

irislogistic

November 4, 2024

```
[5]: import seaborn as sns
import pandas as pd
# Chargement des données Iris
iris = sns.load_dataset('iris')

# Affichage des 5 premières lignes
print(iris.head())
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
[7]: # Vérification des valeurs manquantes
print(iris.isnull().sum())
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

```
[9]: from sklearn.preprocessing import StandardScaler
# Sélection des colonnes numériques (sans la colonne 'species')
iris_features = iris.drop(columns='species')
# Normalisation des données
scaler = StandardScaler()
iris_normalized = pd.DataFrame(scaler.fit_transform(iris_features),
                               columns=iris_features.columns)

# Ajout de la colonne 'species' de nouveau
iris_normalized['species'] = iris['species']
print(iris_normalized.head())
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	-0.900681	1.019004	-1.340227	-1.315444	setosa

1	-1.143017	-0.131979	-1.340227	-1.315444	setosa
2	-1.385353	0.328414	-1.397064	-1.315444	setosa
3	-1.506521	0.098217	-1.283389	-1.315444	setosa
4	-1.021849	1.249201	-1.340227	-1.315444	setosa

```
[47]: from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

data=iris

X = data[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']] #_
    ↪Caractéristiques
y = data['species'] # Variable cible
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Crée un modèle de régression logistique
model = LogisticRegression()

# Entraîne le modèle
model.fit(X_train, y_train)

# Prédiction sur l'ensemble de test
y_pred = model.predict(X_test)

# Évalue le modèle
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

Accuracy: 0.93

```
[51]: data=iris_normalized

X = data[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']] #_
    ↪Caractéristiques
y = data['species'] # Variable cible
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Crée un modèle de régression logistique
model = LogisticRegression()

# Entraîne le modèle
model.fit(X_train, y_train)

# Prédiction sur l'ensemble de test
y_pred = model.predict(X_test)
```

```
# Évalue le modèle
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

Accuracy: 0.97

```
[53]: X = data[['sepal_length', 'sepal_width']] # Caractéristiques
      y = data['species'] # Variable cible
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

      # Crée un modèle de régression logistique
      model = LogisticRegression()

      # Entraîne le modèle
      model.fit(X_train, y_train)

      # Prédiction sur l'ensemble de test
      y_pred = model.predict(X_test)

      # Évalue le modèle
      accuracy = accuracy_score(y_test, y_pred)
      print(f'Accuracy: {accuracy:.2f}')
```

Accuracy: 0.83

```
[ ]:
```