Handling Numerical and Categorial Data

In [220]:

```python
#Rescaling a feature
#You need to rescale the values of a numerical feature to be between two values.
import pandas as pd
import numpy as np
from sklearn import preprocessing

#csv file
url = url = "C:/Users/Prerna/Desktop/ML_jupyter_notenooks/datasets/auto-mpg.csv"
df = pd.read_csv(url)

#Creating a dataframe
dataframe = pd.read_csv(url).fillna(0)
feature= dataframe[['mpg']]

minmax_scale = preprocessing.MinMaxScaler(feature_range=(0, 50))
scale_feature = minmax_scale.fit_transform(feature)
print(scale_feature)

print("Mean:",round(scale_feature.mean()))
print("Standard Deviation:",round(scale_feature.std()))
```

```
[[ 6.64893617]
 [ 6.64893617]
 [ 7.9787234 ]
 [ 6.64893617]
 [ 7.9787234 ]
 [ 9.30851064]
 [ 6.64893617]
 [ 6.64893617]
 [ 7.9787234 ]
 [ 7.9787234 ]
 [10.63829787]
 [ 2.65957447]
 [ 2.65957447]
 [ 3.9893617 ]
 [ 5.31914894]
 [ 5.31914894]
 [ 7.9787234 ]
 [11.96808511]
 [ 9.30851064]
```

In [221]:

```
#Standardizing a Feature
#You want to transform a feature to have a mean of 0 and a standard deviation of 1.

import pandas as pd
import numpy as np
from sklearn import preprocessing

#csv file
url = url = "C:/Users/Prerna/Desktop/ML_jupyter_notenooks/datasets/auto-mpg.csv"

#Creating a dataframe
dataframe = pd.read_csv(url).fillna(0)

feature= dataframe[['mpg']]

#create Scaler
standard_scale = preprocessing.StandardScaler()

#Transform the Feature
scale_feature = standard_scale.fit_transform(feature)

print(scale_feature)
print("Mean:",round(scale_feature.mean()))
print("Standard Deviation:",round(scale_feature.std()))
```

```
[[-1.21885460e+00]
 [-1.21885460e+00]
 [-1.09075062e+00]
 [-1.21885460e+00]
 [-1.09075062e+00]
 [-9.62646649e-01]
 [-1.21885460e+00]
 [-1.21885460e+00]
 [-1.09075062e+00]
 [-1.09075062e+00]
 [-8.34542675e-01]
 [-1.60316652e+00]
 [-1.60316652e+00]
 [-1.47506255e+00]
 [-1.34695857e+00]
 [-1.34695857e+00]
```

```
[-1.09075062e+00]
[-7.06438701e-01]
[-9.62646649e-01]
```

In [222]:

```
#Transforming Features
#You want to make a custom transformation to one or more features.

import pandas as pd
import numpy as np
from sklearn.preprocessing import FunctionTransformer

#csv file
url = url = "C:/Users/Prerna/Desktop/ML_jupyter_notenooks/datasets/auto-mpg.csv"

#Creating a dataframe
dataframe = pd.read_csv(url).fillna(0)

feature= dataframe[['cylinders']]

# Define a simple function
def add_ten(x):
    return x + 10

# Create transformer
ten_transformer = FunctionTransformer(add_ten, validate=True)

# Transform feature matrix
ten_transformer.transform(feature)
```

Out[222]:

```
array([[18],
       [18],
       [18],
       [18],
       [18],
       [18],
       [18],
       [18],
       [18],
       [18],
       [18],
       [18],
       [18],
       [18],
```

```
[18],
[18],
[18],
[18],
[18],
[18],
```

In [223]:

```
#Detecting Outliers with Quartiles

import pandas as pd
import numpy as np

#csv file
url = url = "C:/Users/Prerna/Desktop/ML_jupyter_notenooks/datasets/auto-mpg.csv"

#Creating a dataframe
dataframe = pd.read_csv(url).fillna(0)
anomalies = []
index_val = []
feature_acc = dataframe["acceleration"]

#calculating Q1,Q3,IQR
n = len(dataframe)
q1_val = round((n+1)/4)
q3_val = round(3*(q1_val))
q1 = feature_acc[q1_val]
q3 = feature_acc[q3_val]
iqr = q3 - q1
print("Quartile Q1 is:",q1)
print("Quartile Q3 is:",q3)
print("Inter Quartile Range is:",iqr)
lower_bound = q1 - (iqr * 1.5)
upper_bound = q3 + (iqr * 1.5)
print("The upper bound value is:",upper_bound)
print("The lower bound value is:",lower_bound)

# Generate outliers
for indexval,outlier in enumerate(feature_acc):
    if outlier > upper_bound or outlier < lower_bound:
            index_val.append(indexval)
            anomalies.append(outlier)

print("Outlier Values are:")
print(anomalies)
print("Outlier values indexes are:")
print(index_val)


