

# Quantum Kernel Machine Learning: applicazione alla fisica delle alte energie

Progetto finale, corso di Meccanica Quantistica Avanzata

**Università degli Studi di Trieste**  
**A.A. 2021/2022**

Yasmin Bougammoura  
*Data Science and Scientific  
Computing*

Francesco Tomba  
*Data Science and Scientific  
Computing*

Giacomo Zelbi  
*Fisica Teorica*

# Obiettivo:

L'obiettivo del progetto è applicare **Quantum Machine Learning** a un problema di classificazione riguardante la fisica delle alte energie.

Si testano tecniche di ML - in particolare di **Kernel Support Vector Machine** - confrontando i risultati ottenuti utilizzando usando kernel implementati con tecniche di **computazione quantistica** con quelli ottenuti da uno dei kernel classici più diffusi.

# Il problema da affrontare:

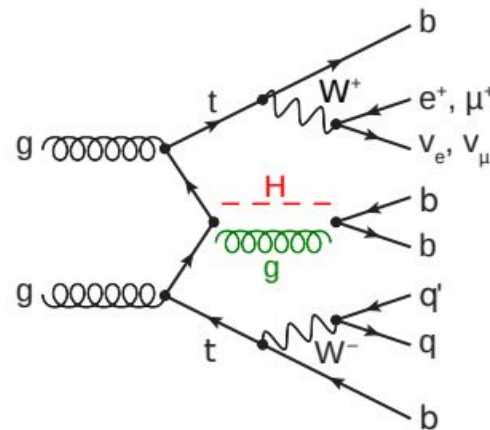
Si considera il processo  $t\bar{t}H(b\bar{b})$  di produzione del bosone di Higgs tramite la collisione di una coppia top quark-antiquark.

In particolare, si vuole riconoscere quando nel processo è stato prodotto un bosone H che è poi decaduto in una coppia b-(anti-b) (**segnale**) e quando invece ciò non è avvenuto (**background**).

La classificazione è binaria.

I dati analizzati provengono da una simulazione e sono scaricabili dal sito:

[https://qml-hep.github.io/qml\\_web/data/](https://qml-hep.github.io/qml_web/data/)



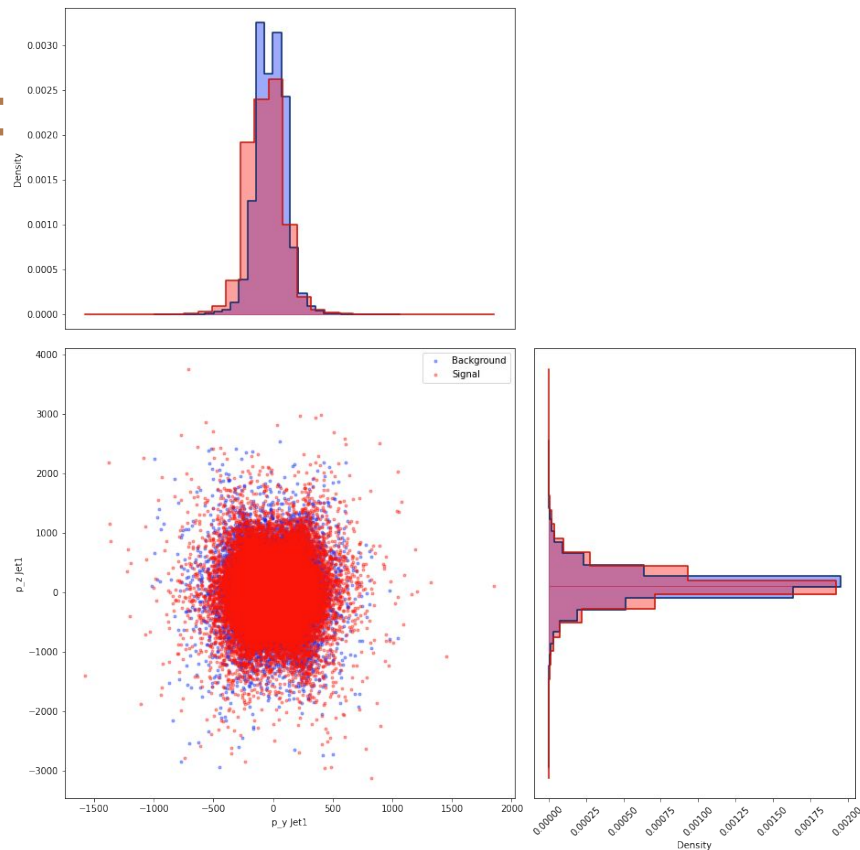
\*Belis, Vasileios, et al. "Higgs analysis with quantum classifiers." *arXiv preprint arXiv:2104.07692* (2021).

