```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

```
from sklearn import preprocessing
```

```
le = preprocessing.LabelEncoder()
data_train = pd.read_csv("play_tennis_train.csv")
data_test  = pd.read_csv("play_tennis_test.csv")
```

	day	outlook	temp	humidity	wind	play
0	D11	Sunny	Mild	Normal	Strong	Yes
1	D12	Overcast	Mild	High	Strong	Yes
2	D13	Overcast	Hot	Normal	Weak	Yes
3	D14	Rain	Mild	High	Strong	No

+ Code + Text

```
data_train_enc = data_train.apply(le.fit_transform)
data_test_enc = data_test.apply(le.fit_transform)
```

```
data_train.head()
```

	day	outlook	temp	humidity	wind	play
0	D1	Sunny	Hot	High	Weak	No
1	D2	Sunny	Hot	High	Strong	No
2	D3	Overcast	Hot	High	Weak	Yes
3	D4	Rain	Mild	High	Weak	Yes
4	D5	Rain	Cool	Normal	Weak	Yes

```
data_test.head()
```

	day	outlook	temp	humidity	wind	play
0	D11	Sunny	Mild	Normal	Strong	Yes
1	D12	Overcast	Mild	High	Strong	Yes
2	D13	Overcast	Hot	Normal	Weak	Yes
3	D14	Rain	Mild	High	Strong	No

```
data_train_enc.head()
```

	day	outlook	temp	humidity	wind	play
0	0	2	1	0	1	0
1	2	2	1	0	0	0
2	3	0	1	0	1	1
3	4	1	2	0	1	1
4	5	1	0	1	1	1

```
data_train_enc['play'].unique()
```

```
array([0, 1])
```

```
x_train = data_train_enc.drop('play',axis = 1)
y_train = data_train_enc.play
```

```
x_train
```

	day	outlook	temp	humidity	wind
0	0	2	1	0	1
1	2	2	1	0	0
2	3	0	1	0	1
3	4	1	2	0	1
4	5	1	0	1	1
5	6	1	0	1	0
6	7	0	0	1	0

```
y_train
```

```
0    0
1    0
2    1
3    1
4    1
5    0
6    1
7    0
8    1
9    1
Name: play, dtype: int64
```

```
from sklearn.naive_bayes import GaussianNB
from sklearn import metrics
```

```
model = GaussianNB()
nbtrain = model.fit(x_train,y_train)
nbtrain
```

```
GaussianNB()
```

```
x_test = data_test_enc.drop('play',axis=1)
y_test = data_test_enc.play
```

```
y_pred = model.predict(x_test)
```

```
y_pred == y_test
```

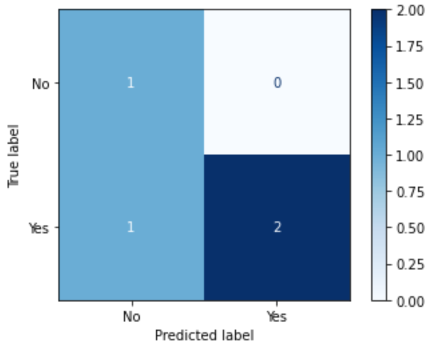
```
0    False
1     True
2     True
3     True
Name: play, dtype: bool
```

```
print("Accuracy : ",metrics.accuracy_score(y_test,y_pred))
```

```
Accuracy :  0.75
```

```
metrics.plot_confusion_matrix(nbtrain, x_test, y_test,
                              display_labels = ['No','Yes'],
                              cmap = plt.cm.Blues)
plt.show()
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarn
  warnings.warn(msg, category=FutureWarning)
```


No
Yes
True label
No
Yes
Predicted label