



# Projected Quantum Support Vector Machine per classificazione di pattern di rumore

20 Marzo 2024

Relatore: Prof. Andrea Giachero

Correlatore: Dott. Roberto Moretti

Candidata: Erika Giacomantonio

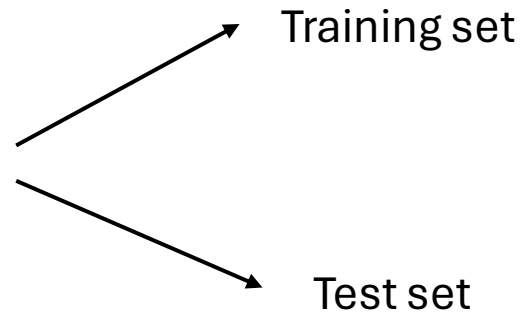
859667

# Machine Learning

Il machine learning (ML) è un settore di studio che fornisce una varietà di modelli in grado di apprendere informazioni dai dati stessi per eseguire operazioni di classificazione, regressione o generazione di dati.

## Dataset:

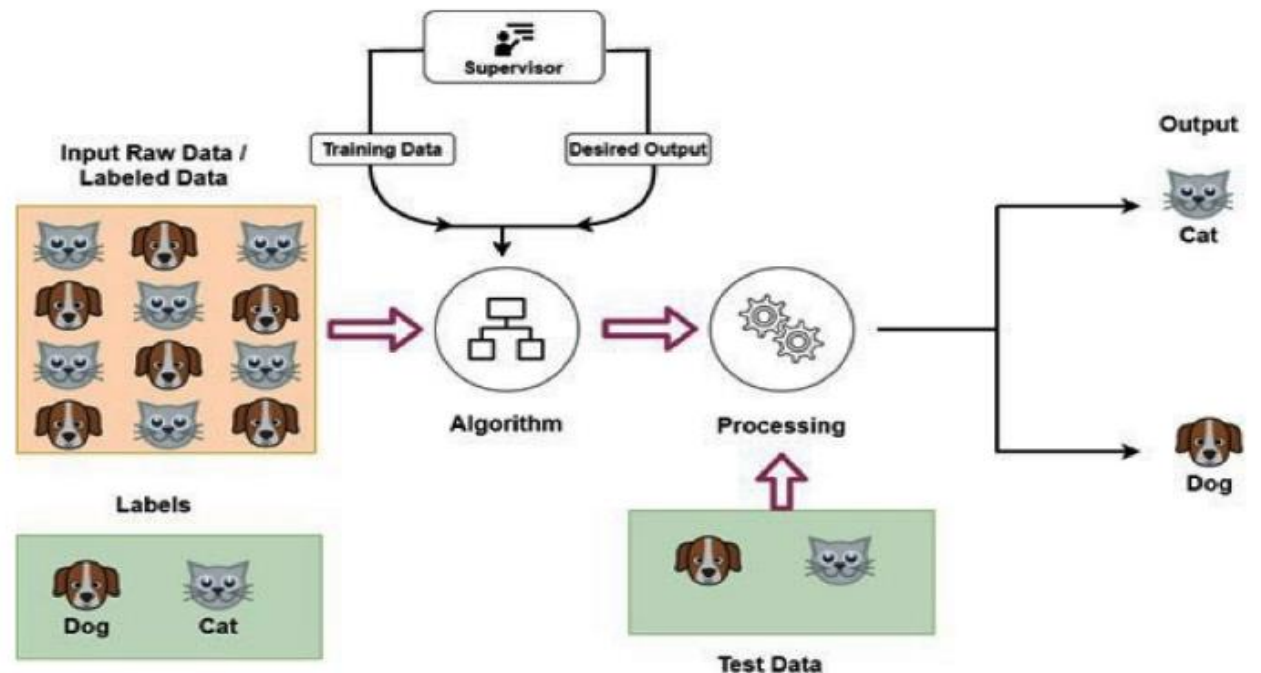
- Pattern
- Feature
- Label



## Supervised Machine Learning

Tipologia di ML che prevede la presenza di labels nel dataset durante la fase di training

$$f(\vec{x}, \vec{P}) = y$$



# Support Vector Machine

È un algoritmo che si basa sull'identificazione di un iperpiano che divide in classi il dataset.

Problema di ottimizzazione:

