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

DATA CLEANING

```
df=pd.DataFrame(np.random.randn(5,3), index=['a','c','e','f','h'], columns=['one','two','three'])
#print(df)
df=df.reindex(['a','b','c','d','e','f','g','h'])
print(df)
```

```
        one       two     three
a  0.603798  0.349102  0.383779
b       NaN       NaN       NaN
c  0.381520  0.785157 -0.872481
d       NaN       NaN       NaN
e -1.357097 -0.406224  2.409487
f  0.315151  0.595545  1.056387
g       NaN       NaN       NaN
h -0.577192 -0.945180  0.406008
```

```
print(df['one'].isnull())
```

```
a    False
b     True
c    False
d     True
e    False
f    False
g     True
h    False
Name: one, dtype: bool
```

```
print("NaN replaced with 0: ")
print(df.fillna(0))
```

```
NaN replaced with 0:
        one       two     three
a -0.720837 -0.172031 -1.126040
b  0.000000  0.000000  0.000000
c  1.767687 -0.591742  1.469700
d  0.000000  0.000000  0.000000
e -0.839010  1.112248  1.111684
f  1.502390 -0.711953 -1.924154
g  0.000000  0.000000  0.000000
h  1.037822 -0.947633 -0.297461
```

```
print(df.fillna(method='pad'))
```

```
        one       two     three
a -0.720837 -0.172031 -1.126040
b -0.720837 -0.172031 -1.126040
c  1.767687 -0.591742  1.469700
d  1.767687 -0.591742  1.469700
e -0.839010  1.112248  1.111684
f  1.502390 -0.711953 -1.924154
g  1.502390 -0.711953 -1.924154
h  1.037822 -0.947633 -0.297461
```

```
print(df.fillna(method='bfill'))
```

```
        one       two     three
a -0.720837 -0.172031 -1.126040
b  1.767687 -0.591742  1.469700
c  1.767687 -0.591742  1.469700
d -0.839010  1.112248  1.111684
e -0.839010  1.112248  1.111684
f  1.502390 -0.711953 -1.924154
g  1.037822 -0.947633 -0.297461
h  1.037822 -0.947633 -0.297461
```

```
print(df.dropna())
```

```
        one       two     three
a -0.720837 -0.172031 -1.126040
c  1.767687 -0.591742  1.469700
e -0.839010  1.112248  1.111684
f  1.502390 -0.711953 -1.924154
h  1.037822 -0.947633 -0.297461
```

```
df = pd.DataFrame( {'one': [10, 20, 30, 40, 50, 2000], 'two': [1000, 0, 10, 40, 50, 60]})
print(df)
print(df.replace({1000: 20, 2000: 60}))
```

```
    one   two
0    10  1000
1    20     0
2    30    10
3    40    40
4    50    50
5  2000    60
   one  two
0   10   20
1   20    0
2   30   10
3   40   40
4   50   50
5   60   60
```

