



MACHINE LEARNING E DATA ANALYSIS PER L'IMPRESA

10 SETTEMBRE 2025 – ore 12:00
Webinar di CSMT Innovation Hub



Introduzione

Evoluzione dell'ottimizzazione dei processi



- Anni '50-'80
 - Ottimizzazione matematica tramite ricerca operativa, controllo statistico e Design of Experiments
 - Teoria dei sistemi per l'analisi delle interconnessioni tra processi
- Anni '80-2000
 - Automazione e robotica, standardizzazione dei processi
 - Software di pianificazione (Enterprise Resource Planning) e controllo dei processi (Manufacturing Execution System)
- 2000-oggi
 - Industria 4.0: digitalizzazione e interconnessione
 - Simulatori di processi industriali e digital-twin
 - **Machine learning per l'analisi dei dati**

Cos'è il Machine Learning

Capacità delle macchine di “apprendere” dai dati **senza essere esplicitamente programmate.**

Apprendere... cosa?

- Rappresentazione delle strutture intrinseche dei dati
- Costruzione di un modello logico-matematico (esplicito o implicito) del processo

Apprendere... come?

- Supervised learning: apprendimento con dati etichettati
- Unsupervised learning: apprendimento senza etichette, rilevamento di schemi e relazioni nascoste
- Reinforcement learning: apprendimento tramite ricompense e penalità

Evoluzione storica del ML

- Anni '50-'70: prime reti neurali e algoritmi simbolici
- Anni '80-'90: algoritmi statistici, SVM, decision tree
- 2000-2020: esplosione big data e deep learning
- Oggi: applicazioni diffuse in azienda e ricerca

```
;; Definiamo una struttura dati chiamata 'ordine' con campi (slots)
;; per il numero d'ordine, il cliente, l'articolo e la quantità.
(defstruct ordine
  numero-ordine
  cliente
  articolo
  quantità)

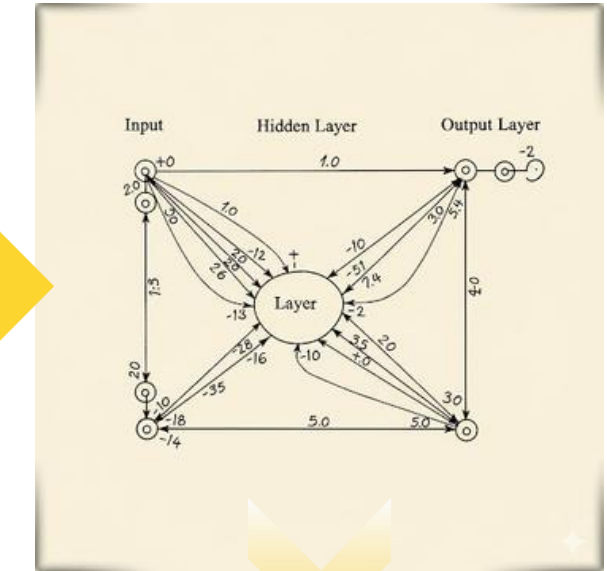
;; Creiamo un'istanza della struttura 'ordine'.
;; Ogni nome di campo (numero-ordine, cliente, ecc.) è un simbolo.
(setq ordine1 (make-ordine :numero-ordine 101
                           :cliente 'Paolo-Rossi
                           :articolo 'computer-portatile
                           :quantità 1))

;; Possiamo accedere ai campi usando una funzione generata automaticamente.
(format t "Numero ordine: ~a%" (ordine-numero-ordine ordine1))
(format t "Cliente: ~a%" (ordine-cliente ordine1))
(format t "Articolo: ~a%" (ordine-articolo ordine1))
(format t "Quantità: ~a%" (ordine-quantità ordine1))

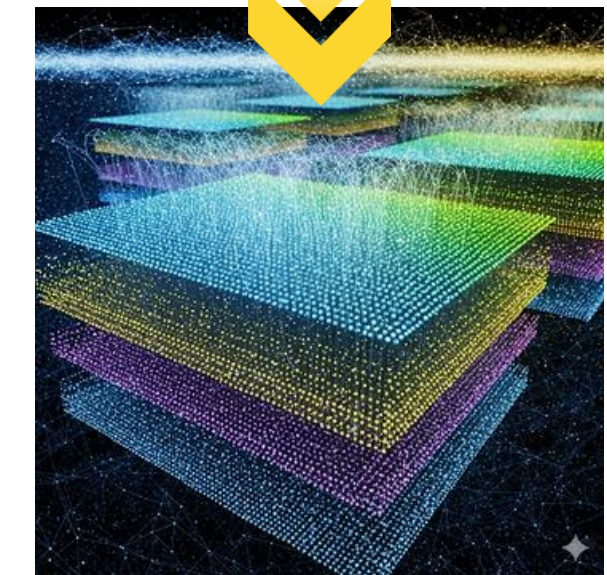
;; Modifichiamo il valore di un campo.
(setf (ordine-quantità ordine1) 2)
(format t "Nuova quantità: ~a%" (ordine-quantità ordine1))

;; Confrontiamo due simboli: i simboli sono confrontati per identità.
(if (eq 'Paolo-Rossi (ordine-cliente ordine1))
    (format t "Il cliente dell'ordine è Paolo Rossi.~%"))
```

Simboli e proposizioni



Connessioni e matrici



Connessioni di strati

Perchè è così importante

- Il Machine Learning non è solo una moda: ha un impatto concreto sul business.
- Permette alle aziende di:
 - capire meglio e più velocemente dati reali:
 - con dimensionalità elevata, non lineari e poco strutturati
 - ottimizzare e automatizzare attività
 - con risorse e in tempi limitati
 - prendere decisioni più informate,
 - anticipare tendenze future.
- In sintesi, trasforma i dati in vantaggio competitivo.

Esempi di applicazioni

- Banche e finanza
 - Identificazione transazioni sospette
 - Valutazione dell'affidabilità creditizia dei clienti
- Marketing
 - Motori di raccomandazione per suggerire prodotti
 - Analisi del comportamento dei clienti
 - Previsione dei comportamenti e delle tendenze di acquisto.
- Sanità
 - Diagnostica medica tramite analisi di immagini, dati genetici, esami clinici
 - Miglioramento dei trattamenti personalizzati
 - Analisi massiva di molecole di farmaci
- Logistica
 - Gestione intelligente dei magazzini per ridurre costi e tempi.
 - Ottimizzazione dei percorsi

Esempi di applicazioni

- Industria
 - Creazione di modelli di processo per
 - aumento della conoscenza del processo
 - simulazione del processo
 - simulazione statistica (es. Montecarlo)
 - ottimizzazione
 - Manutenzione predittiva,
 - Automazione di operazioni con strumenti versatili
 - Controllo di processo e di prodotto
 - Sistemi di visione, analisi e regolazione in tempo reale

Figure coinvolte

- **Alta direzione**
 - La rivoluzione dell'intelligenza digitale è **strategica** e va affrontata a questo livello
- **Direzione intermedia**
 - I progetti di ottimizzazione devono essere **supportati** da una pianificazione attenta
- **Esperto di analisi dei dati**
 - Una figura **esperta delle tecniche di analisi** che sia in grado di eseguire le elaborazioni
- **Ingegneria di prodotto e/o processo, Qualità di prodotto e/o di processo**
 - Insieme all'esperto di analisi elaborano le **strategie di ottimizzazione**
 - Eseguono la **validazione** del modello e pianificano eventuali esperimenti aggiuntivi
- **Produzione**
 - Collabora per la raccolta dei dati, gli esperimenti e la **messa in produzione delle ottimizzazioni**

Competenze richieste di ML



Sono richieste competenze a tutti i livelli per governare correttamente queste tecniche:

- **Alta direzione**
 - Identificare l'opportunità di business del ML
 - Comprendere le implicazioni strategiche e i rischi
 - Allocare le risorse e supportare i progetti a lungo termine
- **Direzione intermedia**
 - Gestione di progetti di data-science (di natura iterativa e sperimentale)
 - Comunicazione della complessità a figure non tecniche
 - Gestione dei gruppi di lavoro multidisciplinari
 - Conoscenza base delle diverse tecniche
- **Produzione**
 - Comprensione e gestione della raccolta dei dati
 - Collaborazione e conoscenza di base delle tecniche per feedback pratico
 - Apertura all'utilizzo di nuove tecnologie

Competenze richieste

Sono richieste competenze a tutti i livelli per governare correttamente queste tecniche:

