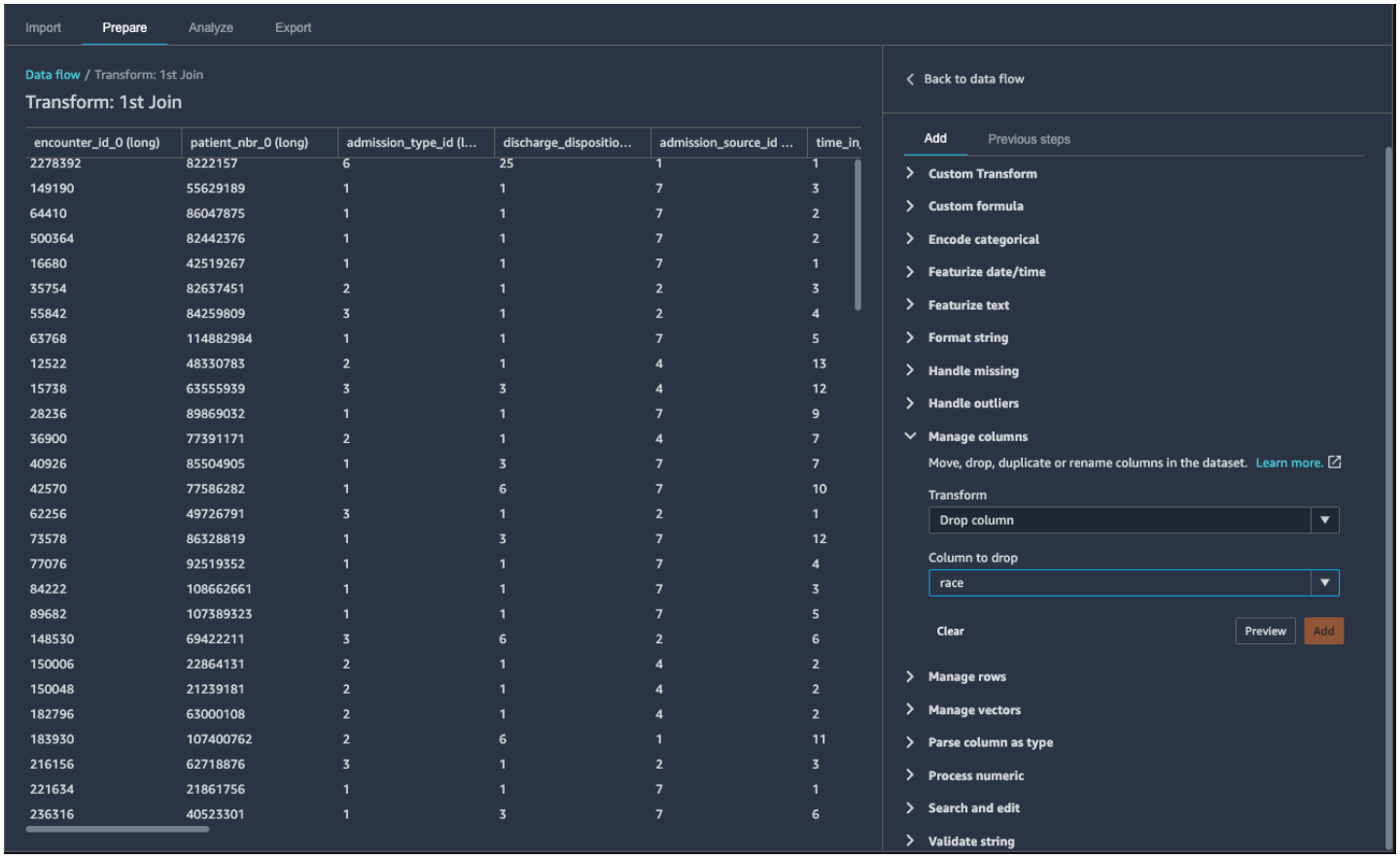


3) Choose `Drop Column` transform and select `age` column to drop

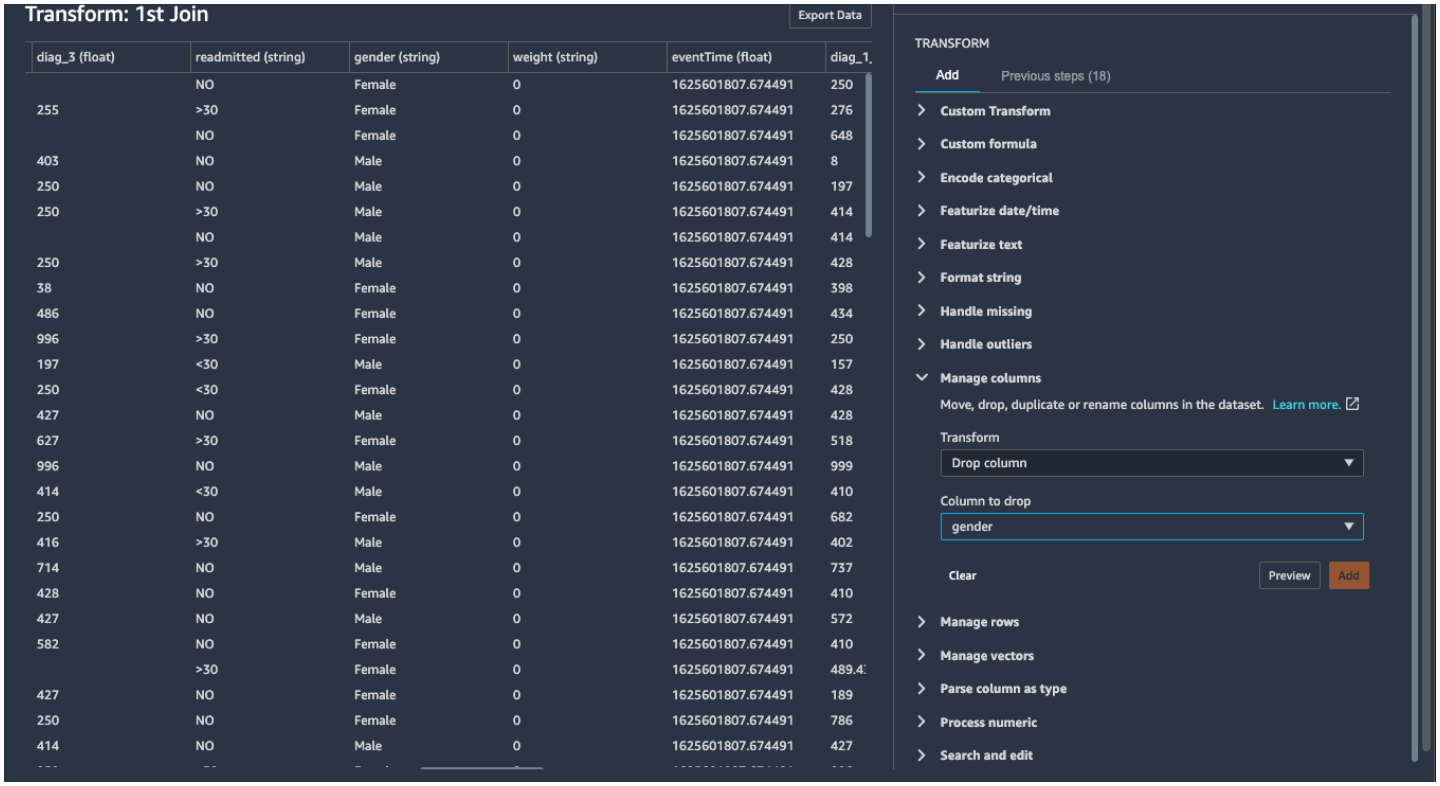


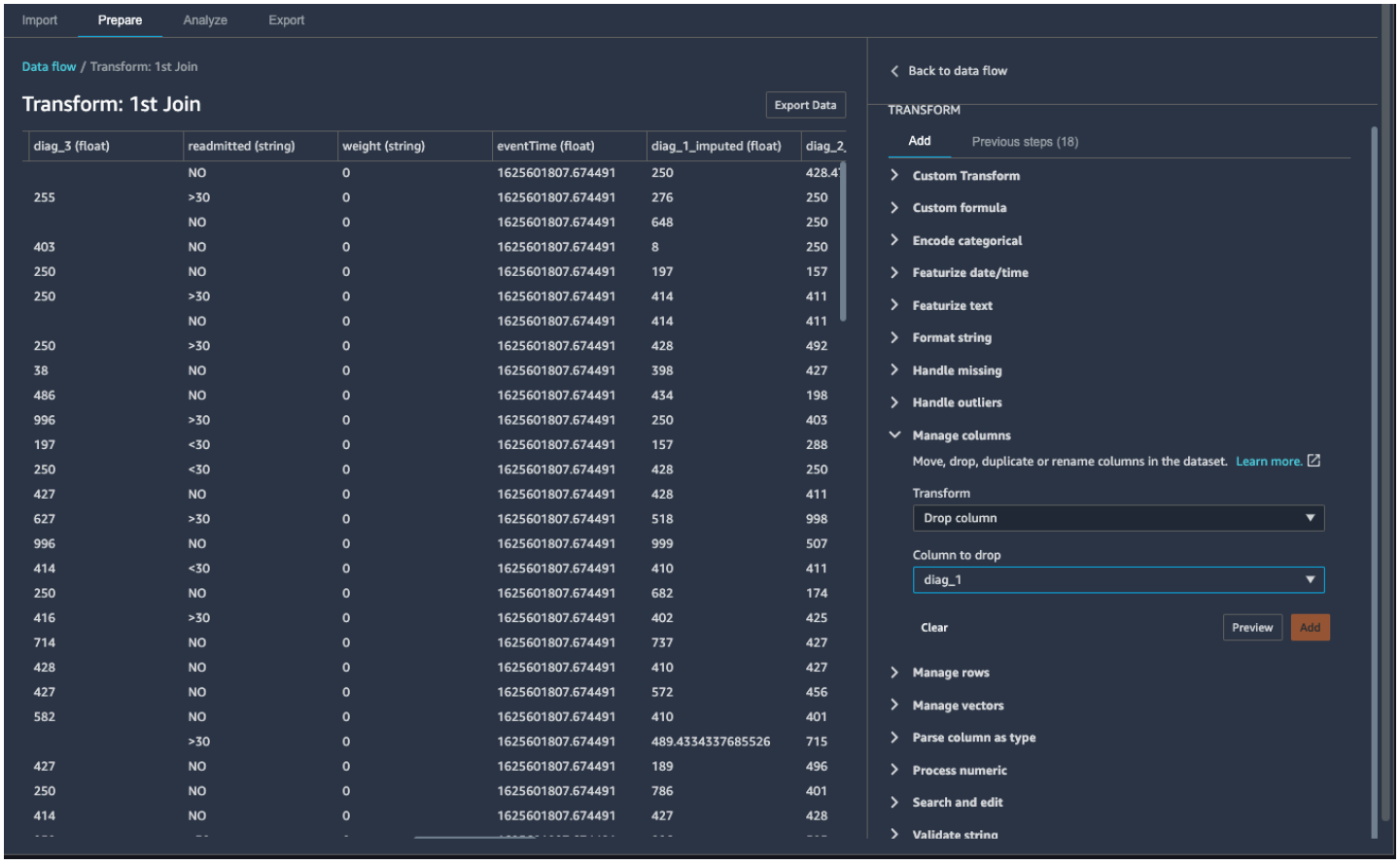
4) Click `Preview` to preview the changes to the data set. Then `Add` to add the changes to flow file.

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

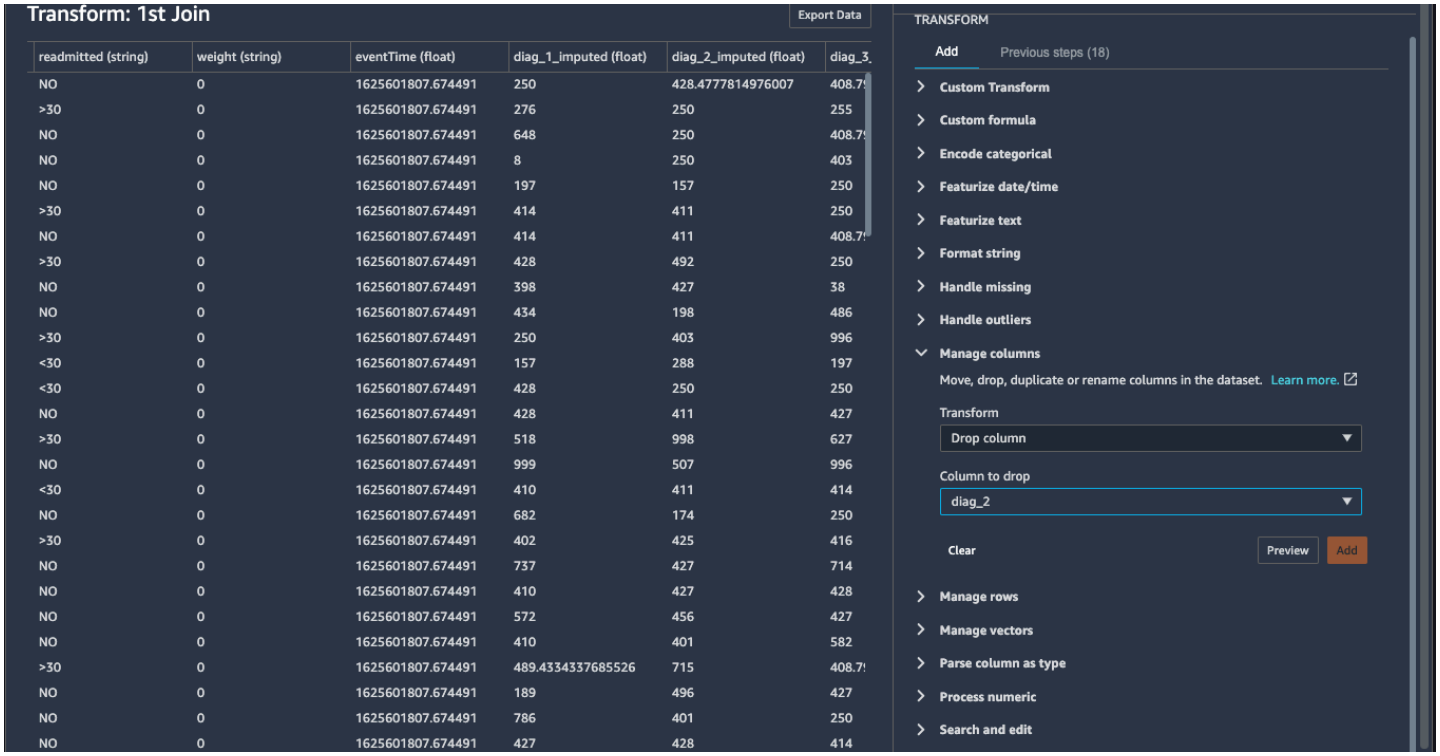

Import
Prepare
Analyze
Export
Data flow / Transform: 1st Join
Transform: 1st Join
encounter_id_0 (long)
patient_nbr_0 (long)
admission_type_id (l...
discharge_dispositio...
admission_source_id ...
time_in
2278392 8222157 6 25 1 1
149190 55629189 1 1 7 3
64410 86047875 1 1 7 2
500364 82442376 1 1 7 2
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
race
Clear
Preview
Add
Manage rows
Manage vectors
Parse column as type
Process numeric
Search and edit
Validate string

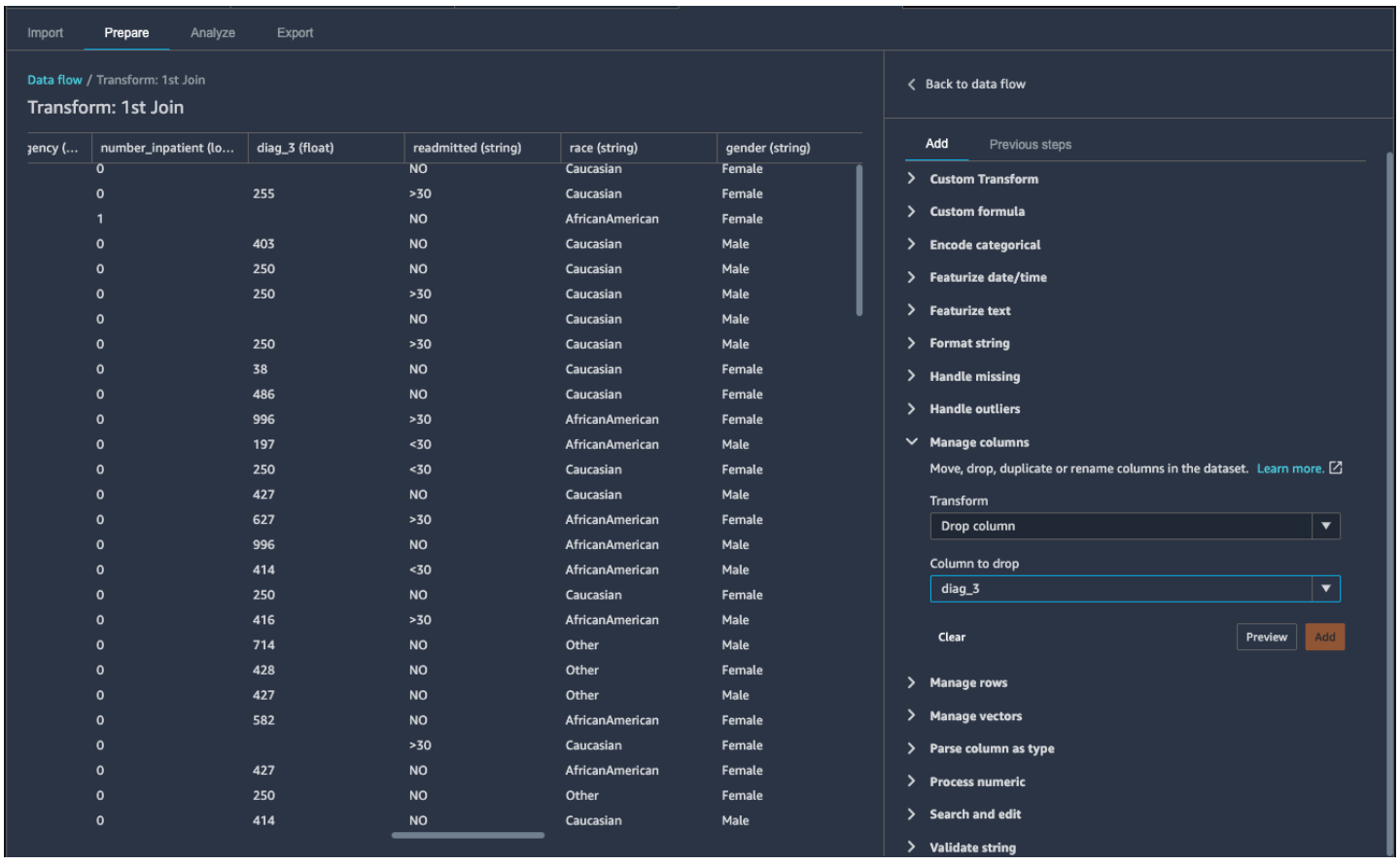

Import
Prepare
Analyze
Export
Data flow / Transform: 1st Join
Back to data flow


Transform: 1st Join
Export Data
diag_3 (float) | readmitted (string) | gender (string) | weight (string) | eventTime (float) | diag_1
NO | Female | 0 | 1625601807.674491 | 250
255 | >30 | Female | 0 | 1625601807.674491 | 276
NO | Female | 0 | 1625601807.674491 | 648
403 | NO | Male | 0 | 1625601807.674491 | 8
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
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
gender
Clear
Preview
Add
Manage rows
Manage vectors
Parse column as type
Process numeric
Search and edit