# Il problema da affrontare:

Per ogni evento di collisione (**osservazione**) sono misurate alcune quantità fisiche dei prodotti di decadimenti.

Queste quantità sono **67** e vengono analizzate congiuntamente, con un'**analisi multivariata**\*.

\*Informazioni più dettagliate:  
[https://qml-hep.github.io/qml\\_web/data/](https://qml-hep.github.io/qml_web/data/)



# Pre-processazione dei dati

**Riduzione della dimensionalità del dataset da 67 a 16 (o 8) features**, usando 3 tecniche:

- a) **AUC**: in base del loro potere di separazione.
- b) **Principal Component Analysis**: diagonalizzazione della matrice di correlazione e proiezione sulla nuova base
- c) **Nystroem**: variante della kernel PCA che massimizza la differenza delle distribuzioni di fondo e segnale.

Obiettivo: eliminare features ridondanti e ridurre la complessità del problema

# Tecnica di analisi: Support Vector Machines

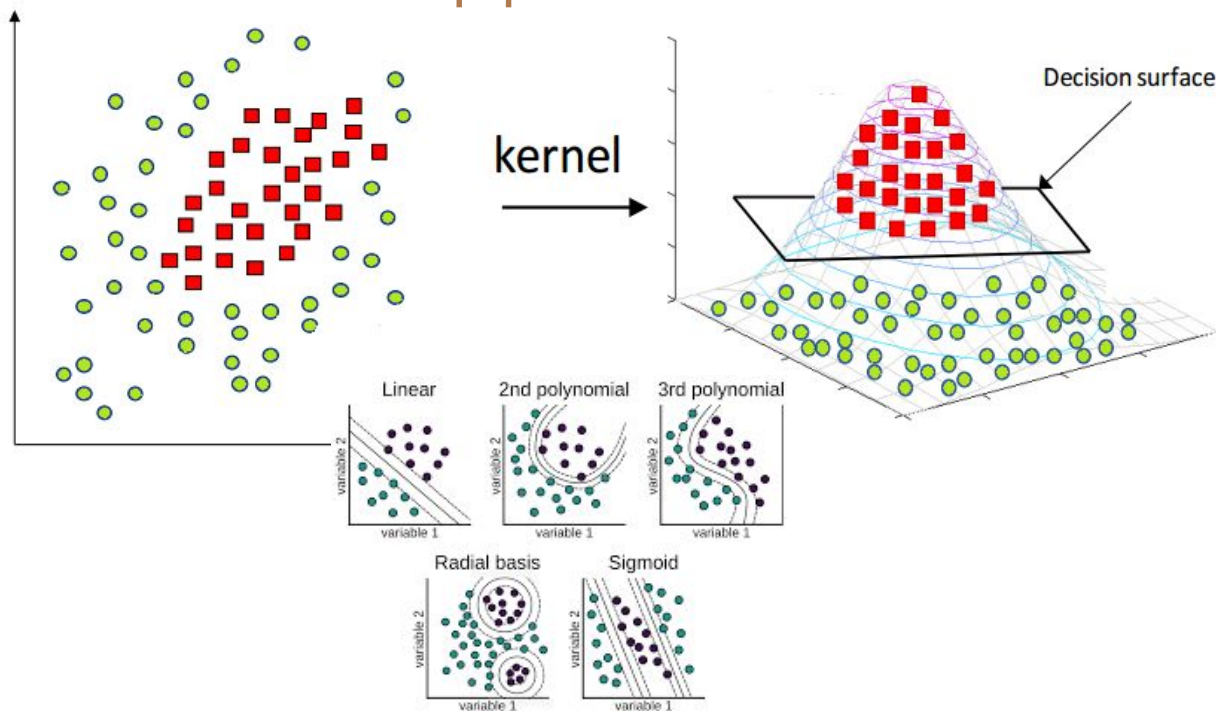
## Introduction to SVM



Per il problema di **classificazione binaria** si sceglie di usare una generalizzazione delle **SVM**.

