

# Assignment 5 – Data Wrangler

## Table of Contents

- I. Data set field description
- II. Analysis and Visualization in Data Wrangler
- III. Transformations using Data Wrangler.

### I. Data Description

WHO estimates that stroke accounts for approximately 11% of all total deaths globally, making it the 2nd most common cause of death.

The following is the healthcare dataset which defines the records of 5110 patients. Each row provides applicable information about the patient - its medical and general background.

Data from this dataset is used to predict whether a patient is likely to suffer from stroke based on input parameters such as their gender, age, variety of diseases, and smoking status.

The following dataset has been taken for Analysis purposes from the website called Kaggle. I give credits to the website and the author who has published this raw dataset.

The link to the dataset and its information is given below.

<https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>

# Assignment 5 – Data Wrangler

Data Attributed will provide information about the data used in the Dataset Excel Sheet i.e., 'healthcare-dataset-stroke-data.csv' that is attached.

1. Field Name – The different names of each field that is defined in the dataset and a brief description of those fields.
  - a. id : Primary – Unique Identifier
  - b. gender : "Male", "Female" or "Other"
  - c. age : Age of the patient.
  - d. hypertension : 0 if the patient does not have hypertension, 1 if the patient has hypertension.
  - e. heart\_disease : 0 if the patient does not have any heart diseases, 1 if the patient has a heart disease.
  - f. ever\_married : Yes OR No
  - g. work\_type : "children", "Govt\_jov", "Never\_worked", "Private" or "Self-employed"
  - h. residence\_type : "Urban" OR "Rural"
  - i. avg\_glucose\_level : Average glucose level in blood
  - j. bmi : Body Mass Index
  - k. smoking\_status : "formerly smoked", "never smoked", "smokes" or "Unknown"

## Assignment 5 – Data Wrangler

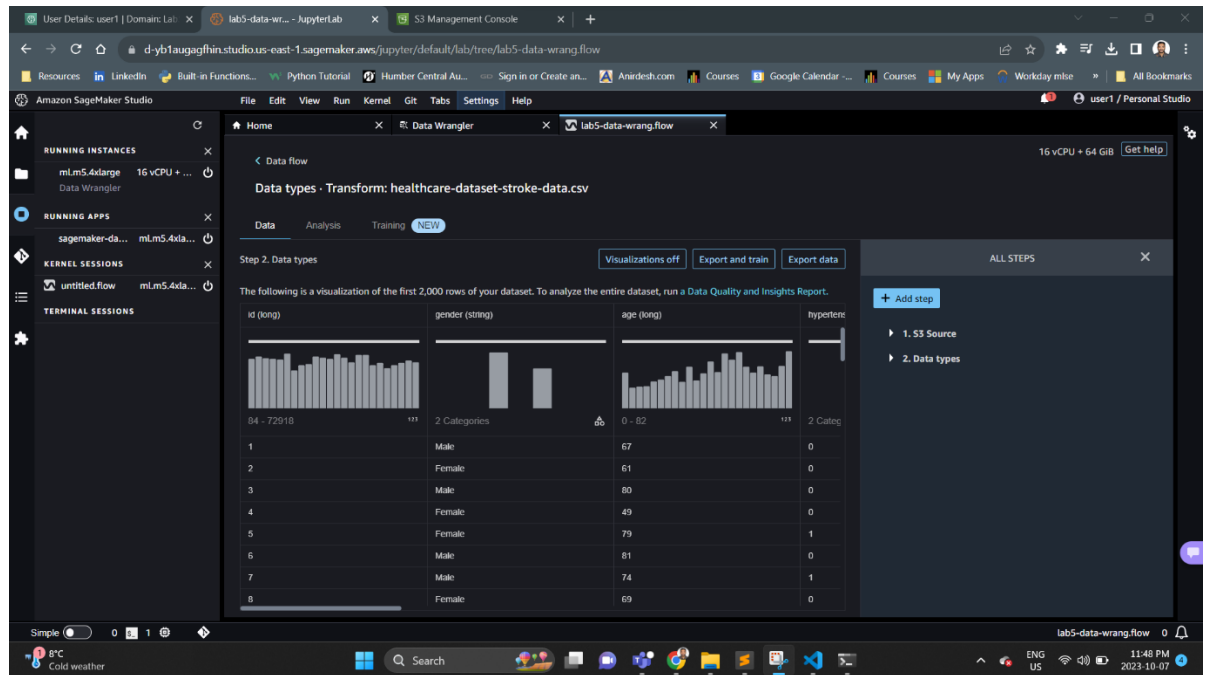
1. stroke : 1 if the patient had a stroke or 0 if he did not.
2. Data Types: Data types for each field are listed below.
  - a. id : Integer
  - b. gender : String
  - c. age : Integer
  - d. hypertension : Integer
  - e. heart\_disease : Integer
  - f. ever\_married : String
  - g. work\_type : String
  - h. residence\_type : String
  - i. avg\_glucose\_level : Decimal
  - j. bmi : Decimal
  - k. smoking\_status : String
  - l. stroke : Integer