DATA PREPROCESSING

```
import pandas as pd
import numpy as np
df=pd.read_csv('/content/titanic.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
cols=['Name','Ticket','Cabin']
df=df.drop(cols, axis=1)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Sex          891 non-null    object 
 4   Age          714 non-null    float64
 5   SibSp        891 non-null    int64  
 6   Parch        891 non-null    int64  
 7   Fare         891 non-null    float64
 8   Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(2)
memory usage: 62.8+ KB
```

```
df=df.dropna()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 0 to 890
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  712 non-null    int64  
 1   Survived     712 non-null    int64  
 2   Pclass       712 non-null    int64  
 3   Sex          712 non-null    object 
 4   Age          712 non-null    float64
 5   SibSp        712 non-null    int64  
 6   Parch        712 non-null    int64  
 7   Fare         712 non-null    float64
 8   Embarked     712 non-null    object 
dtypes: float64(2), int64(5), object(2)
memory usage: 55.6+ KB
```

```
#creating dummy variables
dummies=[]
cols=['Pclass','Sex','Embarked']
for cols in cols:
  dummies.append(pd.get_dummies(df[cols]))
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 0 to 890
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  712 non-null    int64  
 1   Survived     712 non-null    int64  
 2   Pclass       712 non-null    int64  
 3   Sex          712 non-null    object 
 4   Age          712 non-null    float64
 5   SibSp        712 non-null    int64  
 6   Parch        712 non-null    int64  
 7   Fare         712 non-null    float64
 8   Embarked     712 non-null    object 
dtypes: float64(2), int64(5), object(2)
memory usage: 55.6+ KB
```

```
#transfer the 8th col
t_d=pd.concat(dummies, axis=1)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 0 to 890
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  712 non-null    int64  
 1   Survived     712 non-null    int64  
 2   Pclass       712 non-null    int64  
 3   Sex          712 non-null    object 
 4   Age          712 non-null    float64
 5   SibSp        712 non-null    int64  
 6   Parch        712 non-null    int64  
 7   Fare         712 non-null    float64
 8   Embarked     712 non-null    object 
```

```
dtypes: float64(2), int64(5), object(2)
memory usage: 55.6+ KB
```

```
#concatenate values with df
df=pd.concat((df,t_d),axis=1)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 0 to 890
Data columns (total 17 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  712 non-null    int64  
 1   Survived     712 non-null    int64  
 2   Pclass       712 non-null    int64  
 3   Sex          712 non-null    object 
 4   Age          712 non-null    float64
 5   SibSp        712 non-null    int64  
 6   Parch        712 non-null    int64  
 7   Fare         712 non-null    float64
 8   Embarked     712 non-null    object 
 9   1            712 non-null    uint8  
 10  2            712 non-null    uint8  
 11  3            712 non-null    uint8  
 12  female       712 non-null    uint8  
 13  male         712 non-null    uint8  
 14  C            712 non-null    uint8  
 15  Q            712 non-null    uint8  
 16  S            712 non-null    uint8  
dtypes: float64(2), int64(5), object(2), uint8(8)
memory usage: 61.2+ KB
```

```
#remove unwanted cols
df=df.drop(['Pclass','Sex','Embarked'], axis=1)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 0 to 890
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  712 non-null    int64  
 1   Survived     712 non-null    int64  
 2   Age          712 non-null    float64
 3   SibSp        712 non-null    int64  
 4   Parch        712 non-null    int64  
 5   Fare         712 non-null    float64
 6   1            712 non-null    uint8  
 7   2            712 non-null    uint8  
 8   3            712 non-null    uint8  
 9   female       712 non-null    uint8  
 10  male         712 non-null    uint8  
 11  C            712 non-null    uint8  
 12  Q            712 non-null    uint8  
 13  S            712 non-null    uint8  
dtypes: float64(2), int64(4), uint8(8)
memory usage: 44.5 KB
```

Take care of missing data

```
#rereplacing all the missing values
df['Age'] = df['Age'].interpolate()
print(df)
```

```
     PassengerId  Survived   Age  SibSp  Parch     Fare  1  2  3  female  \
0              1         0  22.0      1      0   7.2500  0  0  1       0   
1              2         1  38.0      1      0  71.2833  1  0  0       1   
2              3         1  26.0      0      0   7.9250  0  0  1       1   
3              4         1  35.0      1      0  53.1000  1  0  0       1   
4              5         0  35.0      0      0   8.0500  0  0  1       0   
..           ...       ...   ...    ...    ...      ... .. .. ..     ...   
885          886         0  39.0      0      5  29.1250  0  0  1       1   
886          887         0  27.0      0      0  13.0000  0  1  0       0   
887          888         1  19.0      0      0  30.0000  1  0  0       1   
889          890         1  26.0      0      0  30.0000  1  0  0       0   
890          891         0  32.0      0      0   7.7500  0  0  1       0   

     male  C  Q  S  
0       1  0  0  1  
1       0  1  0  0  
2       0  0  0  1  
3       0  0  0  1  
4       1  0  0  1  
..    ... .. .. ..  
885     0  0  1  0  
886     1  0  0  1  
887     0  0  0  1  
889     1  1  0  0  
890     1  0  1  0  

[712 rows x 14 columns]
```

MinMaxScaler and standardisation

```
from sklearn.preprocessing import MinMaxScaler
data = np.array([[-1,2],[-0.5,6],[0,10],[1,18]])
scaler=MinMaxScaler()
print(scaler.fit(data))
MinMaxScaler()
print(scaler.data_max_)
scaler.fit_transform(data)
```

```
MinMaxScaler()
[ 1. 18.]
```

```
array([[0.  , 0.  ],
       [0.25, 0.25],
       [0.5 , 0.5 ],
       [1.  , 1.  ]])
```

```
from numpy import asarray
from sklearn.preprocessing import StandardScaler
data=asarray([[100,0.01],[0,0.05],[50,0.005],[88,0.07],[4,0.01]])
print(data)
scaler=StandardScaler()
scaled=scaler.fit_transform(data)
print(scaled)
```

```
[[1.0e+02 1.0e-02]
 [0.0e+00 5.0e-02]
 [5.0e+01 5.0e-03]
 [8.8e+01 7.0e-02]
 [4.0e+00 1.0e-02]]
[[ 1.24802352 -0.72648316]
 [-1.17062671  0.80295507]
 [ 0.0386984  -0.91766294]
 [ 0.95778549  1.56767418]
 [-1.0738807  -0.72648316]]
```