```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler


path = "/content/drive/MyDrive/dataset/01.Data Cleaning and Preprocessing.csv"
df = pd.read_csv(path)
df.head(5)
```

	Observation	Y-Kappa	ChipRate	BF-CMratio	BlowFlow	ChipLevel4	T-upperExt-2	T-lowerExt-2
0	31-00:00	23.10	16.520	121.717	1177.607	169.805	358.282	329.545
1	31-01:00	27.60	16.810	79.022	1328.360	341.327	351.050	329.067
2	31-02:00	23.19	16.709	79.562	1329.407	239.161	350.022	329.260
3	31-03:00	23.60	16.478	81.011	1334.877	213.527	350.938	331.142
4	31-04:00	22.90	15.618	93.244	1334.168	243.131	351.640	332.709

5 rows × 23 columns

DATA UNDERSTANDING

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 324 entries, 0 to 323
Data columns (total 23 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Observation      324 non-null    object 
 1   Y-Kappa          324 non-null    float64
 2   ChipRate         319 non-null    float64
 3   BF-CMratio       307 non-null    float64
 4   BlowFlow         308 non-null    float64
 5   ChipLevel4       323 non-null    float64
 6   T-upperExt-2     322 non-null    float64
 7   T-lowerExt-2     322 non-null    float64
 8   UCZAA            299 non-null    float64
 9   WhiteFlow-4      323 non-null    float64
 10  AAWhiteSt-4      173 non-null    float64
 11  AA-Wood-4        323 non-null    float64
 12  ChipMoisture-4   323 non-null    float64
 13  SteamFlow-4      323 non-null    float64
 14  Lower-HeatT-3    322 non-null    float64
 15  Upper-HeatT-3    322 non-null    float64
 16  ChipMass-4       323 non-null    float64
 17  WeakLiquorF      323 non-null    float64
 18  BlackFlow-2      322 non-null    float64
 19  WeakWashF        323 non-null    float64
 20  SteamHeatF-3     322 non-null    float64
 21  T-Top-Chips-4    323 non-null    float64
 22  SulphidityL-4    173 non-null    float64
dtypes: float64(22), object(1)
memory usage: 58.3+ KB
```

```
df.describe()
```

	Y-Kappa	ChipRate	BF-CMratio	BlowFlow	ChipLevel4	T-upperExt-2	lowerE
count	324.000000	319.000000	307.000000	308.000000	323.000000	322.000000	322.000(
mean	20.635370	14.347937	87.464456	1237.837614	258.164483	356.904295	324.020
std	3.070036	1.499095	7.995012	100.593735	87.987452	9.209290	7.621
min	12.170000	9.983000	68.645000	0.000000	0.000000	339.168000	284.633(
25%	18.382500	13.358000	81.823000	1193.215250	213.527000	350.241250	321.420(
50%	20.845000	14.308000	86.739000	1273.138500	271.792000	356.843000	325.669(
75%	23.032500	15.517000	92.372000	1289.196000	321.680000	362.242250	329.175(
max	27.600000	16.958000	121.717000	1351.240000	419.014000	399.135000	337.012(

8 rows × 22 columns

HANDLING MISSING VALUES

```
df.notnull().sum()
```

```
Observation        324
Y-Kappa            324
ChipRate           319
BF-CMratio         307
BlowFlow           308
ChipLevel4         323
T-upperExt-2       322
T-lowerExt-2       322
UCZAA              299
WhiteFlow-4        323
AAWhiteSt-4        173
AA-Wood-4          323
ChipMoisture-4     323
SteamFlow-4        323
Lower-HeatT-3      322
Upper-HeatT-3      322
ChipMass-4         323
WeakLiquorF        323
BlackFlow-2        322
WeakWashF          323
SteamHeatF-3       322
T-Top-Chips-4      323
SulphidityL-4      173
dtype: int64
```

```
df.fillna(value = 0)
```

	Observation	Y-Kappa	ChipRate	BF-CMratio	BlowFlow	ChipLevel4	T-upperExt-2	T lowerExt
0	31-00:00	23.10	16.520	121.717	1177.607	169.805	358.282	329.54
1	31-01:00	27.60	16.810	79.022	1328.360	341.327	351.050	329.06
2	31-02:00	23.19	16.709	79.562	1329.407	239.161	350.022	329.26
3	31-03:00	23.60	16.478	81.011	1334.877	213.527	350.938	331.14
4	31-04:00	22.90	15.618	93.244	1334.168	243.131	351.640	332.70
...	...	...	...	...	...	...	...	.
319	10-16:00	23.75	12.667	93.450	1178.252	276.955	347.286	310.97
320	9-19:00	19.80	12.558	94.352	1184.119	297.071	399.135	319.57
321	9-20:00	23.01	12.550	90.842	1188.517	289.826	373.633	314.59
322	9-21:00	24.32	13.083	88.910	1192.879	318.006	364.081	308.55
323	9-22:00	25.75	13.417	85.451	1186.342	248.312	356.289	310.48

324 rows × 23 columns

DROPPING DUPLICATE VALUES

```
df.drop_duplicates(inplace=True)
print(df)
```

```
    Observation  Y-Kappa  ChipRate  BF-CMratio  BlowFlow  ChipLevel4   \
0      31-00:00    23.10    16.520     121.717  1177.607      169.805
1      31-01:00    27.60    16.810      79.022  1328.360      341.327
2      31-02:00    23.19    16.709      79.562  1329.407      239.161
3      31-03:00    23.60    16.478      81.011  1334.877      213.527
4      31-04:00    22.90    15.618      93.244  1334.168      243.131
..          ...      ...       ...         ...       ...          ...
298    12-09:00    20.90    15.167      84.640  1283.706      339.440
299    12-10:00    24.98       NaN      85.034  1278.345      368.564
300    12-11:00    21.00       NaN      88.013  1307.722      278.842
301    12-12:00    21.40       NaN      85.490  1255.986      273.484
307    31-05:00    20.89    14.308      94.172  1327.832      251.120

     T-upperExt-2   T-lowerExt-2    UCZAA  WhiteFlow-4   ...  SteamFlow-4   \
0          358.282        329.545  1.443       599.253  ...        67.122
1          351.050        329.067  1.549       537.201  ...        60.012
2          350.022        329.260  1.600       549.611  ...        61.304
3          350.938        331.142  1.604       623.362  ...        68.496
4          351.640        332.709    NaN       638.672  ...        70.022
..             ...            ...    ...           ...  ...           ...
298        354.803        311.041  1.635       532.419  ...        65.561
299        357.723        321.387    NaN       520.365  ...        65.729
300        357.438        323.757    NaN       553.070  ...        65.795
301        361.365        322.689    NaN       590.199  ...        71.456
307        351.263        332.485  1.522       631.514  ...        71.286

     Lower-HeatT-3  Upper-HeatT-3   ChipMass-4   WeakLiquorF   BlackFlow-2   \
0          329.432        303.099      175.964      1127.197      1319.039
1          330.823        304.879      163.202       665.975      1297.317
2          329.140        303.383      164.013       677.534      1327.072
3          328.875        302.254      181.487       767.853      1324.461
4          328.352        300.954      183.929       888.448      1343.424
..             ...            ...          ...           ...           ...
298        332.924        307.626      145.299       832.906      1344.708
299        332.523        307.169      151.544       905.639      1344.469
300        331.263        306.400      157.954       908.691      1344.588
301        333.032        308.732      174.069       986.206      1348.747
307        328.699        300.706      180.229       903.605      1323.082

     WeakWashF   SteamHeatF-3   T-Top-Chips-4   SulphidityL-4
0       257.325         54.612         252.077             NaN
1       241.182         46.603         251.406           29.11
2       237.272         51.795         251.335             NaN
3       239.478         54.846         250.312           29.02
4       215.372         54.186         249.916           29.01
..          ...            ...             ...             ...
298     388.911         49.524         251.833           30.29
299     418.979         48.135         251.614           30.47
300     462.712         54.373         251.197             NaN
301     457.313         53.194         251.324           30.46
307     232.729         54.503         250.084             NaN

[301 rows x 23 columns]
```

> Add blockquote

Generate | print hello world using rot13 | Close

```
df.isnull().sum()
```

```
Observation          0
Y-Kappa              0
ChipRate             4
BF-CMratio          14
BlowFlow            13
ChipLevel4           1
T-upperExt-2         1
T-lowerExt-2         1
UCZAA               24
WhiteFlow-4          1
AAWhiteSt-4        141
AA-Wood-4            1
ChipMoisture-4       1
SteamFlow-4          1
Lower-HeatT-3        1
Upper-HeatT-3        1
ChipMass-4           1
WeakLiquorF          1
BlackFlow-2          1
```

```
WeakWashF           1
SteamHeatF-3        1
T-Top-Chips-4       1
SulphidityL-4     141
dtype: int64
```

