Importing the libraries

```
import numpy as np
import pandas as pd
from sklearn.impute import SimpleImputer
```

Importing the dataset

```
dataset = pd.read_csv("car_ad.csv", encoding ="cp1252")
```

```
dataset.head()
```

	car	price	body	mileage	engV	engType	registration	year	model	d
0	Ford	15500.0	crossover	68	2.5	Gas	yes	2010	Kuga	
1	Mercedes-Benz	20500.0	sedan	173	1.8	Gas	yes	2011	E-Class	
2	Mercedes-Benz	35000.0	other	135	5.5	Petrol	yes	2008	CL 550	
	Mercedes-									

Double-click (or enter) to edit

Droping the drive values having NaN

```
dataset = dataset.dropna(subset=['drive'])
```

```
dataset.describe()
```

```
                 price      mileage         engV         year
```

Taking care of missing data

```
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer.fit(dataset[['engV']])
dataset[['engV']] = imputer.transform(dataset[['engV']])
```

```
imputer = SimpleImputer(missing_values=0, strategy='mean')
imputer.fit(dataset[['mileage']])
dataset[['mileage']] = imputer.transform(dataset[['mileage']])
print(dataset)
imputer.fit(dataset[['price']])
dataset[['price']] = imputer.transform(dataset[['price']])
print(dataset[['price']])
```

```
                car    price       body  mileage  engV engType registration  \
0              Ford  15500.0  crossover     68.0   2.5     Gas          yes
1     Mercedes-Benz  20500.0      sedan    173.0   1.8     Gas          yes
2     Mercedes-Benz  35000.0      other    135.0   5.5  Petrol          yes
3     Mercedes-Benz  17800.0        van    162.0   1.8  Diesel          yes
5            Nissan  16600.0  crossover     83.0   2.0  Petrol          yes
...             ...      ...        ...      ...   ...     ...          ...
9571        Hyundai  14500.0  crossover    140.0   2.0     Gas          yes
9572     Volkswagen   2200.0      vagon    150.0   1.6  Petrol          yes
9573  Mercedes-Benz  18500.0  crossover    180.0   3.5  Petrol          yes
9574          Lexus  16999.0      sedan    150.0   3.5     Gas          yes
9575           Audi  22500.0      other     71.0   3.6  Petrol          yes

      year      model  drive
0     2010       Kuga   full
1     2011    E-Class   rear
2     2008     CL 550   rear
3     2012      B 180  front
5     2013     X-Trail   full
...    ...        ...    ...
9571  2011     Tucson  front
9572  1986  Passat B2  front
9573  2008     ML 350   full
9574  2008     ES 350  front
9575  2007         Q7   full

[9065 rows x 10 columns]
        price
0     15500.0
1     20500.0
2     35000.0
3     17800.0
5     16600.0
...       ...
9571  14500.0
9572   2200.0
9573  18500.0
9574  16999.0
9575  22500.0
```

```
[9065 rows x 1 columns]
```

```
print(dataset.describe(include='all'))
```

```
               car          price   body      mileage         engV engType  \
count         9065    9065.000000   9065  9065.000000  9065.000000    9065
unique          83            NaN      6          NaN          NaN       4
top     Volkswagen            NaN  sedan          NaN          NaN  Petrol
freq           879            NaN   3444          NaN          NaN    4181
mean           NaN   16229.235289    NaN   144.568958     2.588607     NaN
std            NaN   24202.479898    NaN    94.391987     5.318369     NaN
min            NaN     259.350000    NaN     1.000000     0.100000     NaN
25%            NaN    5500.000000    NaN    80.000000     1.600000     NaN
50%            NaN    9900.000000    NaN   136.000000     2.000000     NaN
75%            NaN   16800.000000    NaN   195.000000     2.588607     NaN
max            NaN  547800.000000    NaN   999.000000    99.990000     NaN

       registration         year    model  drive
count          9065  9065.000000     9065   9065
unique            2          NaN      863      3
top             yes          NaN  E-Class  front
freq           8542          NaN      185   5188
mean            NaN  2006.638941      NaN    NaN
std             NaN     7.001318      NaN    NaN
min             NaN  1953.000000      NaN    NaN
25%             NaN  2004.000000      NaN    NaN
50%             NaN  2008.000000      NaN    NaN
75%             NaN  2012.000000      NaN    NaN
max             NaN  2016.000000      NaN    NaN
```

