```
import pandas as pd
import numpy as np
import random
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
dataFile = pd.read_csv('car.data')
```

```
dataFile.head()
```

	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

```
dataFile.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1728 entries, 0 to 1727
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   buying    1728 non-null   object
 1   maint     1728 non-null   object
 2   doors     1728 non-null   object
 3   persons   1728 non-null   object
 4   lug_boot  1728 non-null   object
 5   safety    1728 non-null   object
 6   class     1728 non-null   object
dtypes: object(7)
memory usage: 94.6+ KB
```

```
buying =  list(dataFile["buying"])
maint  =  list(dataFile["maint"])
doors  =  list(dataFile["doors"])
persons =  list(dataFile["persons"])
lug_boot =   list(dataFile["lug_boot"])
safety =  list(dataFile["safety"])
car_class =   list(dataFile["class"])
```

```
buying_encoded=le.fit_transform(buying)
maint_encoded=le.fit_transform(maint)
doors_encoded=le.fit_transform(doors)
persons_encoded=le.fit_transform(persons)
lug_boot_encoded=le.fit_transform(lug_boot)
safety_encoded=le.fit_transform(safety)
car_class_encoded=le.fit_transform(car_class)
```

```
data=list(zip(buying_encoded,maint_encoded,doors_encoded,persons_encoded,lug_boot_encoded,safety_encoded))
print(data)
```

```
[(3, 3, 0, 0, 2, 1), (3, 3, 0, 0, 2, 2), (3, 3, 0, 0, 2, 0), (3, 3, 0, 0, 1, 1), (3, 3, 0, 0, 1, 2), (3, 3, 0, 0, 1, 0), (3, 3, 0,
```

```
label=le.fit_transform(car_class_encoded)
print(label)
```

```
[2 2 2 ... 2 1 3]
```

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(data, label, test_size=1382)
```

```
for i in range (3,150):
  neigh = KNeighborsClassifier(n_neighbors=i,p=2)
```

```
neigh.fit(x_train,y_train)
neigh.score(x_test,y_test)
print(i, " " ,neigh.score(x_test,y_test))
```

```
3   0.7901591895803184
4   0.776410998552822
5   0.7829232995658466
6   0.7619392185238785
7   0.7785817655571635
8   0.7677279305354558
9   0.7713458755426917
10   0.7633863965267728
11   0.7597684515195369
12   0.743849493487699
13   0.743849493487699
14   0.7351664254703328
15   0.7344428364688856
16   0.7279305354558611
17   0.7228654124457308
18   0.7199710564399421
19   0.7134587554269175
20   0.7149059334298119
21   0.7127351664254703
22   0.7091172214182344
23   0.7047756874095513
24   0.7091172214182344
25   0.7069464544138929
26   0.7069464544138929
27   0.7069464544138929
28   0.7069464544138929
29   0.703328509406657
30   0.7054992764109985
31   0.7047756874095513
32   0.7040520984081042
33   0.7011577424023154
34   0.7047756874095513
35   0.7069464544138929
36   0.7062228654124457
37   0.7054992764109985
38   0.7040520984081042
39   0.7026049204052098
40   0.703328509406657
41   0.6997105643994211
42   0.7004341534008683
43   0.7018813314037626
44   0.6997105643994211
45   0.7018813314037626
46   0.7011577424023154
47   0.7004341534008683
48   0.7026049204052098
49   0.7026049204052098
50   0.7018813314037626
51   0.7040520984081042
52   0.7040520984081042
53   0.7026049204052098
54   0.703328509406657
55   0.7026049204052098
56   0.7026049204052098
57   0.7040520984081042
58   0.7026049204052098
59   0.7026049204052098
60   0.7040520984081042
61   0.703328509406657
```

```
0.7901591895803184
```

✓ 18 s ukończono o 11:22