

svm-using-car-acceptability

March 30, 2025

```
[5]: import pandas as pd
from sklearn.svm import SVC
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
df=pd.read_csv("/content/car_acceptability.csv")
df
```

```
[5]:
```

	buying	price	maintenance	cost	number of doors	number of persons	\
0		vhigh		vhigh	2		2
1		vhigh		vhigh	2		2
2		vhigh		vhigh	2		2
3		vhigh		vhigh	2		2
4		vhigh		vhigh	2		2
...	
1722		low		low	5		5
1723		low		low	5		5
1724		low		low	5		5
1725		low		low	5		5
1726		low		low	5		5

	lug_boot	safety	evaluation
0	small	med	unacc
1	small	high	unacc
2	med	low	unacc
3	med	med	unacc
4	med	high	unacc
...
1722	med	med	good
1723	med	high	vgood
1724	big	low	unacc
1725	big	med	good
1726	big	high	vgood

[1727 rows x 7 columns]

```
[16]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1727 entries, 0 to 1726
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   buying price          1727 non-null   int64
1   maintenance cost      1727 non-null   int64
2   number of doors        1727 non-null   int64
3   number of persons      1727 non-null   int64
4   lug_boot              1727 non-null   int64
5   safety                1727 non-null   int64
6   evaluation             1727 non-null   object
dtypes: int64(6), object(1)
memory usage: 94.6+ KB
```

```
[6]: df.isnull()
```

```
[6]:
```

	buying price	maintenance cost	number of doors	number of persons	\
0	False	False	False	False	
1	False	False	False	False	
2	False	False	False	False	
3	False	False	False	False	
4	False	False	False	False	
...	
1722	False	False	False	False	
1723	False	False	False	False	
1724	False	False	False	False	
1725	False	False	False	False	
1726	False	False	False	False	

	lug_boot	safety	evaluation
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
...
1722	False	False	False
1723	False	False	False
1724	False	False	False
1725	False	False	False
1726	False	False	False

[1727 rows x 7 columns]

```
[8]: df=df.drop_duplicates()
df
```

```
[8]:      buying price  maintenance cost  number of doors  number of persons \
0           3           3           2           2
1           3           3           2           2
2           3           3           2           2
3           3           3           2           2
4           3           3           2           2
...
1722        1           1           5           5
1723        1           1           5           5
1724        1           1           5           5
1725        1           1           5           5
1726        1           1           5           5
```

```
      lug_boot  safety evaluation
0           2           2      unacc
1           2           0      unacc
2           1           1      unacc
3           1           2      unacc
4           1           0      unacc
...
1722        1           2      good
1723        1           0     vgood
1724        0           1      unacc
1725        0           2      good
1726        0           0     vgood
```

[1727 rows x 7 columns]

```
[7]: l=LabelEncoder()
df['buying price']=l.fit_transform(df['buying price'])
df['maintenance cost']=l.fit_transform(df['maintenance cost'])
df['lug_boot']=l.fit_transform(df['lug_boot'])
df['safety']=l.fit_transform(df['safety'])
df
```

```
[7]:      buying price  maintenance cost  number of doors  number of persons \
0           3           3           2           2
1           3           3           2           2
2           3           3           2           2
3           3           3           2           2
4           3           3           2           2
...
1722        1           1           5           5
1723        1           1           5           5
1724        1           1           5           5
1725        1           1           5           5
1726        1           1           5           5
```

	lug_boot	safety	evaluation
0	2	2	unacc
1	2	0	unacc
2	1	1	unacc
3	1	2	unacc
4	1	0	unacc
...
1722	1	2	good
1723	1	0	vgood
1724	0	1	unacc
1725	0	2	good
1726	0	0	vgood

[1727 rows x 7 columns]

```
[9]: X=df[['buying price','maintenance cost','number of doors','number of
persons','lug_boot','safety']]
Y=df['evaluation']
```

```
[10]: X
```

```
[10]:
```

	buying price	maintenance cost	number of doors	number of persons	\
0	3	3	2	2	
1	3	3	2	2	
2	3	3	2	2	
3	3	3	2	2	
4	3	3	2	2	
...	
1722	1	1	5	5	
1723	1	1	5	5	
1724	1	1	5	5	
1725	1	1	5	5	
1726	1	1	5	5	

	lug_boot	safety
0	2	2
1	2	0
2	1	1
3	1	2
4	1	0
...
1722	1	2
1723	1	0
1724	0	1
1725	0	2
1726	0	0

[1727 rows x 6 columns]

```
[11]: Y
```

```
[11]: 0      unacc
      1      unacc
      2      unacc
      3      unacc
      4      unacc
      ...
     1722    good
     1723   vgood
     1724    unacc
     1725    good
     1726   vgood
      Name: evaluation, Length: 1727, dtype: object
```

```
[12]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
      ↪random_state=42)
```

```
[13]: svm_model=SVC(kernel='linear')
      svm_model.fit(X_train,Y_train)
```

```
[13]: SVC(kernel='linear')
```

```
[14]: Y_pred=svm_model.predict(X_test)
```

```
[15]: print("Accuracy:",accuracy_score(Y_test,Y_pred))
      print("\nconfusion Matrix:",confusion_matrix(Y_test,Y_pred))
```

Accuracy: 0.7138728323699421

confusion Matrix: [[12 0 65 0]
[0 0 15 0]
[2 0 235 0]
[12 0 5 0]]