Data Cleaning Day 2

July 28, 2023

[1]:
```
import numpy as np
import pandas as pd
```

[2]:
```
df=pd.read_csv("C:\\Users\\ASUS\\Desktop\\DS Course\\Data Cleaning\\iris - Missing.csv")
```

[3]:
```
df.head(15)
```

[3]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	NaN	NaN	NaN	NaN	NaN
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	NaN	setosa
8	4.4	2.9	1.4	NaN	NaN
9	4.9	3.1	1.5	NaN	setosa
10	NaN	3.7	1.5	NaN	setosa
11	4.8	3.4	1.6	NaN	setosa
12	4.8	3.0	1.4	NaN	setosa
13	4.3	3.0	1.1	NaN	setosa
14	NaN	NaN	NaN	NaN	setosa

[4]:
```
df.drop("Petal.Width",axis=1,inplace=True) #delete the Petal.Width column because it has more missing data
df
```

[4]:

	Sepal.Length	Sepal.Width	Petal.Length	Species
0	5.1	3.5	1.4	setosa
1	4.9	3.0	1.4	setosa
2	NaN	NaN	NaN	NaN
3	4.6	3.1	1.5	setosa
4	5.0	3.6	1.4	setosa
..	…	…	…	…
145	6.7	3.0	5.2	virginica
146	6.3	2.5	5.0	virginica

```
147           6.5          3.0           5.2  virginica
148           6.2          3.4           5.4  virginica
149           5.9          3.0           5.1  virginica

[150 rows x 4 columns]
```

[ ]:

[ ]:

[5]:
```
#Forward Filling
```

[ ]:

[6]:
```
df.fillna(method="ffill")       #check 2nd row, all the missing data will␣
↪filled with the previous cell data
```

[6]:
```
     Sepal.Length  Sepal.Width  Petal.Length    Species
0             5.1          3.5           1.4     setosa
1             4.9          3.0           1.4     setosa
2             4.9          3.0           1.4     setosa
3             4.6          3.1           1.5     setosa
4             5.0          3.6           1.4     setosa
..            …            …             …       …
145           6.7          3.0           5.2  virginica
146           6.3          2.5           5.0  virginica
147           6.5          3.0           5.2  virginica
148           6.2          3.4           5.4  virginica
149           5.9          3.0           5.1  virginica

[150 rows x 4 columns]
```

[8]:
```
df.fillna(method="pad")
```

[8]:
```
     Sepal.Length  Sepal.Width  Petal.Length    Species
0             5.1          3.5           1.4     setosa
1             4.9          3.0           1.4     setosa
2             4.9          3.0           1.4     setosa
3             4.6          3.1           1.5     setosa
4             5.0          3.6           1.4     setosa
..            …            …             …       …
145           6.7          3.0           5.2  virginica
146           6.3          2.5           5.0  virginica
147           6.5          3.0           5.2  virginica
148           6.2          3.4           5.4  virginica
149           5.9          3.0           5.1  virginica
```

```
[150 rows x 4 columns]
```

[ ]:

[ ]:

[9]:
```
#Backword Filling
```

[ ]:

[10]:
```
df.fillna(method="bfill")   #check 2nd row, all the missing data will filled
 ↪with the next cell data
```

[10]:

	Sepal.Length	Sepal.Width	Petal.Length	Species
0	5.1	3.5	1.4	setosa
1	4.9	3.0	1.4	setosa
2	4.6	3.1	1.5	setosa
3	4.6	3.1	1.5	setosa
4	5.0	3.6	1.4	setosa
..	…	…	…	…
145	6.7	3.0	5.2	virginica
146	6.3	2.5	5.0	virginica
147	6.5	3.0	5.2	virginica
148	6.2	3.4	5.4	virginica
149	5.9	3.0	5.1	virginica

```
[150 rows x 4 columns]
```

[ ]:

[ ]:

[15]:
```
df1=df.iloc[:,:3]     #select all the rows and first 3 columns from df data
 ↪frame
df1.head(15)
```

[15]:

	Sepal.Length	Sepal.Width	Petal.Length
0	5.1	3.5	1.4
1	4.9	3.0	1.4
2	NaN	NaN	NaN
3	4.6	3.1	1.5
4	5.0	3.6	1.4
5	5.4	3.9	1.7
6	4.6	3.4	1.4
7	5.0	3.4	1.5
8	4.4	2.9	1.4
9	4.9	3.1	1.5
10	NaN	3.7	1.5

```
11           4.8          3.4           1.6
12           4.8          3.0           1.4
13           4.3          3.0           1.1
14           NaN          NaN           NaN
```

[14]:
```
#linear interpolation method

df1.interpolate(method="linear")   #2nd row 1 st col cell's answer = (4.60 - 4.
↪90)/2 + 4.90 = 4.75
```

[14]:
```
     Sepal.Length  Sepal.Width  Petal.Length
0             5.10         3.50          1.40
1             4.90         3.00          1.40
2             4.75         3.05          1.45
3             4.60         3.10          1.50
4             5.00         3.60          1.40
..             …            …             …
145           6.70         3.00          5.20
146           6.30         2.50          5.00
147           6.50         3.00          5.20
148           6.20         3.40          5.40
149           5.90         3.00          5.10

[150 rows x 3 columns]
```

[ ]: