Machine Learning con Scikit-Learn

Classificazione dei Fiori Iris

Prof. Massimo Fedeli

IIS Fermi Sacconi Ceci - Ascoli Piceno

29 novembre 2025

Contenuti

Cos'è il Machine Learning?

Definizione

Il **Machine Learning** (apprendimento automatico) è la capacità di un computer di *imparare* dai dati senza essere esplicitamente programmato.

Programmazione tradizionale:

- Regole esplicite
- If-then-else
- Logica predefinita

Machine Learning:

- Impara dai dati
- Trova pattern
- Fa predizioni

Scikit-Learn: La Libreria per ML in Python

Cos'è Scikit-Learn?

Libreria Python per Machine Learning, semplice ed efficiente per analisi dati e data mining.

Caratteristiche principali:

- Open source e gratuita
- Facile da usare
- Integrata con NumPy e Pandas
- Molti algoritmi già implementati

Installazione:

```
pip install scikit-learn
```

Il Dataset Iris

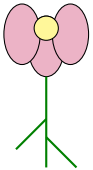
Dataset storico creato da Ronald Fisher nel 1936.

Contenuto:

- 150 fiori di iris
- 3 specie diverse (50 per specie)
- 4 misurazioni per fiore

Le 3 specie:

1. Iris Setosa
2. Iris Versicolor
3. Iris Virginica



Le 4 Caratteristiche (Features)

Features (X) - Cosa misuriamo

Lunghezza Sepalo

Larghezza Sepalo

Lunghezza Petalo

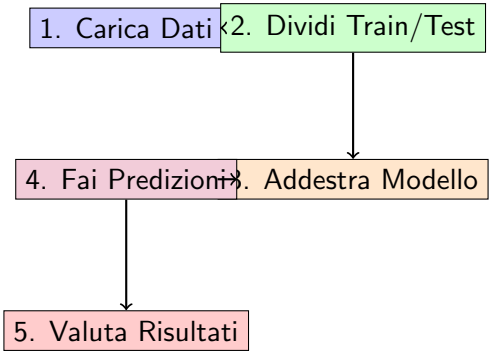
Larghezza Petalo

Target (y) - Cosa prevediamo: la specie (0, 1, o 2)

Esempio di un fiore:

Lung. Sepalo	Larg. Sepalo	Lung. Petalo	Larg. Petalo	Specie
5.1 cm	3.5 cm	1.4 cm	0.2 cm	Setosa (0)

Il Processo di Machine Learning



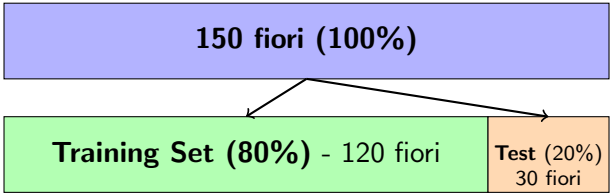
Obiettivo

Creare un modello che impari a classificare i fiori in base alle loro misure.

Dividere i Dati: Train e Test

Perché dividere i dati?

Per verificare se il modello funziona su dati *mai visti prima*.



Training Set: il modello "studia" qui
Test Set: il modello viene "interrogato" qui

Passo 1: Importare le Librerie

```
from sklearn.datasets import load_iris
from sklearn.model_selection import
    train_test_split
from sklearn.tree import
    DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

- `load_iris`: carica il dataset
- `train_test_split`: divide train/test
- `DecisionTreeClassifier`: il modello
- `accuracy_score`: valuta l'accuratezza

Passo 2: Caricare i Dati

```
# Carica il dataset Iris
iris = load_iris()
X = iris.data       # Features (150 righe x 4 colonne)
y = iris.target     # Target (150 valori: 0, 1, o 2)
```

X contiene le misure:

```
[[5.1, 3.5, 1.4, 0.2],
[4.9, 3.0, 1.4, 0.2],
...]
```

y contiene le specie:

```
[0, 0, 0, ..., 1, 1, 1, ..., 2, 2, 2]
```

Passo 3 e 4: Split e Addestramento

```
# Dividi i dati (80% training, 20% test)
X_train, X_test, y_train, y_test =
    train_test_split(
X, y, test_size=0.2, random_state=42
)

# Crea il modello (Decision Tree)
model = DecisionTreeClassifier(random_state
    =42)

# Addestra il modello
model.fit(X_train, y_train)
```

Cosa fa `fit()`?

Il modello "studia" i dati di training per imparare la relazione tra features (X) e specie (y).

Passo 5: Fare Predizioni

```
# Fai predizioni sul test set
y_pred = model.predict(X_test)
```

Cosa succede?

1. Il modello riceve le features di 30 fiori (`X_test`)
2. Per ogni fiore, predice la specie
3. Restituisce un array con 30 predizioni

Esempio

Input: `[[5.1, 3.5, 1.4, 0.2]]`
Output: `[0]` $\rightarrow$ Setosa

Passo 6: Valutare il Modello

```python
# Calcola l'accuratezza
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuratezza: {accuracy:.2%}")
```

Cos'è l'accuratezza?

$$\text{Accuratezza} = \frac{\text{Predizioni Corrette}}{\text{Predizioni Totali}} \times 100$$

Esempio:

- Test set: 30 fiori
- Predizioni corrette: 30
- Accuratezza: $\frac{30}{30} = 100\%$

Il Parametro `random_state`

Cos'è?

Un numero "seme" che controlla la casualità per rendere i risultati **riproducibili**.

Senza random_state:

```
train_test_split(X, y, test_size=0.2)

# Risultati diversi ogni volta!
# Esecuzione 1: accuratezza 96%
# Esecuzione 2: accuratezza 100%
# Esecuzione 3: accuratezza 93%
```

Con random_state=42:

```
train_test_split(X, y, test_size=0.2,
    random_state=42)

# Sempre lo stesso risultato!
```

Riepilogo e Prossimi Passi

Cosa abbiamo imparato:

- Cos'è il Machine Learning
- Come usare scikit-learn
- Il processo completo: dati $\rightarrow$ training $\rightarrow$ predizioni
- Come valutare un modello

Esercizi per casa:

1. Prova altri modelli: `KNeighborsClassifier`, `LogisticRegression`
2. Cambia il `random_state` e osserva i risultati
3. Visualizza la matrice di confusione
4. Prova con altri dataset di scikit-learn

Domande?