Machine Learning Fundamentals

This practical assesses dataset exploration and visualization, and demonstrates how a machine learning model can be trained and evaluated.

Data loading and exploration

Q1: Load the Iris dataset using `sklearn.datasets.load_iris()` and print `feature_names`, `target_names`, and the shape of `data`.

In [1]:
```
# Load the Iris dataset
```

In [2]:
```
#Printing
```

Q2: Provide a short written answer: What type of problem is this (classification/regression)? Why? Write your answer in a Markdown cell below.

Answer (Q2):

Q3: Show the first five rows of the data as a pandas DataFrame with column names from `feature_names`.

In [3]:
```
# Show first 5 rows as DataFrame
```

Q4: Report class distribution (counts for each species)

`Hint`: you can use a pandas Series.

In [ ]:

Visualization

Q5: Create a pair plot (scatter plot matrix) for the four features using matplotlib. Color points by species.

In [4]:
```
# Pair plot
```

Q6: Based on your plot, which two features appear most useful to separate the classes? Explain in one short sentence.

Answer (Q6):

Train - Test - split and KNN modeling

Q7: Split the data into train/test sets (75% train, 25% test). Use `stratify=target` and `random_state=42`. Print shapes.

In [ ]:

In [5]:
```
#Print shapes.
```

Q8: Train a `KNeighborsClassifier` with `n_neighbors=3`. Fit on training data and report test accuracy.

In [6]:
```
# Train KNN
```

In [7]:
```
#and report accuracy
```

Q9: Compare accuracy for `n_neighbors` = 1, 3, 5. Report which k gives best accuracy and why (short sentence).

In [8]:
```
# Compare k values
```

Which k is best and why?

Write your short explanation

Prediction

Q10: Use your trained `KNeighborsClassifier` (use k=3) to predict the class of a new sample with features: `[5.0, 2.9, 1.0, 0.2]`. Show predicted class name.

In [ ]:
```

```