Encoding categorial data

```
from sklearn.preprocessing import LabelEncoder
LE_dataset = LabelEncoder()
dataset['car']= LE_dataset.fit_transform(dataset['car'])
dataset['body']= LE_dataset.fit_transform(dataset['body'])
dataset['engType']= LE_dataset.fit_transform(dataset['engType'])
dataset['registration']= LE_dataset.fit_transform(dataset['registration'])
dataset['registration']= LE_dataset.fit_transform(dataset['registration'])
dataset['drive']= LE_dataset.fit_transform(dataset['drive'])
print(dataset)
```

```
      car    price  body  mileage  engV  engType  registration  year  \
0      23  15500.0     0     68.0   2.5        1             1  2010
1      50  20500.0     3    173.0   1.8        1             1  2011
2      50  35000.0     2    135.0   5.5        3             1  2008
3      50  17800.0     5    162.0   1.8        0             1  2012
5      55  16600.0     0     83.0   2.0        3             1  2013
...   ...      ...   ...      ...   ...      ...           ...   ...
9571   33  14500.0     0    140.0   2.0        1             1  2011
9572   77   2200.0     4    150.0   1.6        3             1  1986
9573   50  18500.0     0    180.0   3.5        3             1  2008
9574   43  16999.0     3    150.0   3.5        1             1  2008
```

```
9575    4  22500.0     2     71.0   3.6        3             1  2007

          model  drive
0          Kuga      1
1       E-Class      2
2        CL 550      2
3         B 180      0
5       X-Trail      1
...         ...    ...
9571     Tucson      0
9572  Passat B2      0
9573     ML 350      1
9574     ES 350      0
9575         Q7      1

[9065 rows x 10 columns]
```

Getting values in variables

```
x = dataset[['car','body','mileage','engV','engType','registration','year','model','dri
print(x)
y = dataset[['price']]
print(y)
```

```
      car  body  mileage  engV  engType  registration  year      model  drive
0      23     0     68.0   2.5        1             1  2010       Kuga      1
1      50     3    173.0   1.8        1             1  2011    E-Class      2
2      50     2    135.0   5.5        3             1  2008     CL 550      2
3      50     5    162.0   1.8        0             1  2012      B 180      0
5      55     0     83.0   2.0        3             1  2013    X-Trail      1
...   ...   ...      ...   ...      ...           ...   ...        ...    ...
9571   33     0    140.0   2.0        1             1  2011     Tucson      0
9572   77     4    150.0   1.6        3             1  1986  Passat B2      0
9573   50     0    180.0   3.5        3             1  2008     ML 350      1
9574   43     3    150.0   3.5        1             1  2008     ES 350      0
9575    4     2     71.0   3.6        3             1  2007         Q7      1

[9065 rows x 9 columns]
        price
0     15500.0
1     20500.0
2     35000.0
3     17800.0
5     16600.0
...       ...
9571  14500.0
9572   2200.0
9573  18500.0
9574  16999.0
9575  22500.0

[9065 rows x 1 columns]
```

Splitting into test and train

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state
print(x_test)
print(x_train)
print(y_train)
print(y_test)
```

```
      car  body     mileage  engV  engType  registration  year  \
6159   59     2    1.000000   1.5        0             1  2007   
2959   39     3   73.000000   1.6        3             1  2008   
2474   74     3  170.000000   3.0        3             1  2004   
5256    4     3  370.000000   2.6        2             1  1997   
8328   56     1  160.000000   1.6        3             1  2005   
...   ...   ...         ...   ...      ...           ...   ...   
2119   70     4  144.568958   2.0        0             1  2016   
1028   74     0  144.568958   4.5        0             1  2016   
6161   12     3  103.000000   1.5        3             1  2005   
7144   76     3   67.000000   1.6        3             1  2012   
762    59     3  310.000000   1.4        3             1  2003   

                model  drive  
6159     Kangoo ãðóç.      0  
2959           Cerato      0  
2474            Camry      0  
5256               A6      0  
8328          Astra H      0  
...               ...    ...  
2119          Outback      1  
1028  Land Cruiser 200      1  
6161             Aveo      0  
7144             2115      0  
762            Symbol      0  

[1813 rows x 9 columns]
      car  body  mileage  engV  engType  registration  year         model  \
2967   23     1     71.0   1.0        3             1  2013        Fiesta   
9494   23     1     70.0   1.6        3             1  2006         Focus   
1605   58     0    115.0   4.8        3             1  2010       Cayenne   
9221   59     4    100.0   1.5        0             1  2009  Kangoo ïàññ.   
5785   55     5     90.0   1.6        1             1  2008          Note   
...   ...   ...      ...   ...      ...           ...   ...           ...   
3046    5     0    200.0   3.0        0             1  2009            X5   
8243   56     4    290.0   1.4        3             1  1994       Astra F   
947    77     3     87.0   1.6        3             1  2011          Polo   
5476   33     0     90.0   2.7        1             1  2008      Santa FE   
248    67     3    109.0   1.8        3             1  2012    Octavia A5   

      drive  
2967      0  
9494      0  
1605      1  
9221      0  
5785      0  
...     ...  
3046      1  
8243      0  
947       0  
5476      1  
```

```
248       0

[7252 rows x 9 columns]
             price
2967  11500.000000
9494   7100.000000
```

✓ 0s completed at 5:00 PM