```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix , ConfusionMatrixDisplay
from sklearn.preprocessing import LabelEncoder
```

```
df=pd.read_csv('/content/Iris.csv')
```

```
df.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Next steps: Generate code with df New interactive sheet

```
Start coding or generate with AI.
```

```
X= df.drop(['Species','Id'],axis=1)
y = df['Species']
```

```
encoder= LabelEncoder()
y=encoder.fit_transform(y)
```

```
X_train,X_test,y_train,y_test = train_test_split(X,y ,test_size=0.2, random_state = 42)
```

```
#scalling
```

```
scaler=StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled= scaler.transform(X_test)
```

```
#model_select and train
```

```
model = LogisticRegression()
model.fit(X_train_scaled,y_train)
```

```
▾ LogisticRegression ⓘ ?
LogisticRegression()
```

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

```
y_train_pred = model.predict(X_train_scaled)
print("Accuracy",accuracy_score(y_train,y_train_pred))
```

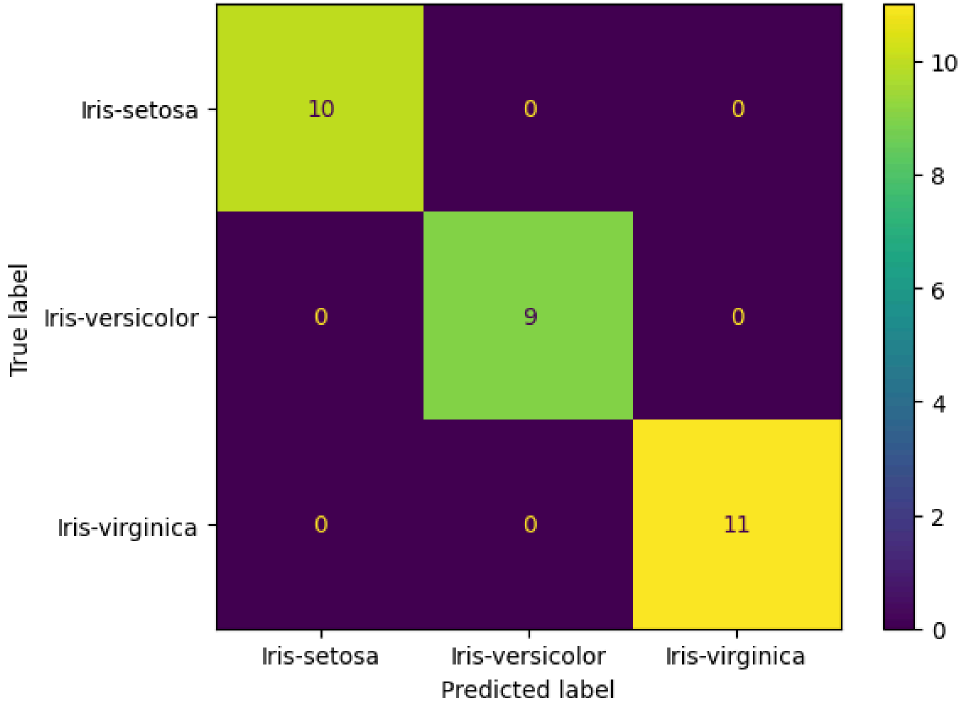
```
Accuracy 0.9666666666666667
```

```
confused = confusion_matrix(y_test,y_pred)
print(confused)
```

```
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

```
display = ConfusionMatrixDisplay(confusion_matrix=confused,display_labels=encoder.classes_)
display.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7be26de1e480>
```



```
# now predict to new flower data

new_flower=pd.DataFrame([[5.1,3.5,1.4,0.2]],columns=X.columns)
new_sclaed=scaler.transform(new_flower)
prediction=model.predict(new_sclaed)
print("new prediction",prediction[0])
```

```
new prediction 0
```

Start coding or generate with AI.

```
# now predict to new flower data

new_flower=pd.DataFrame([[6.5,3.0,5.2,2.0]],columns=X.columns)
new_sclaed=scaler.transform(new_flower)
prediction=model.predict(new_sclaed)
print("new prediction",prediction[0])
```

```
new prediction 2
```