- **Esperto di analisi dei dati**
 - Conoscenza delle tecniche di programmazione (python) e delle relative librerie
 - Capacità di pulire, trasformare ed esplorare i dati
 - Conoscenza degli algoritmi e loro applicazione
 - Capacità di valutazione e validazione dei modelli
 - Capacità di comunicare i risultati anche a persone non esperte
- **Ingegneria di prodotto/processo e Qualità**
 - Conoscenza del dominio oggetto dell'analisi
 - Conoscenza dei metodi di campionamento e misurazione dei dati
 - Capacità di pianificare esperimenti per raccogliere dati e validare le previsioni in un ambiente reale
 - Capacità di fornire un feedback continuo sui risultati dei modelli
 - Capacità di interfacciarsi con la produzione e la direzione

Flusso operativo ML

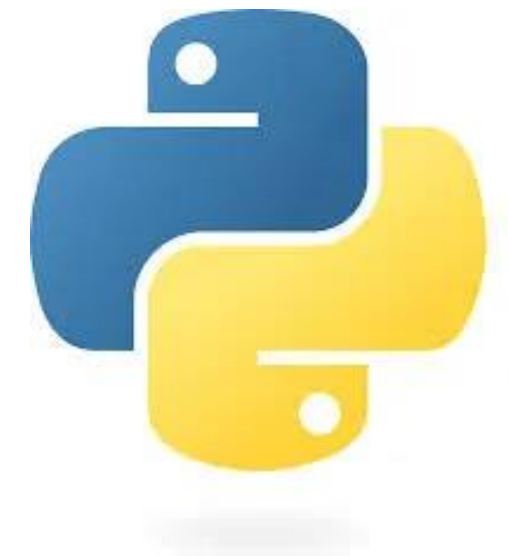
- 
- **Pianificazione e analisi del problema**
 - Obiettivi, risorse e processi, rischi e fattibilità
 - **Raccolta e pre-elaborazione dei dati**
 - Raccolta, esplorazione, pulizia, normalizzazione, feature engineering
 - **Sviluppo del modello**
 - Selezione degli algoritmi ML
 - Cicli di addestramento e validazione, ottimizzazione dell'architettura e degli iperparametri
 - Valutazione delle prestazioni del modello
 - **Deployment e integrazione**
 - Integrazione nell'ambiente di produzione, formazione, validazione
 - **Monitoraggio e manutenzione**
 - Verifica continua delle prestazioni del modello sui nuovi dati
 - Retraining
 - Manutenzione degli strumenti per stabilità e sicurezza



Qualche esempio

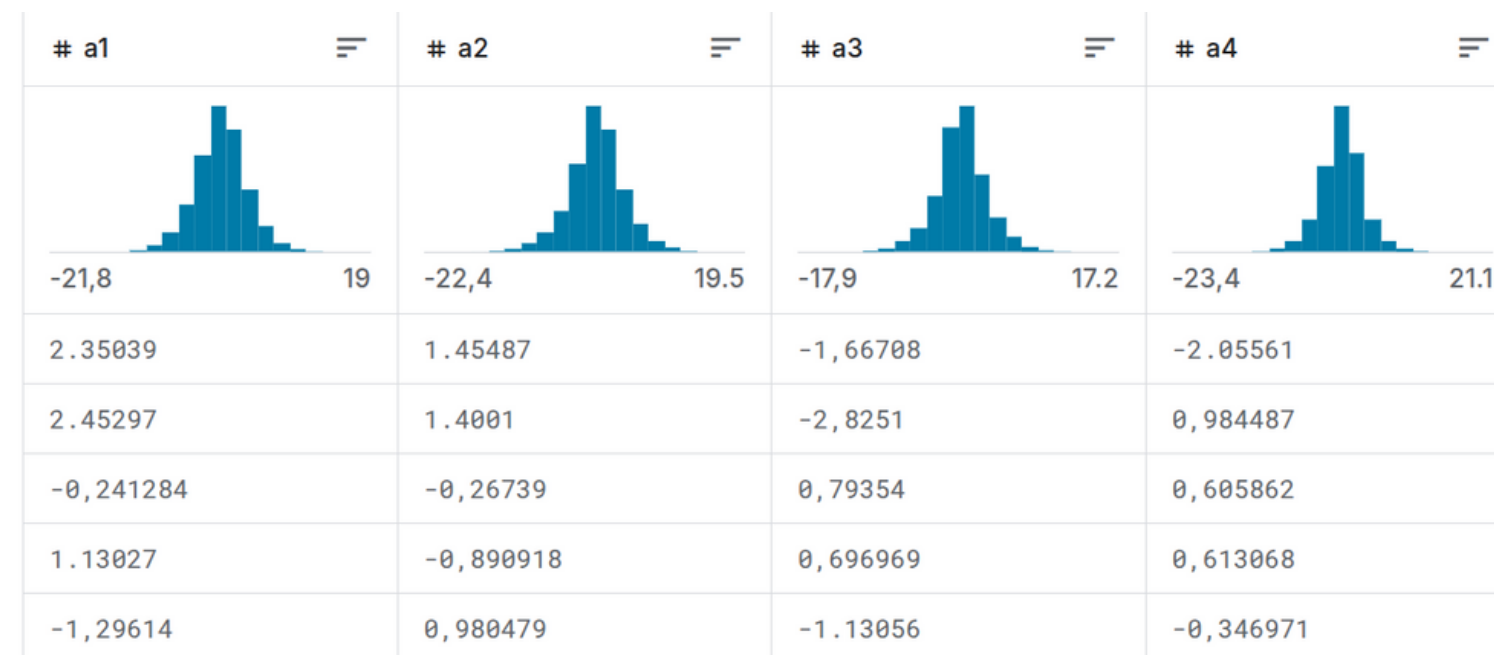
Python

- Costellazione di librerie di ML
- Numpy e Pandas
 - Analisi dei dati in tabelle e array multidimensionali
- Scikit-learn
 - Classificazione, regressione, clustering, ...
- TensorFlow e PyTorch
 - Reti neurali e deep learning
- Matplotlib e Seaborn
 - Visualizzazione dei dati
- Documentazione e comunità di sviluppatori
- LLM in aiuto alla scrittura del codice

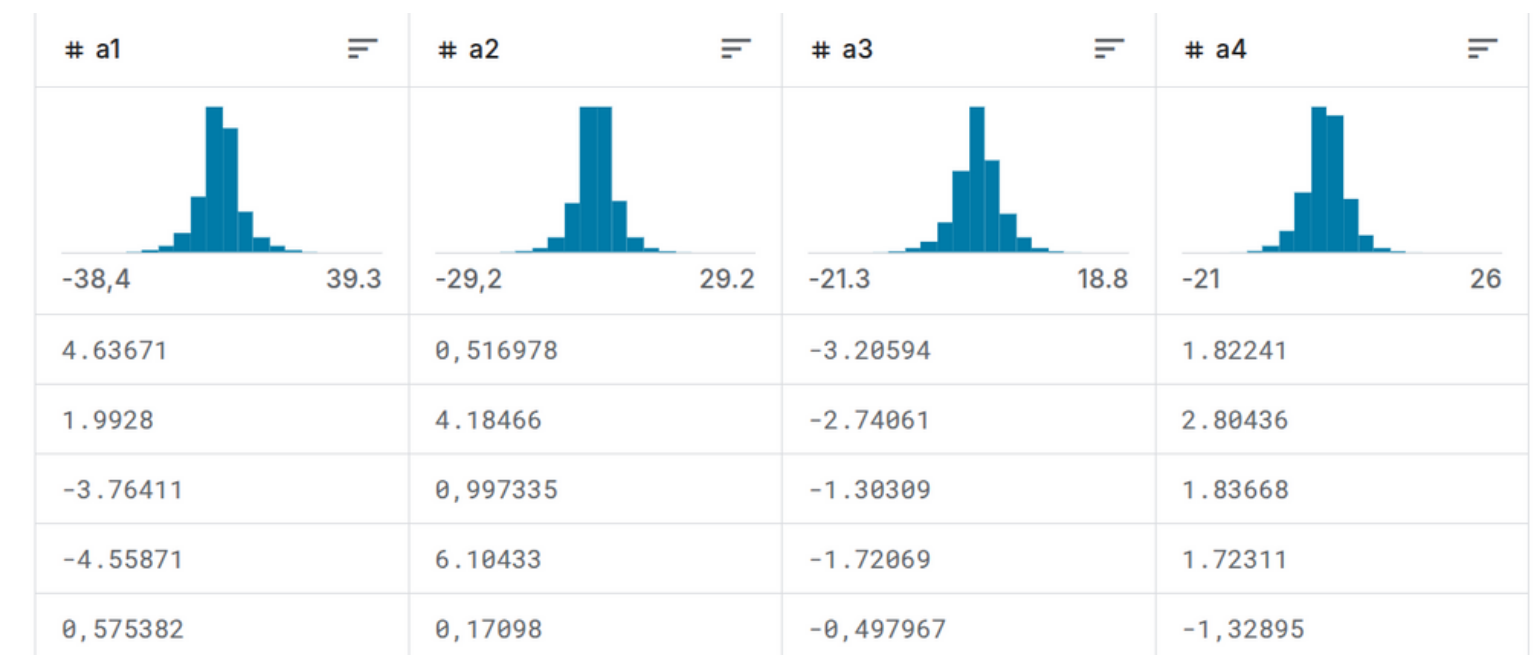


Esempio di analisi dei dati con python

- Analisi delle vibrazioni del cambio
- Fonte dati: 4 sensori di vibrazione posizionati in direzioni diverse.
- Variazione del carico: dallo 0% al 90% (con incrementi del 10%).
- Condizioni del cambio:
 - Sano: 10 file di dati (da 0% a 90% di carico).
 - Dente rotto: 10 file di dati (da 0% a 90% di carico).