- Utilizzo di un iperpiano che divide in 2 lo spazio delle variabili descrittive (**features**).
- A ogni osservazione (punto) viene assegnata un'etichetta (**label**) in base alla zona dello spazio in cui si trova rispetto all'iperpiano.
- Le classi devono essere linearmente separabili.

# Tecnica di analisi: Kernel Support Vector Machines



Le **Kernel SVM** estendono l'algoritmo di **SVM** a casi in cui le classi non sono linearmente separabili.

I dati sono mappati attraverso una **feature map  $\phi$**  in uno spazio di dimensione più alta, dove è possibile dividerli linearmente.

Lo spazio in cui vengono mappate le features può essere uno **spazio di Hilbert**.

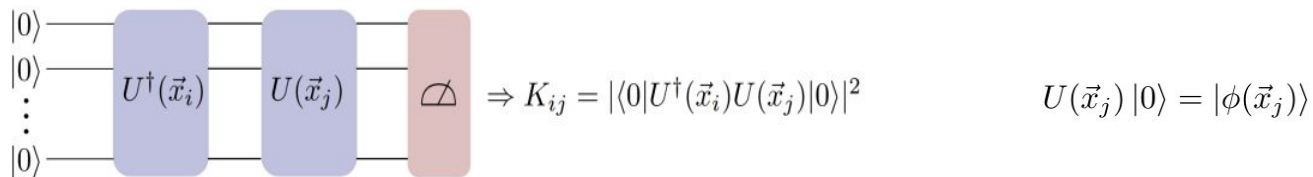
# Tecnica di analisi: Quantum Kernel

Alla **feature map**  $\phi$  è possibile associare un Kernel:  $K(x,y)$  con  $x$  e  $y$  vettori dai dati  $d$ -dimensionali che soddisfa la proprietà:

$$K(\vec{x}, \vec{y}) = | \langle \phi(\vec{x}) | \phi(\vec{y}) \rangle |^2$$

In generale non è necessaria una conoscenza esplicita di  $\phi$  (vedi **rbf kernel**).

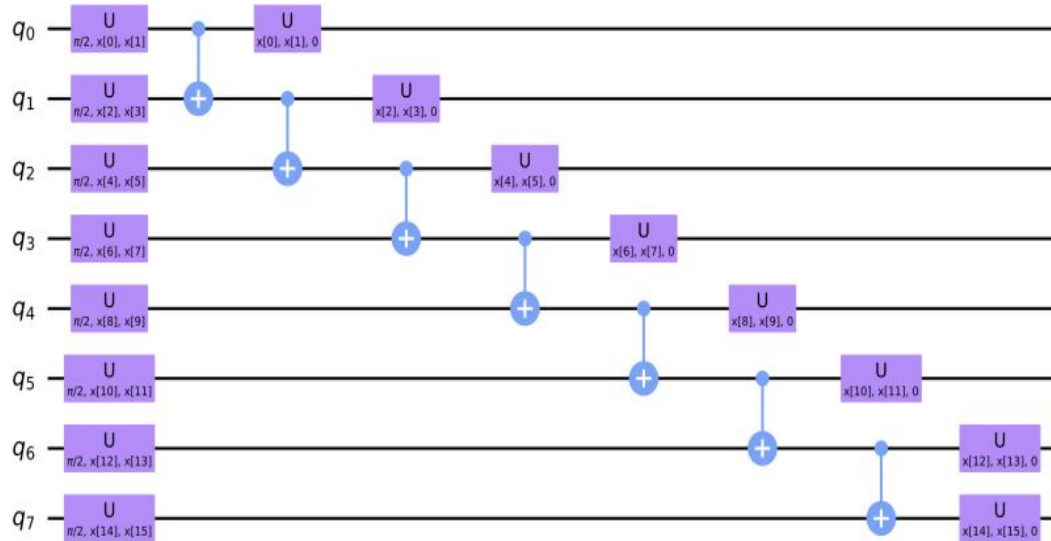
Se  $\Phi : \mathbb{R}^d \rightarrow \mathcal{H}$  spazio di Hilbert, il kernel si può calcolare con l'implementazione di un circuito quantistico, dove le feature maps sono circuiti quantistici parametrizzati sui valori del vettore  $x$ .



