

← → ↻ Not Secure http://10.1.38.19/user/ra2311047010037/lab/tree/DLT/Lab2.ipynb ☆ ⓘ ⋮

File Edit View Run Kernel Tabs Settings Help

Filter files by name 🔍

/ DLT /

Name	Last Modified
• Lab2.ipynb	next year
• Lab3.ipynb	next year

Launcher Lab2.ipynb

Python 3 (ipykernel)

```
[1]: from sklearn.datasets import load_iris
      from sklearn.model_selection import train_test_split
      from sklearn.preprocessing import StandardScaler
      from sklearn.neighbors import KNeighborsClassifier
      from sklearn.metrics import classification_report, accuracy_score

[2]: iris = load_iris()
      X = iris.data
      y = iris.target

[11]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

[12]: scaler = StandardScaler()
      X_train_scaled = scaler.fit_transform(X_train)
      X_test_scaled = scaler.transform(X_test)

[13]: knn = KNeighborsClassifier(n_neighbors=3)

[14]: knn.fit(X_train_scaled, y_train)

[14]: KNeighborsClassifier ⓘ ?
      Parameters
```

Simple 0 3 Python 3 (ipykernel) | Idle Mem: 381.28 MB Mode: Command Ln 1, Col 1 Lab2.ipynb 1

File Edit View Run Kernel Tabs Settings Help

+

📁

⬆

🔄

Filter files by name 🔍

/ DLT /

Name ▲	Last Modified
• 📄 Lab2.ipynb	next year
• 📄 Lab3.ipynb	next year

Launcher Lab2.ipynb

📄 + ✂ 📄 📄 ▶ ■ 🔄 ⏩ Code ▾

Notebook 📄 Python 3 (ipykernel)

[14]: knn.fit(X_train_scaled, y_train)

[14]:

▼ KNeighborsClassifier ⓘ ⓘ

▼ Parameters

📄	n_neighbors	3
📄	weights	'uniform'
📄	algorithm	'auto'
📄	leaf_size	30
📄	p	2
📄	metric	'minkowski'
📄	metric_params	None
📄	n_jobs	None

[15]: y_pred = knn.predict(X_test_scaled)

[16]:

```
print(f"Accuracy: {accuracy_score(y_test, y_pred):.2f}\n")
print("Classification Report:")
print(classification_report(y_test, y_pred, target_names=iris.target_names))
```

FileEditViewRunKernelTabsSettingsHelp

+

📁

📄

🔄

Filter files by name

🔍

/ DLT /

Name	Last Modified
Lab2.ipynb	next year
Lab3.ipynb	next year

Launcher

Lab2.ipynb

📄

+

✂

📄

📄

▶

■

🔄

▶▶

Code

▼

Notebook

Python 3 (ipykernel)

📄

metric_params

None

📄

n_jobs

None

[15]:

y_pred = knn.predict(X_test_scaled)

[16]:

```
print(f"Accuracy: {accuracy_score(y_test, y_pred):.2f}\n")
print("Classification Report:")
print(classification_report(y_test, y_pred, target_names=iris.target_names))
```

Accuracy: 0.98

Classification Report:

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	16
versicolor	1.00	0.94	0.97	18
virginica	0.92	1.00	0.96	11
accuracy			0.98	45
macro avg	0.97	0.98	0.98	45
weighted avg	0.98	0.98	0.98	45

[]: