import numpy as np

from sklearn.datasets import load\_iris

from collections import defaultdict

# Load the Iris dataset

iris = load\_iris()

data = iris['data']

target = iris['target']

target\_names = iris['target\_names']

# Convert the dataset into a list of lists

iris\_data = np.column\_stack((data, target)).tolist()

# Function to compute mean and standard deviation

def compute\_stats(data, indices):

stats = []

for idx in indices:

column\_data = [row[idx] for row in data]

mean = np.mean(column\_data)

std\_dev = np.std(column\_data)

stats.append((mean, std\_dev))

return stats

# Overall mean and standard deviation for each measurement column

overall\_stats = compute\_stats(iris\_data, [0, 1, 2, 3])

# Print overall statistics

print("Overall statistics:")

for i, (mean, std\_dev) in enumerate(overall\_stats):

print(f"Column {i+1}: Mean = {mean:.2f}, Std Dev = {std\_dev:.2f}")

print()

# Separate the data by species

species\_data = defaultdict(list)

for row in iris\_data:

species\_data[target\_names[int(row[4])]].append(row)

# Compute mean and standard deviation for each species and measurement column

species\_stats = {}

for species, data in species\_data.items():

species\_stats[species] = compute\_stats(data, [0, 1, 2, 3])

# Print statistics for each species

for species, stats in species\_stats.items():

print(f"Statistics for {species}:")

for i, (mean, std\_dev) in enumerate(stats):

print(f" Column {i+1}: Mean = {mean:.2f}, Std Dev = {std\_dev:.2f}")

print()

# Determine the "best" measurement for guessing species

# We'll consider the "best" measurement to be the one with the smallest variance within species

variances = defaultdict(list)

for species, stats in species\_stats.items():

for i, (mean, std\_dev) in enumerate(stats):

variances[i].append(std\_dev \*\* 2)

average\_variances = {i: np.mean(var\_list) for i, var\_list in variances.items()}

best\_measurement = min(average\_variances, key=average\_variances.get)

print(f"The best measurement for guessing the Iris species is Column {best\_measurement+1}.")