```
df.isnull().sum().sum()
```

```
352
```

```
numeric_data = df.select_dtypes(include=['number'])

data = numeric_data.fillna(numeric_data.mean())
data
```

	Y-Kappa	ChipRate	BF-CMratio	BlowFlow	ChipLevel4	T-upperExt-2	T-lowerExt-2	UCZAA	W
0	23.10	16.52000	121.717	1177.607	169.805	358.282	329.545	1.443000	
1	27.60	16.81000	79.022	1328.360	341.327	351.050	329.067	1.549000	
2	23.19	16.70900	79.562	1329.407	239.161	350.022	329.260	1.600000	
3	23.60	16.47800	81.011	1334.877	213.527	350.938	331.142	1.604000	
4	22.90	15.61800	93.244	1334.168	243.131	351.640	332.709	1.490588	
...	...	...	...	...	...	...	...	...	
298	20.90	15.16700	84.640	1283.706	339.440	354.803	311.041	1.635000	
299	24.98	14.33867	85.034	1278.345	368.564	357.723	321.387	1.490588	
300	21.00	14.33867	88.013	1307.722	278.842	357.438	323.757	1.490588	
301	21.40	14.33867	85.490	1255.986	273.484	361.365	322.689	1.490588	
307	20.89	14.30800	94.172	1327.832	251.120	351.263	332.485	1.522000	

301 rows × 22 columns

```
import numpy as np
```

```
df.columns
```

```
Index(['Observation', 'Y-Kappa', 'ChipRate', 'BF-CMratio', 'BlowFlow',
       'ChipLevel4 ', 'T-upperExt-2 ', 'T-lowerExt-2  ', 'UCZAA',
       'WhiteFlow-4 ', 'AAWhiteSt-4 ', 'AA-Wood-4  ', 'ChipMoisture-4 ',
       'SteamFlow-4 ', 'Lower-HeatT-3', 'Upper-HeatT-3 ', 'ChipMass-4 ',
       'WeakLiquorF ', 'BlackFlow-2 ', 'WeakWashF ', 'SteamHeatF-3 ',
       'T-Top-Chips-4 ', 'SulphidityL-4 '],
      dtype='object')
```

```
Quantile1 = data.quantile(0.25)
```

Generate a slider using jupyter widgets Close

```
Quantile3 = data.quantile(0.75)

substract = Quantile1 - Quantile3

l_bound = Quantile1 -1.5 * substract
u_bound = Quantile3 +1.5 * substract

data = data[~((data < l_bound) | (data > u_bound).any(axis = 1))]
data
```

	Y-Kappa	ChipRate	BF-CMratio	BlowFlow	ChipLevel4	T-upperExt-2	T-lowerExt-2	UCZAA	Whi
0	NaN	NaN	121.717	NaN	NaN	NaN	NaN	NaN	
1	27.6	16.810	NaN	NaN	NaN	NaN	NaN	NaN	
2	NaN	16.709	NaN	NaN	NaN	NaN	NaN	NaN	
3	NaN	NaN	NaN	1334.877	NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	1334.168	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	
298	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.635	
299	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
300	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
301	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
307	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

301 rows × 22 columns

```
from sklearn.preprocessing import scale

data.describe()
```

	Y-Kappa	ChipRate	BF-CMratio	BlowFlow	ChipLevel4	T-upperExt-2	T-lowerExt-
count	14.000000	16.000000	27.000000	6.000000	18.000000	29.000000	11.00000(
mean	26.105714	16.747000	102.756037	1339.388667	387.192000	373.749517	333.90072
std	0.771758	0.102871	6.917063	7.629669	10.963711	7.330336	1.14118(
min	25.300000	16.542000	96.937000	1334.168000	373.726000	368.547000	332.95300(
25%	25.437500	16.683000	97.782500	1334.410500	379.892750	369.065000	333.31400(
50%	25.885000	16.725500	99.982000	1334.884500	385.974000	370.757000	333.61400(
75%	26.590000	16.819000	105.335000	1343.898000	391.234250	374.752000	333.75650(
max	27.600000	16.958000	121.717000	1351.240000	419.014000	399.135000	337.01200(

8 rows × 22 columns

```
data.matrix = data.values.reshape(-1,1)

from sklearn import preprocessing

result = preprocessing.MinMaxScaler(feature_range=(0,10))
s = result.fit_transform(data)
data
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_data.py:473: RuntimeWa
  data_min = np.nanmin(X, axis=0)
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_data.py:474: RuntimeWa
  data_max = np.nanmax(X, axis=0)
```

	Y-Kappa	ChipRate	BF-CMratio	BlowFlow	ChipLevel4	T-upperExt-2	T-lowerExt-2	UCZAA	Whi
0	NaN	NaN	121.717	NaN	NaN	NaN	NaN	NaN	
1	27.6	16.810	NaN	NaN	NaN	NaN	NaN	NaN	
2	NaN	16.709	NaN	NaN	NaN	NaN	NaN	NaN	

```
data.to_csv("/content/drive/MyDrive/dataset/01.Data Cleaning and Preprocessing.csv",index = False)
```

```
import matplotlib.pyplot as plt

plt.hist(data['BF-CMratio'], bins=20)
plt.title("Histogram BF-CMratio")
plt.xlabel("Values")
plt.ylabel("Frequency")
plt.show()
```