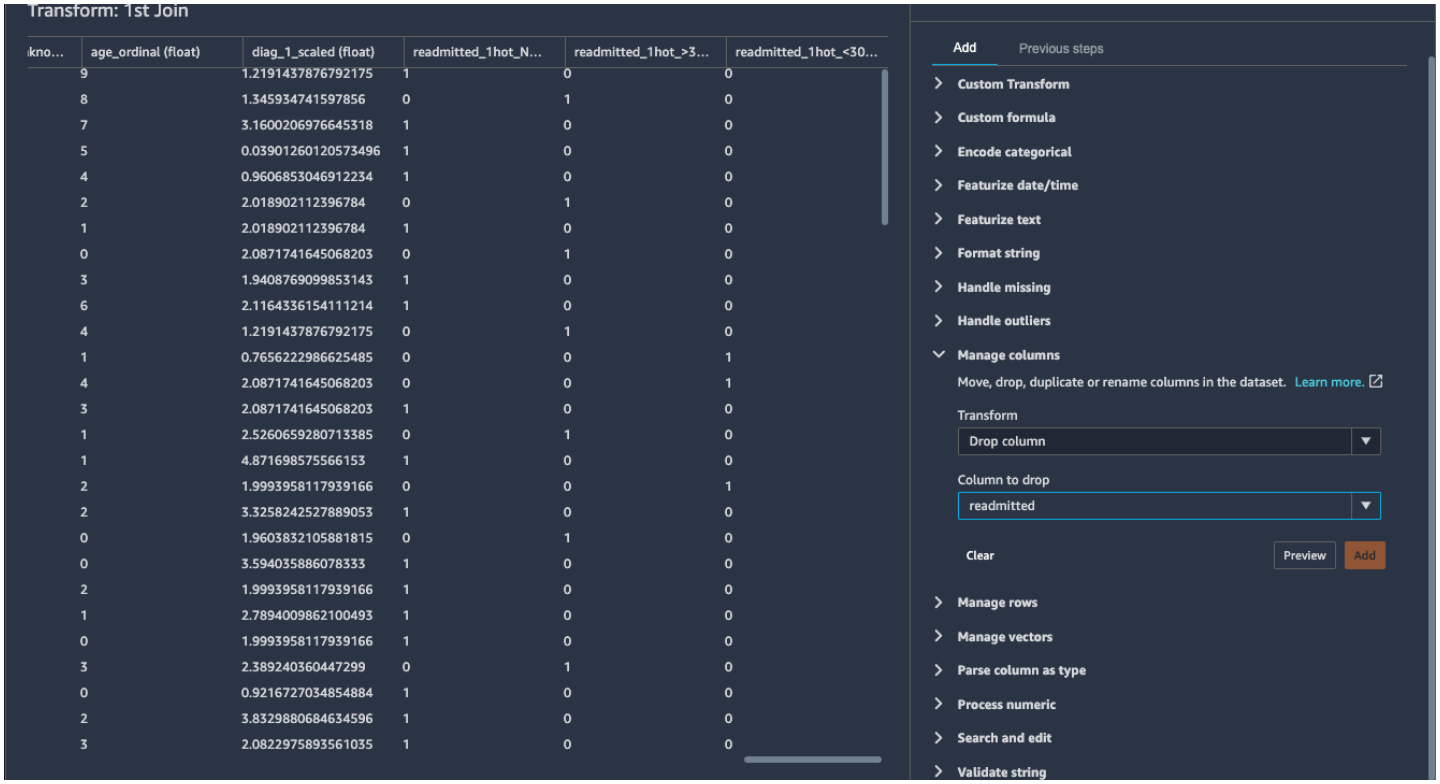

Import
Prepare
Analyze
Export
Data flow / Transform: 1st Join
Transform: 1st Join
Export Data
diag_3 (float) | readmitted (string) | weight (string) | eventTime (float) | diag_1_imputed (float) | diag_2
NO | 0 | 1625601807.674491 | 250 | 428.4
255 | >30 | 0 | 1625601807.674491 | 276 | 250
NO | 0 | 1625601807.674491 | 648 | 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
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
diag_1
Clear
Preview
Add
Manage rows
Manage vectors
Parse column as type
Process numeric
Search and edit
Validate string

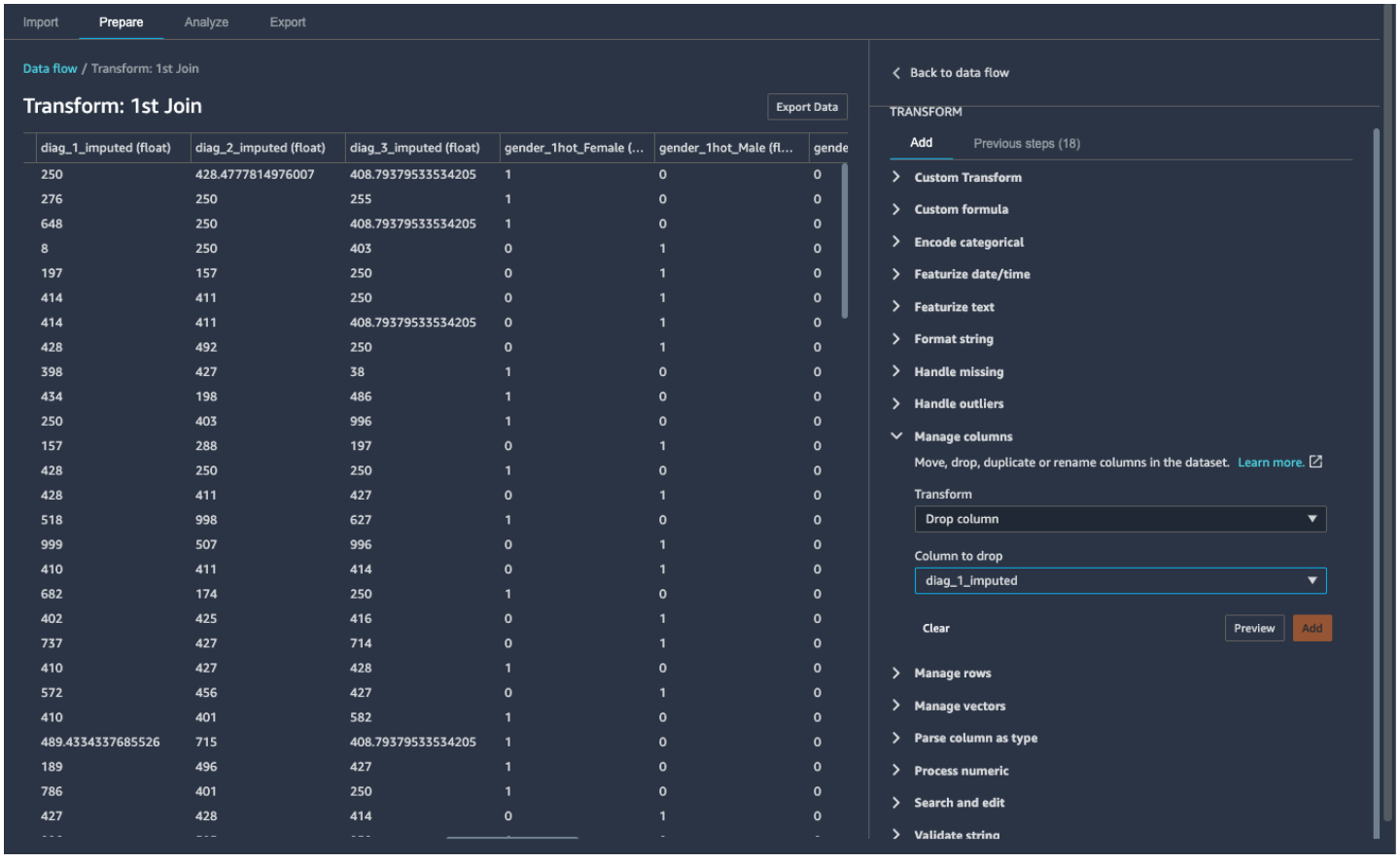

Import
Prepare
Analyze
Export
Data flow / Transform: 1st Join
Back to data flow


Transform: 1st Join
Export Data
readmitted (string) | weight (string) | eventTime (float) | diag_1_imputed (float) | diag_2_imputed (float) | diag_3
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.
Transform
Drop column
Column to drop
diag_2
Clear
Preview
Add
Manage rows
Manage vectors
Parse column as type
Process numeric
Search and edit

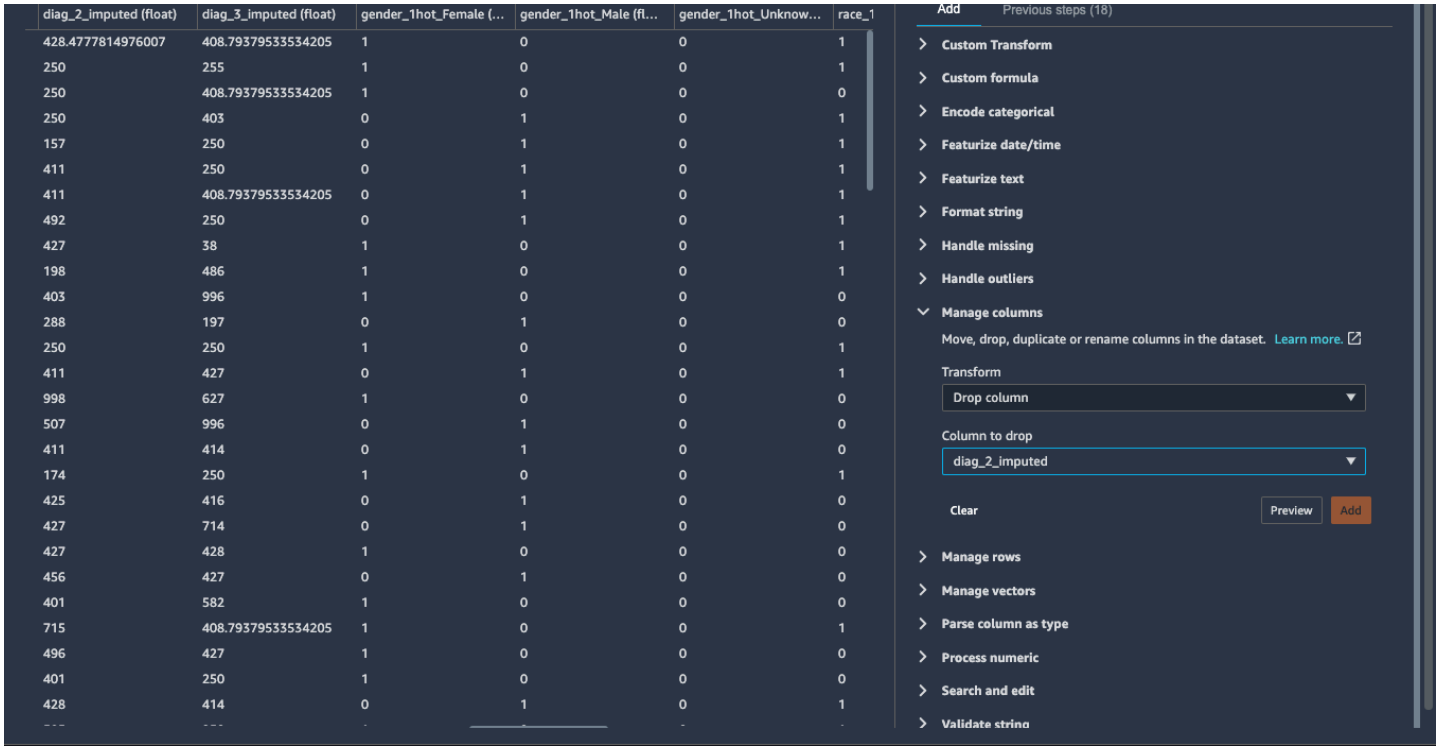

Import
Prepare
Analyze
Export
Data flow / Transform: 1st Join
Transform: 1st Join
gency (... | number_inpatient (lo... | diag_3 (float) | readmitted (string) | race (string) | gender (string)
0 | | NO | Caucasian | Female
0 | 255 | >30 | Caucasian | Female
1 | | NO | AfricanAmerican | Female
0 | 403 | 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
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
diag_3
Clear
Preview
Add
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