The above information describes the dataset according to my understanding and knowledge and will do the analysis on it with Data Wrangler in AWS based on these data definitions.

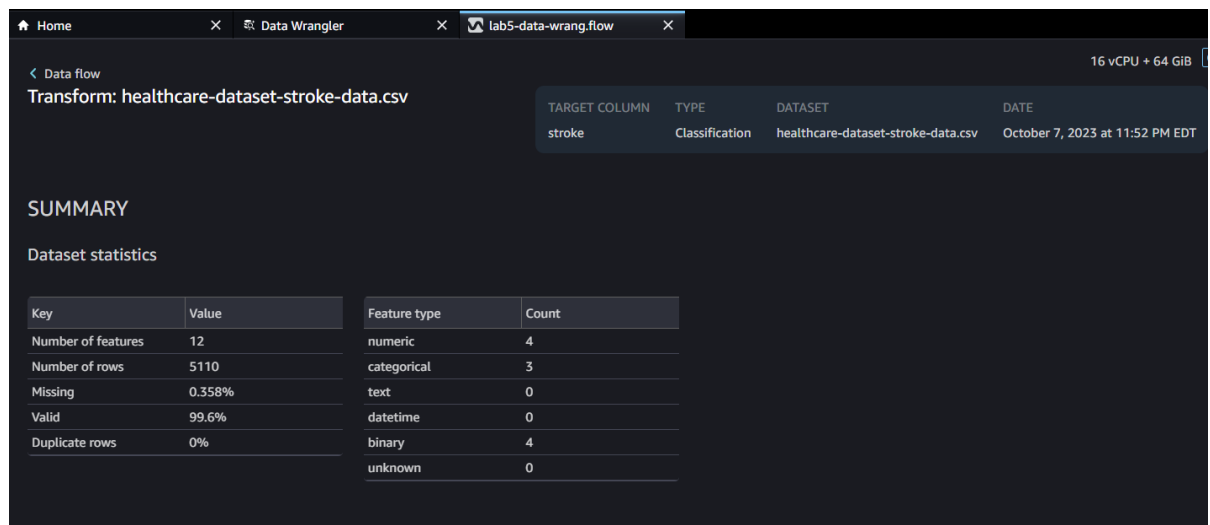
# Assignment 5 – Data Wrangler

## II. Analysis and Visualization in Data Wrangler

1. Load the dataset to S3 bucket, import the dataset to Data Wrangler from S3 bucket and create a flow diagram.



2. Generate a quick report for getting the summary of the dataset using inbuilt feature of Data Wrangler which gives information on missing values and number of outliers.



