```
import requests
import pandas as pd
from sklearn import metrics
from sklearn.metrics import confusion_matrix

%matplotlib inline

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/car/car.data'
data = pd.read_csv(url, names=['buying', 'maint', 'doors', 'persons', 'lug_boot', 'safety', 'class'])
data = data.drop(['persons'], axis=1)
data.head()
```

	buying	maint	doors	lug_boot	safety	class
0	vhigh	vhigh	2	small	low	unacc
1	vhigh	vhigh	2	small	med	unacc
2	vhigh	vhigh	2	small	high	unacc
3	vhigh	vhigh	2	med	low	unacc
4	vhigh	vhigh	2	med	med	unacc

```
print(data['buying'].unique())
print(data['class'].unique())
```

```
['vhigh' 'high' 'med' 'low']
['unacc' 'acc' 'vgood' 'good']
```

```
from sklearn.preprocessing import OrdinalEncoder
buying_price_category = ['vhigh', 'high', 'med', 'low']
maint_cost_category = ['low', 'med', 'high', 'vhigh']
doors_category = ['2', '3', '4', '5more']
person_capacity_category = ['2', '4', 'more']
lug_boot_category = ['small', 'med', 'big']
safety_category = ['low', 'med', 'high']
class_category = ['unacc', 'acc', 'vgood', 'good']
all_categories = [buying_price_category, maint_cost_category,doors_category,lug_boot_category,safety_category,class_category]
oe = OrdinalEncoder(categories= all_categories)
```

```
tf_data = oe.fit_transform( data)
X = tf_data[:,1:]
y = tf_data[:,0]

case = [['med', 'high', '4', 'big', 'high', 'good']]
tf_case = oe.fit_transform(case)
```

+ Code + Text

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state=143)


from sklearn.tree import DecisionTreeClassifier
DT_classifier = DecisionTreeClassifier( criterion='gini', max_depth= 8, min_samples_split= 7)
DT_classifier.fit(X_train, y_train)
```

```
DecisionTreeClassifier(max_depth=8, min_samples_split=7)
```

```
y_pred = DT_classifier.predict(X_test)


print(confusion_matrix(y_test, y_pred))
print(metrics.classification_report(y_test, y_pred))
```

```
[[50 43 30 10]
 [68 10 41  9]
 [43 26 17 31]
 [41 20 59 21]]
              precision    recall  f1-score   support

         0.0       0.25      0.38      0.30       133
         1.0       0.10      0.08      0.09       128
         2.0       0.12      0.15      0.13       117
         3.0       0.30      0.15      0.20       141

    accuracy                           0.19       519
   macro avg       0.19      0.19      0.18       519
weighted avg       0.19      0.19      0.18       519
```

```
case_pred = DT_classifier.predict(tf_case[:,1:])
```

```
print(buying_price_category[int(case_pred[0])])
```

```
low
```