```

```
Quartile Q1 is: 13.9
Quartile Q3 is: 17.3
Inter Quartile Range is: 3.4000000000000004
The upper bound value is: 22.400000000000002
The lower bound value is: 8.8
Outlier Values are:
[8.0, 8.5, 8.5, 23.5, 23.7, 24.6, 24.8]
Outlier values indexes are:
[0, 1, 2, 394, 395, 396, 397]
```

In [224]:

```
#Handling Outliers

#1) dropping the values of outliers from dataframe

clean_df=dataframe.loc[(dataframe['acceleration'] >lower_bound) & (dataframe['acceleration']<upper_bound)]
clean_df


```

Out[224]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
3	14.0	8	454.0	220	4354	9.0	70	1	chevrolet impala
4	15.0	8	400.0	150	3761	9.5	70	1	chevrolet monte carlo
5	16.0	8	400.0	230	4278	9.5	73	1	pontiac grand prix
6	14.0	8	455.0	225	4425	10.0	70	1	pontiac catalina
7	14.0	8	455.0	225	3086	10.0	70	1	buick estate wagon (sw)
8	15.0	8	383.0	170	3563	10.0	70	1	dodge challenger se
9	15.0	8	429.0	198	4341	10.0	70	1	ford galaxie 500
10	17.0	8	302.0	140	3449	10.5	70	1	ford torino
11	11.0	8	350.0	180	3664	11.0	73	1	oldsmobile omega
12	11.0	8	429.0	208	4633	11.0	72	1	mercury marquis
13	12.0	8	455.0	225	4951	11.0	73	1	buick electra 225 custom
14	13.0	8	360.0	175	3821	11.0	73	1	amc ambassador brougham
15	13.0	8	440.0	215	4735	11.0	73	1	chrysler new yorker brougham
16	15.0	8	318.0	150	3399	11.0	73	1	dodge dart custom
17	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
18	16.0	8	400.0	180	4220	11.1	77	1	pontiac grand prix lj
19	18.1	8	302.0	139	3205	11.2	78	1	ford futura
20	28.8	6	173.0	115	2595	11.3	79	1	chevrolet citation
21	15.5	8	350.0	170	4165	11.4	77	1	chevrolet monte carlo landau

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
22	32.7	6	168.0	132	2910	11.4	80	3	datsun 280-zx
23	12.0	8	383.0	180	4955	11.5	71	1	dodge monaco (sw)
24	12.0	8	429.0	198	4952	11.5	73	1	mercury marquis brougham
25	14.0	8	304.0	150	3672	11.5	73	1	amc matador
26	14.0	8	400.0	175	4464	11.5	71	1	pontiac catalina brougham
27	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
28	16.0	8	400.0	170	4668	11.5	75	1	pontiac catalina
29	17.0	8	304.0	150	3672	11.5	72	1	amc ambassador sst
30	32.0	4	135.0	84	2295	11.6	82	1	dodge rampage
31	13.0	8	302.0	129	3169	12.0	75	1	ford mustang ii
32	13.0	8	350.0	165	4274	12.0	72	1	chevrolet impala
...	...	...	...	...	...	...	...	...	...
364	32.8	4	78.0	52	1985	19.4	78	3	mazda glc deluxe
365	37.0	4	85.0	65	1975	19.4	81	3	datsun 210 mpg
366	15.0	6	250.0	72	3158	19.5	75	1	ford maverick
367	20.0	4	140.0	90	2408	19.5	72	1	chevrolet vega
368	21.0	4	120.0	87	2979	19.5	72	2	peugeot 504 (sw)
369	21.0	4	140.0	72	2401	19.5	73	1	chevrolet vega
370	29.0	4	68.0	49	1867	19.5	73	2	fiat 128
371	30.0	4	79.0	70	2074	19.5	71	2	peugeot 304
372	28.0	4	112.0	88	2605	19.6	82	1	chevrolet cavalier
373	30.7	6	145.0	76	3160	19.6	81	2	volvo diesel
374	36.4	5	121.0	67	2950	19.9	80	2	audi 5000s (diesel)
375	24.3	4	151.0	90	3003	20.1	80	1	amc concord
376	25.4	5	183.0	77	3530	20.1	79	2	mercedes benz 300d
377	28.1	4	141.0	80	3230	20.4	81	2	peugeot 505s turbo diesel

