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

# load data from file
data = pd.read_csv("car.csv",
                   names=['buying', 'maint', 'doors', 'persons', 'lug_boot', 'safety', 'cl
data
```

Out[1]:

	buying	maint	doors	persons	lug_boot	safety	class
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc
...	...	...	...	...	...	...	...
1723	low	low	5more	more	med	med	good
1724	low	low	5more	more	med	high	vgood
1725	low	low	5more	more	big	low	unacc
1726	low	low	5more	more	big	med	good
1727	low	low	5more	more	big	high	vgood

1728 rows × 7 columns

In [2]:
```
from sklearn.preprocessing import OrdinalEncoder

# prepare data - convert categories to numbers
enc = OrdinalEncoder()
data[['maint', 'doors', 'lug_boot', 'safety', 'class']] = \
    enc.fit_transform(data[['maint','doors', 'lug_boot', 'safety', 'class']])
enc.categories_
```

Out[2]:
```
[array(['high', 'low', 'med', 'vhigh'], dtype=object),
 array(['2', '3', '4', '5more'], dtype=object),
 array(['big', 'med', 'small'], dtype=object),
 array(['high', 'low', 'med'], dtype=object),
 array(['acc', 'good', 'unacc', 'vgood'], dtype=object)]
```

In [3]:
```
data
```

Out[3]:

	buying	maint	doors	persons	lug_boot	safety	class
0	vhigh	3.0	0.0	2	2.0	1.0	2.0
1	vhigh	3.0	0.0	2	2.0	2.0	2.0
2	vhigh	3.0	0.0	2	2.0	0.0	2.0
3	vhigh	3.0	0.0	2	1.0	1.0	2.0
4	vhigh	3.0	0.0	2	1.0	2.0	2.0
...	...	...	...	...	...	...	...

	buying	maint	doors	persons	lug_boot	safety	class
1723	low	1.0	3.0	more	1.0	2.0	1.0
1724	low	1.0	3.0	more	1.0	0.0	3.0
1725	low	1.0	3.0	more	0.0	1.0	2.0
1726	low	1.0	3.0	more	0.0	2.0	1.0
1727	low	1.0	3.0	more	0.0	0.0	3.0

1728 rows × 7 columns

In [4]:
```
# remove persons column as it will not be used
data.drop(['persons'], axis=1, inplace=True)
x = np.array(data.drop(['buying'], axis=1))
y = np.array(data['buying'])

# train and test model using decision tree
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import mean_absolute_error

xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2)

model = DecisionTreeClassifier()
model.fit(xtrain, ytrain)
predictions = model.predict(xtest)

print("Accuracy of DecisionTreeClassifier : ",
      model.score(xtest, predictions))
```

```
Accuracy of DecisionTreeClassifier :  1.0
```

In [5]:
```
# predict for maintenance = High, number of doors = 4,
# lug boot size = big, safety = high, class value = Good
xnew = [[0, 2, 0, 0, 1]]
ynew = model.predict(xnew)
print("Predicted price (decision tree):",ynew)
```

```
Predicted price (decision tree): ['low']
```

In [6]:
```
# try MLP model
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score

classifier = MLPClassifier(hidden_layer_sizes=(64,16), max_iter=500,
                           activation = 'relu',solver='adam')
classifier.fit(xtrain, ytrain)
y_pred = classifier.predict(xtest)
print("Accuracy of MLPClassifier : ", accuracy_score(y_pred, ytest))
```

```
Accuracy of MLPClassifier :  0.2774566473988439
```

In [7]:
```
# predict for maintenance = High, number of doors = 4,
# lug boot size = big, safety = high, class value = Good
Xnew = [[0, 2, 0, 0, 1]]
Ynew = classifier.predict(Xnew)
print("Predicted price (MLP):",Ynew)
```

```
Predicted price (MLP): ['med']
```

In [ ]: