



```
In [1]: import numpy as np
import pandas as pd
```

One Hot Encoding

One hot encoding is one method of converting data to prepare it for an algorithm and get a better prediction. With one-hot, we convert each categorical value into a new categorical column and assign a binary value of 1 or 0 to those columns. Each integer value is represented as a binary vector.

```
In [2]: df=pd.read_csv('cars.csv')
```

```
In [3]: df.sample(5)
```

```
Out[3]:
```

	brand	km_driven	fuel	owner	selling_price
6802	Hyundai	70000	Petrol	Second Owner	200000
2349	Maruti	46000	Petrol	Second Owner	195000
4309	Maruti	120000	Petrol	Third Owner	125000
2822	Maruti	79000	Diesel	First Owner	825000
472	Tata	110000	Diesel	Second Owner	200000

```
In [4]: df['brand'].value_counts()
```

```
Out[4]:
```

Maruti	2448
Hyundai	1415
Mahindra	772
Tata	734
Toyota	488
Honda	467
Ford	397
Chevrolet	230
Renault	228
Volkswagen	186
BMW	120
Skoda	105
Nissan	81
Jaguar	71
Volvo	67
Datsun	65
Mercedes-Benz	54
Fiat	47
Audi	40
Loading [MathJax]/extensions/Safe.js	34

```

Jeep          31
Mitsubishi    14
Force         6
Land          6
Isuzu         5
Kia           4
Ambassador    4
Daewoo        3
MG            3
Ashok         1
Opel          1
Peugeot       1
Name: brand, dtype: int64

```

```
In [5]: df['brand'].nunique()
```

```
Out[5]: 32
```

```
In [6]: df['fuel'].value_counts()
```

```

Out[6]: Diesel    4402
        Petrol    3631
        CNG       57
        LPG       38
        Name: fuel, dtype: int64

```

```
In [7]: df['owner'].value_counts()
```

```

Out[7]: First Owner      5289
        Second Owner    2105
        Third Owner     555
        Fourth & Above Owner 174
        Test Drive Car     5
        Name: owner, dtype: int64

```

One Hot Encoding using pandas

```
In [53]: pd.get_dummies(df, columns=['fuel', 'owner'])
```

```

Out[53]:

```

	brand	km_driven	selling_price	fuel_CNG	fuel_Diesel	fuel_LPG	fuel_Petrol	owner_First Owner	owner_Fourth & Above Owner
0	Maruti	145500	450000	0	1	0	0	1	0
1	Skoda	120000	370000	0	1	0	0	0	0
2	Honda	140000	158000	0	0	0	1	0	0
3	Hyundai	127000	225000	0	1	0	0	1	0
4	Maruti	120000	130000	0	0	0	1	1	0
...
8123	Hyundai	110000	320000	0	0	0	1	1	0
8124	Hyundai	119000	135000	0	1	0	0	0	1
8125	Maruti	120000	382000	0	1	0	0	1	0
8126	Tata	25000	290000	0	1	0	0	1	0
Loading [MathJax]/extensions/Safe.js		25000	290000	0	1	0	0	1	0

8128 rows × 12 columns

k-1 OneHotEncoding

```
In [9]: pd.get_dummies(df, columns=['fuel', 'owner'], drop_first=True)
```

```
Out[9]:
```

	km_driven	selling_price	fuel_Diesel	fuel_LPG	fuel_Petrol	owner_Fourth & Above Owner	owner_Second Owner	owner_Test Drive Car	owr
0	145500	450000	1	0	0	0	0	0	
1	120000	370000	1	0	0	0	1	0	
2	140000	158000	0	0	1	0	0	0	
3	127000	225000	1	0	0	0	0	0	
4	120000	130000	0	0	1	0	0	0	
...
8123	110000	320000	0	0	1	0	0	0	
8124	119000	135000	1	0	0	1	0	0	
8125	120000	382000	1	0	0	0	0	0	
8126	25000	290000	1	0	0	0	0	0	
8127	25000	290000	1	0	0	0	0	0	

8128 rows × 40 columns

```
In [ ]:
```

One Hot Encoding using sklearn

```
In [10]: 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_size=0.2,
```

```
In [11]: df.head()
```

```
Out[11]:
```

	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 [13]: x_train
```

```
Out[13]:
```

	brand	km_driven	fuel	owner
--	-------	-----------	------	-------

	brand	km_driven	fuel	owner
3042	Hyundai	60000	LPG	First Owner
1520	Tata	150000	Diesel	Third Owner
2611	Hyundai	110000	Diesel	Second Owner
3544	Mahindra	28000	Diesel	Second Owner
4138	Maruti	15000	Petrol	First Owner
...
4931	Tata	70000	Diesel	Third Owner
3264	Ford	100000	Diesel	Second Owner
1653	Hyundai	90000	Petrol	Second Owner
2607	Volkswagen	90000	Diesel	First Owner
2732	Hyundai	110000	Petrol	First Owner

6502 rows × 4 columns

In [14]: x_test

Out[14]:

	brand	km_driven	fuel	owner
3558	Hyundai	40000	Diesel	First Owner
233	Mahindra	70000	Diesel	First Owner
7952	Maruti	5000	Petrol	First Owner
572	Maruti	120000	Petrol	Third Owner
6960	Lexus	20000	Petrol	First Owner
...
7576	Fiat	100000	Diesel	Third Owner
1484	Maruti	120000	Petrol	Third Owner
1881	Maruti	40000	Diesel	First Owner
4917	Hyundai	2350	Petrol	First Owner
5934	Hyundai	80000	Diesel	Second Owner

1626 rows × 4 columns

In [12]: `from sklearn.preprocessing import OneHotEncoder`

In [38]: `ohe=OneHotEncoder(drop='first', sparse=False, dtype=np.int32)`

In [39]: `x_train_new=ohe.fit_transform(x_train[['fuel', 'owner']])`
`x_test_new=ohe.fit_transform(x_test[['fuel', 'owner']])`

In [40]: x_train_new

Out[40]: array([[0, 1, 0, ..., 0, 0, 0],
[0, ..., 0, 0, 1],

```
[1, 0, 0, ..., 1, 0, 0],
...,
[0, 0, 1, ..., 1, 0, 0],
[1, 0, 0, ..., 0, 0, 0],
[0, 0, 1, ..., 0, 0, 0]])
```

```
In [41]: np.hstack((x_train[['brand', 'km_driven']].values, x_train_new))
```

```
Out[41]: array([[ 'Hyundai', 600000, 0, ..., 0, 0, 0],
      [ 'Tata', 150000, 1, ..., 0, 0, 1],
      [ 'Hyundai', 110000, 1, ..., 1, 0, 0],
      ...,
      [ 'Hyundai', 900000, 0, ..., 1, 0, 0],
      [ 'Volkswagen', 90000, 1, ..., 0, 0, 0],
      [ 'Hyundai', 110000, 0, ..., 0, 0, 0]], dtype=object)
```

```
In [42]: np.hstack((x_train[['brand', 'km_driven']].values, x_train_new)).shape
```

```
Out[42]: (6502, 9)
```

One Hot Encoding with top Categories of brand column

```
In [47]: counts=df['brand'].value_counts()
```

```
In [48]: df['brand'].nunique()
threshold=100
```

```
In [51]: repl=counts[counts<threshold].index
```

```
In [52]: pd.get_dummies(df['brand']).replace(repl, 'uncommon')
```

```
Out[52]:
```

	Ambassador	Ashok	Audi	BMW	Chevrolet	Daewoo	Datsun	Fiat	Force	Ford	...	Mitsubishi	Nissan	C
0	0	0	0	0	0	0	0	0	0	0	...	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	
...	
8123	0	0	0	0	0	0	0	0	0	0	...	0	0	
8124	0	0	0	0	0	0	0	0	0	0	...	0	0	
8125	0	0	0	0	0	0	0	0	0	0	...	0	0	
8126	0	0	0	0	0	0	0	0	0	0	...	0	0	
8127	0	0	0	0	0	0	0	0	0	0	...	0	0	

8128 rows × 32 columns

Thank you

© [Mt Learners](#) 2022. All rights reserved.