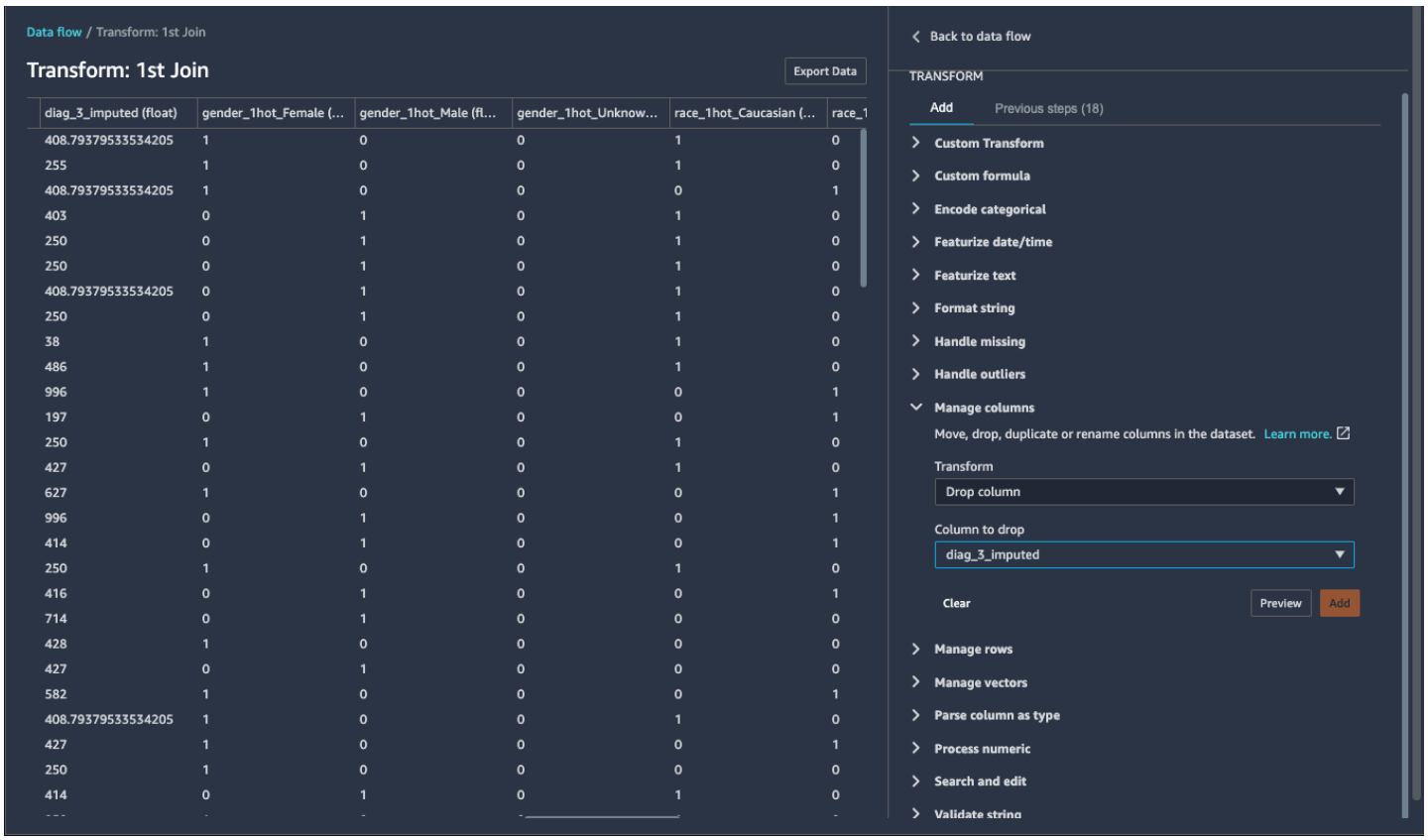

Transform: 1st Join
kno...
age_ordinal (float)
diag_1_scaled (float)
readmitted_1hot_N...
readmitted_1hot_>3...
readmitted_1hot_<30...
9 1.2191437876792175 1 0 0
8 1.345934741597856 0 1 0
7 3.1600206976645318 1 0 0
5 0.03901260120573496 1 0 0
4 0.9606853046912234 1 0 0
2 2.018902112396784 0 1 0
1 2.018902112396784 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
readmitted
Clear
Preview
Add
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
Export Data
diag_1_imputed (float)
diag_2_imputed (float)
diag_3_imputed (float)
gender_1hot_Female (...
gender_1hot_Male (fl...
gende
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
diag_1_imputed
Clear
Preview
Add
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_Unknow...	race_1
428.4777814976007	408.79379533534205	1	0	0	1
250	255	1	0	0	1
250	408.79379533534205	1	0	0	0
250	403	0	1	0	1
157	250	0	1	0	1
411	250	0	1	0	1
411	408.79379533534205	0	1	0	1
492	250	0	1	0	1
427	38	1	0	0	1
198	486	1	0	0	1
403	996	1	0	0	0
288	197	0	1	0	0
250	250	1	0	0	1
411	427	0	1	0	1
998	627	1	0	0	0
507	996	0	1	0	0
411	414	0	1	0	0
174	250	1	0	0	1
425	416	0	1	0	0
427	714	0	1	0	0
427	428	1	0	0	0
456	427	0	1	0	0
401	582	1	0	0	0
715	408.79379533534205	1	0	0	1
496	427	1	0	0	0
401	250	1	0	0	0
428	414	0	1	0	1

