Handling null values

In [1]:
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
data = pd.read_csv('Windspeed.csv')
data.head()
```

Out[2]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.4	10.788	sunny
1	07-05-2020	36.7	NaN	sunny
2	08-05-2020	NaN	6.880	NaN
3	NaN	30.4	NaN	cloudy
4	10-05-2020	NaN	19.055	rainy

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22 entries, 0 to 21
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Date         18 non-null     object
 1   Temperature  16 non-null     float64
 2   Windspeed    16 non-null     float64
 3   Status       16 non-null     object
dtypes: float64(2), object(2)
memory usage: 832.0+ bytes
```

In [4]:
```
data.isnull().sum()
```

Out[4]:
```
Date           4
Temperature    6
Windspeed      6
Status         6
dtype: int64
```

Filling a common value to all missing data

Let's try filling 0 to all the missing data

In [5]:
```
data.fillna(0)
```

Out[5]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.4	10.788	sunny
1	07-05-2020	36.7	0.000	sunny
2	08-05-2020	0.0	6.880	0
3	0	30.4	0.000	cloudy
4	10-05-2020	0.0	19.055	rainy
5	11-05-2020	24.2	13.900	sunny
6	12-05-2020	22.7	0.000	0
7	13-05-2020	35.4	10.788	sunny
8	0	26.3	8.658	0
9	15-05-2020	22.4	14.884	sunny
10	0	0.0	0.000	sunny
11	17-05-2020	23.5	16.555	rainy
12	18-05-2020	0.0	25.664	0
13	19-05-2020	30.2	14.258	cloudy
14	20-05-2020	0.0	17.256	rainy
15	0	33.4	0.000	sunny
16	22-05-2020	0.0	12.222	0
17	23-05-2020	28.5	13.525	sunny
18	24-05-2020	27.6	5.899	0
19	25-05-2020	26.7	0.000	sunny
20	26-05-2020	34.4	16.258	rainy
21	27-05-2020	33.9	14.363	cloudy

Adding missing data to individual columns

The same method can be used to add missing data for various columns differently. We just need to pass a dictionary as below.

In [6]:
```
data.fillna({
    "Temperature":0,
    "Windspeed":5,
    "Status":"sunny"
})
```

Out[6]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.4	10.788	sunny
1	07-05-2020	36.7	5.000	sunny
2	08-05-2020	0.0	6.880	sunny
3	NaN	30.4	5.000	cloudy
4	10-05-2020	0.0	19.055	rainy
5	11-05-2020	24.2	13.900	sunny
6	12-05-2020	22.7	5.000	sunny
7	13-05-2020	35.4	10.788	sunny
8	NaN	26.3	8.658	sunny
9	15-05-2020	22.4	14.884	sunny
10	NaN	0.0	5.000	sunny

Forward fill (row)

Forward fill is a method to forward the data from the row above the missing value. Thus all the missing value will get filled with the value above. If there are multiple missing values consecutively , they will also get filled with the same value of the above available data.

In [7]:
```
data.fillna(method="ffill")
```

Out[7]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.4	10.788	sunny
1	07-05-2020	36.7	10.788	sunny
2	08-05-2020	36.7	6.880	sunny
3	08-05-2020	30.4	6.880	cloudy
4	10-05-2020	30.4	19.055	rainy
5	11-05-2020	24.2	13.900	sunny
6	12-05-2020	22.7	13.900	sunny
7	13-05-2020	35.4	10.788	sunny
8	13-05-2020	26.3	8.658	sunny
9	15-05-2020	22.4	14.884	sunny
10	15-05-2020	22.4	14.884	sunny
11	17-05-2020	23.5	16.555	rainy
12	18-05-2020	23.5	25.664	rainy
13	19-05-2020	30.2	14.258	cloudy
14	20-05-2020	30.2	17.256	rainy
15	20-05-2020	33.4	17.256	sunny
16	22-05-2020	33.4	12.222	sunny
17	23-05-2020	28.5	13.525	sunny
18	24-05-2020	27.6	5.899	sunny
19	25-05-2020	26.7	5.899	sunny
20	26-05-2020	34.4	16.258	rainy
21	27-05-2020	33.9	14.363	cloudy

Backward fill (row)

Similar to that for forward fill, backward fill also fills the data bus as the name suggest, this fills the data from back, i.e. from bottom.

So the missing data will be filled from the existing data below

In [8]:
```
data.fillna(method='bfill')
```

Out[8]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.4	10.788	sunny
1	07-05-2020	36.7	6.880	sunny
2	08-05-2020	30.4	6.880	cloudy
3	10-05-2020	30.4	19.055	cloudy
4	10-05-2020	24.2	19.055	rainy
5	11-05-2020	24.2	13.900	sunny
6	12-05-2020	22.7	10.788	sunny
7	13-05-2020	35.4	10.788	sunny
8	15-05-2020	26.3	8.658	sunny
9	15-05-2020	22.4	14.884	sunny
10	17-05-2020	23.5	16.555	sunny
11	17-05-2020	23.5	16.555	rainy
12	18-05-2020	30.2	25.664	cloudy
13	19-05-2020	30.2	14.258	cloudy
14	20-05-2020	33.4	17.256	rainy
15	22-05-2020	33.4	12.222	sunny
16	22-05-2020	28.5	12.222	sunny
17	23-05-2020	28.5	13.525	sunny
18	24-05-2020	27.6	5.899	sunny
19	25-05-2020	26.7	16.258	sunny
20	26-05-2020	34.4	16.258	rainy
21	27-05-2020	33.9	14.363	cloudy

Limiting the forward/backward fill

We can limit the number of rows or columns getting filled.

In [9]:
```
data.fillna(method="ffill", limit=1)
```

Out[9]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.4	10.788	sunny
1	07-05-2020	36.7	10.788	sunny
2	08-05-2020	36.7	6.880	sunny
3	08-05-2020	30.4	6.880	cloudy
4	10-05-2020	30.4	19.055	rainy
5	11-05-2020	24.2	13.900	sunny
6	12-05-2020	22.7	13.900	sunny
7	13-05-2020	35.4	10.788	sunny
8	13-05-2020	26.3	8.658	sunny
9	15-05-2020	22.4	14.884	sunny
10	15-05-2020	22.4	14.884	sunny
11	17-05-2020	23.5	16.555	rainy
12	18-05-2020	23.5	25.664	rainy
13	19-05-2020	30.2	14.258	cloudy
14	20-05-2020	30.2	17.256	rainy
15	20-05-2020	33.4	17.256	sunny
16	22-05-2020	33.4	12.222	sunny
17	23-05-2020	28.5	13.525	sunny
18	24-05-2020	27.6	5.899	sunny
19	25-05-2020	26.7	5.899	sunny
20	26-05-2020	34.4	16.258	rainy
21	27-05-2020	33.9	14.363	cloudy

Filling with Pandas objects

There are many Pandas objects like df.sum(), df.max(), etc. we can fill the missing values with these too

In [10]:
```
data.fillna(data.mean())
```

Out[10]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.40000	10.788000	sunny
1	07-05-2020	36.70000	13.809563	sunny
2	08-05-2020	29.48125	6.880000	NaN
3	NaN	30.40000	13.809563	cloudy
4	10-05-2020	29.48125	19.055000	rainy
5	11-05-2020	24.20000	13.900000	sunny
6	12-05-2020	22.70000	13.809563	NaN
7	13-05-2020	35.40000	10.788000	sunny
8	NaN	26.30000	8.658000	NaN
9	15-05-2020	22.40000	14.884000	sunny
10	NaN	29.48125	13.809563	sunny

Filling for specific range of columns

We can do filling for a specific range of column too as:

In [11]:
```
data.fillna(data.mean()['Temperature':'Windspeed'])
```

Out[11]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.40000	10.788000	sunny
1	07-05-2020	36.70000	13.809563	sunny
2	08-05-2020	29.48125	6.880000	NaN
3	NaN	30.40000	13.809563	cloudy
4	10-05-2020	29.48125	19.055000	rainy
5	11-05-2020	24.20000	13.900000	sunny
6	12-05-2020	22.70000	13.809563	NaN
7	13-05-2020	35.40000	10.788000	sunny
8	NaN	26.30000	8.658000	NaN
9	15-05-2020	22.40000	14.884000	sunny
10	NaN	29.48125	13.809563	sunny

Interpolate missing value

We can interpolate missing values based on different methods. This is done by an object in DataFrame as interpolate() . By default, interpolate() does linear interpolation.

In [12]:
```
data.interpolate()
```

Out[12]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.40	10.7880	sunny
1	07-05-2020	36.70	8.8340	sunny
2	08-05-2020	33.55	6.8800	NaN
3	NaN	30.40	12.9675	cloudy
4	10-05-2020	27.30	19.0550	rainy
5	11-05-2020	24.20	13.9000	sunny
6	12-05-2020	22.70	12.3440	NaN
7	13-05-2020	35.40	10.7880	sunny
8	NaN	26.30	8.6580	NaN
9	15-05-2020	22.40	14.8840	sunny
10	NaN	22.95	15.7195	sunny
11	17-05-2020	23.50	16.5550	rainy
12	18-05-2020	26.85	25.6640	NaN
13	19-05-2020	30.20	14.2580	cloudy
14	20-05-2020	31.80	17.2560	rainy
15	NaN	33.40	14.7390	sunny
16	22-05-2020	30.95	12.2220	NaN
17	23-05-2020	28.50	13.5250	sunny
18	24-05-2020	27.60	5.8990	NaN
19	25-05-2020	26.70	11.0785	sunny
20	26-05-2020	34.40	16.2580	rainy
21	27-05-2020	33.90	14.3630	cloudy