$$\vec{x} \in \mathbb{R}^n, y \in \{+1, -1\}$$

$$\phi: \mathbb{R}^n \rightarrow F \quad \dim(F) > n$$

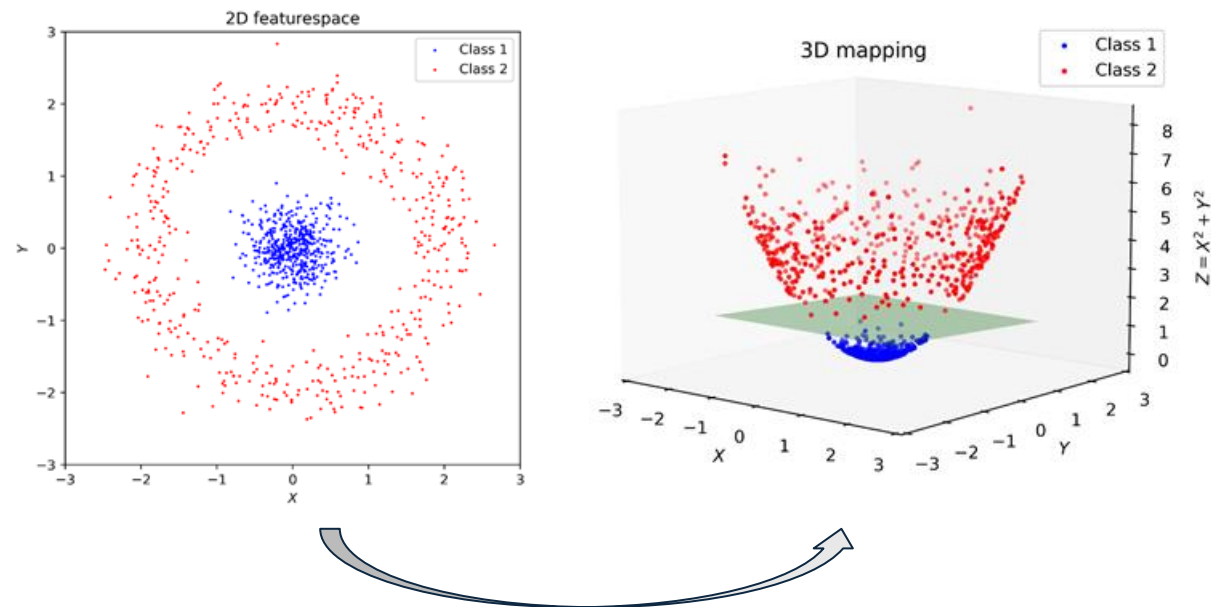
$$\vec{w}^T \cdot \phi(\vec{x}) - b = 0$$

$$f(c_1, c_2, \dots, c_n) = \sum_i c_i - \frac{1}{2} \sum_{ij} y_i c_i y_j c_j \langle \phi(\vec{x}_i), \phi(\vec{x}_j) \rangle$$

$$\text{Con } \vec{w} = \sum_i c_i y_i \phi(\vec{x}_i)$$



**Kernel:**  $K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$



Kernel lineare:

$$K(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j$$

Kernel polinomiale:

$$K(\vec{x}_i, \vec{x}_j) = (\gamma \vec{x}_i \cdot \vec{x}_j + r)^d$$

Kernel Gaussiano:

$$K(\vec{x}_i, \vec{x}_j) = \exp \left( -\gamma \|\vec{x}_i - \vec{x}_j\|^2 + C \right)$$

# Quantum Computing

Il quantum computing sfrutta i qubit per eseguire calcoli

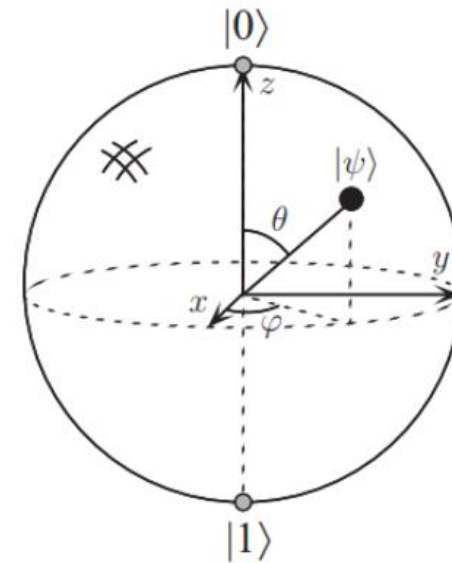
## Qubit

unità fondamentale  
del quantum computing

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

$$B = \left\{ |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle$$



Sfera di Bloch

# Quantum Gates

Lo stato di un qubit può essere controllato tramite i quantum gates

Sono rappresentati da matrici  $U$  tali che:  $UU^\dagger = I$

Gates a singolo qubit

- **Hadamard Gate**

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

stato iniziale	H
$ 0\rangle$	$\frac{ 0\rangle+ 1\rangle}{\sqrt{2}} :=  +\rangle$
$ 1\rangle$	$\frac{ 0\rangle- 1\rangle}{\sqrt{2}} :=  -\rangle$

- **Rotation gates**

Descrivono delle rotazioni attorno ad un asse della sfera di Bloch di un parametro  $\theta$

Gate a multi-qubit

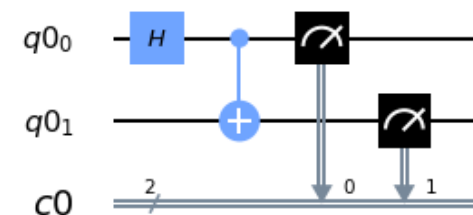
- **CNOT gate**

$$|A, B\rangle \longrightarrow |A, A \oplus B\rangle$$

stato iniziale	CNOT
$ 00\rangle$	$ 00\rangle$
$ 01\rangle$	$ 01\rangle$
$ 10\rangle$	$ 11\rangle$
$ 11\rangle$	$ 10\rangle$

## Circuito quantistico

formato da un insieme di qubits e quantum gates



# Quantum Support Vector Machine e Quantum Kernel

**Quantum Machine Learning:** è un'area del quantum computing che si occupa di potenziare modelli già esistenti di ML oppure di crearne di nuovi sfruttando i principi fondamentali della meccanica quantistica.

