



```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
```

```
In [2]: df = pd.DataFrame({"actual y":['+ve','+ve','+ve','-ve','-ve','-ve','+ve','-ve','+ve',
"predicted y":['+ve','+ve','+ve','+ve','+ve','+ve','-ve','-ve','+ve',
df
```

```
Out[2]:
```

	actual y	predicted y
0	+ve	+ve
1	+ve	+ve
2	+ve	+ve
3	-ve	+ve
4	-ve	+ve
5	-ve	+ve
6	+ve	-ve
7	-ve	-ve
8	+ve	-ve
9	-ve	-ve

	actual y	predicted y
0	+ve	+ve
1	+ve	+ve
2	+ve	+ve
3	-ve	+ve
4	-ve	+ve
5	-ve	+ve
6	+ve	-ve
7	-ve	-ve
8	+ve	-ve
9	-ve	-ve

```
In [3]: y = df["actual y"]
ypred = df["predicted y"]
```

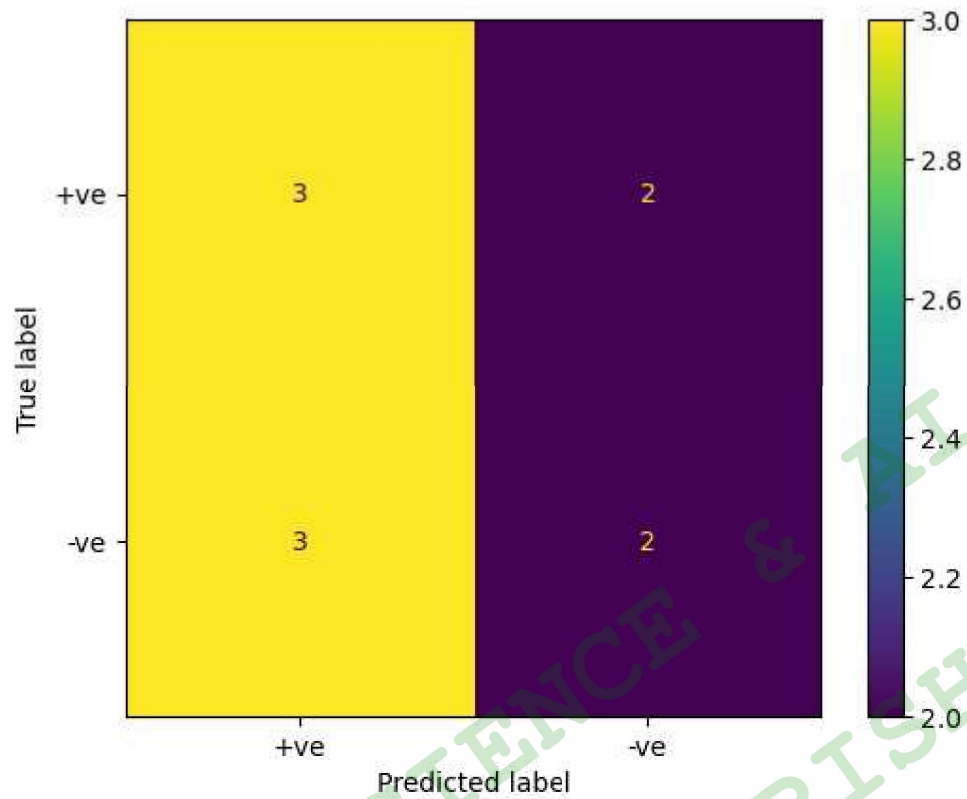
```
In [4]: from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y,ypred)
accuracy
```

```
Out[4]: 0.5
```

```
In [5]: from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y,ypred)
cm
```

```
Out[5]: array([[3, 2],
[3, 2]], dtype=int64)
```

```
In [6]: from sklearn.metrics import ConfusionMatrixDisplay
ConfusionMatrixDisplay(cm,display_labels = df["actual y"].unique()).plot()
plt.show()
```



```
In [7]: from sklearn.metrics import classification_report
print(classification_report(y,ypred))
```

	precision	recall	f1-score	support
+ve	0.50	0.60	0.55	5
-ve	0.50	0.40	0.44	5
accuracy			0.50	10
macro avg	0.50	0.50	0.49	10
weighted avg	0.50	0.50	0.49	10