L'approccio computazionale è **ibrido**, perché nell'algoritmo SVM si integra un componente valutato da un computer quantistico.



# Quantum Feature maps: U2Reuploading



[1] Belis, Vasilis & González-Castillo, Samuel & Reissel, Christina & Vallecorsa, Sofia & Combarro, Elias & Dissertori, Günther & Reiter, Florentin. (2021). Higgs analysis with quantum classifiers. EPJ Web of Conferences. 251. 03070. 10.1051/epjconf/202125103070.

I 16 valori del vettore  $x$  parametrizzano dei gate  $U$  implementando una rotazione nello spazio  $\mathbb{C}^2$

L'obiettivo è codificare  $x$  nei coefficienti della sovrapposizione di più stati.

In principio quindi può codificare un vettore appartenente a  $\mathbb{R}^d$  utilizzando  $\log_2(d)$  qubit.

Vengono utilizzati  $2 \cdot \log_2(d)$  qubit per codificare  $d$  features, perchè viene generato un circuito meno profondo. (ref. [1])

Il codice che genera il circuito della mappa è tratto dalla seguente repository

[https://github.com/QML-HEP/ae\\_qml](https://github.com/QML-HEP/ae_qml)

# Quantum Feature maps:

## Pauli Feature Map

Il circuito necessita di **1 qubit per ogni feature** di cui fare l'encoding.

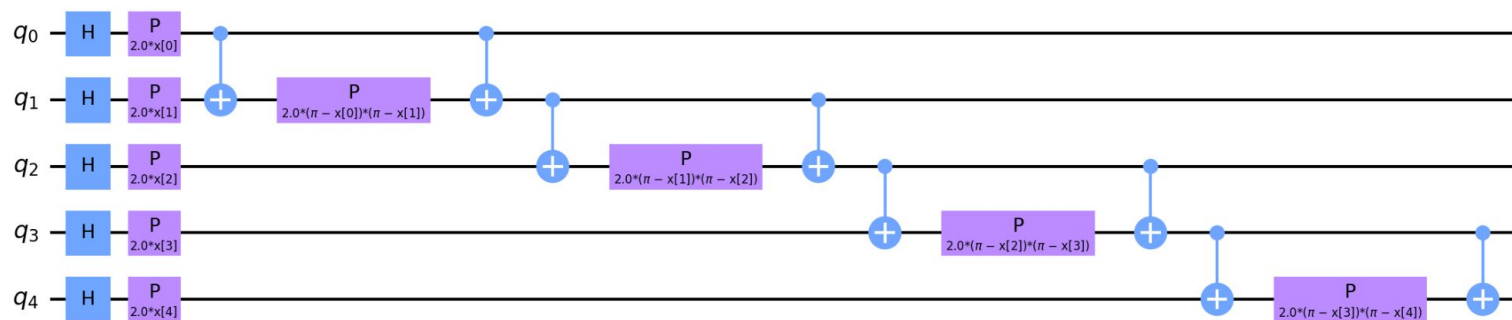
Queste feature maps implementano l'encoding di un vettore  $x$  in  $\mathbf{R}^n$  attraverso la costruzione del seguente circuito.

$$U_{\Phi(\vec{x})} = \exp \left( i \sum_{S \subseteq [n]} \phi_S(\vec{x}) \prod_{i \in S} P_i \right) \quad \phi_S(\vec{x}) = \begin{cases} x_0 & \text{if } k = 1 \\ \prod_{j \in S} (\pi - x_j) & \text{otherwise} \end{cases}$$

Dove  $\mathbf{S}$  è una combinazione di lunghezza al massimo  $k = 1, \dots, n$  e  $P_i$  è una matrice di pauli  $\{I, X, Y, Z\}$  che agisce su  $q_i$ .

$S$  determina l'entanglement tra diversi qubit, mentre la lunghezza massima  $k$  delle combinazioni determina lo scaling del numero di gate di singolo qubit necessari per l'implementazione del circuito.

