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

df=pd.read_csv('car_evaluation.csv')
```

In [2]:
```
df
```

Out[2]:

	vhigh	vhigh.1	2	2.1	small	low	unacc
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
...	...	...	...	...	...	...	...
1722	low	low	5more	more	med	med	good
1723	low	low	5more	more	med	high	vgood
1724	low	low	5more	more	big	low	unacc
1725	low	low	5more	more	big	med	good
1726	low	low	5more	more	big	high	vgood

1727 rows × 7 columns

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
vhigh      0
vhigh.1    0
2          0
2.1        0
small      0
low        0
unacc      0
dtype: int64
```

In [5]:
```
df.duplicated().sum()
```

Out[5]:
```
np.int64(0)
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1727 entries, 0 to 1726
Data columns (total 7 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   vhigh    1727 non-null   object
 1   vhigh.1  1727 non-null   object
 2   2        1727 non-null   object
 3   2.1      1727 non-null   object
 4   small    1727 non-null   object
 5   low      1727 non-null   object
 6   unacc    1727 non-null   object
dtypes: object(7)
memory usage: 94.6+ KB
```

In [7]:
```
cols=['buying','maintainence','doors','persons','lug_boot','safety','clas
df.columns=cols
```

In [8]:
```
df
```

Out[8]:

	buying	maintainence	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
...	...	...	...	...	...	...	...
1722	low	low	5more	more	med	med	good
1723	low	low	5more	more	med	high	vgood
1724	low	low	5more	more	big	low	unacc
1725	low	low	5more	more	big	med	good
1726	low	low	5more	more	big	high	vgood

1727 rows × 7 columns

In [12]:
```
for i in df.columns:
    print(df[i].value_counts(),'\n')
```

```
buying
high     432
med      432
low      432
vhigh    431
Name: count, dtype: int64

maintainence
high     432
med      432
low      432
vhigh    431
Name: count, dtype: int64

doors
3        432
4        432
5more    432
2        431
Name: count, dtype: int64

persons
4       576
more    576
2       575
Name: count, dtype: int64

lug_boot
med      576
big      576
small    575
Name: count, dtype: int64

safety
med     576
high    576
low     575
Name: count, dtype: int64

class
unacc    1209
acc       384
good       69
vgood      65
Name: count, dtype: int64
```

In [16]:
```
from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

for i in df.columns:
    df[i]=le.fit_transform(df[i])
```

In [17]:
```
df
```

Out[17]:

	buying	maintainence	doors	persons	lug_boot	safety	class
0	3	3	0	0	2	2	2
1	3	3	0	0	2	0	2
2	3	3	0	0	1	1	2
3	3	3	0	0	1	2	2
4	3	3	0	0	1	0	2
...	...	...	...	...	...	...	...
1722	1	1	3	2	1	2	1
1723	1	1	3	2	1	0	3
1724	1	1	3	2	0	1	2
1725	1	1	3	2	0	2	1
1726	1	1	3	2	0	0	3

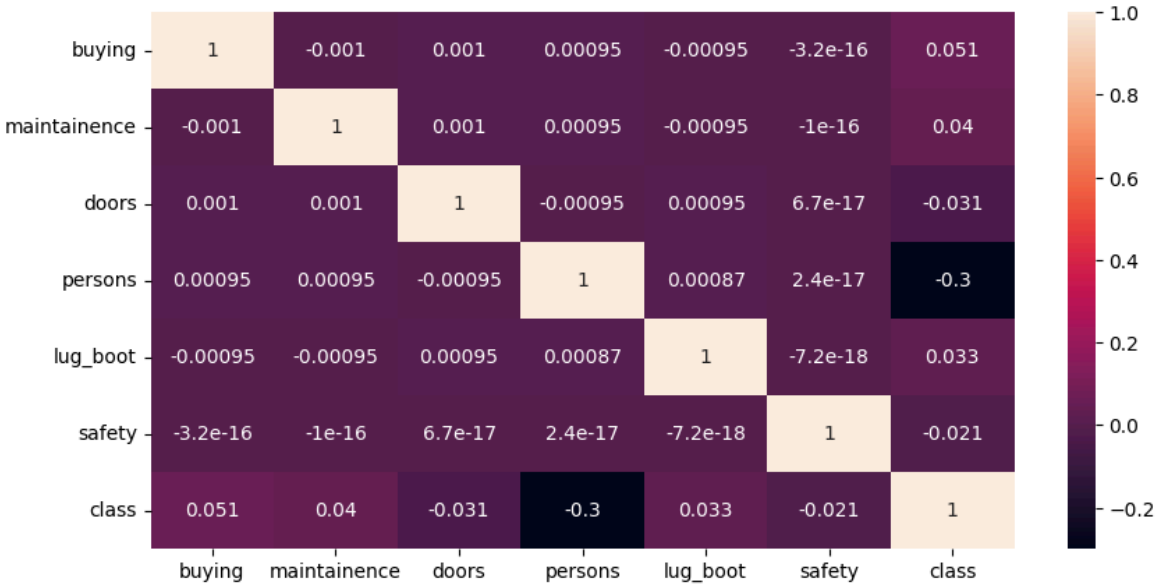
1727 rows × 7 columns

In [35]:
```
corr=df.corr()

import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10,5))
sns.heatmap(corr,annot=True)
```

Out[35]: <Axes: >



In [19]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

model=RandomForestClassifier()
```

In [21]:
```
x=df.drop(['class'],axis=1)
y=df['class']
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_s
```

In [22]:
```
model.fit(x_train,y_train)
```

Out[22]:
```
RandomForestClassifier()
```

In [25]:
```
from sklearn.metrics import classification_report

y_pred=model.predict(x_test)
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.93      0.95      0.94        87
           1       0.94      0.94      0.94        16
           2       0.99      0.98      0.98       230
           3       1.00      0.92      0.96        13

    accuracy                           0.97       346
   macro avg       0.96      0.95      0.96       346
weighted avg       0.97      0.97      0.97       346
```