Time interpolate

data.interpolate(method="time")

Different interpolations: Linear interpolation, Barycentric interpolation, Pchip interpolation, Akima interpolation, Spline interpolation, Polynomial interpolation

Interpolation direction

Similar to ffill and bfill interpolation can also be directed.

In [13]:
```
data.interpolate(limit=1, limit_direction='backward')
```

Out[13]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.40	10.7880	sunny
1	07-05-2020	36.70	8.8340	sunny
2	08-05-2020	33.55	6.8800	NaN
3	NaN	30.40	12.9675	cloudy
4	10-05-2020	27.30	19.0550	rainy
5	11-05-2020	24.20	13.9000	sunny
6	12-05-2020	22.70	12.3440	NaN
7	13-05-2020	35.40	10.7880	sunny
8	NaN	26.30	8.6580	NaN
9	15-05-2020	22.40	14.8840	sunny
10	NaN	22.95	15.7195	sunny
11	17-05-2020	23.50	16.5550	rainy
12	18-05-2020	26.85	25.6640	NaN
13	19-05-2020	30.20	14.2580	cloudy
14	20-05-2020	31.80	17.2560	rainy
15	NaN	33.40	14.7390	sunny
16	22-05-2020	30.95	12.2220	NaN
17	23-05-2020	28.50	13.5250	sunny
18	24-05-2020	27.60	5.8990	NaN
19	25-05-2020	26.70	11.0785	sunny
20	26-05-2020	34.40	16.2580	rainy
21	27-05-2020	33.90	14.3630	cloudy

Limit area of interpolation

We can also restrict our missing value to be filled with inside or outside values

Inside

In [14]:
```
data.interpolate(limit_direction='both', limit_area='inside', limit=1)
```

Out[14]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.40	10.7880	sunny
1	07-05-2020	36.70	8.8340	sunny
2	08-05-2020	33.55	6.8800	NaN
3	NaN	30.40	12.9675	cloudy
4	10-05-2020	27.30	19.0550	rainy
5	11-05-2020	24.20	13.9000	sunny
6	12-05-2020	22.70	12.3440	NaN
7	13-05-2020	35.40	10.7880	sunny
8	NaN	26.30	8.6580	NaN
9	15-05-2020	22.40	14.8840	sunny
10	NaN	22.95	15.7195	sunny
11	17-05-2020	23.50	16.5550	rainy
12	18-05-2020	26.85	25.6640	NaN
13	19-05-2020	30.20	14.2580	cloudy
14	20-05-2020	31.80	17.2560	rainy
15	NaN	33.40	14.7390	sunny
16	22-05-2020	30.95	12.2220	NaN
17	23-05-2020	28.50	13.5250	sunny
18	24-05-2020	27.60	5.8990	NaN
19	25-05-2020	26.70	11.0785	sunny
20	26-05-2020	34.40	16.2580	rainy
21	27-05-2020	33.90	14.3630	cloudy

Outside

In [15]:
```
data.interpolate(limit_direction='both', limit_area='outside', limit=1)
```

Out[15]:

	Date	Temperature	Windspeed	Status
0	06-05-2020	35.4	10.788	sunny
1	07-05-2020	36.7	NaN	sunny
2	08-05-2020	NaN	6.880	NaN
3	NaN	30.4	NaN	cloudy
4	10-05-2020	NaN	19.055	rainy
5	11-05-2020	24.2	13.900	sunny
6	12-05-2020	22.7	NaN	NaN
7	13-05-2020	35.4	10.788	sunny
8	NaN	26.3	8.658	NaN
9	15-05-2020	22.4	14.884	sunny
10	NaN	NaN	NaN	sunny
11	17-05-2020	23.5	16.555	rainy
12	18-05-2020	NaN	25.664	NaN
13	19-05-2020	30.2	14.258	cloudy
14	20-05-2020	NaN	17.256	rainy
15	NaN	33.4	NaN	sunny
16	22-05-2020	NaN	12.222	NaN
17	23-05-2020	28.5	13.525	sunny
18	24-05-2020	27.6	5.899	NaN
19	25-05-2020	26.7	NaN	sunny
20	26-05-2020	34.4	16.258	rainy
21	27-05-2020	33.9	14.363	cloudy