ML Pipeline: Iris Dataset Implementation

1. The Dataset

The Iris dataset [cite: 79] is the "hello world" of ML. It contains 150 samples with four numerical features: sepal length, sepal width, petal length, and petal width, mapped to three target species [cite: 79, 99].

2. Implementation Code

This pipeline demonstrates proper splitting and scaling to prevent data leakage [cite: 24, 78]:

```
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Load data
iris = load_iris()
X, y = iris.data, iris.target

# 1. Split data (80/20)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# 2. Fit Scaler on training data ONLY
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)

# 3. Apply transformation to test data
X_test_scaled = scaler.transform(X_test)

print(f"Train shape: {X_train_scaled.shape}")
print(f"Test shape: {X_test_scaled.shape}")
```

3. Key Takeaways

- **Fit on Train:** Use `fit_transform()` on training data to learn parameters [cite: 65, 75].
- **Transform on Test:** Use `transform()` on testing data to avoid leakage [cite: 26, 67].
- **Verification:** Always check that scaling parameters are derived only from the training subset [cite: 105].