# Quantum Feature maps: ZZFeatureMap

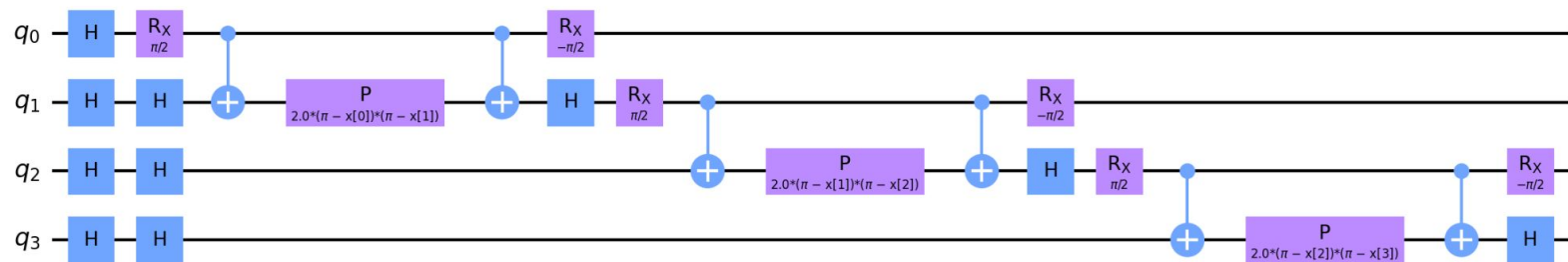


- Combinazioni di features di lunghezza  $k = 2$
- Abbiamo usato un **entanglement lineare**, preservando cioè solo le combinazioni del tipo

$$\{ (0), (1), \dots, (n), (0,1), (1,2), \dots, (n-1,n) \}$$

La scelta è dovuta al costo in termini di tempi di CPU richiesti dal simulatore offerto da Qiskit.

# Quantum Feature maps: XYFeatureMap



- Combinazioni di features di lunghezza  $k = 2$
- **entanglement lineare**

# Training, test e metriche di confronto

I dati utilizzati dall'algoritmo ibrido sono stati ridotti: **500 osservazioni per il training e 500 per il test.**

La decisione è dovuta al costo computazionale della valutazione del kernel quantistico. Per training e test dei modelli servono  **$\Theta(n^2)$  valutazioni** del Kernel.

Riassumendo:

- **8 o 16 features** estratte con AUC, NYS, PCA
- un kernel classico, tre kernel quantistici

Le metriche di confronto per valutare l'efficacia dei modelli sono state:

1. il punteggio **ROC-AUC**
2. le loro **Matrici di Confusione** sui dataset di **Training** e **Test**, non presente negli articoli consultati.

# Risultati:

## 8 features 3 runs

Model	T train [s]		T test [s]		AUC train		AUC test	
-----	-----		-----		-----		-----	
auc+rbf	0.06 +- 0.01	0.01	0.01 +- 0.00	0.00	0.71 +- 0.01	0.01	0.64 +- 0.01	0.01
auc+u2	5.13 +- 0.17	0.17	5.07 +- 0.02	0.02	0.66 +- 0.02	0.02	0.64 +- 0.00	0.00
auc+xy	13.23 +- 0.21	0.21	13.67 +- 0.31	0.31	0.80 +- 0.01	0.01	0.64 +- 0.01	0.01
auc+zz	14.46 +- 0.43	0.43	14.19 +- 0.27	0.27	0.79 +- 0.01	0.01	0.63 +- 0.00	0.00
pca+rbf	0.11 +- 0.00	0.00	0.01 +- 0.00	0.00	0.77 +- 0.00	0.00	0.58 +- 0.01	0.01
pca+u2	5.52 +- 0.58	0.58	5.09 +- 0.04	0.04	0.78 +- 0.00	0.00	0.58 +- 0.02	0.02
pca+xy	13.30 +- 0.03	0.03	13.09 +- 0.04	0.04	0.66 +- 0.44	0.44	0.52 +- 0.08	0.08
pca+zz	14.56 +- 0.48	0.48	14.37 +- 0.26	0.26	0.67 +- 0.47	0.47	0.53 +- 0.06	0.06
nys+rbf	0.11 +- 0.00	0.00	0.01 +- 0.00	0.00	0.73 +- 0.01	0.01	0.58 +- 0.02	0.02
nys+u2	4.97 +- 0.04	0.04	5.02 +- 0.11	0.11	0.56 +- 0.11	0.11	0.54 +- 0.08	0.08
nys+xy	13.49 +- 0.49	0.49	13.47 +- 0.35	0.35	0.63 +- 0.04	0.04	0.59 +- 0.02	0.02
nys+zz	14.28 +- 0.12	0.12	14.04 +- 0.05	0.05	0.62 +- 0.03	0.03	0.58 +- 0.02	0.02