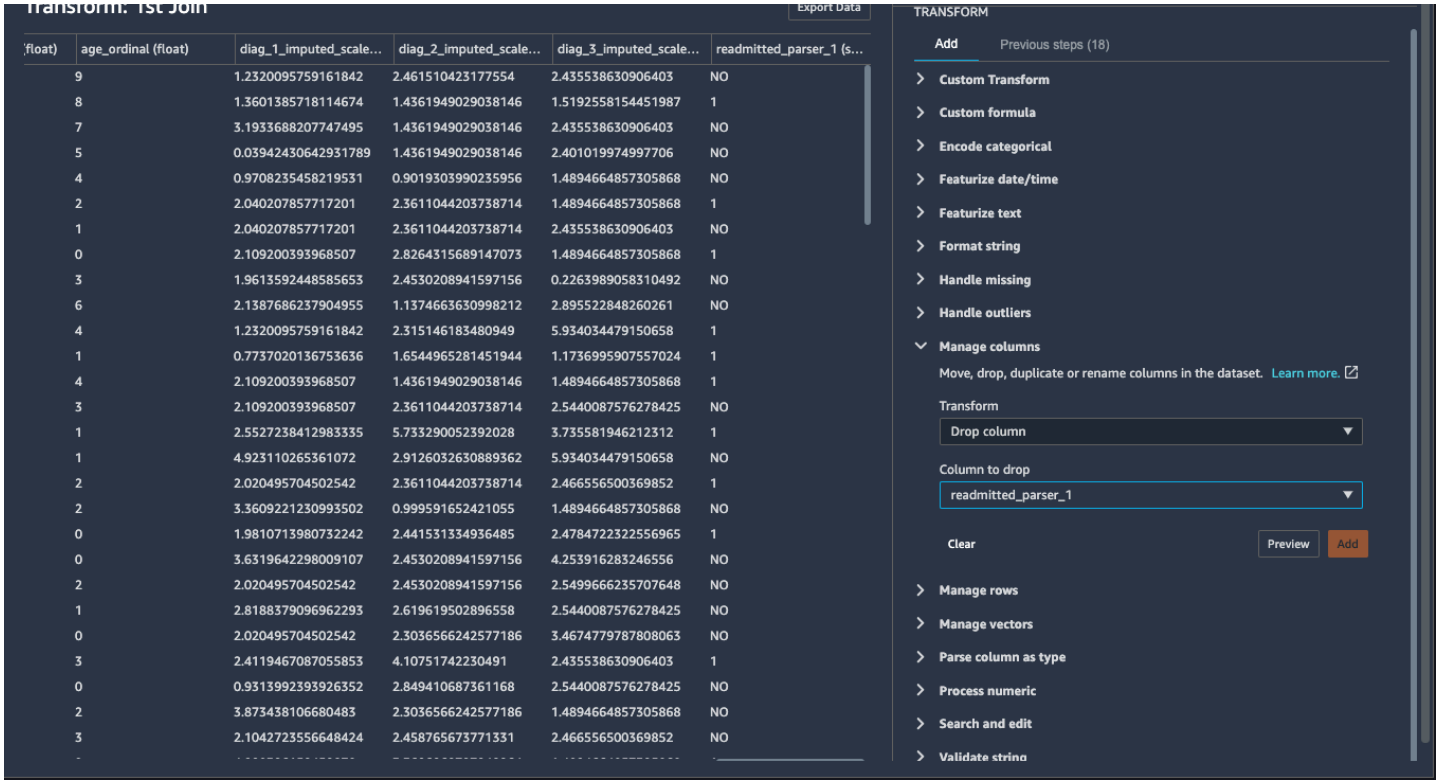


Data flow / Transform: 1st Join

Transform: 1st Join

Export Data

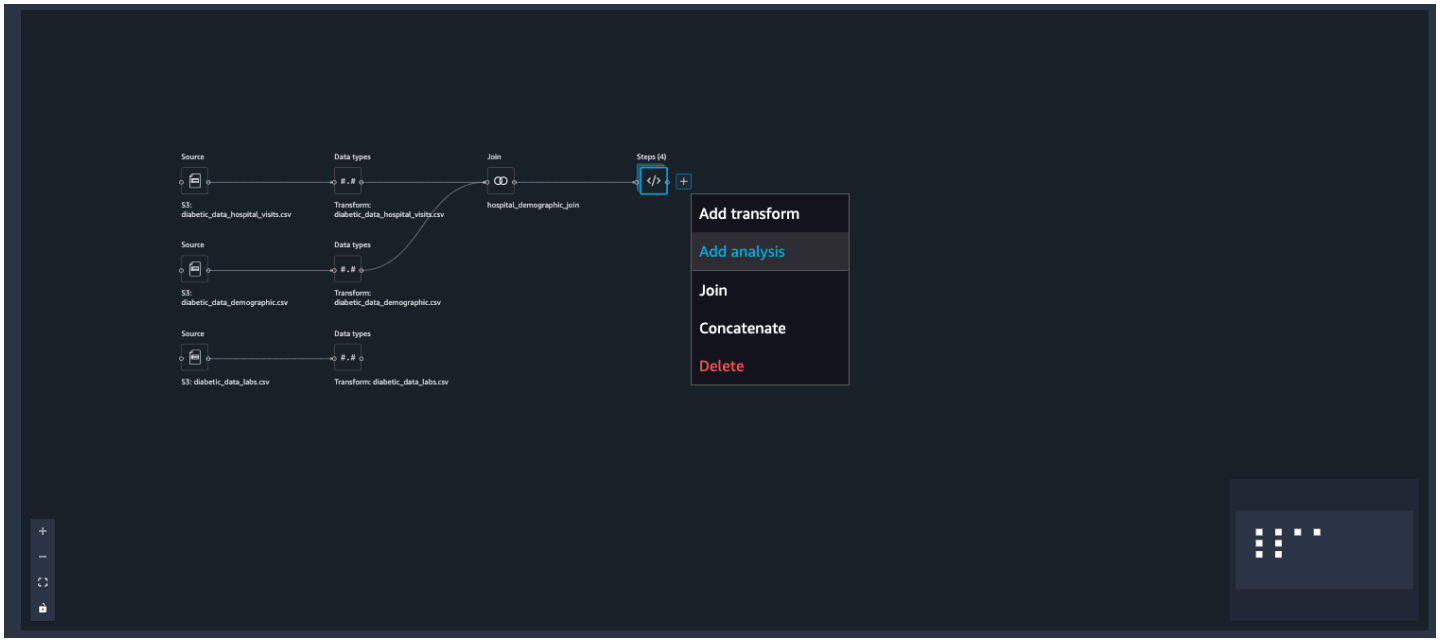
diag_3_imputed (float)	gender_1hot_Female (...	gender_1hot_Male (fl...	gender_1hot_Unknow...	race_1hot_Caucasian (...	race_1
408.79379533534205	1	0	0	1	0
255	1	0	0	1	0
408.79379533534205	1	0	0	0	1
403	0	1	0	1	0
250	0	1	0	1	0
250	0	1	0	1	0
408.79379533534205	0	1	0	1	0
250	0	1	0	1	0
38	1	0	0	1	0
486	1	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



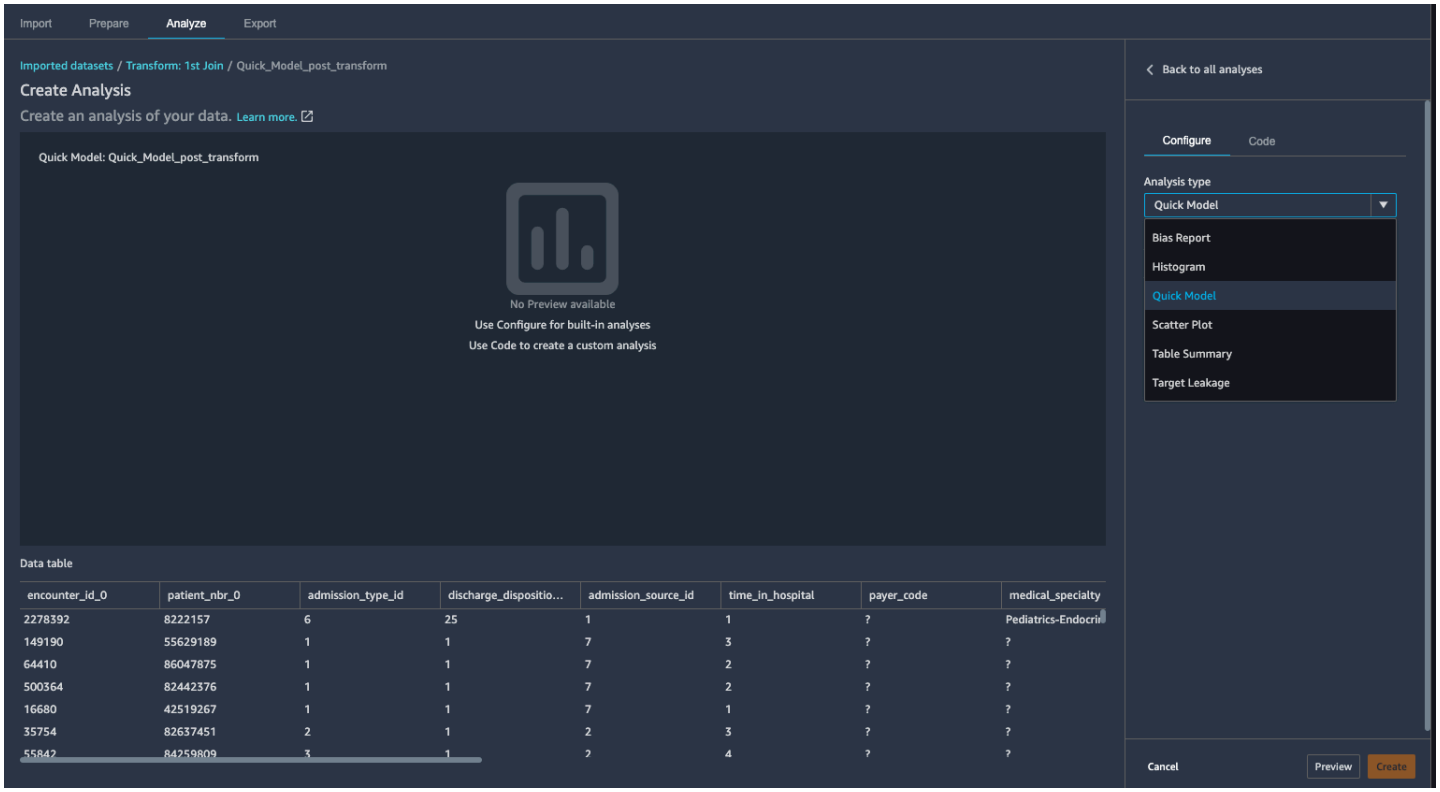
Quick Model

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.

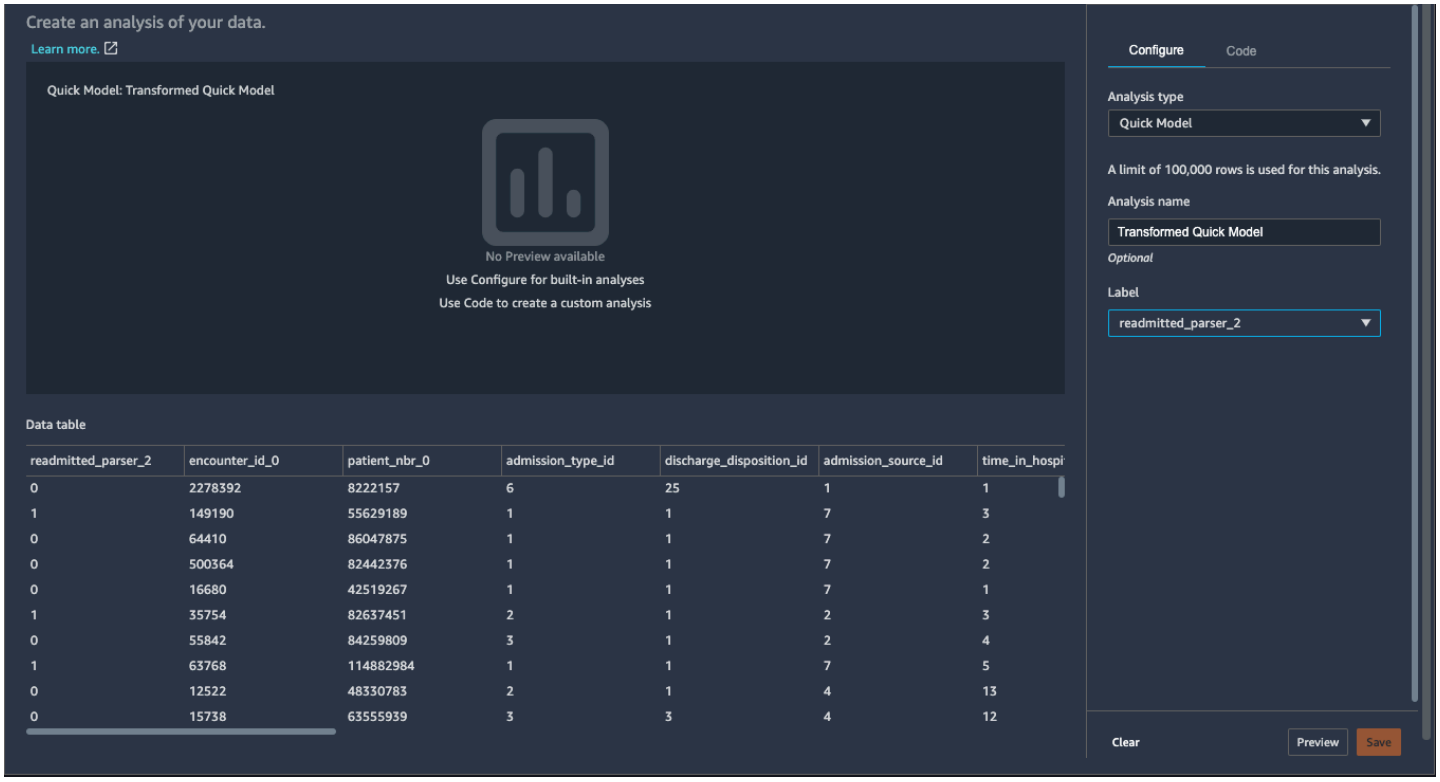
1) Click + sign next to Join flow icon and choose `Add analysis`