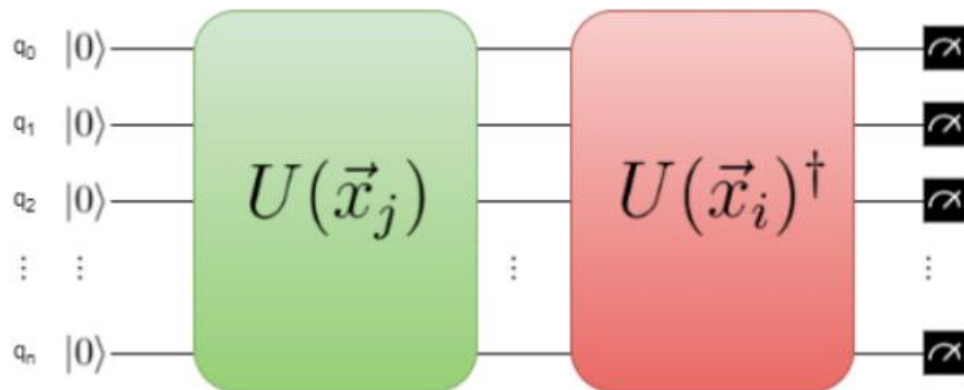
Nel Quantum Support Vector Machine la feature map è uno stato quantistico

$$|\phi(\vec{x}_i)\rangle \langle \phi(\vec{x}_j)| = U(\vec{x}_i) |0^n\rangle \langle 0^n| U(\vec{x}_j)^\dagger$$

**Fidelity quantum kernel:**

$$K(\vec{x}_i, \vec{x}_j) = |\langle \phi(\vec{x}_i) | \phi(\vec{x}_j) \rangle|^2 = |\langle 0^n | U(\vec{x}_i) U(\vec{x}_j)^\dagger | 0^n \rangle|^2$$

Restituisce la probabilità di riottenere lo stato iniziale in seguito all'applicazione della feature map



**Projected quantum kernel:**

$$K(\vec{x}_i, \vec{x}_j) = \exp \left( -\gamma \sum_{k=1}^n \|\rho_k(\vec{x}_i) - \rho_k(\vec{x}_j)\|_2^2 \right)$$

Permette una riduzione dello spazio di Hilbert che contiene lo stato del sistema a più qubit

# Exponential Concentration

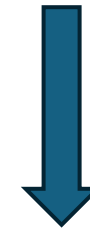
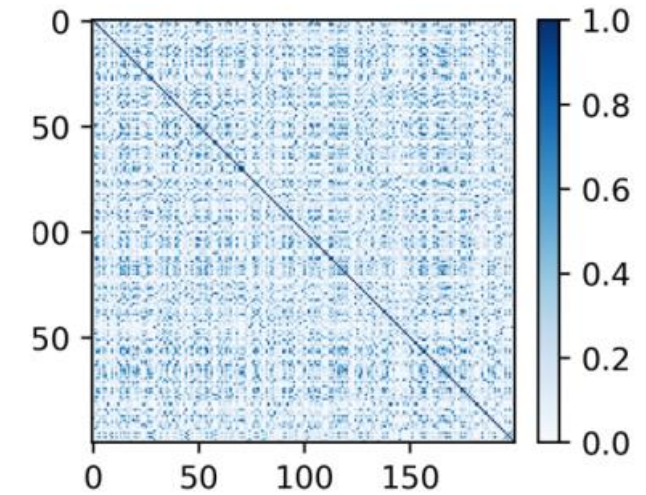
Il Fidelity Quantum kernel è limitato dal problema dell'exponential concentration

All'aumentare del numero di qubit della fmap il valore della funzione di kernel valutata su elementi diversi di un dataset si addensa attorno ad un valore fissato  $\mu \rightarrow$  la varianza al di fuori della diagonale diminuisce

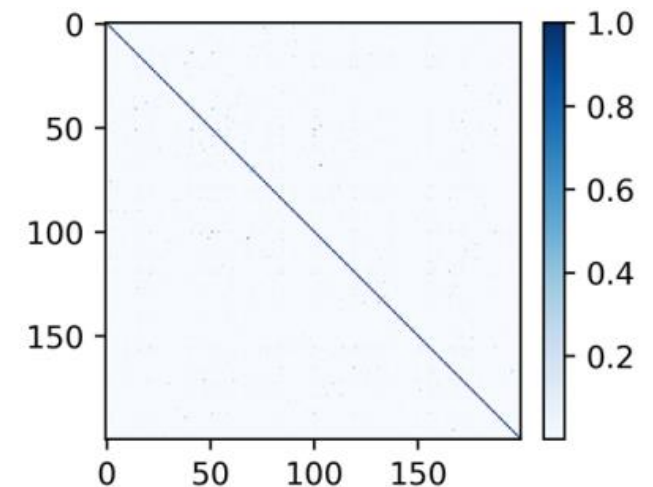
## Conseguenze:

Aumenta precisione richiesta dal dispositivo

È un problema che va mitigato poiché ci troviamo nella NISQ-era (Noise Intermediate Scale Quantum-era)



Caso in cui  
 $\mu=0$





# Dataset

insieme di distribuzioni di stati finali ottenuti in seguito all'esecuzione di un circuito quantistico arbitrario su due diversi computer quantistici simulati, soggetti a diverse tipologie di errore.

## Features:

0: conteggi  $|00\rangle$

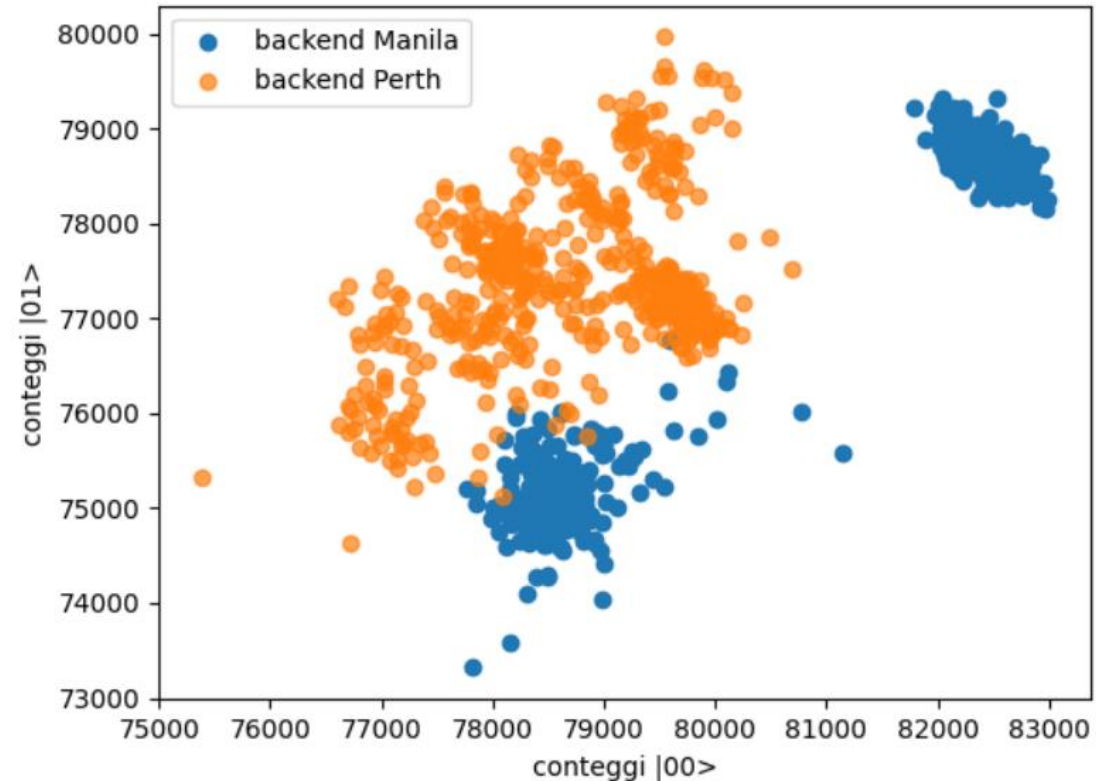
1: conteggi  $|01\rangle$

2: conteggi  $|10\rangle$

3: conteggi  $|11\rangle$

Accuratezze di classificazione:

samples	kernel gaussiano	kernel lineare
500	89,95%	76,90%
600	89,52%	74,72%
700	89,51%	76,23%
800	88,78%	76,19%
900	90,63%	74,60%
1000	88,83%	75,58%





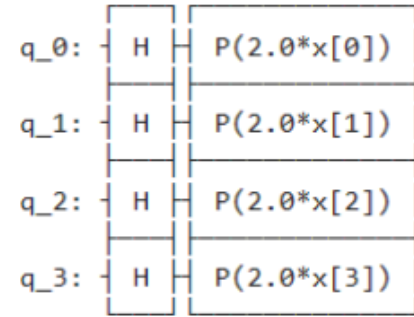
# Quantum feature maps con fidelity quantum kernel

Accuratezze di classificazione:

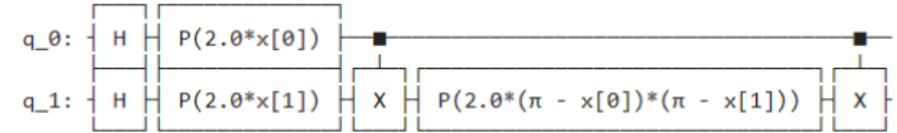
samples	Z	ZZ	fmap 1	fmap 2
500	90,09%	89,88%	83,43%	89,88%
600	89,20%	89,20%	81,90%	89,72%
700	89,28 %	89,18%	81,56%	89,38%
800	88,36%	88,12%	82,23%	88,69%
900	90,29%	90,29%	83,49%	90,29%
1000	88,59%	87,84%	84,17 %	88%

