LAB – 3: PANDAS FILLING OUT MISSING VALUES

PANDAS:

Pandas is a Python software library for data analysis and manipulation widely used in data science and machine learning applications enabling cleansing, merging, and reshaping of data structures for big datasets, handling various file formats including Excel, JSON, Parquet, SQL database tables or queries, and CSV files.

Installation: pip install pandas

Here is a Pandas methods for filling out missing values:

1. **isnull():** Checks for missing values in a DataFrame or Series.

   Syntax: df.isnull()

2. **dropna():** Removes rows with missing values from a DataFrame.

   Syntax: df.dropna()

3. **fillna():** Fills missing values in a DataFrame or Series with a specified value.

   Syntax: df.fillna(value)

4. **fillna() “pad” method:** Fills missing values with the previous non-null value along the specified axis.
   Syntax: df.fillna(method='pad')

5. **fillna() “bfill” method:** Fills missing values with the next non-null value along the specified axis.

   Syntax: df.fillna(method='bfill')

6. **sum():** Computes the sum of values along a specified axis

   Syntax: df.sum()

7. **replace():** Replaces values with other values in a DataFrame or Series.

   Syntax: df.replace(to_replace, value)

8. **fillna() “mean”:** Fills missing values with the mean of the non-null values in a DataFrame or Series.

   Syntax: df.fillna(df.mean())

9. **fillna() “median”:** Fills missing values with the median of the non-null values in a DataFrame or Series.

   Syntax: df.fillna(df.median())

10. **fillna() “mode”:** Fills missing values with the mode (most frequent value) in a DataFrame or Series.

    Syntax: df.fillna(df.mode())

11. **corr():** Computes the correlation matrix of a DataFrame.
    Syntax: df.corr()

Pandas Filling out Missing Values

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
dict_data = {
    'Field 1':[88, 90, 80 , 70],
    'Field 2': [30, 45, 50, np.nan],
    'Field 3':[np.nan, 40, 80, np.nan]
}

df_dict_data = pd.DataFrame(dict_data)
df_dict_data
```

Out[2]:

	Field 1	Field 2	Field 3
0	88	30.0	NaN
1	90	45.0	40.0
2	80	50.0	80.0
3	70	NaN	NaN

In [3]:

```
df_dict_data.isnull()
```

Out[3]:

	Field 1	Field 2	Field 3
0	False	False	True
1	False	False	False
2	False	False	False
3	False	True	True

In [4]:

```
df_dict_data.dropna()
```

Out[4]:

	Field 1	Field 2	Field 3
1	90	45.0	40.0
2	80	50.0	80.0

In [5]:

```
bool_series = pd.isnull(df_dict_data["Field 3"])
bool_series
```

Out[5]:

```
0     True
1    False
2    False
```

```
3     True
Name: Field 3, dtype: bool
```

In [6]:

```
df_dict_data[bool_series]
```

Out[6]:

	Field 1	Field 2	Field 3
0	88	30.0	NaN
3	70	NaN	NaN

Filling missing value with some constant (0) using fillna()

In [7]:

```
df_dict_data.fillna(23)
```

Out[7]:

	Field 1	Field 2	Field 3
0	88	30.0	23.0
1	90	45.0	40.0
2	80	50.0	80.0
3	70	23.0	23.0

Filling a missing value with previous value

In [8]:

```
df_dict_data.fillna(method='pad', axis=1)
```

Out[8]:

	Field 1	Field 2	Field 3
0	88.0	30.0	30.0
1	90.0	45.0	40.0
2	80.0	50.0	80.0
3	70.0	70.0	70.0

Filling a missing value with ahead value

In [9]:

```
df_dict_data.fillna(method='bfill')
```

Out[9]:

	Field 1	Field 2	Field 3
0	88	30.0	40.0
1	90	45.0	40.0
2	80	50.0	80.0
3	70	NaN	NaN

Playing with CSV

In [10]:

```
diabetes_df = pd.read_csv("https://raw.githubusercontent.com/YBI-Foundation/Dataset/main/Diabetes%20Missing%20Data.csv")
```

In [11]:

```
diabetes_df.head()
```

Out[11]:

	Pregnant	Glucose	Diastolic_BP	Skin_Fold	Serum_Insulin	BMI	Diabetes_Pedigree	Age	Class
0	6	148.0	72.0	35.0	NaN	33.6	0.627	50	1
1	1	85.0	66.0	29.0	NaN	26.6	0.351	31	0
2	8	183.0	64.0	NaN	NaN	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1

In [12]:

```
diabetes_df.shape
```

Out[12]:

```
(768, 9)
```

In [13]:

```
diabetes_df.isnull().sum()
```

Out[13]:

```
Pregnant              0
Glucose               5
Diastolic_BP         35
Skin_Fold           227
Serum_Insulin       374
BMI                  11
Diabetes_Pedigree     0
Age                   0
Class                 0
dtype: int64
```

In [14]:

```
diabetes_df.isnull().sum().sum()
```

Out[14]:

```
652
```

In [15]:

```
diabetes_df.dropna().head()
```

Out[15]:

	Pregnant	Glucose	Diastolic_BP	Skin_Fold	Serum_Insulin	BMI	Diabetes_Pedigree	Age	Class
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1
6	3	78.0	50.0	32.0	88.0	31.0	0.248	26	1
8	2	197.0	70.0	45.0	543.0	30.5	0.158	53	1

	Pregnant	Glucose	Diastolic_BP	Skin_Fold	Serum_Insulin	BMI	Diabetes_Pedigree	Age	Class
13	1	189.0	60.0	23.0	846.0	30.1	0.398	59	1

In [16]:

