



UNIVERSITÀ  
degli STUDI  
di CATANIA



# Image Classification with CIFAR-10

Homework 2 – Machine Learning

---

Alessia Tripolone 1000059364

# Descrizione del Dataset : *CIFAR-10*

- **60.000 immagini** a colori di dimensione 32x32x3

<u>SET</u>	<u>NUMERO IMMAGINI</u>
Training (allenamento)	50.000
Test	10.000
Classi	10

- Ogni immagine è assegnata a una delle **10 classi**
- Le immagini sono memorizzate come tensori 3D (altezza, larghezza, canali)

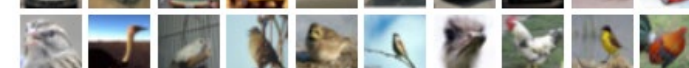
airplane



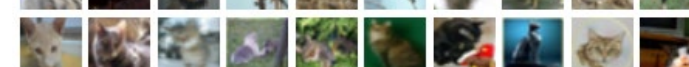
automobile



bird



cat



deer



dog



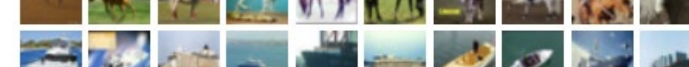
frog



horse



ship



truck



# Preprocessing dei Dati

## ➤ FLATTENING DELLE IMMAGINI:

- Ogni immagine è trasformata in un **vettore di 3072 elementi** ( $32 \times 32 \times 3$ ) usando `.reshape()`

## ➤ RIDUZIONE DEL DATASET:

- Per evitare tempi troppo lunghi nella fase di ottimizzazione dei modelli, ho selezionato:
  - **10.000 immagini** per il training
  - **2.000 immagini** per il test

## ➤ STANDARDIZZAZIONE:

- Applicata con `StandardScaler` per portare ogni feature su scala standard
- Evita che attributi con range ampio dominino l'addestramento

# Modelli Considerati

Sono stati testati 4 classificatori supervisionati:

MODELLI	PARAMETRI GRID SEARCH
SVM	$C = [0.1, 1]$ , kernel = [linear, rbf]
Logistic Regression	$C = [0.1, 1, 10]$ , solver = lbfgs
KNN	n_neighbors = [3, 5, 7], weights = [uniform, distance]
Decision Tree	max_depth = [5, 7], criterion = [entropy]

# Ricerca degli Iperparametri

## ➤ GRID SEARCH CON CROSS-VALIDATION

- Funzione `grid_search()`
- Utilizza GridSearchCV di Scikit-Learn con **scoring = accuracy**
- Per ogni modello:
  - Trova la miglior configurazione
  - Calcola l'**accuracy** in **cross-validation**
  - Restituisce il modello addestrato con i parametri ottimali
- Output salvato in un dizionario **risultati**, con:
  - **MODELLO**: miglior modello
  - **CONFIGURAZIONE**: iperparametri ottimali
  - **ACCURACY\_VALIDATION**: score



# Selezione del Miglior Modello

## ➤ Scelta del miglior classificatore

- Confronto delle performance su validation set
- Selezione del classificatore con la **miglior accuracy**

Allenamento: SVM

Best config: {'C': 1, 'kernel': 'rbf'}

Accuracy (cross-val): 0.4672

Allenamento: Logistic Regression

Best config: {'C': 0.1, 'solver': 'lbfgs'}

Accuracy (cross-val): 0.3139

Allenamento: KNN

Best config: {'n\_neighbors': 7, 'weights': 'distance'}

Accuracy (cross-val): 0.3055

Allenamento: Decision Tree

Best config: {'criterion': 'entropy', 'max\_depth': 7}

Accuracy (cross-val): 0.2539

- In questo caso, la miglior configurazione in cross-validation è stata ottenuta da:



Support Vector Machine (SVM) con kernel RBF  
Parametri ottimali trovati: C = 1, kernel = 'rbf'  
Accuracy su validation set: 0.4672



UNIVERSITÀ  
degli STUDI  
di CATANIA

# Valutazione sul Test Set

---

## ➤ Funzione prestazioni()

- Calcola le metriche standard su `X_test_std` e `y_test_reduced`

Include:

- **ACCURACY**
- **PRECISION**
- **RECALL**
- **F1 SCORE**
- **CLASSIFICATION\_REPORT PER CLASSE**
- **CONFUSION\_MATRIX**



# Prestazioni ottenute:

Report dettagliato per classe:					
		precision	recall	f1-score	support
	0	0.54	0.54	0.54	196
Accuracy: 0.4675	1	0.56	0.55	0.55	198
Precision: 0.4626	2	0.32	0.32	0.32	195
Recall: 0.4675	3	0.34	0.32	0.33	199
F1 Score: 0.4631	4	0.34	0.32	0.33	198
	5	0.40	0.32	0.36	185
	6	0.48	0.59	0.53	216
	7	0.54	0.49	0.51	193
	8	0.56	0.68	0.62	217
	9	0.51	0.51	0.51	203
	accuracy			0.47	2000
	macro avg	0.46	0.46	0.46	2000
	weighted avg	0.46	0.47	0.46	2000





# Matrice di Confusione

## ➤ FUNZIONE: PLOT\_CONFUSION\_MATRIX\_MULTICLASS()

- Heatmap con **Seaborn**
- Etichette per le 10 classi
- Permette di analizzare le **classi più confuse**
  - Es: errori frequenti tra **gatto** e **cane**, **nave** e **camion**
- Utile per capire dove il modello fatica maggiormente

