One hot encoding and Feature Scaling

In [1]:

```python
import pandas as pd
import numpy as np
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Loding the dataset

In [2]:

```python
data = sns.load_dataset('titanic')
data.head()
```

Out[2]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_mal
0	0	3	male	22.0	1	0	7.2500	S	Third	man	Tru
1	1	1	female	38.0	1	0	71.2833	C	First	woman	Fals
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	Fals
3	1	1	female	35.0	1	0	53.1000	S	First	woman	Fals
4	0	3	male	35.0	0	0	8.0500	S	Third	man	Tru

In [3]:

```python
data.dropna(inplace=True)
data.drop_duplicates(inplace=True)
```

In [4]:

```
data
```

Out[4]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_m
1	1	1	female	38.0	1	0	71.2833	C	First	woman	Fa
3	1	1	female	35.0	1	0	53.1000	S	First	woman	Fa
6	0	1	male	54.0	0	0	51.8625	S	First	man	T
10	1	3	female	4.0	1	1	16.7000	S	Third	child	Fa
11	1	1	female	58.0	0	0	26.5500	S	First	woman	Fa
...	...	...	...	...	...	...	...	...	...	...	
871	1	1	female	47.0	1	1	52.5542	S	First	woman	Fa
872	0	1	male	33.0	0	0	5.0000	S	First	man	T
879	1	1	female	56.0	0	1	83.1583	C	First	woman	Fa
887	1	1	female	19.0	0	0	30.0000	S	First	woman	Fa
889	1	1	male	26.0	0	0	30.0000	C	First	man	T

181 rows × 15 columns

One Hot Encoding

In [5]:

```
data['SEX'] = pd.get_dummies(data=data.sex, drop_first=True)
```

In [6]:

```
data.head(3)
```

Out[6]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_mal
1	1	1	female	38.0	1	0	71.2833	C	First	woman	Fals
3	1	1	female	35.0	1	0	53.1000	S	First	woman	Fals
6	0	1	male	54.0	0	0	51.8625	S	First	man	Tru

Converting the Entire dataframe

In [7]:

```
data_No_Dummies = pd.get_dummies(data=data)
```

In [8]:

```
data_No_Dummies
```

Out[8]:

e	sibsp	parch	fare	adult_male	alone	SEX	sex_female	...	deck_C	deck_D	deck_E	d
0	1	0	71.2833	False	False	0	1	...	1	0	0	
0	1	0	53.1000	False	False	0	1	...	1	0	0	
0	0	0	51.8625	True	True	1	0	...	0	0	1	
0	1	1	16.7000	False	False	0	1	...	0	0	0	
0	0	0	26.5500	False	True	0	1	...	1	0	0	
..	...	...	...	...	...	...	...	...	...	...	...	
0	1	1	52.5542	False	False	0	1	...	0	1	0	
0	0	0	5.0000	True	True	1	0	...	0	0	0	
0	0	1	83.1583	False	False	0	1	...	1	0	0	
0	0	0	30.0000	False	True	0	1	...	0	0	0	
0	0	0	30.0000	True	True	1	0	...	1	0	0	

In [9]:

```
data_No_Dummies.shape
```
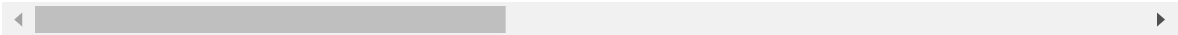
Out[9]:

(181, 32)

In [10]:

```
data_With_Dummies = pd.get_dummies(data=data, drop_first=True)
data_With_Dummies.head()
```

Out[10]:

	survived	pclass	age	sibsp	parch	fare	adult_male	alone	SEX	sex_male	...	wh
1	1	1	38.0	1	0	71.2833	False	False	0	0	...	
3	1	1	35.0	1	0	53.1000	False	False	0	0	...	
6	0	1	54.0	0	0	51.8625	True	True	1	1	...	
10	1	3	4.0	1	1	16.7000	False	False	0	0	...	
11	1	1	58.0	0	0	26.5500	False	True	0	0	...	

5 rows × 25 columns

In [11]:

```
data_With_Dummies.shape
```

Out[11]:

```
(181, 25)
```

In [12]:

```python
from sklearn.preprocessing import MinMaxScaler, StandardScaler, RobustScaler, LabelEncode
```

Label encoder

In [13]:

```python
le = LabelEncoder()
```

In [14]:

```
data
```

Out[14]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_m
1	1	1	female	38.0	1	0	71.2833	C	First	woman	Fa
3	1	1	female	35.0	1	0	53.1000	S	First	woman	Fa
6	0	1	male	54.0	0	0	51.8625	S	First	man	T
10	1	3	female	4.0	1	1	16.7000	S	Third	child	Fa
11	1	1	female	58.0	0	0	26.5500	S	First	woman	Fa
...	...	...	...	...	...	...	...	...	...	...	
871	1	1	female	47.0	1	1	52.5542	S	First	woman	Fa
872	0	1	male	33.0	0	0	5.0000	S	First	man	T
879	1	1	female	56.0	0	1	83.1583	C	First	woman	Fa
887	1	1	female	19.0	0	0	30.0000	S	First	woman	Fa
889	1	1	male	26.0	0	0	30.0000	C	First	man	T

181 rows × 16 columns

In [15]:

```python
data['EMBARKED'] = le.fit_transform(data.embarked)
```

In [16]:

```
data
```

Out[16]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_m
1	1	1	female	38.0	1	0	71.2833	C	First	woman	Fa
3	1	1	female	35.0	1	0	53.1000	S	First	woman	Fa
6	0	1	male	54.0	0	0	51.8625	S	First	man	T
10	1	3	female	4.0	1	1	16.7000	S	Third	child	Fa
11	1	1	female	58.0	0	0	26.5500	S	First	woman	Fa
...	...	...	...	...	...	...	...	...	...	...	
871	1	1	female	47.0	1	1	52.5542	S	First	woman	Fa
872	0	1	male	33.0	0	0	5.0000	S	First	man	T
879	1	1	female	56.0	0	1	83.1583	C	First	woman	Fa
887	1	1	female	19.0	0	0	30.0000	S	First	woman	Fa
889	1	1	male	26.0	0	0	30.0000	C	First	man	T

181 rows × 17 columns

In [17]:

```
data.embarked.unique()
```

Out[17]:

```
array(['C', 'S', 'Q'], dtype=object)
```

Data Scaling or Data Normalising

Minmax Scaler

In [18]:

```
minmax = MinMaxScaler()
```

In [19]:

```python
data[['age']]
```

Out[19]:

	age
1	38.0
3	35.0
6	54.0
10	4.0
11	58.0
...	...
871	47.0
872	33.0
879	56.0
887	19.0
889	26.0

181 rows × 1 columns

In [20]:

```python
data[['age']].values
```

Out[20]:

```
array([[38.  ],
       [35.  ],
       [54.  ],
       [ 4.  ],
       [58.  ],
       [34.  ],
       [28.  ],
       [19.  ],
       [49.  ],
       [65.  ],
       [45.  ],
       [29.  ],
       [25.  ],
       [23.  ],
       [46.  ],
       [71.  ],
       [23.  ],
       [21.  ],
```

In [21]:

```python
age = data[['age']].values
```

In [22]:

```
age
```

```
       [40.  ],
       [58.  ],
       [35.  ],
       [37.  ],
       [63.  ],
       [19.  ],
       [36.  ],
       [ 2.  ],
       [50.  ],
       [ 0.92],
       [17.  ],
       [30.  ],
       [24.  ],
       [18.  ],
       [31.  ],
       [40.  ],
       [36.  ],
       [36.  ],
       [16.  ],
       [45.5 ],
```

In [23]:

```
data['AGE'] = minmax.fit_transform(age)
```

In [24]:

```
data.head()
```

Out[24]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_ma
1	1	1	female	38.0	1	0	71.2833	C	First	woman	Fal
3	1	1	female	35.0	1	0	53.1000	S	First	woman	Fal
6	0	1	male	54.0	0	0	51.8625	S	First	man	Tr
10	1	3	female	4.0	1	1	16.7000	S	Third	child	Fal
11	1	1	female	58.0	0	0	26.5500	S	First	woman	Fal

In [25]:

```
data.AGE.min()
```

Out[25]:

```
0.0
```

In [26]:

```
data.AGE.max()
```

Out[26]:

```
1.0
```

Standard Scaler

In [27]:

```python
sc = StandardScaler()
```

In [28]:

```python
data['AGE_SC'] = sc.fit_transform(age)
```

In [29]:

```python
data
```

Out[29]:

ived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
1	1	female	35.0	1	0	53.1000	S	First	woman	False	C
0	1	male	54.0	0	0	51.8625	S	First	man	True	E
1	3	female	4.0	1	1	16.7000	S	Third	child	False	G
1	1	female	58.0	0	0	26.5500	S	First	woman	False	C
...	...	...	...	...	...	...	...	...	...	...	...
1	1	female	47.0	1	1	52.5542	S	First	woman	False	D
0	1	male	33.0	0	0	5.0000	S	First	man	True	B
1	1	female	56.0	0	1	83.1583	C	First	woman	False	C
1	1	female	19.0	0	0	30.0000	S	First	woman	False	B
1	1	male	26.0	0	0	30.0000	C	First	man	True	C

‹ 19 columns
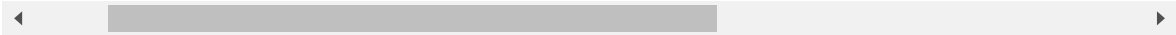
In [30]:

```python
data.AGE_SC.min()
```

Out[30]:

-2.221892714597594

In [31]:

```python
data.AGE_SC.max()
```

Out[31]:

2.83190075622482

Tasks

1. Try other Scaling methods

2. Try all these methods in car sales dataset