

# Esame Introduction to Machine Learning

23 Giugno 2026

**Istruzioni:** Scrivere su ogni pagina nome, cognome numero di matricola e firmare. Usare penna o blu o nera (NO MATITA). Sui fogli protocollo, identificare ogni risposta con il numero associato. Durata esame: 60 minuti.

## Domande Teoria (21 punti)

### 1. Lazy learning.

Si spieghi l'algoritmo di k-Nearest Neighbors sia per la task di classificazione che quella di regressione. (7 punti)

### 2. Ranking e selezione di feature.

Si spieghi la motivazione per la selezione di feature e si illustrino i metodi principali. (7 punti)

### 3. Reti neurali: perceptrone multistrato.

Spiegare brevemente e chiaramente il Perceptrone Multistrato: motivazione, obiettivo, architettura (con schema grafico), equazioni per l'addestramento e per l'utilizzo. Si definisca il gradiente di una funzione. Qual è il ruolo del gradiente nell'addestramento di MLP? (spiegare). (7 punti)

## Domande Pratica (10 punti)

### 4. Commenta le seguenti linee di codice. Cosa ti aspetti di ottenere? (3 punti)

```
1 from sklearn.datasets import load_iris
2 import pandas as pd
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5
6
7 iris = load_iris()
8 iris_df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
9 plt.figure(figsize=(8, 5))
10 sns.histplot(data=iris_df, x='sepal length (cm)', kde=True)
11 plt.title('Distribution of Sepal Length (cm)')
12 plt.xlabel('Sepal Length (cm)')
13 plt.ylabel('Frequency')
14 plt.show()
```

### 5. Commenta le seguenti linee di codice. A cosa serve function1? (3 punti)

```
1 import numpy as np
2
3 def function1(X, y):
4     assert X.shape[1] == 1
5     assert X.shape[0] > 0
6     x_bar = np.sum(X)/X.shape[0]
7     y_bar = np.sum(y)/y.shape[0]
8     value1 = np.sum((X - x_bar).reshape(-1,1)*(y - y_bar).reshape(-1,1))
9     value2 = np.sum((X - x_bar).reshape(-1,1)*(X - x_bar).reshape(-1,1))
10    out2 = value1 / value2
11    out1 = y_bar - out2 * x_bar
12    return out1, out2
```

6. Commenta le seguenti linee di codice. Cosa manca per valutare l'accuratezza di `clf`? (4 punti)

```
1 import pandas as pd
2 import numpy as np
3 from sklearn.datasets import load_iris
4 from sklearn.model_selection import train_test_split
5 from sklearn.tree import DecisionTreeClassifier
6 from sklearn.metrics import accuracy_score
7
8 db = load_iris()
9 X = db['data']
10 y = db['target']
11
12 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
13
14 clf = DecisionTreeClassifier(max_depth=2)
15 clf.fit(X_train, y_train)
16
17 acc = accuracy_score(y_train, clf.predict(X_train))
18 print(acc)
```