Si ottengono accuratezze  
paragonabili ai SVM classici  
con la Z, ZZ e fmap 2

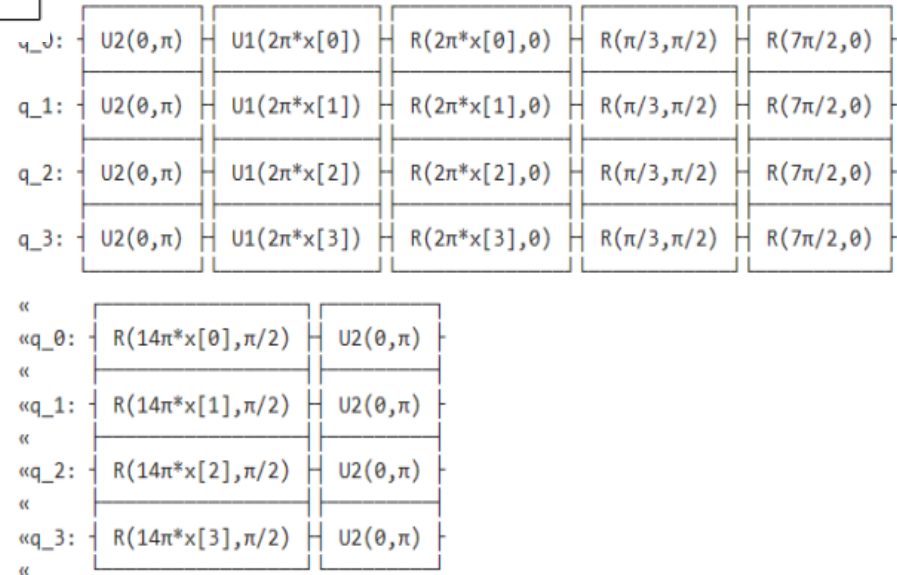
Z feature map



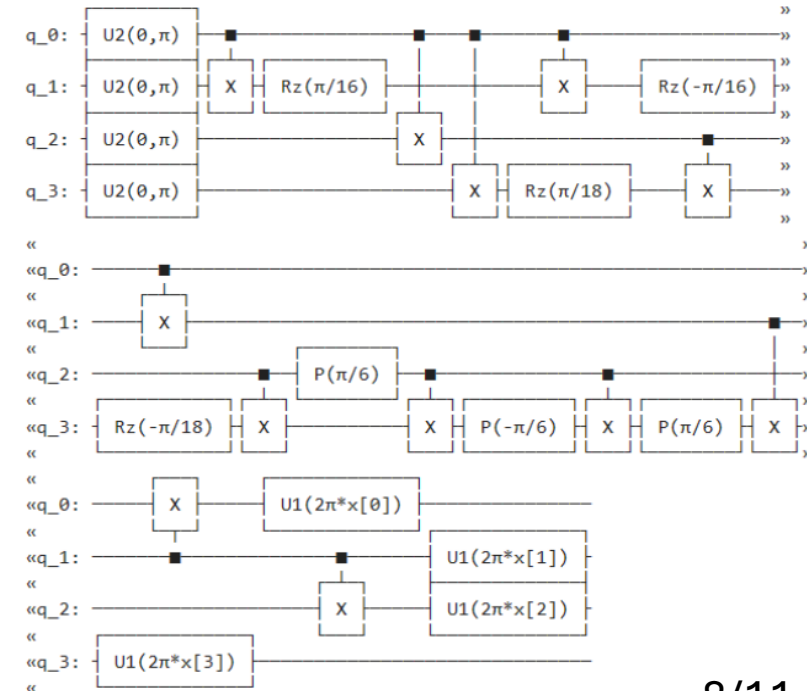
ZZ feature map



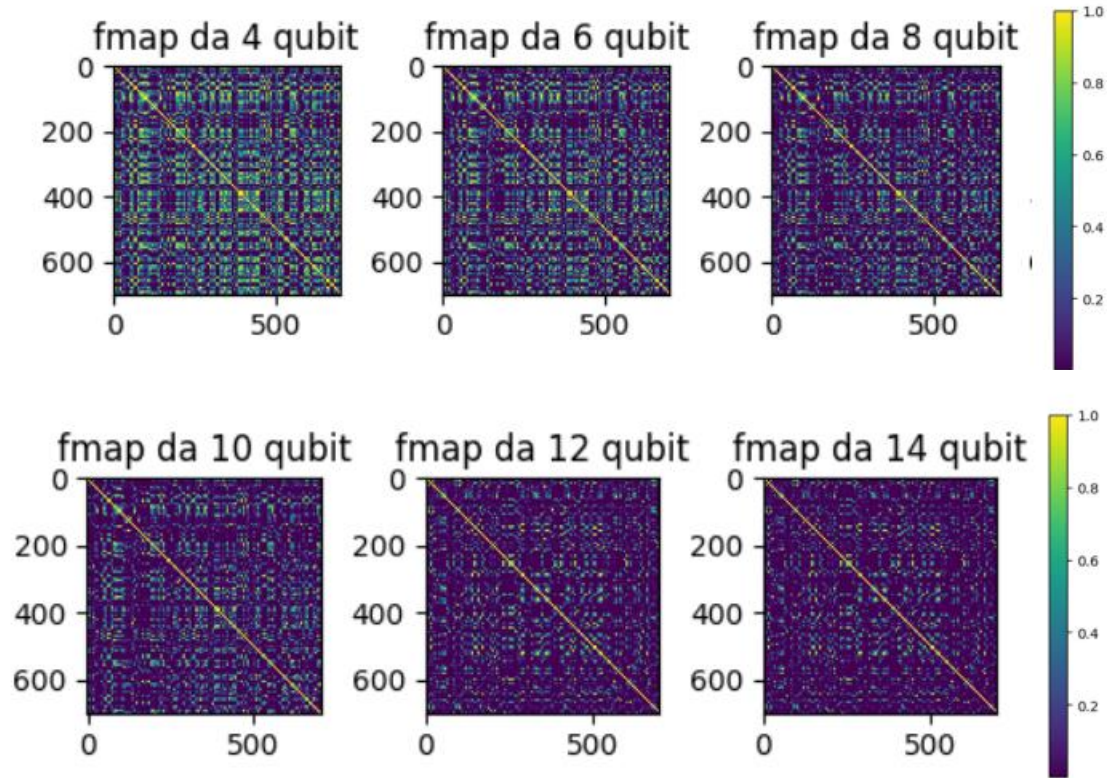
fmap 1 senza entanglement



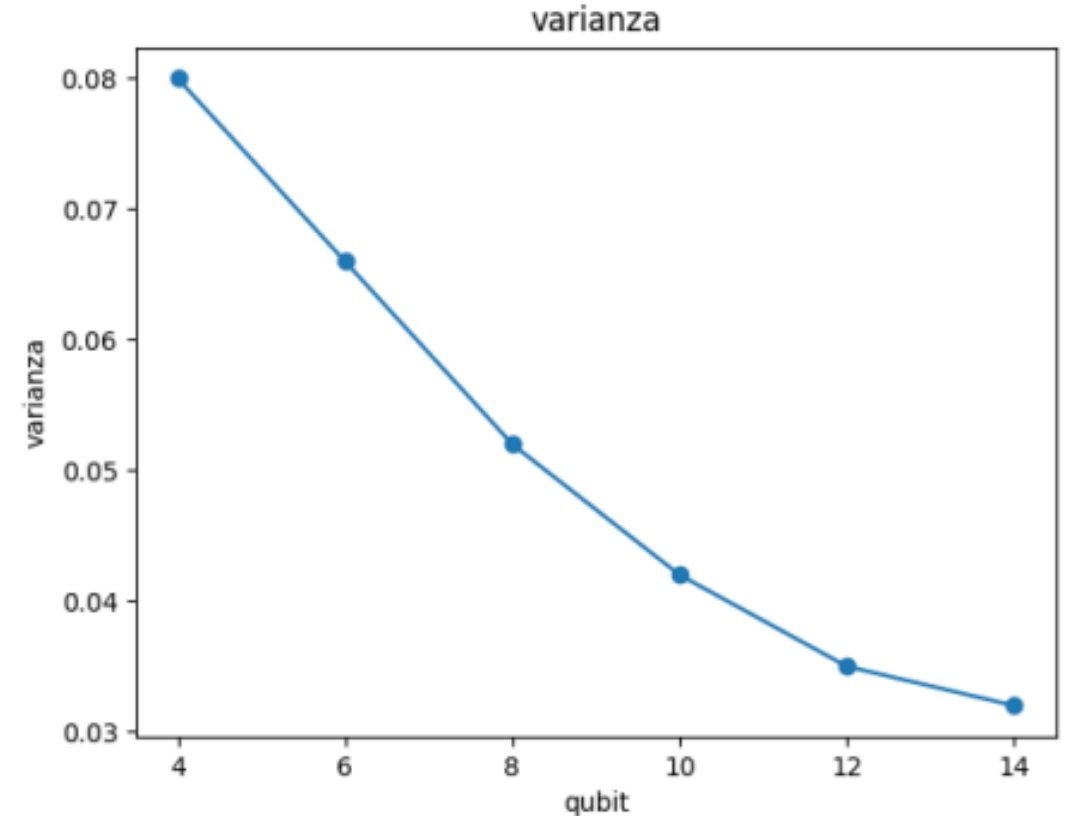
