```
# load the required libraries
import pandas as pd

# give path to your dataset(csv) file
df = pd.read_csv(r'/iris.csv')

print(df)
```

```
     sepal.length  sepal.width  petal.length  petal.width    variety
0             5.1          3.5           1.4          0.2     Setosa
1             4.9          3.0           1.4          0.2     Setosa
2             4.7          3.2           1.3          0.2     Setosa
3             4.6          3.1           1.5          0.2     Setosa
4             5.0          3.6           1.4          0.2     Setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  Virginica
146           6.3          2.5           5.0          1.9  Virginica
147           6.5          3.0           5.2          2.0  Virginica
148           6.2          3.4           5.4          2.3  Virginica
149           5.9          3.0           5.1          1.8  Virginica

[150 rows x 5 columns]
```

```
df.isnull()
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

```
df.isnull().any()
```

```
sepal.length    False
```

```
sepal.width     False
petal.length    False
petal.width     False
variety         False
dtype: bool
```

```
# count Column wise missing value using isnull()
df.isnull().sum()
```

```
sepal.length    0
sepal.width     0
petal.length    0
petal.width     0
variety         0
dtype: int64
```

```
# count row wise missing value using isnull()
df.isnull().sum(axis=1)
```

```
0      0
1      0
2      0
3      0
4      0
      ..
145    0
146    0
147    0
148    0
149    0
Length: 150, dtype: int64
```

```
# count Not a Number Values
df.isna().sum()
```

```
sepal.length    0
sepal.width     0
petal.length    0
petal.width     0
variety         0
dtype: int64
```

```
# count of missing values of a specific column
df.variety.isnull().sum()
```

```
0
```

```
# groupby count of missing values of a column
df.groupby(['sepal.length'])['sepal.width'].apply(lambda x: x
```

```
sepal.length
4.3    0
```

```
4.4    0
4.5    0
4.6    0
4.7    0
4.8    0
4.9    0
5.0    0
5.1    0
5.2    0
5.3    0
5.4    0
5.5    0
5.6    0
5.7    0
5.8    0
5.9    0
6.0    0
6.1    0
6.2    0
6.3    0
6.4    0
6.5    0
6.6    0
6.7    0
6.8    0
6.9    0
7.0    0
7.1    0
7.2    0
7.3    0
7.4    0
7.6    0
7.7    0
7.9    0
Name: sepal.width, dtype: int64
```

```
# To check the datatype of each column
df.dtypes
```

```
sepal.length    float64
sepal.width     float64
petal.length    float64
petal.width     float64
variety          object
dtype: object
```

```
# To change the datatype (data type of petal.lengthchanged t
df['petal.length'] = df['petal.length'].astype("int")
df.dtypes
```

```
sepal.length    float64
sepal.width     float64
petal.length      int64
petal.width     float64
variety          object
dtype: object
```

```
# data normalization
from sklearn import preprocessing, datasets


iris = datasets.load_iris()



# load iris dataset into DataFrame Object df
df = pd.DataFrame(iris.data, columns=iris.feature_names)

# print first 5 values of dataframe
df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

```
x = df[['sepal length (cm)']].values.astype(float)

# Create a minimum and maximum processor object
min_max_scaler = preprocessing.MinMaxScaler()

# Create an object to transform the data to fit minmax proces
x_scaled = min_max_scaler.fit_transform(x)

# Run the normalizer on the dataframe
df_normalized = pd.DataFrame(x_scaled)

print(df_normalized)
```

```
             0
0     0.222222
1     0.166667
2     0.111111
3     0.083333
4     0.194444
..         ...
145   0.666667
```

```
146  0.555556
147  0.611111
148  0.527778
149  0.444444

[150 rows x 1 columns]
```

```
# iris dataset used for label encoding
df = pd.read_csv(r'/iris.csv')
```

```
from sklearn import preprocessing
```

```
# Observe the unique values for the variety column(of flowers
df['variety'].unique()
```

```
array(['Setosa', 'Versicolor', 'Virginica'], dtype=object)
```

```
# define label_encoder object knows how to understand word la
label_encoder = preprocessing.LabelEncoder()
```

```
# Encode labels in column 'variety'.
df['variety']= label_encoder.fit_transform(df['variety'])
```

```
# Observe the "label encoded" unique values for the variety c
df['variety'].unique()
```

```
array([0, 1, 2])
```

✓ 0s completed at 19:24