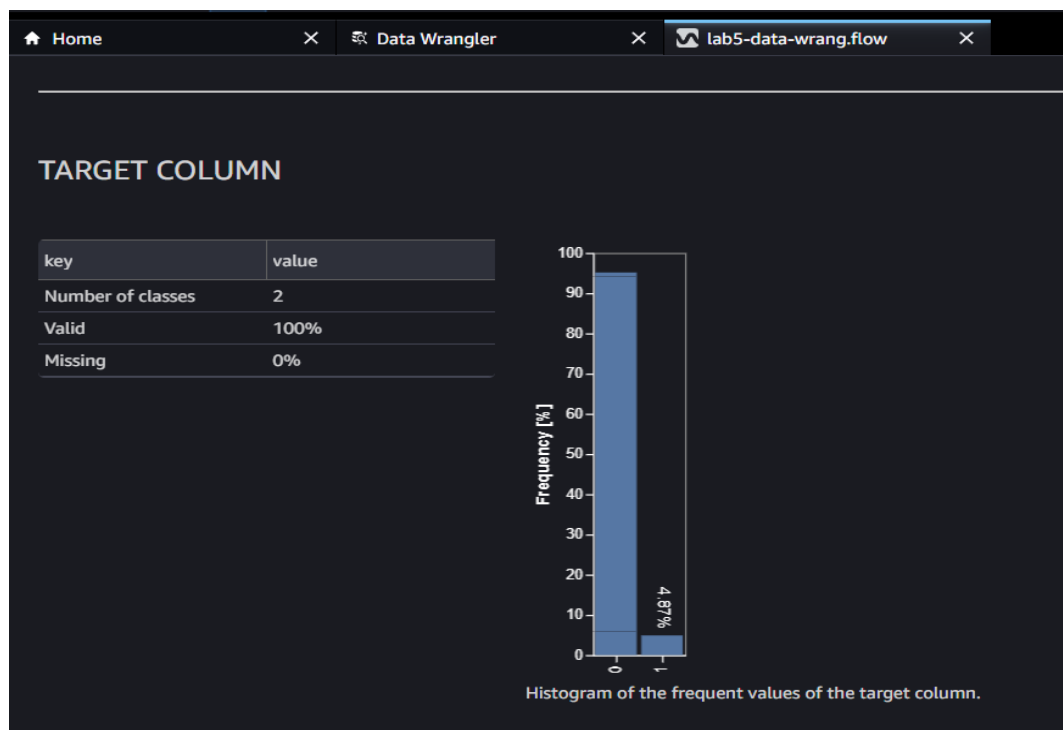
# Assignment 5 – Data Wrangler

Table Summary: TableSummary Refresh

Last generated: Sunday, October 8, 2023 at 12:10 AM EDT

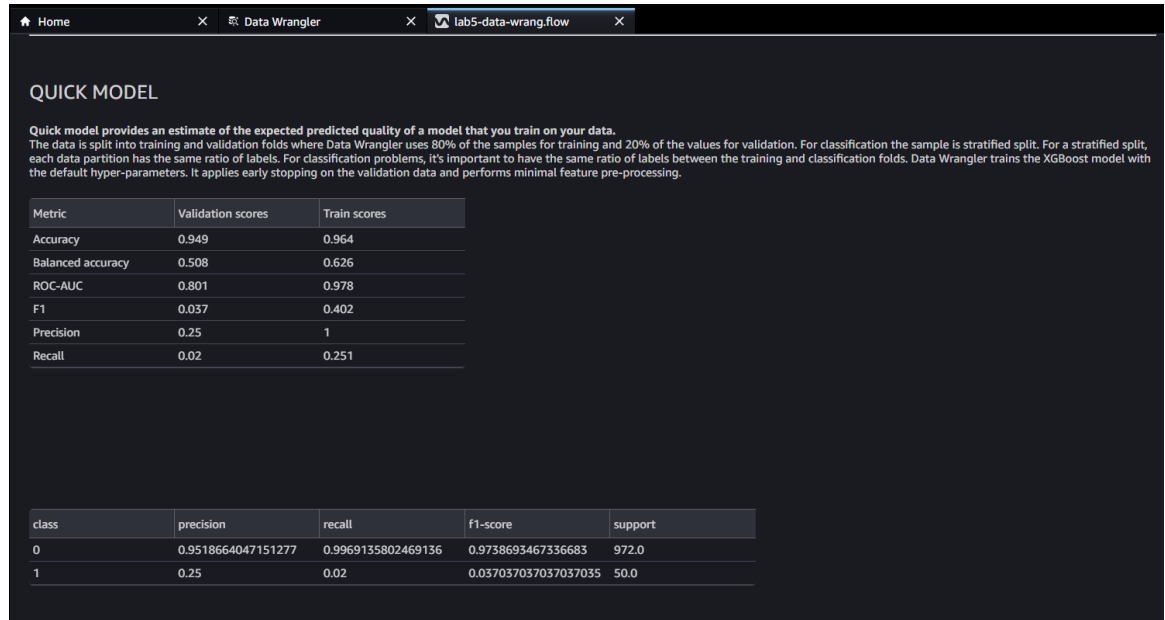
summary	id	gender	age	hypertension	heart_diseas
count	5110	5110	5110	5110	5110
mean	36517.82935420744	None	43.21526418786693	0.0974559686888454	0.05401174
stddev	21161.72162482715	None	22.633865752854746	0.296606674233791	0.22606298
min	67	Female	0	0	0
max	72940	Other	82	1	1