fmap 2 con entanglement



# Studio dell'exponential concentration



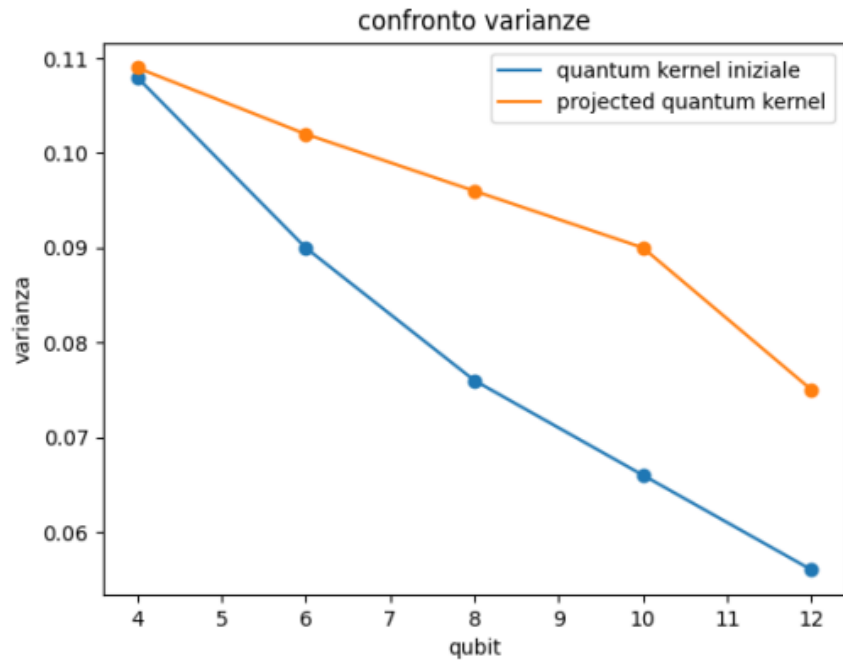
Gli elementi al di fuori della diagonale della matrice di kernel si addensano a  $\mu = 0$



La varianza degli elementi al di fuori della diagonale decresce