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
378	23.0	4	151.0	?	3035	20.5	82	1	amc concord dl
379	26.0	4	91.0	70	1955	20.5	71	1	plymouth cricket
380	26.0	4	97.0	46	1835	20.5	70	2	volkswagen 1131 deluxe sedan
381	29.9	4	98.0	65	2380	20.7	81	1	ford escort 2h
382	15.0	6	250.0	72	3432	21.0	75	1	mercury monarch
383	17.0	6	231.0	110	3907	21.0	75	1	buick century
384	18.0	6	250.0	78	3574	21.0	76	1	ford granada ghia
385	26.0	4	97.0	46	1950	21.0	73	2	volkswagen super beetle
386	32.0	4	71.0	65	1836	21.0	74	3	toyota corolla 1200
387	43.1	4	90.0	48	1985	21.5	78	2	volkswagen rabbit custom diesel
388	44.3	4	90.0	48	2085	21.7	80	2	vw rabbit c (diesel)
389	30.0	4	146.0	67	3250	21.8	80	2	mercedes-benz 240d
390	19.0	4	120.0	88	3270	21.9	76	2	peugeot 504
391	24.5	4	98.0	60	2164	22.1	76	1	chevrolet woody
392	23.9	8	260.0	90	3420	22.2	79	1	oldsmobile cutlass salon brougham
393	29.0	4	85.0	52	2035	22.2	76	1	chevrolet chevette

391 rows × 9 columns

In [225]:

```
#2) mark them as outliers and include it as a feature
new_df = dataframe
new_df["Outlier"] = np.where((dataframe['acceleration'] > lower_bound)
                             & (dataframe['acceleration'] < upper_bound), 0, 1)
new_df
```

Out[225]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name	Outlier
0	14.0	8	340.0	160	3609	8.0	70	1	plymouth 'cuda 340	1
1	14.0	8	440.0	215	4312	8.5	70	1	plymouth fury iii	1
2	15.0	8	390.0	190	3850	8.5	70	1	amc ambassador dpl	1
3	14.0	8	454.0	220	4354	9.0	70	1	chevrolet impala	0
4	15.0	8	400.0	150	3761	9.5	70	1	chevrolet monte carlo	0
5	16.0	8	400.0	230	4278	9.5	73	1	pontiac grand prix	0
6	14.0	8	455.0	225	4425	10.0	70	1	pontiac catalina	0
7	14.0	8	455.0	225	3086	10.0	70	1	buick estate wagon (sw)	0
8	15.0	8	383.0	170	3563	10.0	70	1	dodge challenger se	0
9	15.0	8	429.0	198	4341	10.0	70	1	ford galaxie 500	0
10	17.0	8	302.0	140	3449	10.5	70	1	ford torino	0
11	11.0	8	350.0	180	3664	11.0	73	1	oldsmobile omega	0
12	11.0	8	429.0	208	4633	11.0	72	1	mercury marquis	0
13	12.0	8	455.0	225	4951	11.0	73	1	buick electra 225 custom	0
14	13.0	8	360.0	175	3821	11.0	73	1	amc ambassador brougham	0
15	13.0	8	440.0	215	4735	11.0	73	1	chrysler new yorker brougham	0
16	15.0	8	318.0	150	3399	11.0	73	1	dodge dart custom	0
17	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite	0
18	16.0	8	400.0	180	4220	11.1	77	1	pontiac grand prix lj	0
19	18.1	8	302.0	139	3205	11.2	78	1	ford futura	0
20	28.8	6	173.0	115	2595	11.3	79	1	chevrolet citation	0

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name	Outlier
21	15.5	8	350.0	170	4165	11.4	77	1	chevrolet monte carlo landau	0
22	32.7	6	168.0	132	2910	11.4	80	3	datsun 280-zx	0
23	12.0	8	383.0	180	4955	11.5	71	1	dodge monaco (sw)	0
24	12.0	8	429.0	198	4952	11.5	73	1	mercury marquis brougham	0
25	14.0	8	304.0	150	3672	11.5	73	1	amc matador	0
26	14.0	8	400.0	175	4464	11.5	71	1	pontiac catalina brougham	0
27	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320	0
28	16.0	8	400.0	170	4668	11.5	75	1	pontiac catalina	0
29	17.0	8	304.0	150	3672	11.5	72	1	amc ambassador sst	0
...	...	...	...	...	...	...	...	...	...	...
368	21.0	4	120.0	87	2979	19.5	72	2	peugeot 504 (sw)	0
369	21.0	4	140.0	72	2401	19.5	73	1	chevrolet vega	0
370	29.0	4	68.0	49	1867	19.5	73	2	fiat 128	0
371	30.0	4	79.0	70	2074	19.5	71	2	peugeot 304	0
372	28.0	4	112.0	88	2605	19.6	82	1	chevrolet cavalier	0
373	30.7	6	145.0	76	3160	19.6	81	2	volvo diesel	0
374	36.4	5	121.0	67	2950	19.9	80	2	audi 5000s (diesel)	0
375	24.3	4	151.0	90	3003	20.1	80	1	amc concord	0
376	25.4	5	183.0	77	3530	20.1	79	2	mercedes benz 300d	0
377	28.1	4	141.0	80	3230	20.4	81	2	peugeot 505s turbo diesel	0
378	23.0	4	151.0	?	3035	20.5	82	1	amc concord dl	0
379	26.0	4	91.0	70	1955	20.5	71	1	plymouth cricket	0
380	26.0	4	97.0	46	1835	20.5	70	2	volkswagen 1131 deluxe sedan	0
381	29.9	4	98.0	65	2380	20.7	81	1	ford escort 2h	0
382	15.0	6	250.0	72	3432	21.0	75	1	mercury monarch	0
383	17.0	6	231.0	110	3907	21.0	75	1	buick century	0