## 3. Target Column selection – ‘stroke’ analysis

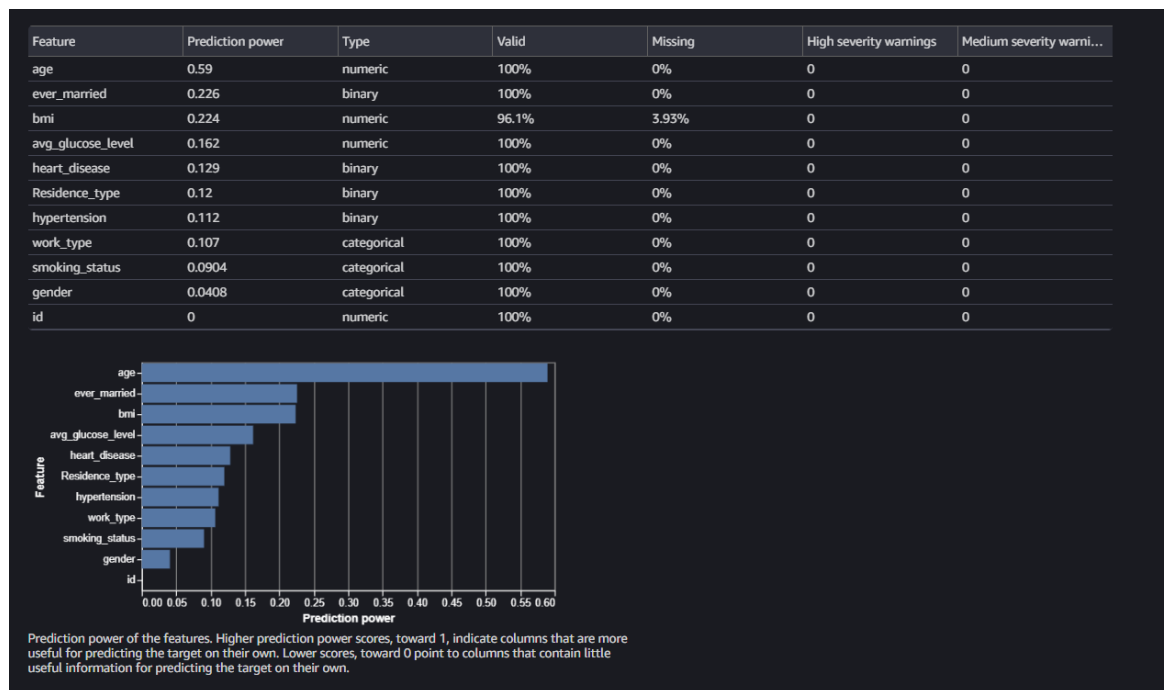


## 4. Data Wrangler can run a quick model on the dataset and give insights into the stroke prediction and also creates visualization.