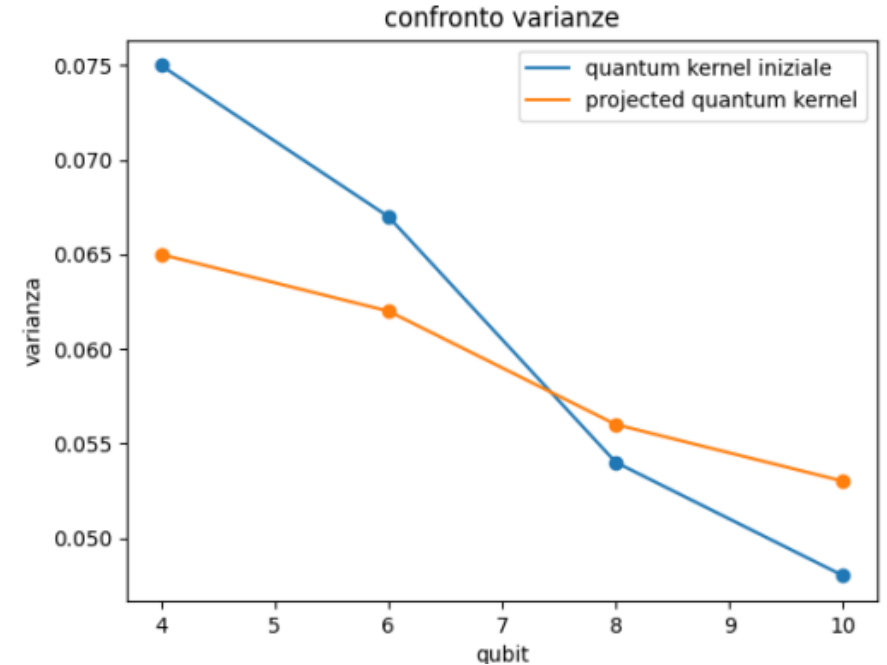
# Mitigazione dell'exponential concentration

Confronto varianze di una riga della matrice di kernel



n	varianza fidelity	varianza projected	aumento percentuale
4	0,108	0,109	0,92 %
6	0,09	0,102	13,33 %
8	0,076	0,096	26,32 %
10	0,066	0,09	36,36 %
12	0,056	0,075	33,93 %