Con
guasto



Senza
guasto

Esempio di analisi dei dati con python

```
#Importiamo le librerie da utilizzare:
import numpy as np #algebra lineare
import pandas as pd #data processing, lettura file CSV
import os #lettura cartelle
import matplotlib.pyplot as plt #grafici
import seaborn as sns #grafici
from scipy import fft, signal, stats #analisi spettrale e statistica

#importiamo i dati di tutti i file in un'unica matrice
#colonne: nr. campione, valore sensore a[1..4], stato e carico
dfs = []
for dirname, _, filenames in tqdm(os.walk('/kaggle/input')):
    for filename in tqdm(filenames, leave=False):
        ...
df = pd.concat(dfs).reset_index().rename(columns={'index': 'sample_index'})
```

È complicato?

Perchè non usare un LLM?

- Un buon prompt può farci risparmiare ore di lavoro
- Una volta importato il codice deve essere testato ed eventualmente modificato
 - Le risposte non sono perfette perchè il prompt o il contesto sono imprecisi o per errori dell'LLM
- Devo essere in grado di capire il codice

in python ho 20 file da importare in una matrice, i file hanno nome [b|h]frequenzaHzCarico.csv, voglio ottenere una matrice con una colonna numero progressivo del file, poi le colonne contenute nel file, una colonna con b o h e una colonna con il Carico

Per importare i tuoi 20 file CSV in Python e creare una matrice strutturata come richiesto, puoi seguire un processo che prevede l'iterazione sui file, l'estrazione delle informazioni dai nomi dei file e la combinazione dei dati in un unico DataFrame Pandas.

3. Implementazione del codice

Ecco un esempio di script Python che esegue l'operazione descritta. Assicurati che i file CSV si trovino nella stessa directory dello script o di specificare il percorso corretto.

Python

```
import pandas as pd
import glob
import re

# Crea una lista di tutti i file CSV che corrispondono al pattern
file_pattern = '[b|h]*.csv'
file_list = glob.glob(file_pattern)

# Inizializza una lista per i DataFrame
df_list = []
file_counter = 1
```

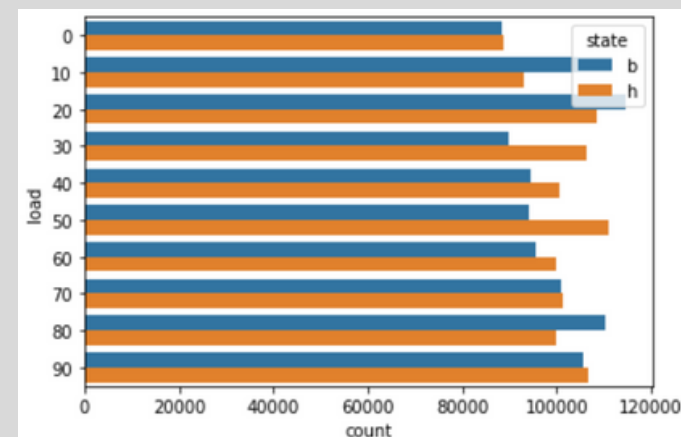
Cominciamo con data-esploration

- Comprendere i dati, identificare i problemi, formulare ipotesi

```
df.sample(3, random_state=42)
```

	sample_index	a1	a2	a3	a4	state	load
1583597	50926	-2.07267	-3.895440	-0.040142	-5.86337	h	10
1197075	83220	15.33900	-0.266811	-2.544860	6.91818	h	70
17675	17675	4.01382	1.450290	1.473320	-1.63767	b	10

```
sns.countplot(data=sensor_readings[sensor_readings.sensor=='a1'], y='load', hue='state', )
```

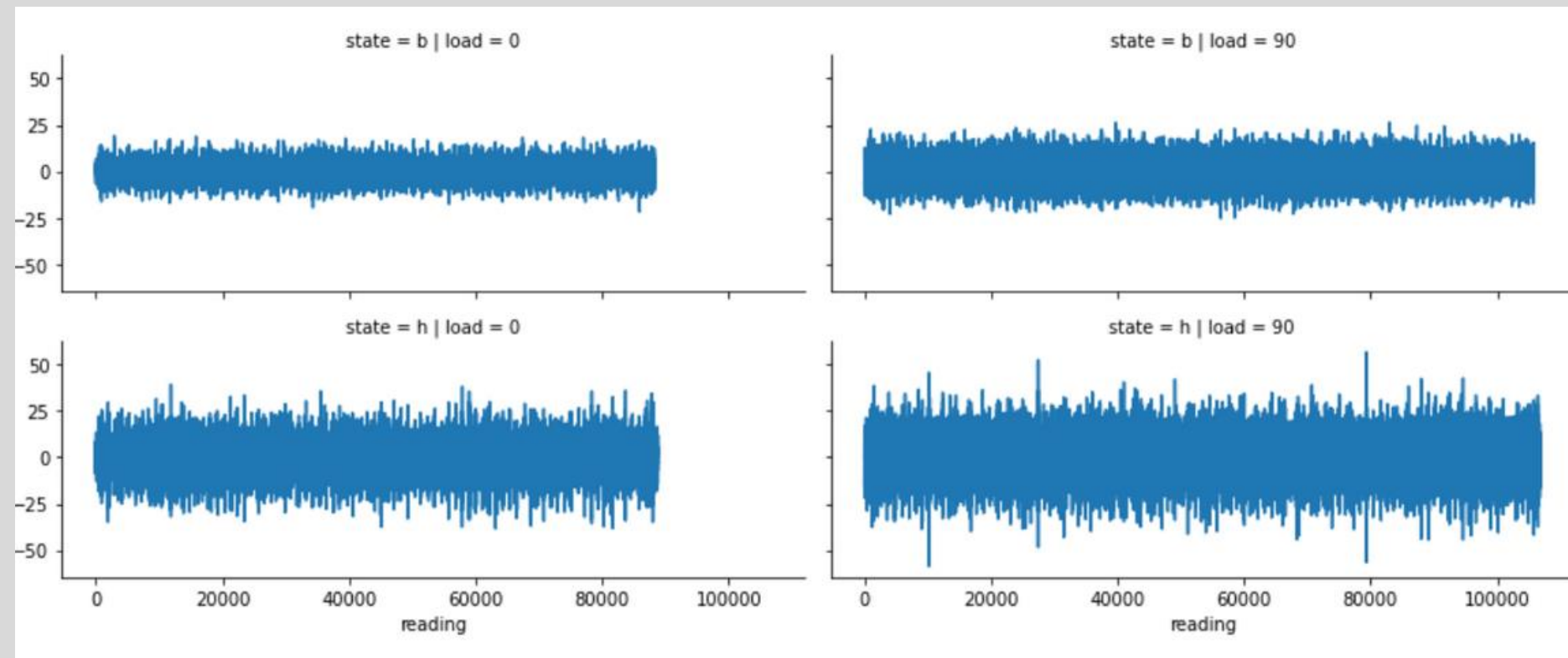


```
lowest_samples = df.groupby(['state', 'load']).sample_index.count().min()
print(f'min_num_campioni = {lowest_samples}, {lowest_samples/30/60:0.2f} minuti')
```

```
min_num_campioni = 88320, 49.07 minuti
```

Visualizziamo i dati

```
#costruiamo un grafico per confrontare sano/dente rotto con carico 0 e 90
g = sns.FacetGrid(data=pd.concat([rdg(sensor_readings, load=0,
sensor='a1'),rdg(sensor_readings, load=90,
sensor='a1')] ),col='load',row='state',height=2.5,aspect=2.5)
g.map(plt.plot, 'reading')
plt.show()
```