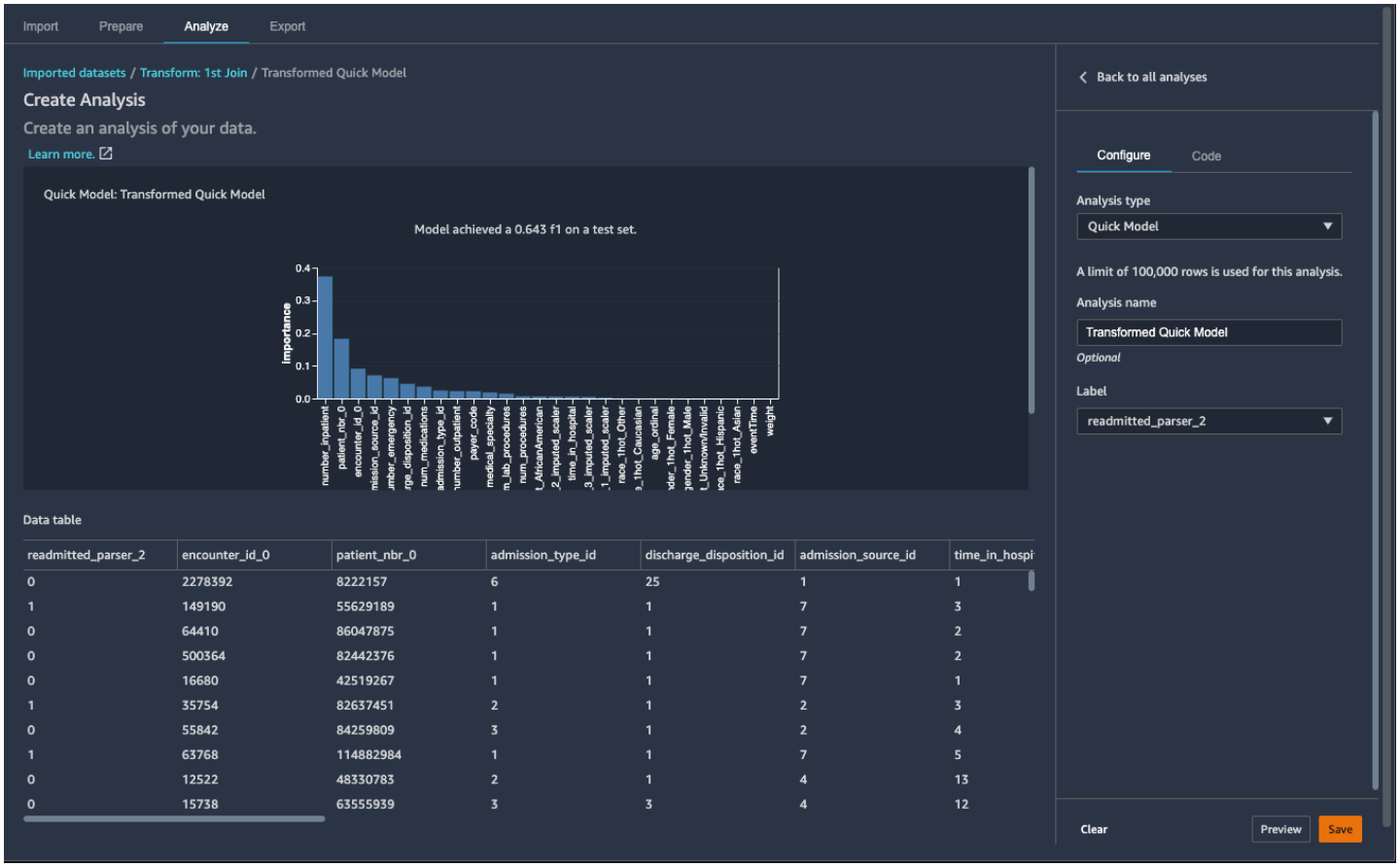
2) Select `Quick Model` from the list of Analysis types on the right panel.



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



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



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`.

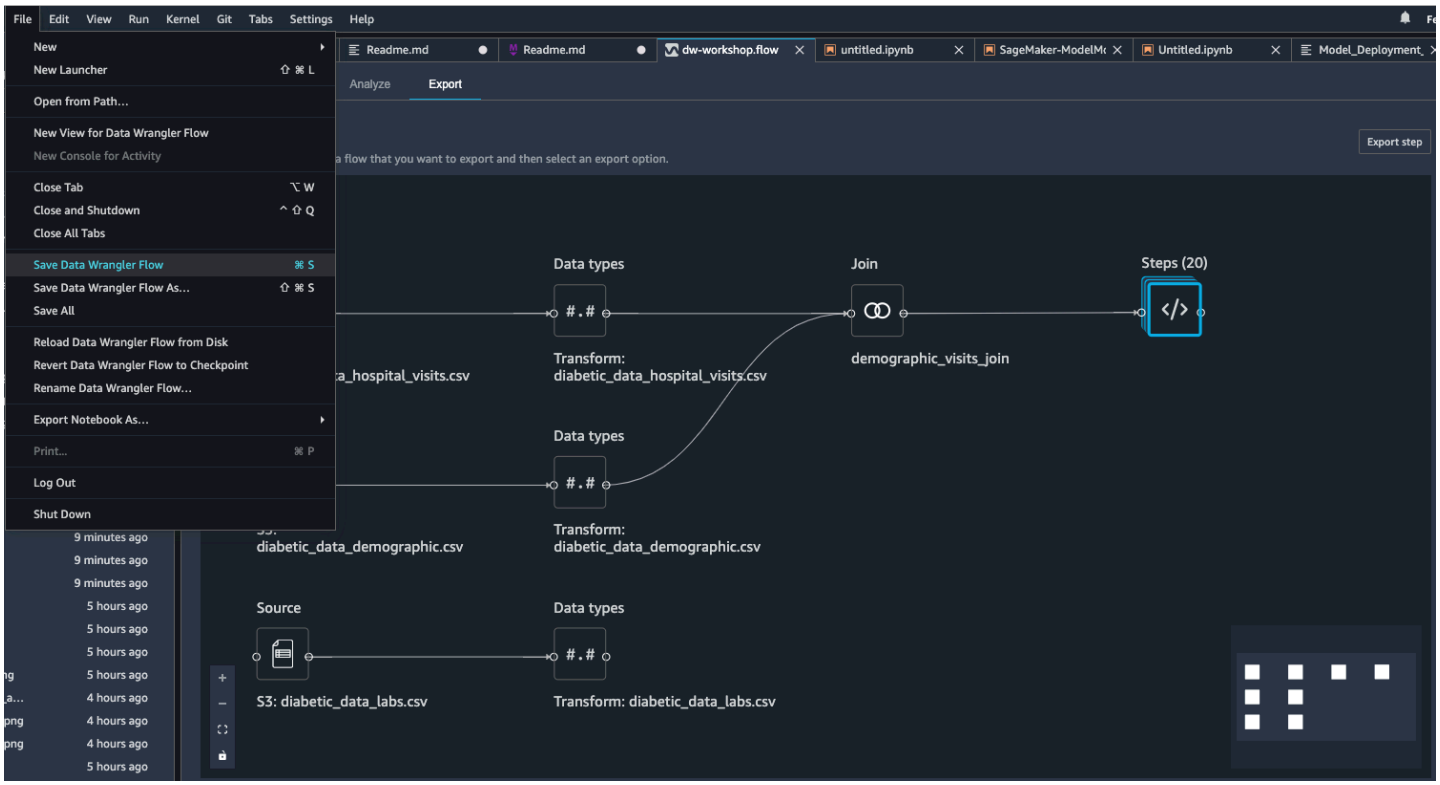
As you can see, Data Wrangler helped us to quickly experiment the transformations of input features and improve our model accuracy (without us building any ML models). Using this feature, Data Scientists can iterate through applicable transformations until they see desired transformed dataset that would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