Studio condotto su un dataset di dimensioni ridotte



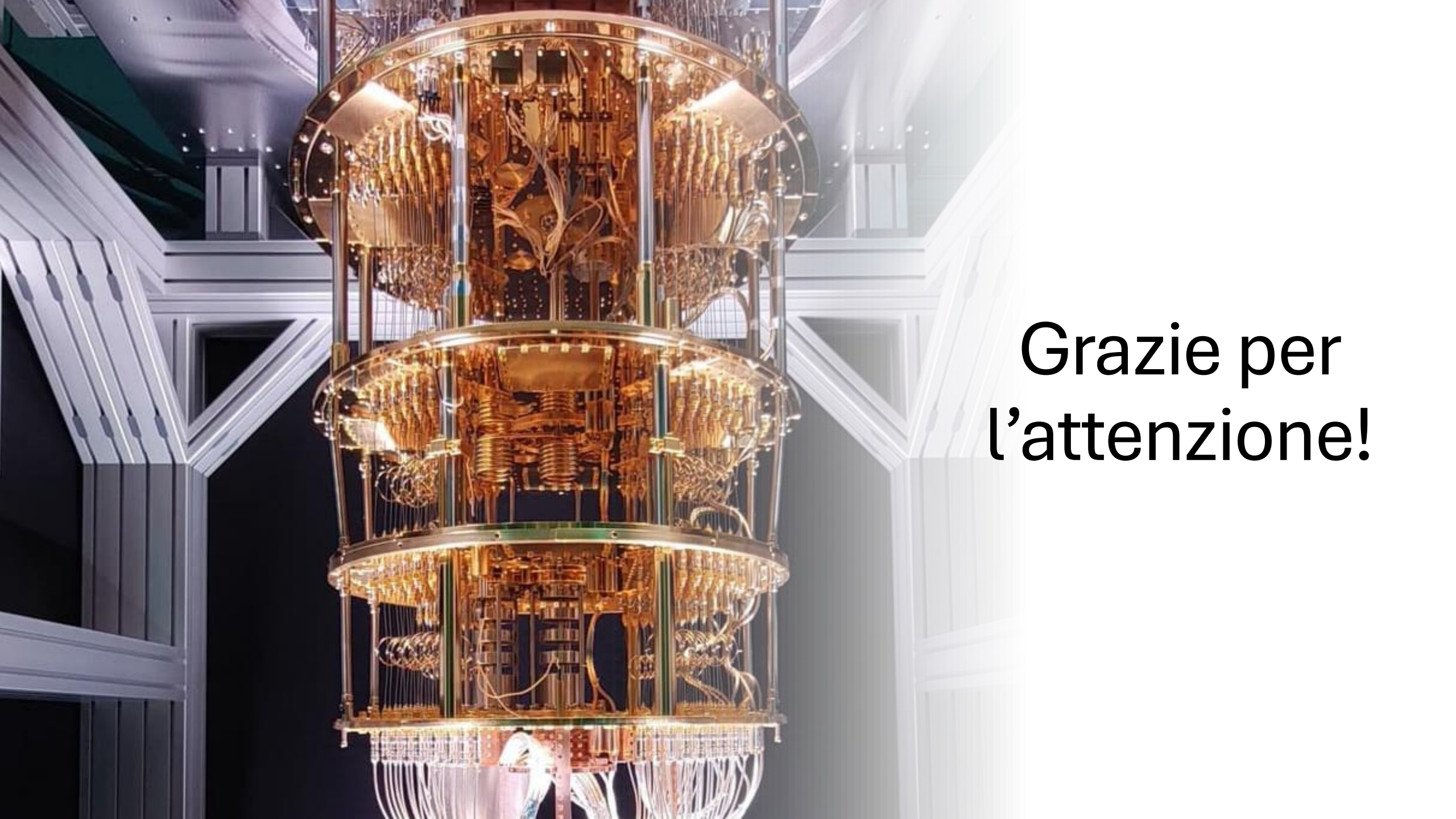
n	accuratezza fidelity	accuratezza projected
4	86,97%	86,67%
6	86 %	86,67 %
8	86,67 %	86,67%
10	88,57%	86,67%

n	varianza fidelity	varianza projected	aumento percentuale
8	0,054	0,056	3,70 %
10	0,041	0,053	29,27 %

# Conclusioni

- Il Quantum Machine Learning è una branca interessata per l'analisi dati che è stata applicata con successo
- Ad oggi le limitazioni hardware richiedono strategie di mitigazione degli errori
- Il Projected quantum kernel è un modello più adatto a limitare l'effetto del rumore dei dispositivi NISQ



A large, ornate, multi-tiered chandelier with a complex mechanical interior, set against a background of a modern building's structural framework. The chandelier features multiple tiers of glass and metal, with intricate mechanical components visible within. The background shows a modern building's structural framework with white beams and a dark interior. The text "Grazie per l'attenzione!" is overlaid on the right side of the image.

Grazie per  
l'attenzione!

# Problema di ottimizzazione SVM

$$M = \min_{1 \leq i \leq n} y_i(\vec{w}^T \cdot \phi(\vec{x}_i) - b) \longrightarrow \text{margini}$$

Problema di ottimizzazione:

$$\max_{\vec{w}, \varepsilon, b} M$$

$$\sum_{j=1}^N w_j^2 = 1$$

$$y_i(\vec{w}^T \cdot \phi(\vec{x}_i) - b) \geq M(1 - \varepsilon_i)$$

$$\varepsilon_i \geq 0, \sum_{i=1}^n \varepsilon_i \leq C$$

Equivalente a



$$f(c_1, c_2, \dots, c_n) = \sum_i c_i - \frac{1}{2} \sum_{ij} y_i c_i y_j c_j \langle \phi(\vec{x}_i), \phi(\vec{x}_j) \rangle$$

$$0 \leq c_i \leq C$$

$$\sum_i c_i y_i = 0$$

$$f(\vec{x}) = \text{sign}(\vec{w}^T \cdot \phi(\vec{x}_i) - b)$$

$$f(\vec{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i K(\vec{x}_i, \vec{x}_j) - b\right)$$