```
diabetes_df = diabetes_df.drop(['Serum_Insulin'], axis=1)
diabetes_df.head()
```

Out[16]:

	Pregnant	Glucose	Diastolic_BP	Skin_Fold	BMI	Diabetes_Pedigree	Age	Class
0	6	148.0	72.0	35.0	33.6	0.627	50	1
1	1	85.0	66.0	29.0	26.6	0.351	31	0
2	8	183.0	64.0	NaN	23.3	0.672	32	1
3	1	89.0	66.0	23.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	43.1	2.288	33	1

In [17]:

```
diabetes_df["Glucose"].isnull().sum()
```

Out[17]:

```
5
```

In [18]:

```
diabetes_df.replace(to_replace=np.nan, value=-99).tail()
```

Out[18]:

	Pregnant	Glucose	Diastolic_BP	Skin_Fold	BMI	Diabetes_Pedigree	Age	Class
763	10	101.0	76.0	48.0	32.9	0.171	63	0
764	2	122.0	70.0	27.0	36.8	0.340	27	0
765	5	121.0	72.0	23.0	26.2	0.245	30	0
766	1	126.0	60.0	-99.0	30.1	0.349	47	1
767	1	93.0	70.0	31.0	30.4	0.315	23	0

In [19]:

```
diabetes_df[20:25]
```

Out[19]:

	Pregnant	Glucose	Diastolic_BP	Skin_Fold	BMI	Diabetes_Pedigree	Age	Class
20	3	126.0	88.0	41.0	39.3	0.704	27	0
21	8	99.0	84.0	NaN	35.4	0.388	50	0
22	7	196.0	90.0	NaN	39.8	0.451	41	1
23	9	119.0	80.0	35.0	29.0	0.263	29	1
24	11	143.0	94.0	33.0	36.6	0.254	51	1

In [20]:

```
bool_series = pd.isnull(diabetes_df["Glucose"])
```

In [21]:

```
diabetes_df[bool_series]
```

Out[21]:

	Pregnant	Glucose	Diastolic_BP	Skin_Fold	BMI	Diabetes_Pedigree	Age	Class
75	1	NaN	48.0	20.0	24.7	0.140	22	0
182	1	NaN	74.0	20.0	27.7	0.299	21	0
342	1	NaN	68.0	35.0	32.0	0.389	22	0
349	5	NaN	80.0	32.0	41.0	0.346	37	1
502	6	NaN	68.0	41.0	39.0	0.727	41	1

In [22]:

```
bool_series = pd.notnull(diabetes_df["Glucose"])
```

In [23]:

```
diabetes_df[bool_series].head()
```

Out[23]:

	Pregnant	Glucose	Diastolic_BP	Skin_Fold	BMI	Diabetes_Pedigree	Age	Class
0	6	148.0	72.0	35.0	33.6	0.627	50	1
1	1	85.0	66.0	29.0	26.6	0.351	31	0
2	8	183.0	64.0	NaN	23.3	0.672	32	1
3	1	89.0	66.0	23.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	43.1	2.288	33	1

In [24]:

```
diabetes_df["Diastolic_BP"].fillna("3", inplace=True)
```

In [25]:

```
diabetes_df.isnull().sum()
```

Out[25]:

```
Pregnant                0
Glucose                 5
Diastolic_BP            0
Skin_Fold             227
BMI                    11
Diabetes_Pedigree       0
Age                     0
Class                   0
dtype: int64
```

Filling out with missing value with Mean

In [26]:

```
diabetes_df["Glucose"].fillna(value=diabetes_df["Glucose"].mean(), inplace=True)
```

In [27]:

```
diabetes_df.isnull().sum()
```

Out[27]:

```
Pregnant                0
Glucose                 0
Diastolic_BP            0
Skin_Fold             227
BMI                    11
```

```
Diabetes_Pedigree       0
Age                     0
Class                   0
dtype: int64
```

Filling out missing value with Median

In [28]:

```
diabetes_df["Skin_Fold"].fillna(value=diabetes_df["Skin_Fold"].median(), inplace=True)
```

In [29]:

```
diabetes_df.isnull().sum()
```

Out[29]:

```
Pregnant               0
Glucose                0
Diastolic_BP           0
Skin_Fold              0
BMI                   11
Diabetes_Pedigree      0
Age                    0
Class                  0
dtype: int64
```

Filling out missing value with Mode

In [30]:

```
diabetes_df["BMI"].fillna(value=diabetes_df["BMI"].median(), inplace=True)
```

In [31]:

```
diabetes_df.isnull().sum()
```

Out[31]:

```
Pregnant               0
Glucose                0
Diastolic_BP           0
Skin_Fold              0
BMI                    0
Diabetes_Pedigree      0
Age                    0
Class                  0
dtype: int64
```

In [32]:

```
diabetes_df.corr()
```

```
<ipython-input-32-0aaddaacdfa9>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  diabetes_df.corr()
```

Out[32]:

	Pregnant	Glucose	Skin_Fold	BMI	Diabetes_Pedigree	Age	Class
Pregnant	1.000000	0.127911	0.081770	0.021559	-0.033523	0.544341	0.221898
Glucose	0.127911	1.000000	0.192686	0.231128	0.137060	0.266534	0.492928
Skin_Fold	0.081770	0.192686	1.000000	0.543205	0.102188	0.126107	0.214873

	Pregnant	Glucose	Skin_Fold	BMI	Diabetes_Pedigree	Age	Class
BMI	0.021559	0.231128	0.543205	1.000000	0.153438	0.025597	0.312038
Diabetes_Pedigree	-0.033523	0.137060	0.102188	0.153438	1.000000	0.033561	0.173844
Age	0.544341	0.266534	0.126107	0.025597	0.033561	1.000000	0.238356
Class	0.221898	0.492928	0.214873	0.312038	0.173844	0.238356	1.000000