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

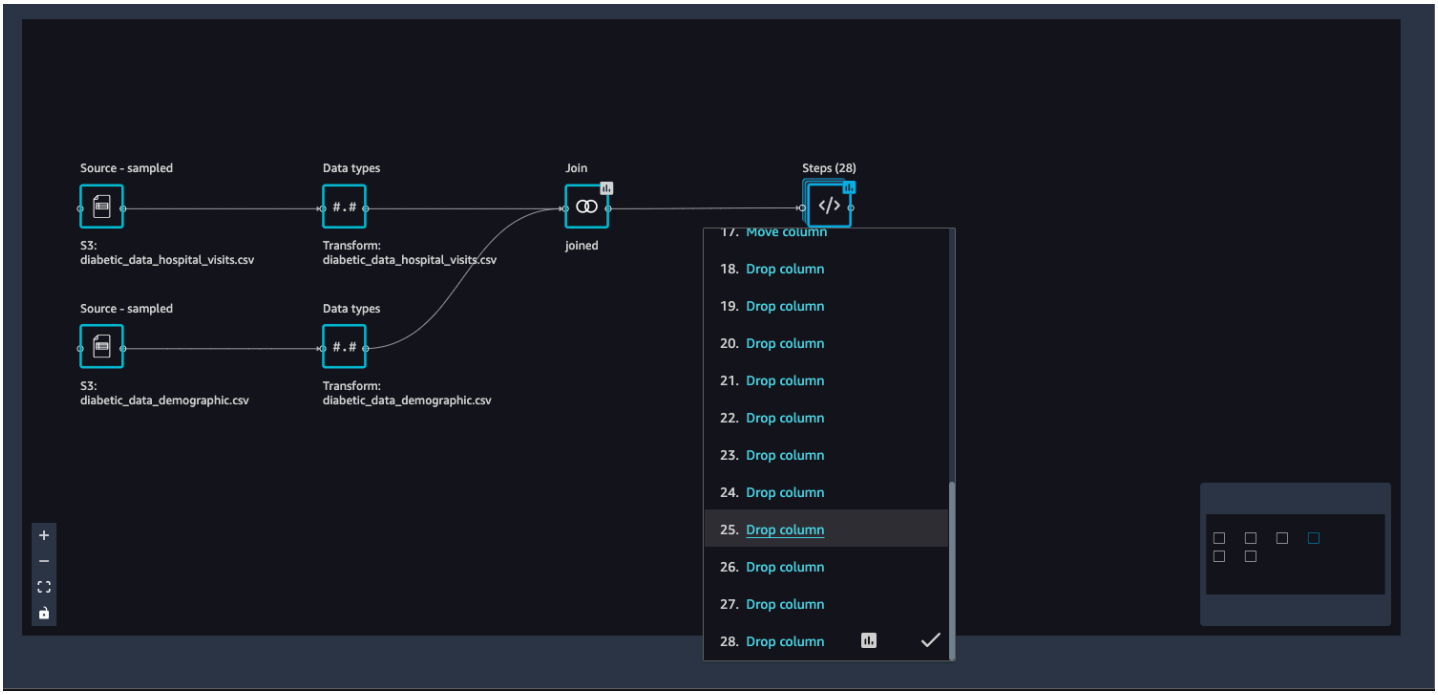
Export Options

We are now ready to export dataflow for further processing.

1) Save the DW flow file as shown below



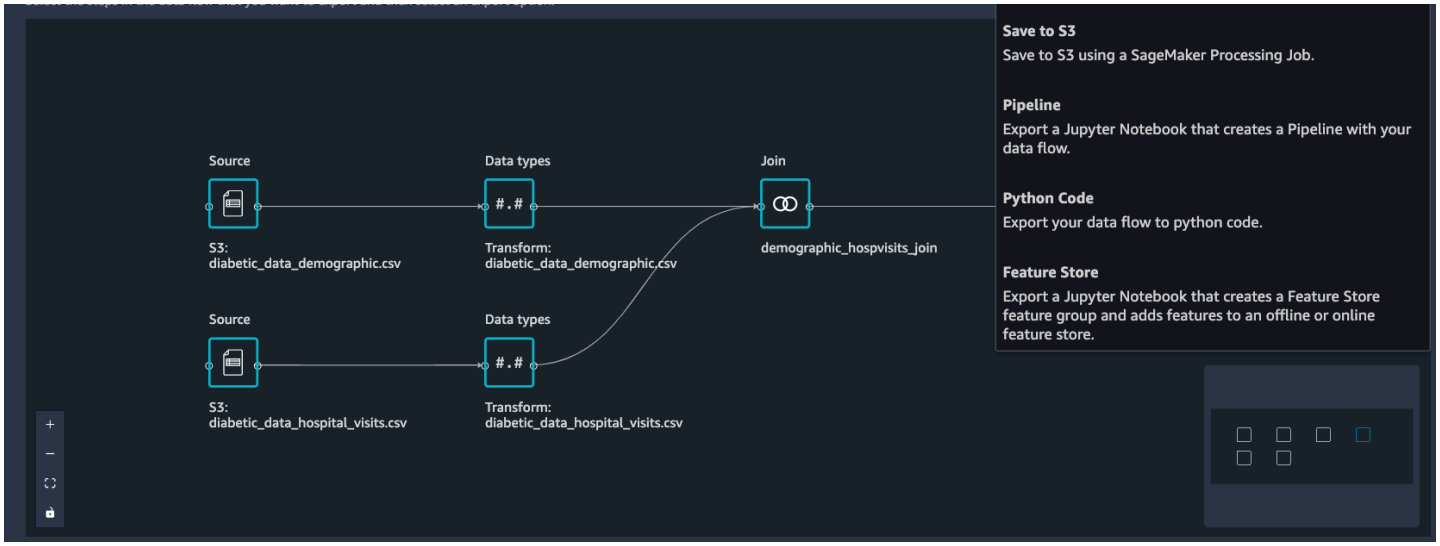
1) Click `Export` tab and select `Steps` icon to reveal all the DW flow steps. Click the last step to mark it as check (as shown in figure below)



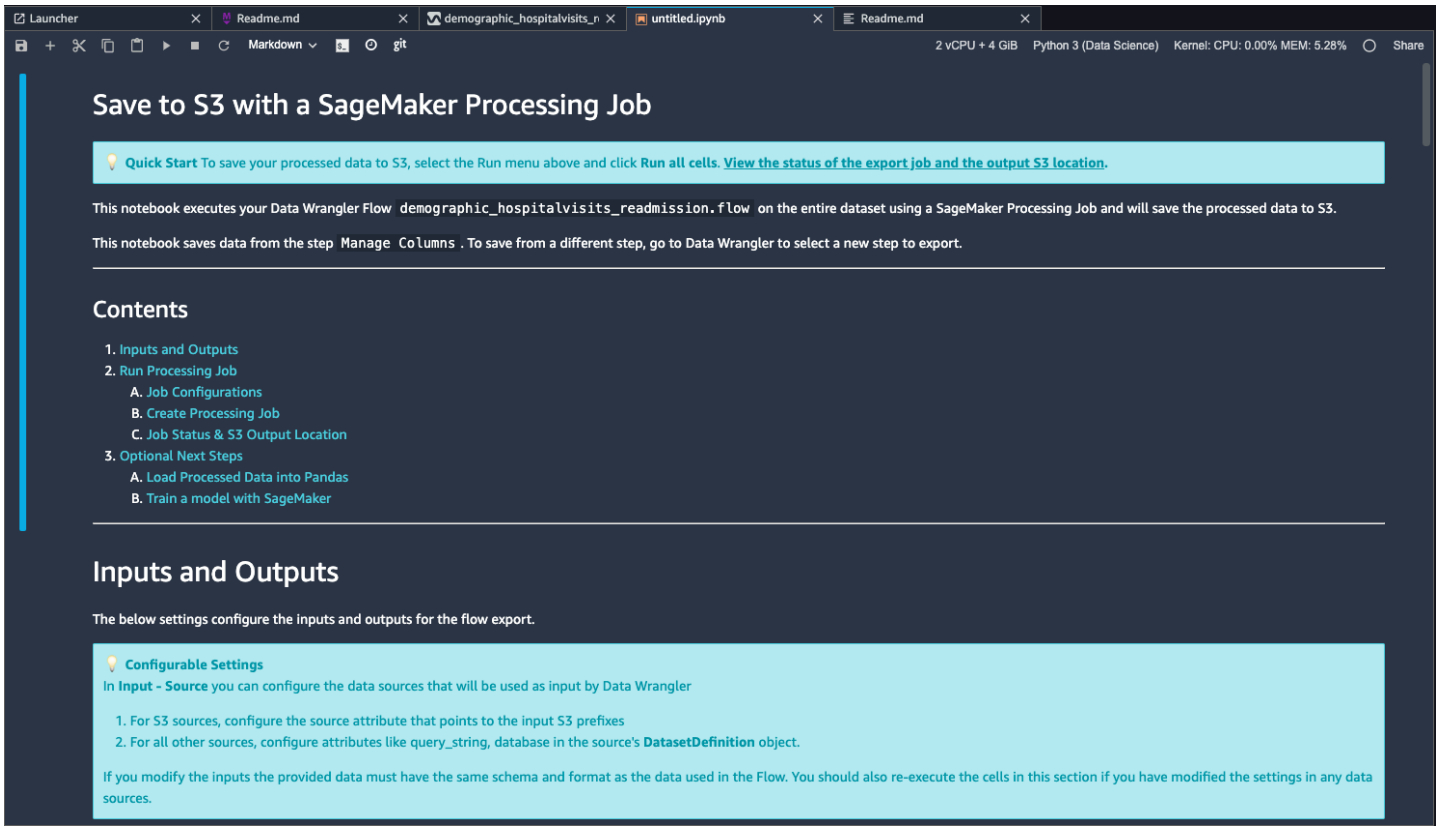
2) Click `Export step` to reveal the export options. You currently have 4 export options

- `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.



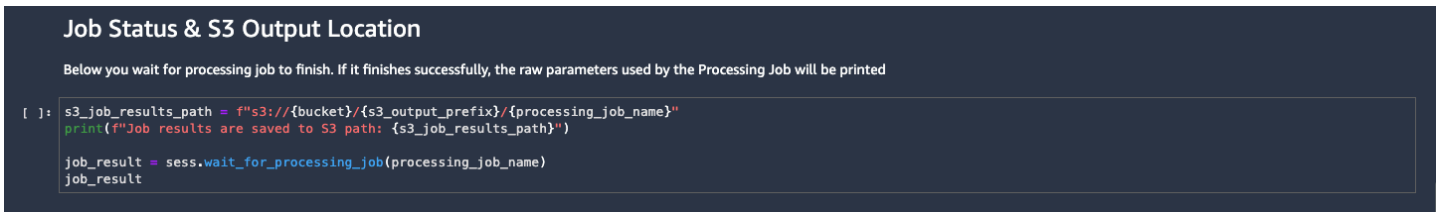
3. Processing and Training Jobs for Model building

Processing Job submission

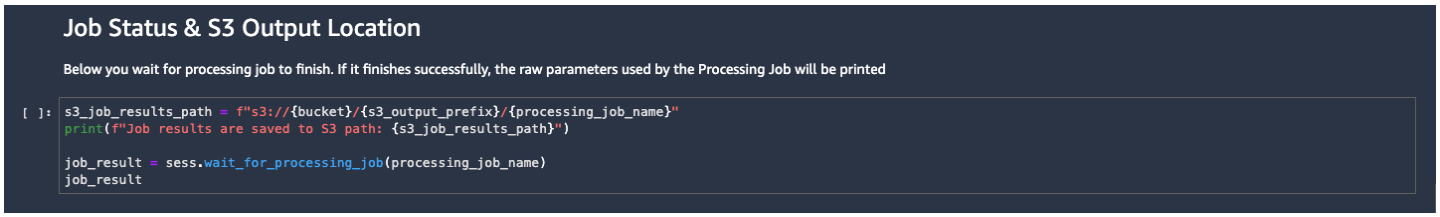
1) We are now ready to submit a SageMaker Processing Job using the data flow file. Run all the cells upto

1) 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 SagaMaker processing job by provisioning managed infrastructure and running the required DataWrangler docker container on that infrastructure.