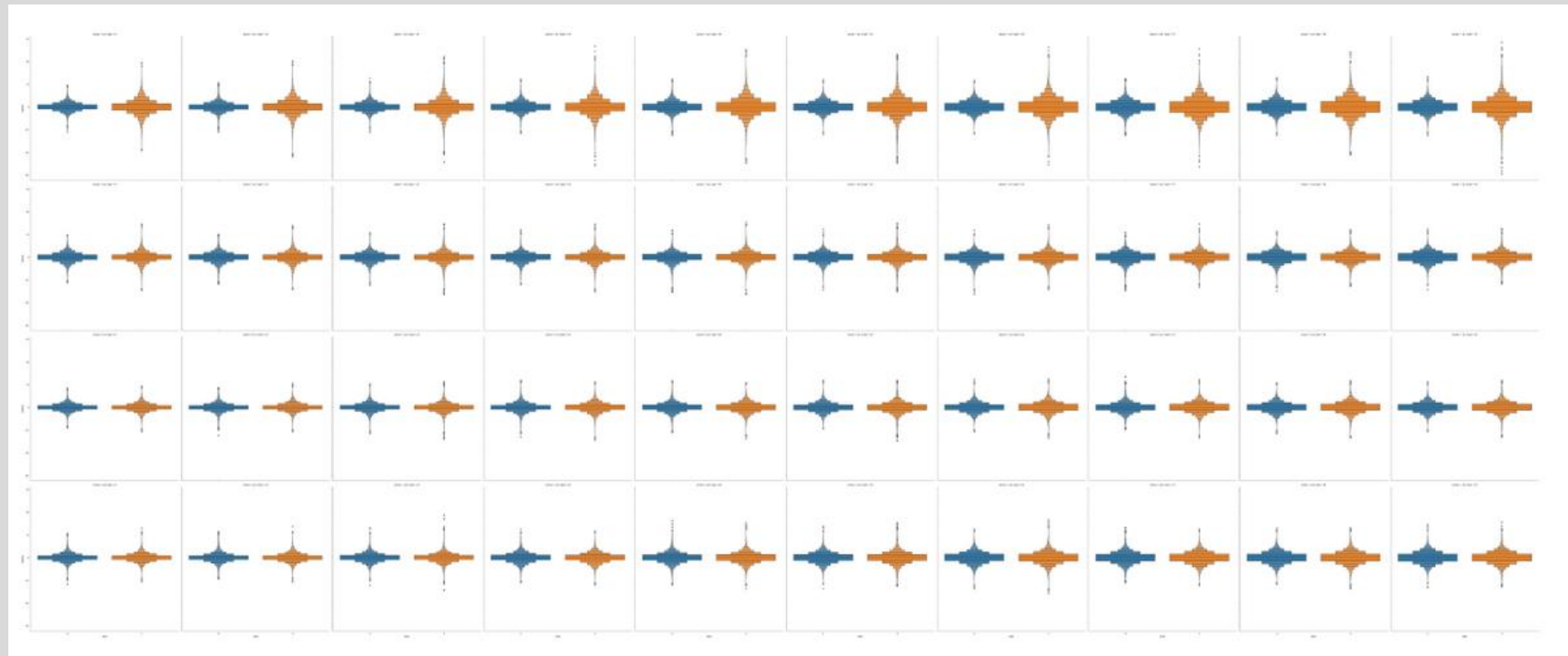


#Si nota l'ampiezza diversa tra le due categorie!

Visualizziamo i dati

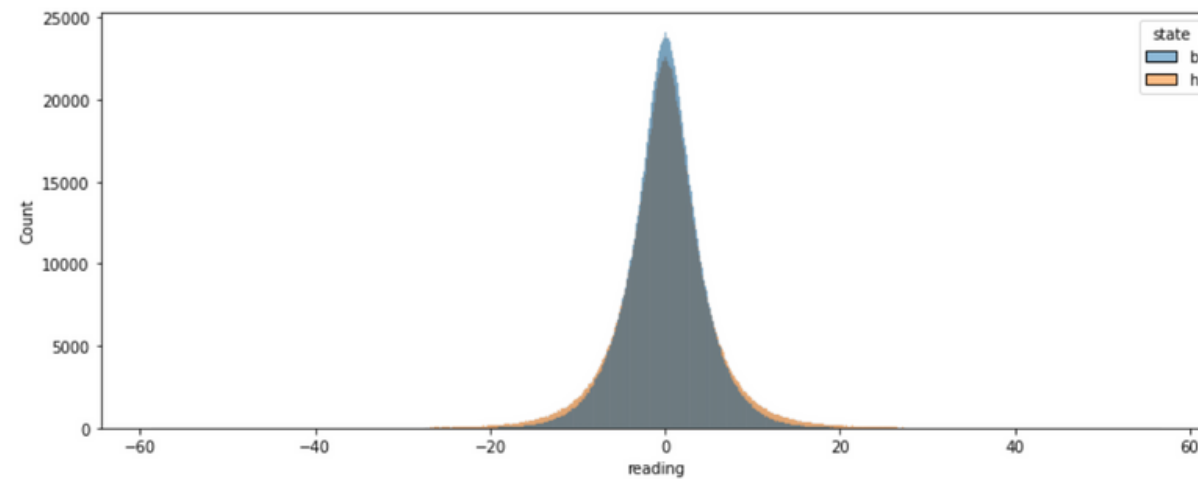
#è una caratteristica solo dei carichi a 0 e 90? Verifichiamo:

```
sns.catplot(data=sensor_readings,col='load', row='sensor',x='state',  
y='reading',kind='boxen',height=10)
```

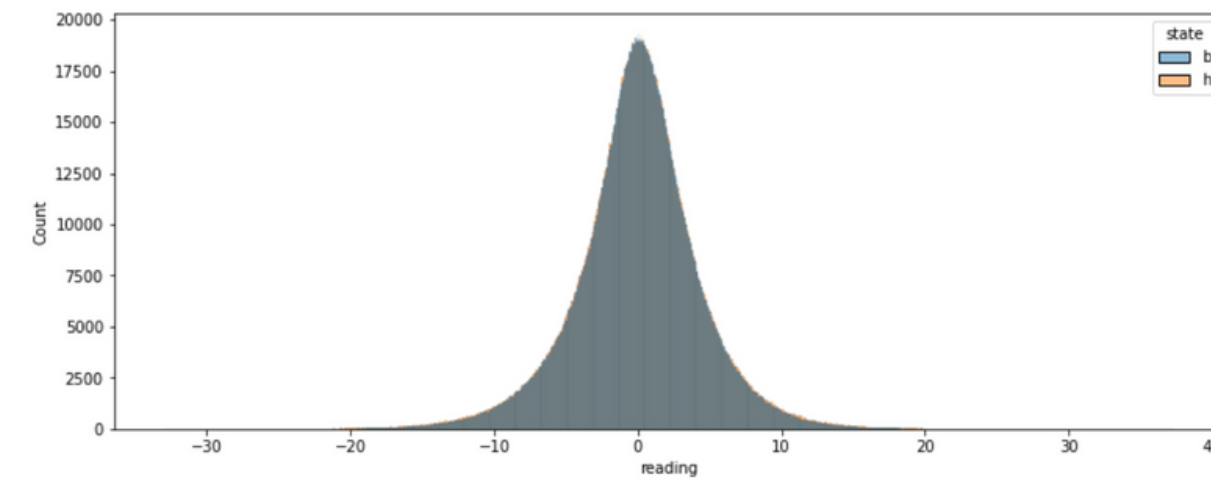


#La caratteristica si mantiene per tutti i carichi
#Il primo sensore è quello che mostra maggiori differenze