# Assignment 5 – Data Wrangler



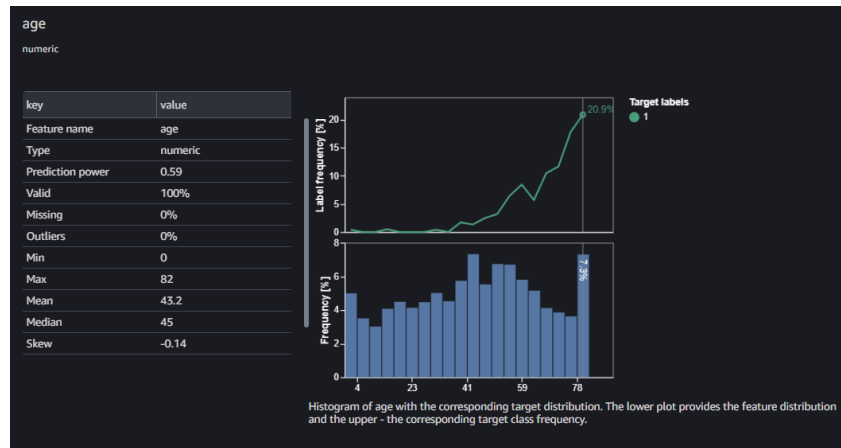
## 5. Feature Summary Report



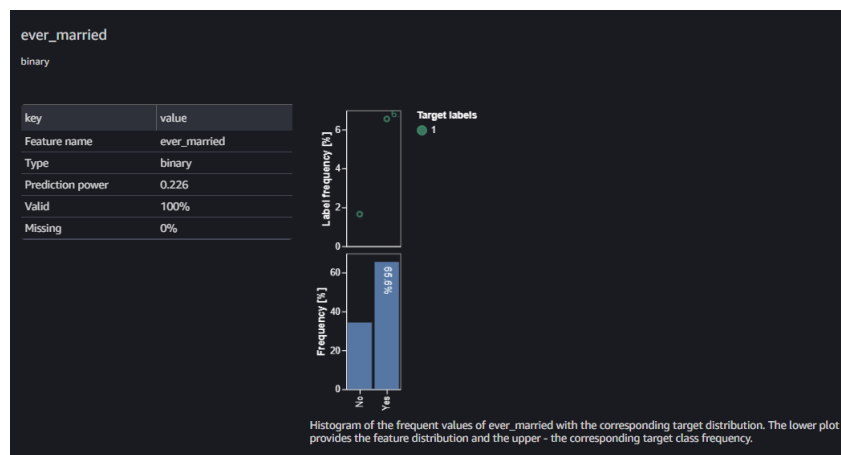
# Assignment 5 – Data Wrangler

6. Feature that are most important for consideration into predicting the output variable stroke.

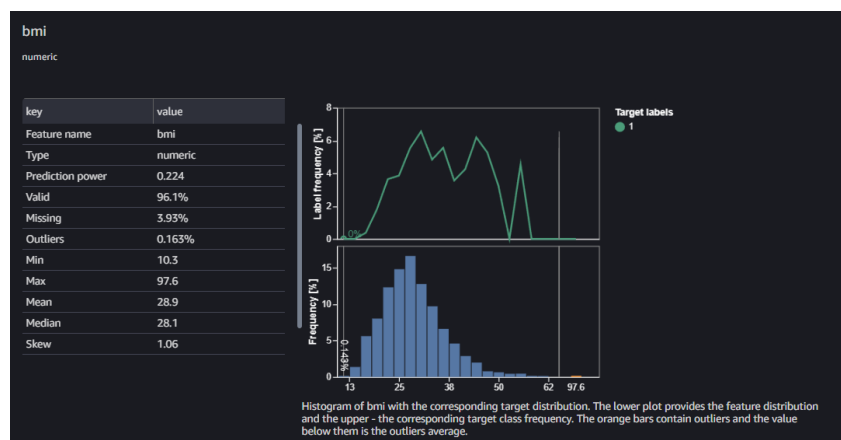
## a. Age Distribution



## b. Ever Married



## c. BMI Levels



# Assignment 5 – Data Wrangler

## III. Transformations using Data Wrangler.

### 1. Drop Columns Transformations.

Step 3. Drop column

The following is a visualization of the first 2,000 rows of your dataset. To analyze the entire dataset, run a Data Quality and Insights Report.

age (long)	hypertension (long)	heart_disease (long)	ever_married (string)
67	0	1	Yes
61	0	0	Yes
80	0	1	Yes
49	0	0	Yes
79	1	0	Yes
81	0	0	Yes
74	1	1	Yes
69	0	0	No
59	0	0	Yes
78	0	0	Yes
81	1	0	Yes

Columns to drop: id, gender, smoking\_status

Columns – ‘id’, ‘gender’ and ‘smoke\_status’ were dropped as they did not have any significant impact on the model taking consideration of Feature summary report.

### 2. Identify Missing Values using custom code.

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

Clear Preview Add

Output

```
1 <class 'pandas.core.frame.DataFrame'>
2 RangeIndex: 5110 entries, 0 to 5109
3 Data columns (total 9 columns):
4 #   Column              Non-Null Count  Dtype
5 ---  ---
6 0   age                  5110 non-null   int64
7 1   hypertension          5110 non-null   int64
8 2   heart_disease         5110 non-null   int64
9 3   ever_married          5110 non-null   object
10 4   work_type             5110 non-null   object
11 5   Residence_type        5110 non-null   object
12 6   avg_glucose_level     5110 non-null   float64
13 7   bmi                   4909 non-null   float64
14 8   stroke                5110 non-null   int64
15 dtypes: float64(2), int64(4), object(3)
16 memory usage: 359.4+ KB
17
```

Running a python code `df.info()` we found the information that the ‘bmi’ column has some missing values.



# Assignment 5 – Data Wrangler

## 3. Impute Missing Values with Mean of the column BMI.

The screenshot shows the Data Wrangler interface for a dataset named 'healthcare-dataset-stroke-data.csv'. The 'Impute' step is selected, and the 'bmi' column is chosen for imputation. The imputing strategy is set to 'Mean'. The interface includes a preview of the first 2,000 rows of the dataset, a histogram of the 'bmi' column, and a sidebar with steps: 2. Data types, 3. Drop column, and 4. Impute.

residence_type (string)	avg_glucose_level (float)	bmi (float)	stroke (long)
ban	228.69	36.6	1
rral	282.21	28.893236911794673	1
rral	105.92	32.5	1
ban	171.23	34.4	1
rral	174.12	24	1
ban	186.21	29	1
rral	70.09	27.4	1
ban	94.39	22.8	1
rral	76.15	28.893236911794673	1
ban	58.57	24.2	1
rral	80.43	20.7	1

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

Clear
Output
Preview
Insert

1 <class 'pandas.core.frame.DataFrame'>
2 RangeIndex: 5110 entries, 0 to 5109
3 Data columns (total 9 columns):
4 #   Column              Non-Null Count  Dtype
5 ---  ---
6 0   age                  5110 non-null   int64
7 1   hypertension          5110 non-null   int64
8 2   heart_disease         5110 non-null   int64
9 3   ever_married          5110 non-null   object
10 4   work_type             5110 non-null   object
11 5   Residence_type         5110 non-null   object
12 6   avg_glucose_level     5110 non-null   float64
13 7   bmi                   5110 non-null   float64
14 8   stroke                5110 non-null   int64
15 dtypes: float64(2), int64(4), object(3)
16 memory usage: 359.4+ KB
17
```

As the 'bmi' column is important for the prediction of the target column 'stroke' we are not removing those columns instead we would impute the missing values of bmi with the mean of the column.

# Assignment 5 – Data Wrangler

## 4. Perform One Hot Encoding on categorical variables.

Step 5: OHE

The following is a visualization of the first 2,000 rows of your dataset. To analyze the entire dataset, run a Data Quality and Insights Report.

age (long)	hypertension (long)	heart_disease (long)	ever_married (string)
67	0	1	Yes
61	0	0	Yes
80	0	1	Yes
49	0	0	Yes
79	1	0	Yes
81	0	0	Yes
74	1	1	Yes
69	0	0	No
59	0	0	Yes
78	0	0	Yes
81	1	0	Yes

Optional

Python (Pandas)

Using Python (Pandas) requires your dataset to fit in memory and only uses a single instance in batch computation. It is ideal for smaller datasets less than 2GB and experimentation but we recommend Python (PySpark) or Python (User-Defined Function) for production use-cases.

```
1 # Table is available as variable "df"
2 import pandas as pd
3
4 dummies = []
5 cols =
6   ['ever_married', 'work_type', 'Residence_type']
7   for col in cols:
8     dummies.append(pd.get_dummies(df[col]))
9
10 encoded = pd.concat(dummies, axis=1)
11 df = pd.concat((df, encoded), axis=1)
```

Clear Preview Update

6. Selection

As there are multiple categorical variables in the dataset and I wish to use those variables as features in a machine learning model I have applied One-hot-encoding on few categories such as 'ever\_married', 'work\_type' and 'Residence\_type'.

## 5. Select Required columns for further analysis.

Step 5: OHE

The following is a visualization of the first 2,000 rows of your dataset. To analyze the entire dataset, run a Data Quality and Insights Report.

age (long)	hypertension (long)	heart_disease (long)	ever_married (string)
67	0	1	Yes
61	0	0	Yes
80	0	1	Yes
49	0	0	Yes
79	1	0	Yes
81	0	0	Yes
74	1	1	Yes
69	0	0	No
59	0	0	Yes
78	0	0	Yes
81	1	0	Yes

CUSTOM TRANSFORM

Use PySpark, Pandas, or PySpark (SQL) to define custom transformations. [Learn more](#)

Name

Selection

Optional

SQL (PySpark SQL)

```
1 /* Table is available as variable "df" */
2 SELECT age, hypertension, heart_disease,
3        avg_glucose_level, bmi, stroke, No, Yes, Govt_job,
4        Never_worked, Private, Self-employed, children, Rural,
5        Urban
6 FROM df;
```

Clear Preview Insert

After performing One-hot-encoding the columns that I thought that are required for the input to the machine learning model are been retained and the remaining column have been dropped. I have used PySpark SQL for querying the data frame and select what is required.