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name	Outlier
384	18.0	6	250.0	78	3574	21.0	76	1	ford granada ghia	0
385	26.0	4	97.0	46	1950	21.0	73	2	volkswagen super beetle	0
386	32.0	4	71.0	65	1836	21.0	74	3	toyota corolla 1200	0
387	43.1	4	90.0	48	1985	21.5	78	2	volkswagen rabbit custom diesel	0
388	44.3	4	90.0	48	2085	21.7	80	2	vw rabbit c (diesel)	0
389	30.0	4	146.0	67	3250	21.8	80	2	mercedes-benz 240d	0
390	19.0	4	120.0	88	3270	21.9	76	2	peugeot 504	0
391	24.5	4	98.0	60	2164	22.1	76	1	chevrolet woody	0
392	23.9	8	260.0	90	3420	22.2	79	1	oldsmobile cutlass salon brougham	0
393	29.0	4	85.0	52	2035	22.2	76	1	chevrolet chevette	0
394	23.0	4	97.0	54	2254	23.5	72	2	volkswagen type 3	1
395	43.4	4	90.0	48	2335	23.7	80	2	vw dasher (diesel)	1
396	44.0	4	97.0	52	2130	24.6	82	2	vw pickup	1
397	27.2	4	141.0	71	3190	24.8	79	2	peugeot 504	1

398 rows × 10 columns

In [226]:

```
#3)Finally, we can transform the feature to dampen the effect of the outlier
import numpy as np
new_df["mean_accelaration"] = np.where(new_df['Outlier']==1,
                        np.mean(new_df['acceleration']),new_df['acceleration'])
new_df
```

Out[226]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name	Outlier	mean_accelaration
0	14.0	8	340.0	160	3609	8.0	70	1	plymouth 'cuda 340	1	15.56809
1	14.0	8	440.0	215	4312	8.5	70	1	plymouth fury iii	1	15.56809
2	15.0	8	390.0	190	3850	8.5	70	1	amc ambassador dpl	1	15.56809
3	14.0	8	454.0	220	4354	9.0	70	1	chevrolet impala	0	9.00000
4	15.0	8	400.0	150	3761	9.5	70	1	chevrolet monte carlo	0	9.50000
5	16.0	8	400.0	230	4278	9.5	73	1	pontiac grand prix	0	9.50000
6	14.0	8	455.0	225	4425	10.0	70	1	pontiac catalina	0	10.00000
7	14.0	8	455.0	225	3086	10.0	70	1	buick estate wagon (sw)	0	10.00000
8	15.0	8	383.0	170	3563	10.0	70	1	dodge challenger se	0	10.00000
9	15.0	8	429.0	198	4341	10.0	70	1	ford galaxie 500	0	10.00000
10	17.0	8	302.0	140	3449	10.5	70	1	ford torino	0	10.50000
11	11.0	8	350.0	180	3664	11.0	73	1	oldsmobile omega	0	11.00000
12	11.0	8	429.0	208	4633	11.0	72	1	mercury marquis	0	11.00000
13	12.0	8	455.0	225	4951	11.0	73	1	buick electra 225 custom	0	11.00000
14	13.0	8	360.0	175	3821	11.0	73	1	amc ambassador brougham	0	11.00000
15	13.0	8	440.0	215	4735	11.0	73	1	chrysler new yorker brougham	0	11.00000
16	15.0	8	318.0	150	3399	11.0	73	1	dodge dart custom	0	11.00000
17	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite	0	11.00000
18	16.0	8	400.0	180	4220	11.1	77	1	pontiac grand prix lj	0	11.10000
19	18.1	8	302.0	139	3205	11.2	78	1	ford futura	0	11.20000
20	28.8	6	173.0	115	2595	11.3	79	1	chevrolet citation	0	11.30000