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

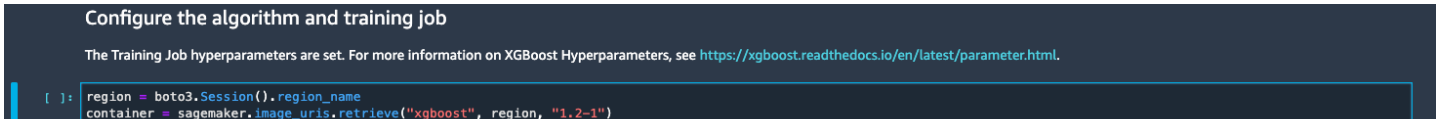


3) You can also check the status of the submitted processing job from Amazon SageMaker Console as shown below



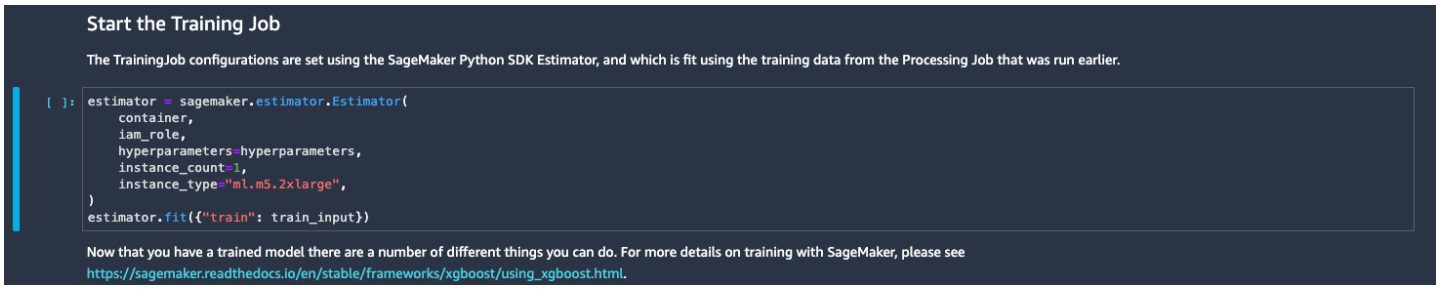
Train a model with Amazon SageMaker

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.

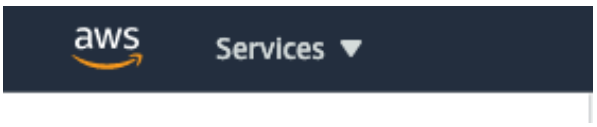


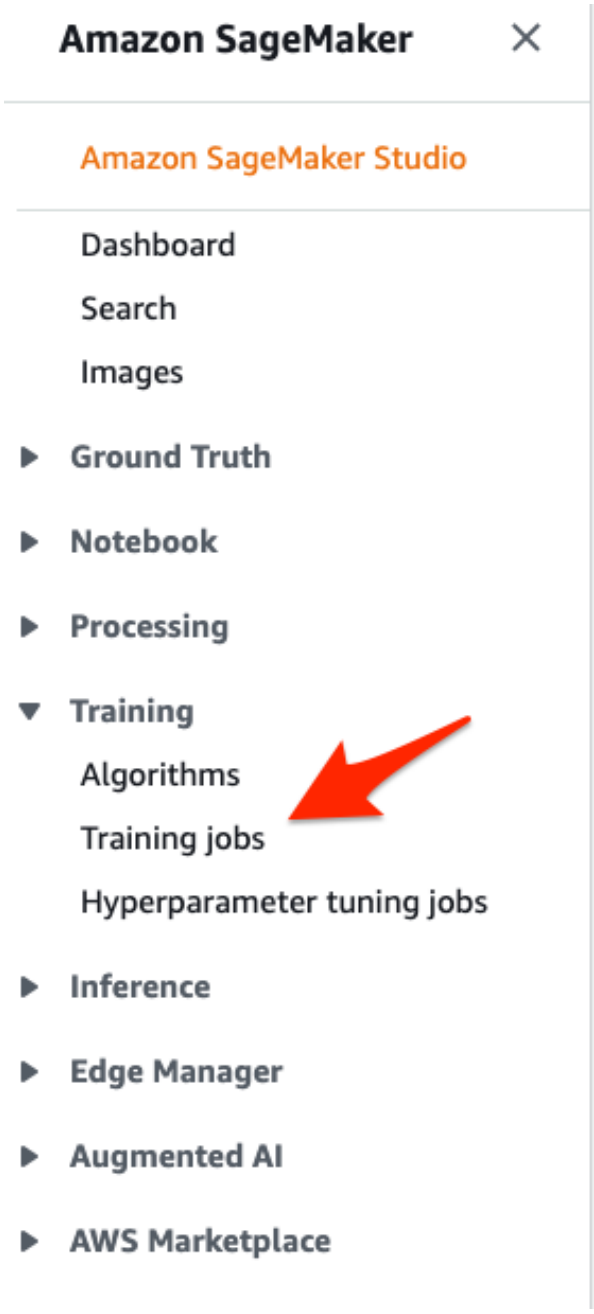
```
hyperparameters = {
    "max_depth":"5",
    "objective": "binary:logistic",
    "num_round": "10",
}
train_content_type = (
    "application/x-parquet" if output_content_type.upper() == "PARQUET"
    else "text/csv"
)
train_input = sagemaker.inputs.TrainingInput(
    s3_data=s3_training_input_path,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



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

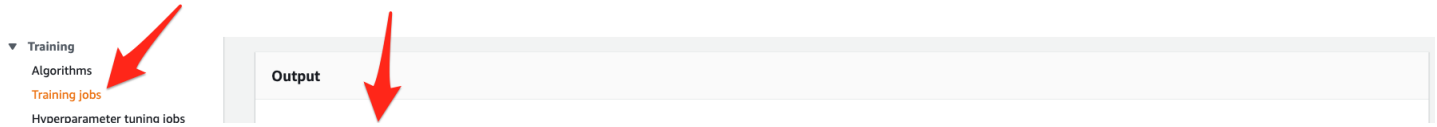


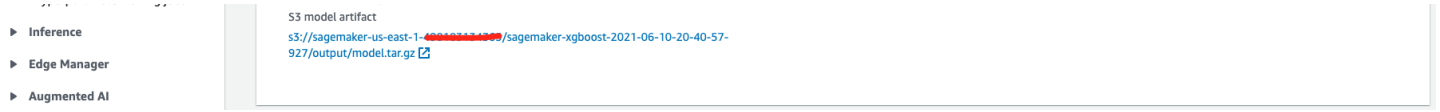


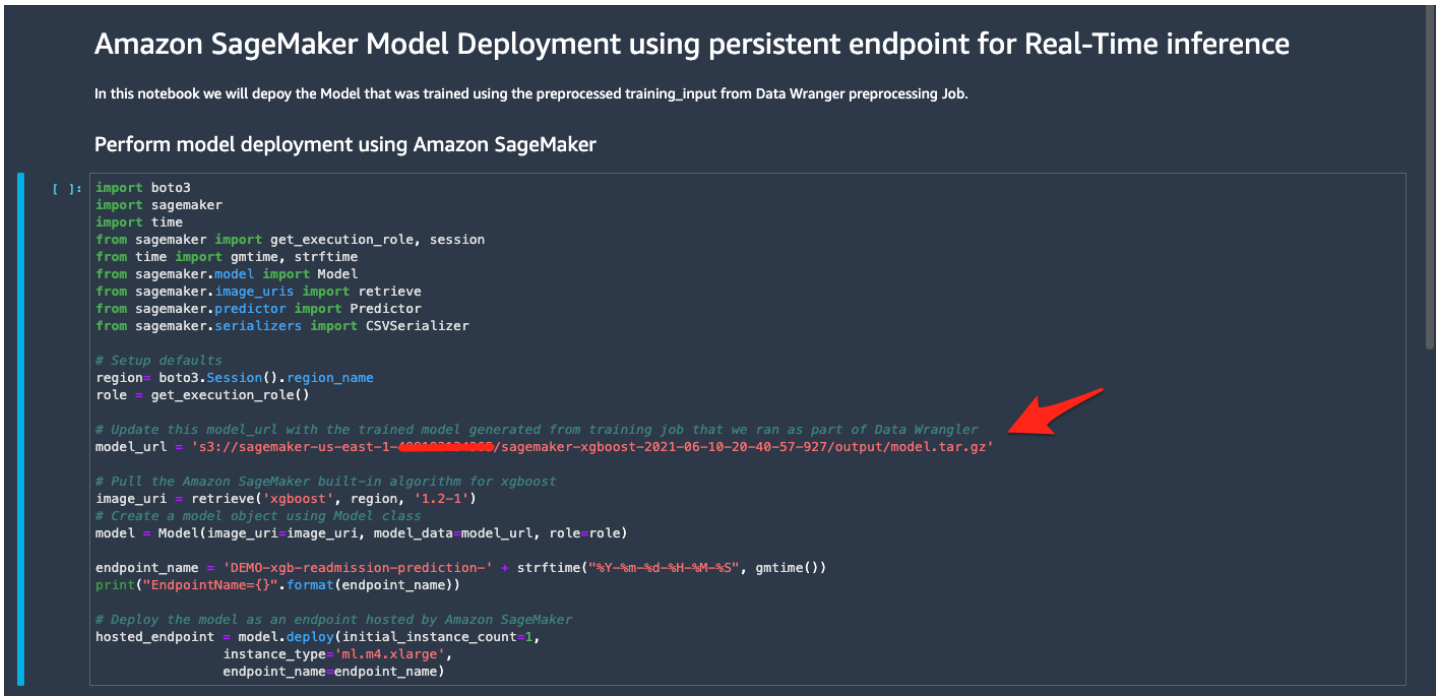
4. Host trained Model for real time inference

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.







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

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.