Import Library | pandas | numpy

In [71]:
```
import numpy as np
import pandas as pd
```

Import Datasets

In [72]:
```
df = pd.read_csv('cars.csv')
```

In [73]:
```
df.sample(5)
```

Out[73]:

	brand	km_driven	fuel	owner	selling_price
2230	Mahindra	50000	Diesel	Second Owner	250000
5904	Honda	50000	Diesel	First Owner	1050000
2114	Maruti	46000	Petrol	Fourth & Above Owner	75000
6940	Mahindra	90000	Diesel	First Owner	550000
4665	Honda	85000	Petrol	Third Owner	476999

In [74]:
```
df['owner'].value_counts()
```

Out[74]:
```
owner
First Owner             5289
Second Owner            2105
Third Owner              555
Fourth & Above Owner     174
Test Drive Car             5
Name: count, dtype: int64
```

In [75]:
```
df['brand'].value_counts()
df['brand'].nunique()
```

Out[75]:
```
32
```

In [76]:
```
df['fuel'].value_counts()
```

Out[76]:
```
fuel
Diesel    4402
Petrol    3631
CNG         57
LPG         38
Name: count, dtype: int64
```

1. One Hot Encoding Using Pandas

In [117]:
```
pd.get_dummies(df,columns=['fuel','owner'])
```

Out[117]:

	brand	km_driven	selling_price	fuel_CNG	fuel_Diesel	fuel_LPG	fuel_Petrol	owner_First Owner	owner_ &
0	Maruti	145500	450000	False	True	False	False	True	
1	Skoda	120000	370000	False	True	False	False	False	
2	Honda	140000	158000	False	False	False	True	False	
3	Hyundai	127000	225000	False	True	False	False	True	
4	Maruti	120000	130000	False	False	False	True	True	
...	...	...	...	...	...	...	...	...	
8123	Hyundai	110000	320000	False	False	False	True	True	
8124	Hyundai	119000	135000	False	True	False	False	False	
8125	Maruti	120000	382000	False	True	False	False	True	
8126	Tata	25000	290000	False	True	False	False	True	
8127	Tata	25000	290000	False	True	False	False	True	

8128 rows × 12 columns

2. K-1 One Hot Encoding

In [118]:
```
pd.get_dummies(df,columns=['fuel','owner'],drop_first=True)
```

Out[118]:

	brand	km_driven	selling_price	fuel_Diesel	fuel_LPG	fuel_Petrol	owner_Fourth & Above Owner	owner_Second Owner
0	Maruti	145500	450000	True	False	False	False	False
1	Skoda	120000	370000	True	False	False	False	True
2	Honda	140000	158000	False	False	True	False	False
3	Hyundai	127000	225000	True	False	False	False	False
4	Maruti	120000	130000	False	False	True	False	False
...	...	...	...	...	...	...	...	...
8123	Hyundai	110000	320000	False	False	True	False	False
8124	Hyundai	119000	135000	True	False	False	True	False
8125	Maruti	120000	382000	True	False	False	False	False
8126	Tata	25000	290000	True	False	False	False	False
8127	Tata	25000	290000	True	False	False	False	False

8128 rows × 10 columns

3. One hot encoding Using Sklearn

In [79]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(df.iloc[:,0:4],df.iloc[:,-1],test_si
```

In [80]:
```
df.head()
```

Out[80]:

	brand	km_driven	fuel	owner	selling_price
0	Maruti	145500	Diesel	First Owner	450000
1	Skoda	120000	Diesel	Second Owner	370000
2	Honda	140000	Petrol	Third Owner	158000
3	Hyundai	127000	Diesel	First Owner	225000
4	Maruti	120000	Petrol	First Owner	130000

In [81]:
```
X_train.head()
```

Out[81]:

	brand	km_driven	fuel	owner
5571	Hyundai	35000	Diesel	First Owner
2038	Jeep	60000	Diesel	First Owner
2957	Hyundai	25000	Petrol	First Owner
7618	Mahindra	130000	Diesel	Second Owner
6684	Hyundai	155000	Diesel	First Owner

In [82]:
```
X_test.head()
```

Out[82]:

	brand	km_driven	fuel	owner
606	Hyundai	80000	Petrol	First Owner
7575	Mahindra	70000	Diesel	Second Owner
7705	Toyota	68089	Petrol	First Owner
4305	Hyundai	70000	Petrol	Second Owner
2685	Mahindra	97000	Diesel	Second Owner

Import OHE: SK Learn

In [109]:
```
from sklearn.preprocessing import OneHotEncoder
```

In [110]:
```
ohe = OneHotEncoder(drop='first',sparse=False,dtype=np.int32)
```

In [111]:
```
X_train_new = ohe.fit_transform(X_train[['fuel','owner']]).toarray()
```

```
C:\Users\ASUS\anaconda3\Lib\site-packages\sklearn\preprocessing\_encoders.py:972: Fu
tureWarning: `sparse` was renamed to `sparse_output` in version 1.2 and will be remo
ved in 1.4. `sparse_output` is ignored unless you leave `sparse` to its default valu
e.
  warnings.warn(

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[111], line 1
----> 1 X_train_new = ohe.fit_transform(X_train[['fuel','owner']]).toarray()

AttributeError: 'numpy.ndarray' object has no attribute 'toarray'
```

```
In [112]: X_test_new = ohe.transform(X_test[['fuel','owner']]).toarray()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[112], line 1
----> 1 X_test_new = ohe.transform(X_test[['fuel','owner']]).toarray()

AttributeError: 'numpy.ndarray' object has no attribute 'toarray'
```

```
In [113]: X_train_new
```

```
Out[113]: array([[1, 0, 0, ..., 0, 0, 0],
                 [1, 0, 0, ..., 0, 0, 0],
                 [0, 0, 1, ..., 0, 0, 0],
                 ...,
                 [0, 0, 1, ..., 0, 0, 0],
                 [1, 0, 0, ..., 1, 0, 0],
                 [1, 0, 0, ..., 0, 0, 0]])
```

```
In [114]: X_train_new.shape
```

```
Out[114]: (6502, 7)
```

```
In [116]: np.hstack((X_train[['brand','km_driven']].values,X_train_new))
```

```
Out[116]: array([['Hyundai', 35000, 1, ..., 0, 0, 0],
                 ['Jeep', 60000, 1, ..., 0, 0, 0],
                 ['Hyundai', 25000, 0, ..., 0, 0, 0],
                 ...,
                 ['Tata', 15000, 0, ..., 0, 0, 0],
                 ['Maruti', 32500, 1, ..., 1, 0, 0],
                 ['Isuzu', 121000, 1, ..., 0, 0, 0]], dtype=object)
```

```
In [ ]:
```