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name	Outlier	mean_accelaration
21	15.5	8	350.0	170	4165	11.4	77	1	chevrolet monte carlo landau	0	11.40000
22	32.7	6	168.0	132	2910	11.4	80	3	datsun 280-zx	0	11.40000
23	12.0	8	383.0	180	4955	11.5	71	1	dodge monaco (sw)	0	11.50000
24	12.0	8	429.0	198	4952	11.5	73	1	mercury marquis brougham	0	11.50000
25	14.0	8	304.0	150	3672	11.5	73	1	amc matador	0	11.50000
26	14.0	8	400.0	175	4464	11.5	71	1	pontiac catalina brougham	0	11.50000
27	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320	0	11.50000
28	16.0	8	400.0	170	4668	11.5	75	1	pontiac catalina	0	11.50000
29	17.0	8	304.0	150	3672	11.5	72	1	amc ambassador sst	0	11.50000
...	...	...	...	...	...	...	...	...	...	...	...
368	21.0	4	120.0	87	2979	19.5	72	2	peugeot 504 (sw)	0	19.50000
369	21.0	4	140.0	72	2401	19.5	73	1	chevrolet vega	0	19.50000
370	29.0	4	68.0	49	1867	19.5	73	2	fiat 128	0	19.50000
371	30.0	4	79.0	70	2074	19.5	71	2	peugeot 304	0	19.50000
372	28.0	4	112.0	88	2605	19.6	82	1	chevrolet cavalier	0	19.60000
373	30.7	6	145.0	76	3160	19.6	81	2	volvo diesel	0	19.60000
374	36.4	5	121.0	67	2950	19.9	80	2	audi 5000s (diesel)	0	19.90000
375	24.3	4	151.0	90	3003	20.1	80	1	amc concord	0	20.10000
376	25.4	5	183.0	77	3530	20.1	79	2	mercedes benz 300d	0	20.10000
377	28.1	4	141.0	80	3230	20.4	81	2	peugeot 505s turbo diesel	0	20.40000
378	23.0	4	151.0	?	3035	20.5	82	1	amc concord dl	0	20.50000
379	26.0	4	91.0	70	1955	20.5	71	1	plymouth cricket	0	20.50000
380	26.0	4	97.0	46	1835	20.5	70	2	volkswagen 1131 deluxe sedan	0	20.50000
381	29.9	4	98.0	65	2380	20.7	81	1	ford escort 2h	0	20.70000
382	15.0	6	250.0	72	3432	21.0	75	1	mercury monarch	0	21.00000

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name	Outlier	mean_accelaration
383	17.0	6	231.0	110	3907	21.0	75	1	buick century	0	21.00000
384	18.0	6	250.0	78	3574	21.0	76	1	ford granada ghia	0	21.00000
385	26.0	4	97.0	46	1950	21.0	73	2	volkswagen super beetle	0	21.00000
386	32.0	4	71.0	65	1836	21.0	74	3	toyota corolla 1200	0	21.00000
387	43.1	4	90.0	48	1985	21.5	78	2	volkswagen rabbit custom diesel	0	21.50000
388	44.3	4	90.0	48	2085	21.7	80	2	vw rabbit c (diesel)	0	21.70000
389	30.0	4	146.0	67	3250	21.8	80	2	mercedes-benz 240d	0	21.80000
390	19.0	4	120.0	88	3270	21.9	76	2	peugeot 504	0	21.90000
391	24.5	4	98.0	60	2164	22.1	76	1	chevrolet woody	0	22.10000
392	23.9	8	260.0	90	3420	22.2	79	1	oldsmobile cutlass salon brougham	0	22.20000
393	29.0	4	85.0	52	2035	22.2	76	1	chevrolet chevette	0	22.20000
394	23.0	4	97.0	54	2254	23.5	72	2	volkswagen type 3	1	15.56809
395	43.4	4	90.0	48	2335	23.7	80	2	vw dasher (diesel)	1	15.56809
396	44.0	4	97.0	52	2130	24.6	82	2	vw pickup	1	15.56809
397	27.2	4	141.0	71	3190	24.8	79	2	peugeot 504	1	15.56809

398 rows × 11 columns

In [227]:

```
#Deleting Observations with Missing Values with dropna()

import pandas as pd

#csv file
url = 'C:/Users/Prerna/Desktop/ML_jupyter_notenooks/datasets/titanic.csv'

#Creating a dataframe
dataframe = pd.read_csv(url)
dataframe.dropna(inplace=True)

dataframe
```

Out[227]:

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st	29.00	female	1	1
1	Allison, Miss Helen Loraine	1st	2.00	female	0	1
2	Allison, Mr Hudson Joshua Creighton	1st	30.00	male	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.00	female	0	1
4	Allison, Master Hudson Trevor	1st	0.92	male	1	0
5	Anderson, Mr Harry	1st	47.00	male	1	0
6	Andrews, Miss Kornelia Theodosia	1st	63.00	female	1	1
7	Andrews, Mr Thomas, jr	1st	39.00	male	0	0
8	Appleton, Mrs Edward Dale (Charlotte Lamson)	1st	58.00	female	1	1
9	Artagaveytia, Mr Ramon	1st	71.00	male	0	0
10	Astor, Colonel John Jacob	1st	47.00	male	0	0
11	Astor, Mrs John Jacob (Madeleine Talmadge Force)	1st	19.00	female	1	1
15	Baxter, Mrs James (Helene DeLaudeniere Chaput)	1st	50.00	female	1	1
16	Baxter, Mr Quigg Edmond	1st	24.00	male	0	0
17	Beattie, Mr Thomson	1st	36.00	male	0	0
18	Beckwith, Mr Richard Leonard	1st	37.00	male	1	0
19	Beckwith, Mrs Richard Leonard (Sallie Monypeny)	1st	47.00	female	1	1

