

```
In [1]: import pandas as pd
import requests

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/car/car.data"
df = pd.read_csv(url )
df.columns = ["buying", "maint", "doors", "persons", "lug_boot", "safety", "class"]
df.head(5)
```

Out[1]:

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

```
In [2]: # encode the Labels:
# vhigh=4 high=3 med=2 low=1
# 5more=6 more =5
# small=1 med=2 big=3
# unacc=1 acc=2 good=3 vgood=4
```

```
df = df.replace('vhigh',4)
df = df.replace('high',3)
df = df.replace('med',2)
df = df.replace('low',1)
df = df.replace('5more',6)
df = df.replace('more',5)
df = df.replace('small',1)
df = df.replace('big',3)
df = df.replace('vgood',4)
df = df.replace('good',3)
df = df.replace('acc',2)
df = df.replace('unacc',1)
df.describe
```

```
Out[2]: <bound method NDFrame.describe of          buying  maint  doors  persons  lug_boot
safety  class
0          4      4      2      2          1      2      1
1          4      4      2      2          1      3      1
2          4      4      2      2          2      1      1
3          4      4      2      2          2      2      1
4          4      4      2      2          2      3      1
...      ...      ...      ...      ...      ...      ...      ...
1722       1      1      6      5          2      2      3
1723       1      1      6      5          2      3      4
1724       1      1      6      5          3      1      1
1725       1      1      6      5          3      2      3
1726       1      1      6      5          3      3      4
```

```
[1727 rows x 7 columns]>
```

```
In [3]: from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix

X = df[["maint", "doors", "lug_boot", "safety", "class"]]
y = df["buying"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.8, random
```

```
In [4]: DT_classifier = DecisionTreeClassifier( criterion= 'gini', max_depth= 5, min_samp
DT_classifier.fit(X_train, y_train)

y_pred = DT_classifier.predict(X_test)

confusion_matrix(y_test, y_pred)
a = confusion_matrix(y_test, y_pred).ravel()
# print accuracy rate
print((a[0]+a[5]+a[10]+a[15])/a.sum())

0.3154848046309696
```

```
In [5]: from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix

RF_classifier = RandomForestClassifier(n_estimators=5,criterion= 'gini',max_depth

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.7, random
RF_classifier.fit(X_train, y_train)

y_predrf = RF_classifier.predict(X_test)

a = confusion_matrix(y_test, y_predrf).ravel()
print((a[0]+a[5]+a[10]+a[15])/a.sum())

0.29280397022332505
```

```
In [6]: # Maintenance = High
# Number of doors = 4
# Lug Boot Size = Big
# Safety = High
# Class Value = Good

data = [[3,4,3,3,3]]

# Create the pandas DataFrame
ndf = pd.DataFrame(data, columns = ["maint", "doors", "lug_boot", "safety", "class"])

# print dataframe.

print(DT_classifier.predict(ndf))

print(RF_classifier.predict(ndf))

[1]
[1]
```

```
In [ ]: # both predict it's low price
```