Visualizziamo i dati



Istogramma di tutti i
sensori

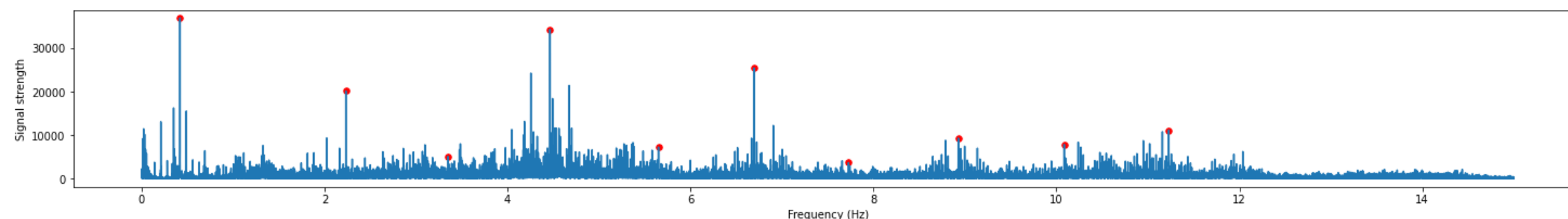
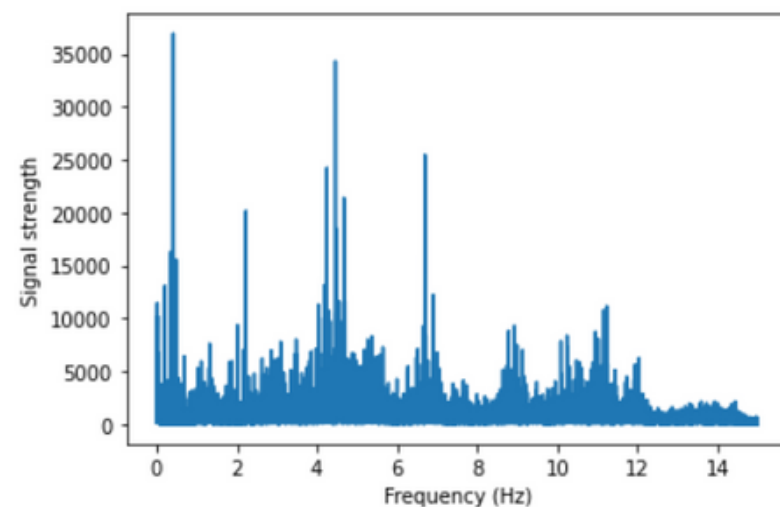


Istogramma sensori 2, 3,
4i

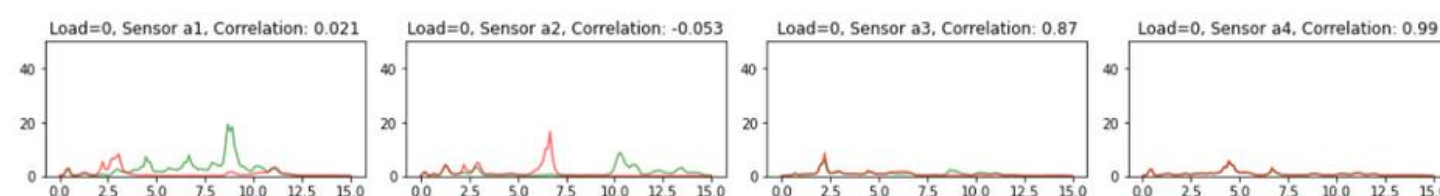
- Questa semplice analisi ci mostra che possiamo utilizzare la misurazione eseguita dal primo sensore per diagnosticare la rottura di un dente del cambio
- Ci mostra anche che la vibrazione tipica è principalmente in una sola direzione, quella del primo sensore

La statistica è il nostro strumento principale, soprattutto l'analisi della
posizione e della varianza dei dati

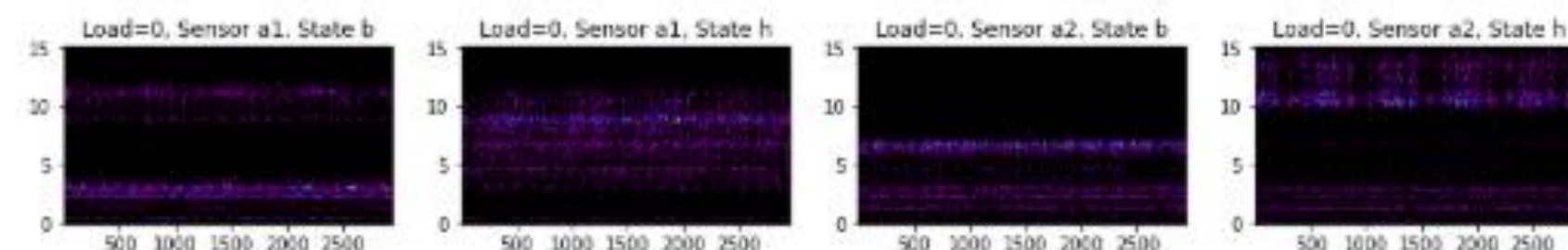
Approfondiamo con l'analisi in frequenza



Identifichiamo i valori di picco



Confronto in frequenza tra sensori diversi

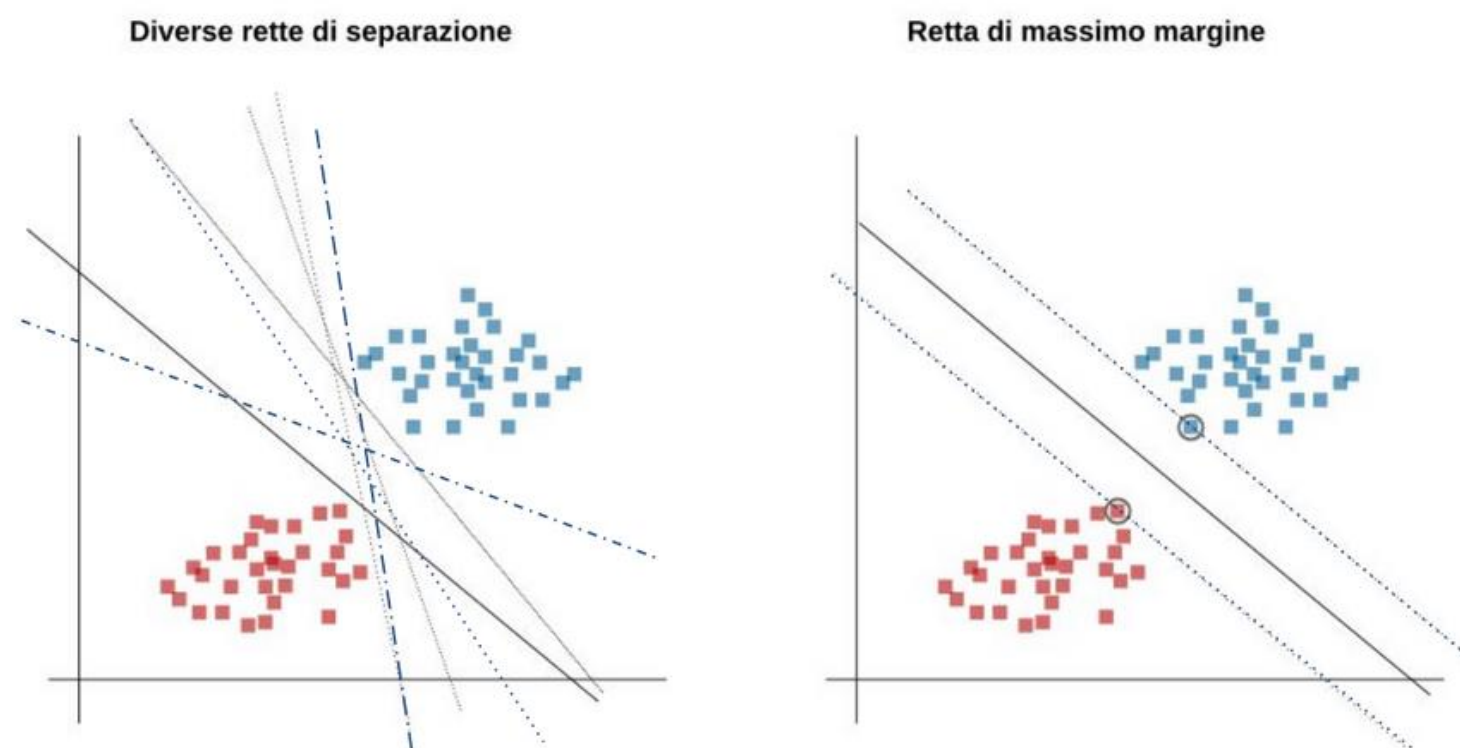


Spettrogramma

- Dal confronto si evidenzia che il funzionamento normale ha un picco nel sensore 1 a circa 9Hz che manca nel funzionamento con guasto
- Il sensore 2 ha un picco a circa 6Hz nel sensore 2, non presente nel funzionamento normale

Esempio di machine-learning: SVM

- Support Vector Machine
 - È un algoritmo di apprendimento supervisionato utilizzato principalmente per problemi di classificazione e regressione.
 - L'obiettivo di una SVM è trovare il "miglior" confine decisionale (chiamato iperpiano) che separi le classi di dati, massimizzando il margine tra esse.
- A differenza di altri algoritmi, la SVM non si limita a trovare una linea qualsiasi, ma cerca quella che ha la massima distanza dai punti dati più vicini di ciascuna classe.
- Lo scopo è trovare un modello in grado di generalizzare



Costruzione SVM: suddivisione dati

```
#Selezione delle caratteristiche in input
training_features = ['a1', 'a2', 'a3', 'a4', 'load']
#output
label = ['failure']

#Diamo nomi più semplici
x = gear_data[training_features]
y = gear_data[label]

#Randomizziamo, utile per poi dividere in dataset di train e test per verificare
overfitting
x,y = shuffle(x,y)

#dividiamo utilizzando il 33% dei dati come test
x1_train, x1_test, y1_train, y1_test = train_test_split(x1, y1, test_size=0.33,
random_state=42)
```

Costruzione SVM: valutazione

```
#Creiamo il classificatore con kernel rbf (Radial Basis Function)
classifier = svm.SVC(kernel='rbf', gamma='auto', C=1.5)

#Addestriamo il classificatore
classifier.fit(x1_train, y1_train.values.ravel())

#Verifichiamo con i dati di test
y_pred = classifier.predict(x1_test)

#Vediamo i risultati della classificazione: classi 0 e 1
print(classification_report(y1_test, y_pred))
```

	precision	recall	f1-score	support
0.0	0.57	0.57	0.57	9869
1.0	0.58	0.58	0.58	9931
accuracy			0.57	19800
macro avg	0.57	0.57	0.57	19800
weighted avg	0.57	0.57	0.57	19800

57% di precisione, non sufficiente

Come posso migliorare il sistema?

Per esempio con una trasformazione dei dati:

- Selezione dei dati solo del primo sensore (riduzione della dimensionalità)
- Analisi dei dati in frequenza (trasformazione di dominio)
- Modifica dei parametri del classificatore

Esempi di altre metodologie

- **Learning Decision Tree**

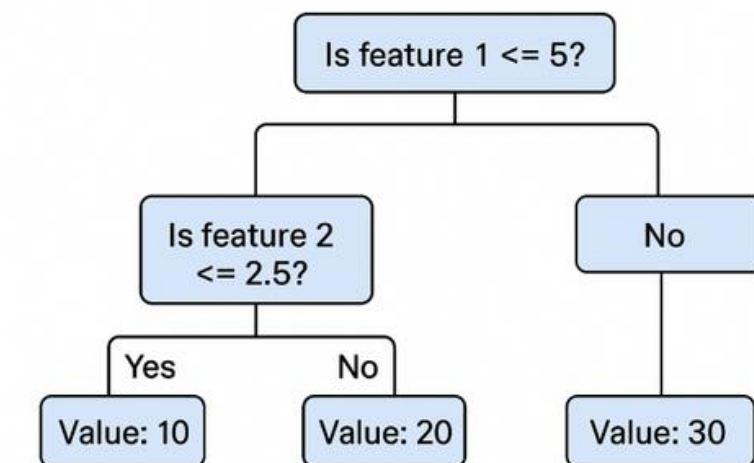
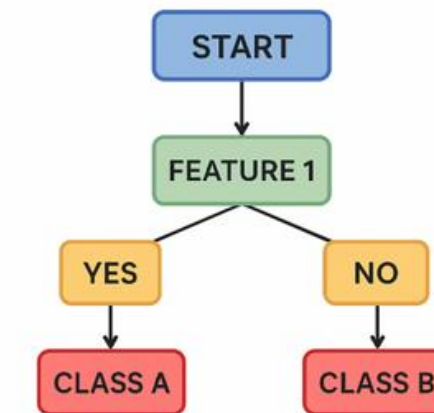
- Problemi di classificazione e regressione
- Si parte dal considerare tutti i dati e poi si eseguono partizioni ricorsive dei dati sulla base della migliore caratteristica
- Criticità: overfitting

- **Random Forest**

- Utilizza un insieme di alberi decisionali (Forest)
- Ogni albero viene addestrato su un sottoinsieme dei dati
- Votazione a maggioranza per classificazione o media per regressione

- **Gradient boosting**

- L'albero decisionale viene addestrato in fasi sui residui (errori) della fase precedente
- È un sistema sequenziale e incrementale
- Criticità: overfitting



Esempi di altre metodologie

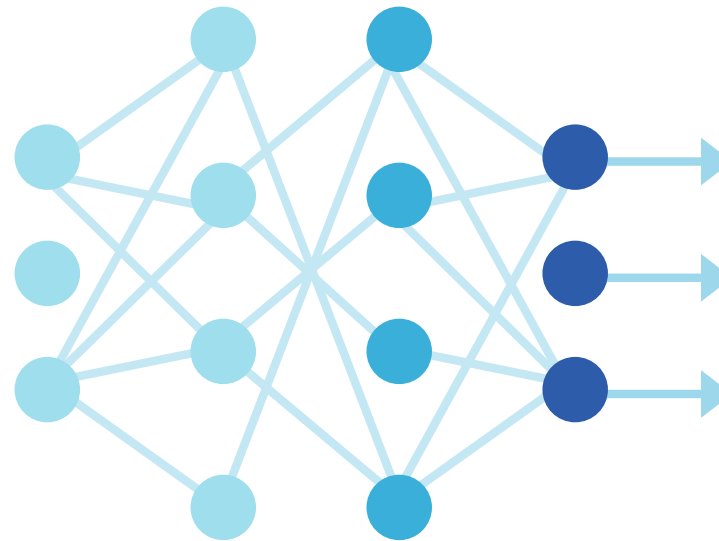
- **Isolation Forest**
 - Problemi di individuazione delle anomalie
 - Algoritmo non supervisionato, non richiede addestramento
 - Si partizionano i dati in un albero e si misura la lunghezza del percorso
- **Sequential Discounting Auto-Regression**
 - Problemi di previsione di anomalie da serie temporali
 - Algoritmo non supervisionato
 - Utile per l'analisi online
 - Modello auto-regressivo che rileva anomalie quando non riesce più a prevedere i dati successivi
- **Principal component analysis**
- ...

**La ricerca ha creato
numerosi modelli di ML, il
compito del data-analyst è
di selezionare e studiare il
modello migliore
per il caso specifico.**

Deep Learning



Dati storici



Elaborazione



Previsione

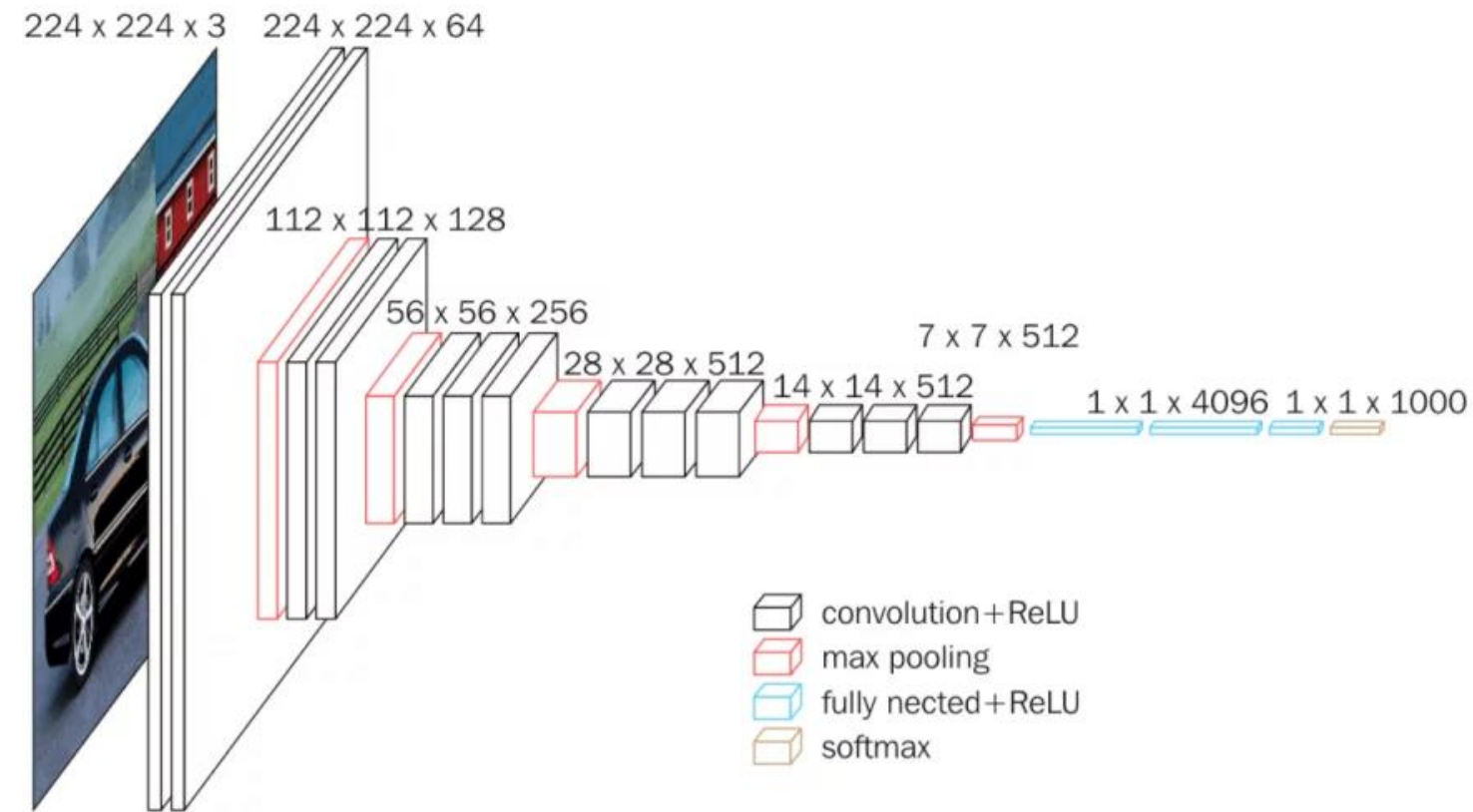
- Reti neurali a più strati
 - Risolto il problema della “scomparsa del gradiente” su reti profonde
 - Nuove architetture, funzioni di attivazione, addestramento
 - Potenza di calcolo (GPU)
 - Disponibilità di grandi quantità di dati
- Efficaci nel trovare schemi complessi e non lineari nei dati
- Più complesse da impostare, molto potenti
- Architetture specializzate

Deep Learning: principali tipologie

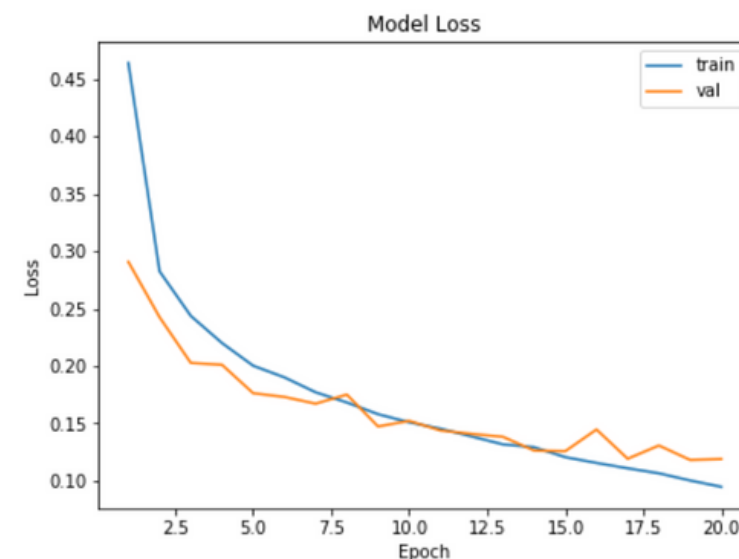
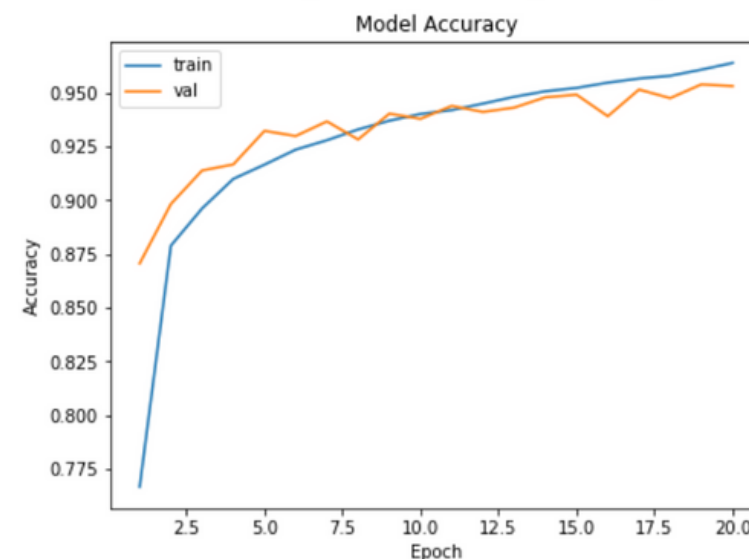
- Strato di **input**
- Strati nascosti
 - **Fully connected**
 - Ogni neurone in uno strato nascosto è connesso a tutti i neuroni dello strato precedente.
 - Questo strato applica una trasformazione lineare ai dati in ingresso e poi li passa attraverso una funzione di attivazione (ad es. ReLU, Sigmoid) per introdurre non linearità e permettere al modello di imparare schemi complessi.
 - Le reti "profonde" hanno più di uno di questi strati.
 - **Convolutionali (CNN)**
 - Non sono adatti per dati tabellari aziendali standard, ma sono un building block fondamentale nelle reti profonde per la computer vision.
 - Usano dei "filtri" per scansionare l'immagine e rilevare caratteristiche come bordi, angoli e forme.
 - **Ricorrenti (RNN)**
 - Sono specializzati nel processare dati sequenziali (come serie storiche aziendali, testo o audio).
 - A differenza di altri strati, hanno una "memoria" che permette loro di considerare l'ordine degli input.
 - Sono ideali per la previsione di serie storiche, come i prezzi delle azioni o le vendite mensili.
- Strato di **output**
 - Classificazione binaria o multicalsse
 - Regressione

Esempio: cat or dog?

Conforme o non conforme?

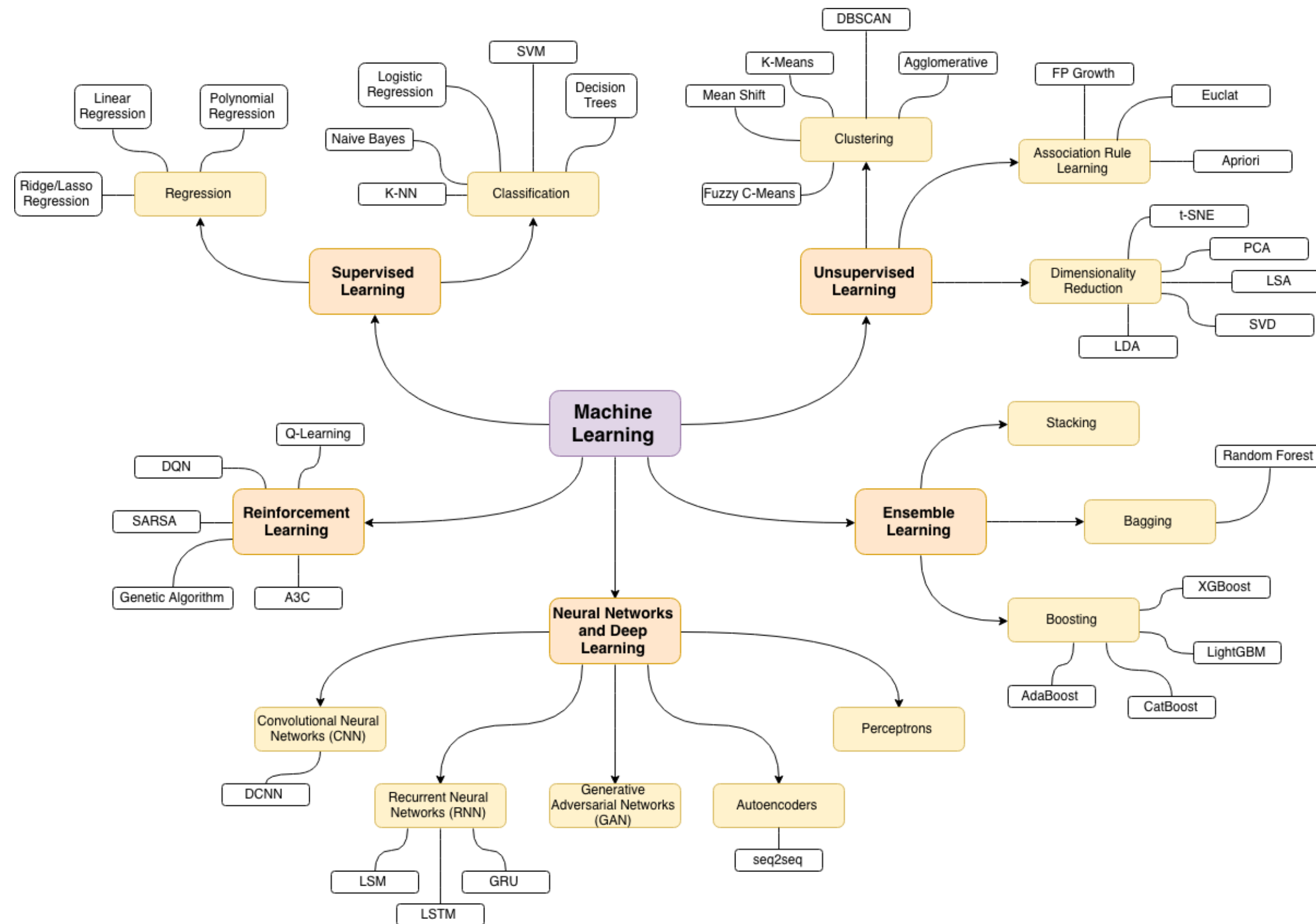


Result of VGG16 using transfer learning on 20 epochs.

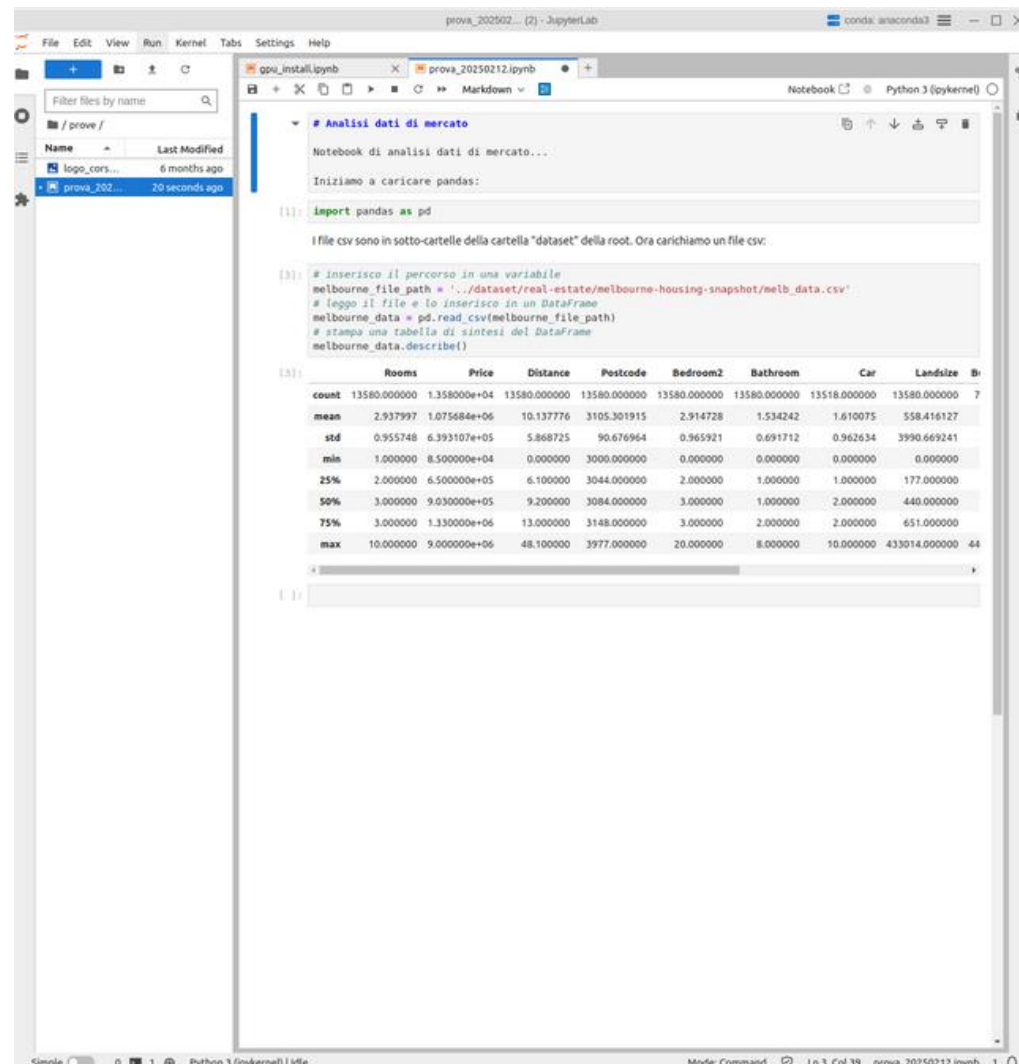


- Le reti DL possono essere anche molto complesse
- L'architettura va progettata con cura (a volte anche per tentativi)
 - Strati, funzioni di attivazione, iperparametri
- Quanti dati servono? >>> ...transfer learning
- Complicato? Perchè non farlo fare a un algoritmo?
 - NAS: **Neural Architecture Search**
 - Eseguono un'ottimizzazione valutando tutte le possibili architetture
 - Utilizzano Reinforcement learning, algoritmi genetici, ricerca casuale o bayesiana
 - Criticità: potenza di calcolo, tempi

Una galassia di algoritmi



Come documentare l'analisi? I notebook



- Una documentazione accurata è fondamentale per garantire la **tracciabilità** e la **verifica** dell'analisi dei dati. Integrandola nella gestione della conoscenza aziendale, si rende il lavoro riutilizzabile e condivisibile.
- I **notebook** (come Jupyter) sono lo strumento ideale per la documentazione. Permettono di combinare codice, visualizzazioni e testo descrittivo in un unico ambiente.
- Questo approccio offre due vantaggi chiave:
 - **Sincronia**: la documentazione viene creata durante l'analisi, non in un secondo momento, garantendo che sia sempre aggiornata e allineata con i passaggi eseguiti.
 - **Efficacia**: la documentazione non è un'attività separata, ma è il programma di analisi stesso. Il codice, i grafici e le spiegazioni diventano un'unica entità coerente.



"L'IA è la nuova elettricità. Trasformerà ogni settore e creerà nuovi lavori. Non si tratta dell'IA che sostituisce le persone. Si tratta dell'IA che abilita le persone."

Andrew Ng

MIND

Dal 2 ottobre 2025
IN AULA E ONLINE

MACHINE LEARNING PER L'INNOVAZIONE E LA DIGITALIZZAZIONE AZIENDALE



Un corso per migliorare i **processi** aziendali, prevedere **trend** di mercato, ottimizzare la gestione del **rischio** e supportare decisioni **data-driven**, con strumenti concreti di data analysis e machine learning.

02/10/25 ○ ——— ○ ——— ○ **19/02/26**

LEZIONI / ESERCITAZIONI / PROJECT WORK FINALE

ARGOMENTI

FONDAMENTI DI PYTHON PER IL BUSINESS

Librerie per analisi dati e creazione di dashboard interattive.

MODELLI DI REGRESSIONE E PREVISIONE PER L'OTTIMIZZAZIONE AZIENDALE

Regressione lineare, Random Forest e metriche di valutazione.

OTTIMIZZAZIONE DEI MODELLI REGRESSIVI E SIMULAZIONE

Iper-parametri e tecniche di cross-validation per migliorare la generalizzazione.

FEATURE ENGINEERING PER MIGLIORARE L'ACCURATEZZA DEI MODELLI

Selezione, trasformazione e riduzione della dimensionalità.

DEEP LEARNING PER LA QUALITÀ E AUTOMAZIONE INDUSTRIALE

Implementazione di reti neurali per dati strutturati e immagini.

ANALISI DI SERIE TEMPORALI PER FORECASTING AZIENDALE

Modelli ARIMA, SARIMA e RNN per la previsione di trend produttivi.



MODALITÀ ACTION LEARNING

- 11 LEZIONI
- 63 ORE DI FORMAZIONE
- 36 ORE IN PRESENZA | 27 ORE ONLINE
- ESERCITAZIONI PRATICHE
- PROJECT WORK FINALE



REQUISITI DI ACCESSO

- Diploma tecnico di scuola superiore con conoscenze di statistica di base (requisito minimo)
- Preferibile laurea scientifica
- Buona padronanza di Excel o conoscenze base di linguaggi di programmazione
- Esperienza pregressa nell'analisi dati
- Almeno 3 anni di esperienza aziendale (consigliato)

DOTAZIONI TECNICHE RICHIESTE

- PC personale con almeno 4 GB di RAM e processore quad-core
- Installazione di JupyterLab

APRI LA SCHEDA TECNICA

FINANZIAMENTI

Il corso è finanziabile attraverso le risorse, i piani e gli avvisi dei diversi fondi interprofessionali per i dipendenti e i dirigenti delle aziende (ad es. Fondimpresa e Fondirigenti). Oltre al bando CCIIAA di Brescia dedicato alla formazione professionale.



CSMT INNOVATION HUB
VIA BRANZE 45 BRESCIA
WWW.CSMT.IT
INFO@CSMT.IT
030 6595108



CONTATTI

Andrea Pasotti

Head of Technical Training &
Technology Transfer Senior Project Manager
a.pasotti@csmt.it | +39 371 1688928

Licia Zagni

Head of Sales & Marketing
l.zagni@csmt.it | T. +39 030 6595110 | M. +39 328 4505280