	Name	PClass	Age	Sex	Survived	SexCode
20	Behr, Mr Karl Howell	1st	26.00	male	1	0
21	Birnbaum, Mr Jakob	1st	25.00	male	0	0
22	Bishop, Mr Dickinson H	1st	25.00	male	1	0
23	Bishop, Mrs Dickinson H (Helen Walton)	1st	19.00	female	1	1
24	Bjornstrm-Steffansson, Mr Mauritz Hakan	1st	28.00	male	1	0
25	Blackwell, Mr Stephen Weart	1st	45.00	male	0	0
26	Blank, Mr Henry	1st	39.00	male	1	0
27	Bonnell, Miss Caroline	1st	30.00	female	1	1
28	Bonnell, Miss Elizabeth	1st	58.00	female	1	1
30	Bowen, Miss Grace Scott	1st	45.00	female	1	1
31	Bowerman, Miss Elsie Edith	1st	22.00	female	1	1
33	Brady, Mr John Bertram	1st	41.00	male	0	0
34	Brandeis, Mr Emil	1st	48.00	male	0	0
...	...	...	...	...	...	...
1264	Turkula, Mrs Hedvig	3rd	63.00	female	1	1
1269	Van der Planke, Miss Augusta	3rd	18.00	female	0	1
1270	Van der Planke, Mr Jules	3rd	31.00	male	0	0
1271	Van der Planke, Mrs Jules	3rd	31.00	female	0	1
1272	Van der Planke, Mr Leon	3rd	15.00	male	0	0
1273	Van der Steen, Mr Leo Peter	3rd	28.00	male	0	0
1274	Van de Velde, Mr John Joseph	3rd	36.00	male	0	0
1275	Vandewalle, Mr Nestor Cyriel	3rd	28.00	male	0	0
1276	Van Impe, Miss Catharine	3rd	10.00	female	0	1
1277	Van Impe, Mr Jean Baptiste	3rd	36.00	male	0	0
1278	Van Impe, Mrs Jean Baptiste	3rd	30.00	female	0	1
1279	Vartunian, Mr David	3rd	22.00	male	1	0

	Name	PClass	Age	Sex	Survived	SexCode
1281	Vendel, Mr Olof Wdvin	3rd	29.00	male	0	0
1282	Vereruysse, Mr Victor	3rd	47.00	male	0	0
1283	Vestrom, Miss Hulda Amanda Adolfina	3rd	14.00	female	0	1
1284	Vonk, Mr Jenko	3rd	22.00	male	0	0
1291	Widegren, Mr Charles Peter	3rd	51.00	male	0	0
1292	Wiklund, Mr Jacob Alfred	3rd	18.00	male	0	0
1293	Wilkes, Mrs Ellen	3rd	45.00	female	1	1
1297	Williams, Mr Leslie	3rd	28.00	male	0	0
1298	Windelov, Mr Einar	3rd	21.00	male	0	0
1299	Wirz, Mr Albert	3rd	27.00	male	0	0
1301	Wittevrongel, Mr Camiel	3rd	36.00	male	0	0
1303	Yasbeck, Mr Antoni	3rd	27.00	male	0	0
1304	Yasbeck, Mrs Antoni	3rd	15.00	female	1	1
1308	Zakarian, Mr Artun	3rd	27.00	male	0	0
1309	Zakarian, Mr Maprieder	3rd	26.00	male	0	0
1310	Zenni, Mr Philip	3rd	22.00	male	0	0
1311	Lievens, Mr Rene	3rd	24.00	male	0	0
1312	Zimmerman, Leo	3rd	29.00	male	0	0

756 rows × 6 columns

In [21]:

```
#Encoding Nominal Categorical Features
#You have a feature with nominal classes (Categorial data) that has no intrinsic ordering (Sex: Male,female)
#One-hot encode the feature using scikit-learn's LabelBinarizer

import pandas as pd
from sklearn.preprocessing import LabelBinarizer

#csv file
url = url = "C:/Users/Prerna/Desktop/ML_jupyter_notenooks/datasets/titanic.csv"
df = pd.read_csv(url)

#Creating a dataframe
dataframe = pd.read_csv(url).fillna(0)

dataframe = dataframe.drop('SexCode',axis=1)
print(dataframe.dtypes)

# Create one-hot encoder
one_hot = LabelBinarizer()

# One-hot encode feature
dataframe['Sex'] = one_hot.fit_transform(dataframe['Sex'])

# View feature classes and use the classes_ method to output the classes
print(one_hot.classes_)

dataframe
```

```
Name         object
PClass       object
Age         float64
Sex          object
Survived      int64
dtype: object
['female' 'male']
```

Out[21]:

	Name	PClass	Age	Sex	Survived
0	Allen, Miss Elisabeth Walton	1st	29.00	0	1
1	Allison, Miss Helen Loraine	1st	2.00	0	0

