ML Project 2: Classification with the Iris Dataset

Now let’s try classification: predict the species of an iris flower based on its measurements. The Iris dataset is a classic ML dataset included with scikit-learn.

Code: Classification in PyCharm

Create a new file (e.g., ml\_classification.py):

python

import numpy as np

import matplotlib.pyplot as plt

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix

import seaborn as sns

# Step 1: Load the Iris dataset

iris = load\_iris()

X = iris.data[:, :2] # Use only the first two features for easy plotting

y = iris.target

# Step 2: Split data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

# Step 3: Train a logistic regression model

model = LogisticRegression()

model.fit(X\_train, y\_train)

# Step 4: Predict and evaluate

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy: {accuracy:.2f}")

# Step 5: Plot decision boundaries

x\_min, x\_max = X[:, 0].min() - 1, X[:, 0].max() + 1

y\_min, y\_max = X[:, 1].min() - 1, X[:, 1].max() + 1

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, 0.02), np.arange(y\_min, y\_max, 0.02))

Z = model.predict(np.c\_[xx.ravel(), yy.ravel()])

Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z, alpha=0.3)

plt.scatter(X[:, 0], X[:, 1], c=y, edgecolor='k', label='Data Points')

plt.xlabel(iris.feature\_names[0])

plt.ylabel(iris.feature\_names[1])

plt.title('Iris Classification with Logistic Regression')

plt.show()

# Step 6: Confusion matrix

cm = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix')

plt.show()

What’s Happening?

* Data: The Iris dataset has 150 samples, 4 features, and 3 classes (species).
* Model: LogisticRegression classifies samples into one of the three species.
* Visualization: Plot decision boundaries (using only 2 features for simplicity) and a confusion matrix to evaluate performance.

Your Turn

1. Use all four features (X = iris.data) and see if accuracy improves.
2. Try a different classifier (e.g., from sklearn.svm import SVC; model = SVC()).
3. Experiment with the test size in train\_test\_split.