	Name	PClass	Age	Sex	Survived
2	Allison, Mr Hudson Joshua Creighton	1st	30.00	1	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.00	0	0
4	Allison, Master Hudson Trevor	1st	0.92	1	1
5	Anderson, Mr Harry	1st	47.00	1	1
6	Andrews, Miss Kornelia Theodosia	1st	63.00	0	1
7	Andrews, Mr Thomas, jr	1st	39.00	1	0
8	Appleton, Mrs Edward Dale (Charlotte Lamson)	1st	58.00	0	1
9	Artagaveytia, Mr Ramon	1st	71.00	1	0
10	Astor, Colonel John Jacob	1st	47.00	1	0
11	Astor, Mrs John Jacob (Madeleine Talmadge Force)	1st	19.00	0	1
12	Aubert, Mrs Leontine Pauline	1st	0.00	0	1
13	Barkworth, Mr Algernon H	1st	0.00	1	1
14	Baumann, Mr John D	1st	0.00	1	0
15	Baxter, Mrs James (Helene DeLaudeniere Chaput)	1st	50.00	0	1
16	Baxter, Mr Quigg Edmond	1st	24.00	1	0
17	Beattie, Mr Thomson	1st	36.00	1	0
18	Beckwith, Mr Richard Leonard	1st	37.00	1	1
19	Beckwith, Mrs Richard Leonard (Sallie Monypeny)	1st	47.00	0	1
20	Behr, Mr Karl Howell	1st	26.00	1	1
21	Birnbaum, Mr Jakob	1st	25.00	1	0
22	Bishop, Mr Dickinson H	1st	25.00	1	1
23	Bishop, Mrs Dickinson H (Helen Walton)	1st	19.00	0	1
24	Bjornstrm-Steffansson, Mr Mauritz Hakan	1st	28.00	1	1
25	Blackwell, Mr Stephen Weart	1st	45.00	1	0
26	Blank, Mr Henry	1st	39.00	1	1
27	Bonnell, Miss Caroline	1st	30.00	0	1

	Name	PClass	Age	Sex	Survived
28	Bonnell, Miss Elizabeth	1st	58.00	0	1
29	Borebank, Mr John James	1st	0.00	1	0
...	...	...	...	...	...
1283	Vestrom, Miss Hulda Amanda Adolfina	3rd	14.00	0	0
1284	Vonk, Mr Jenko	3rd	22.00	1	0
1285	Ware, Mr Frederick	3rd	0.00	1	0
1286	Warren, Mr Charles William	3rd	0.00	1	0
1287	Wazli, Mr Yousif	3rd	0.00	1	0
1288	Webber, Mr James	3rd	0.00	1	0
1289	Wennerstrom, Mr August Edvard	3rd	0.00	1	1
1290	Wenzel, Mr Linhart	3rd	0.00	1	0
1291	Widegren, Mr Charles Peter	3rd	51.00	1	0
1292	Wiklund, Mr Jacob Alfred	3rd	18.00	1	0
1293	Wilkes, Mrs Ellen	3rd	45.00	0	1
1294	Willer, Mr Aaron	3rd	0.00	1	0
1295	Willey, Mr Edward	3rd	0.00	1	0
1296	Williams, Mr Howard Hugh	3rd	0.00	1	0
1297	Williams, Mr Leslie	3rd	28.00	1	0
1298	Windelov, Mr Einar	3rd	21.00	1	0
1299	Wirz, Mr Albert	3rd	27.00	1	0
1300	Wiseman, Mr Phillippe	3rd	0.00	1	0
1301	Wittevrongel, Mr Camiel	3rd	36.00	1	0
1302	Yalsevac, Mr Ivan	3rd	0.00	1	1
1303	Yasbeck, Mr Antoni	3rd	27.00	1	0
1304	Yasbeck, Mrs Antoni	3rd	15.00	0	1
1305	Youssef, Mr Gerios	3rd	0.00	1	0

	Name	PClass	Age	Sex	Survived
1306	Zabour, Miss Hileni	3rd	0.00	0	0
1307	Zabour, Miss Tamini	3rd	0.00	0	0
1308	Zakarian, Mr Artun	3rd	27.00	1	0
1309	Zakarian, Mr Maprieder	3rd	26.00	1	0
1310	Zenni, Mr Philip	3rd	22.00	1	0
1311	Lievens, Mr Rene	3rd	24.00	1	0
1312	Zimmerman, Leo	3rd	29.00	1	0

1313 rows × 5 columns

In [23]:

```
#Encoding Ordinal Categorical Features
#You have an ordinal categorical feature(PClass: 1st,2nd,3rd)
#Use pandas DataFrame’s replace method to transform string labels to numerical equivalents

import pandas as pd

#csv file
url = url = "C:/Users/Prerna/Desktop/ML_jupyter_notenooks/datasets/titanic.csv"
df = pd.read_csv(url)

#Creating a dataframe
dataframe = pd.read_csv(url).fillna(0)

# Create mapper
scale_mapper = {"1st":1,"2nd":2,"3rd":3}

# Replace feature values with scale
dataframe["PClass"] = dataframe["PClass"].replace(scale_mapper)
dataframe
```

Out[23]:

	Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1	29.00	female	1	1
1	Allison, Miss Helen Loraine	1	2.00	female	0	1
2	Allison, Mr Hudson Joshua Creighton	1	30.00	male	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1	25.00	female	0	1
4	Allison, Master Hudson Trevor	1	0.92	male	1	0
5	Anderson, Mr Harry	1	47.00	male	1	0
6	Andrews, Miss Kornelia Theodosia	1	63.00	female	1	1
7	Andrews, Mr Thomas, jr	1	39.00	male	0	0
8	Appleton, Mrs Edward Dale (Charlotte Lamson)	1	58.00	female	1	1
9	Artagaveytia, Mr Ramon	1	71.00	male	0	0
10	Astor, Colonel John Jacob	1	47.00	male	0	0
11	Astor, Mrs John Jacob (Madeleine Talmadge Force)	1	19.00	female	1	1

	Name	PClass	Age	Sex	Survived	SexCode
12	Aubert, Mrs Leontine Pauline	1	0.00	female	1	1
13	Barkworth, Mr Algernon H	1	0.00	male	1	0
14	Baumann, Mr John D	1	0.00	male	0	0
15	Baxter, Mrs James (Helene DeLaudeniere Chaput)	1	50.00	female	1	1
16	Baxter, Mr Quigg Edmond	1	24.00	male	0	0
17	Beattie, Mr Thomson	1	36.00	male	0	0
18	Beckwith, Mr Richard Leonard	1	37.00	male	1	0
19	Beckwith, Mrs Richard Leonard (Sallie Monypeny)	1	47.00	female	1	1
20	Behr, Mr Karl Howell	1	26.00	male	1	0
21	Birnbaum, Mr Jakob	1	25.00	male	0	0
22	Bishop, Mr Dickinson H	1	25.00	male	1	0
23	Bishop, Mrs Dickinson H (Helen Walton)	1	19.00	female	1	1
24	Bjornstrm-Steffansson, Mr Mauritz Hakan	1	28.00	male	1	0
25	Blackwell, Mr Stephen Weart	1	45.00	male	0	0
26	Blank, Mr Henry	1	39.00	male	1	0
27	Bonnell, Miss Caroline	1	30.00	female	1	1
28	Bonnell, Miss Elizabeth	1	58.00	female	1	1
29	Borebank, Mr John James	1	0.00	male	0	0
...	...	...	...	...	...	...
1283	Vestrom, Miss Hulda Amanda Adolfina	3	14.00	female	0	1
1284	Vonk, Mr Jenko	3	22.00	male	0	0
1285	Ware, Mr Frederick	3	0.00	male	0	0
1286	Warren, Mr Charles William	3	0.00	male	0	0
1287	Wazli, Mr Yousif	3	0.00	male	0	0
1288	Webber, Mr James	3	0.00	male	0	0
1289	Wennerstrom, Mr August Edvard	3	0.00	male	1	0

	Name	PClass	Age	Sex	Survived	SexCode
1290	Wenzel, Mr Linhart	3	0.00	male	0	0
1291	Widegren, Mr Charles Peter	3	51.00	male	0	0
1292	Wiklund, Mr Jacob Alfred	3	18.00	male	0	0
1293	Wilkes, Mrs Ellen	3	45.00	female	1	1
1294	Willer, Mr Aaron	3	0.00	male	0	0
1295	Willey, Mr Edward	3	0.00	male	0	0
1296	Williams, Mr Howard Hugh	3	0.00	male	0	0
1297	Williams, Mr Leslie	3	28.00	male	0	0
1298	Windelov, Mr Einar	3	21.00	male	0	0
1299	Wirz, Mr Albert	3	27.00	male	0	0
1300	Wiseman, Mr Phillippe	3	0.00	male	0	0
1301	Wittevrongel, Mr Camiel	3	36.00	male	0	0
1302	Yalsevac, Mr Ivan	3	0.00	male	1	0
1303	Yasbeck, Mr Antoni	3	27.00	male	0	0
1304	Yasbeck, Mrs Antoni	3	15.00	female	1	1
1305	Youssef, Mr Gerios	3	0.00	male	0	0
1306	Zabour, Miss Hileni	3	0.00	female	0	1
1307	Zabour, Miss Tamini	3	0.00	female	0	1
1308	Zakarian, Mr Artun	3	27.00	male	0	0
1309	Zakarian, Mr Maprieder	3	26.00	male	0	0
1310	Zenni, Mr Philip	3	22.00	male	0	0
1311	Lievens, Mr Rene	3	24.00	male	0	0
1312	Zimmerman, Leo	3	29.00	male	0	0

1313 rows × 6 columns

In [ ]:
```
1
```