- I migliori risultati vengono ottenuti con la selezione delle variabili tramite AUC
- I modelli hanno le medesime prestazioni
- Il tempo di calcolo delle mappe **xy** e **zz** sono di un fattore 3 maggiori di quello della mappa **u2**

# Risultati:

## 16 features 3 runs

Model	T train [s]		T test [s]		AUC train		AUC test	
-----	-----		-----		-----		-----	
auc+rbf	0.07 +- 0.00		0.01 +- 0.00		0.68 +- 0.01		0.62 +- 0.04	
auc+u2	9.72 +- 2.03		9.60 +- 1.78		0.66 +- 0.01		0.61 +- 0.04	
auc+xy	120.80 +- 0.45		119.51 +- 0.79		0.98 +- 0.00		0.59 +- 0.01	
auc+zz	129.62 +- 20.72		128.30 +- 19.37		0.98 +- 0.00		0.59 +- 0.01	
pca+rbf	0.12 +- 0.00		0.01 +- 0.00		0.87 +- 0.01		0.60 +- 0.02	
pca+u2	8.32 +- 0.19		8.28 +- 0.11		0.66 +- 0.46		0.53 +- 0.08	
pca+xy	119.46 +- 0.58		119.47 +- 0.45		1.00 +- 0.00		0.47 +- 0.02	
pca+zz	112.05 +- 2.44		112.77 +- 1.26		1.00 +- 0.00		0.47 +- 0.01	
nys+rbf	0.11 +- 0.00		0.01 +- 0.00		0.79 +- 0.04		0.58 +- 0.00	
nys+u2	8.29 +- 0.15		8.24 +- 0.04		0.47 +- 0.11		0.47 +- 0.05	
nys+xy	121.23 +- 2.78		124.39 +- 4.97		0.66 +- 0.03		0.59 +- 0.01	
nys+zz	113.45 +- 1.65		113.23 +- 0.71		0.57 +- 0.13		0.53 +- 0.08	

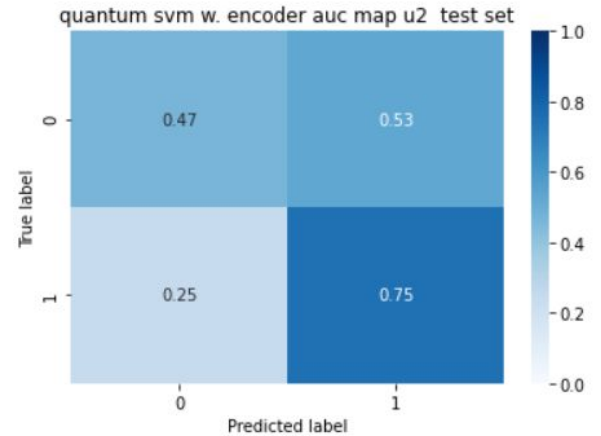
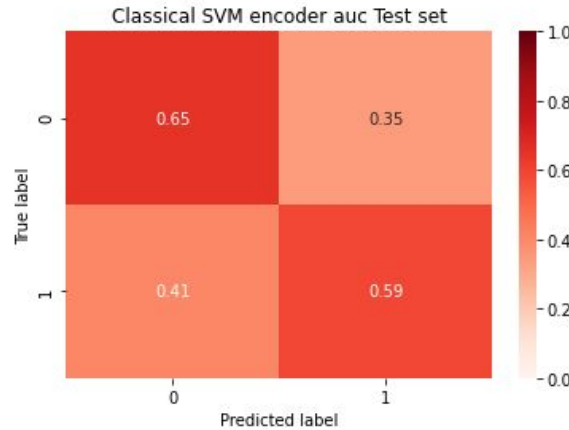
- I migliori risultati vengono ottenuti con la selezione delle variabili tramite AUC
- I modelli migliori sono **auc+rbf**, **auc+u2**
- Il divario tra tempi di calcolo dei modelli si allarga.
- I tempi di xy e zz aumentano di un fattore 10 rispetto all'esperimento su 8 features

# Risultati:

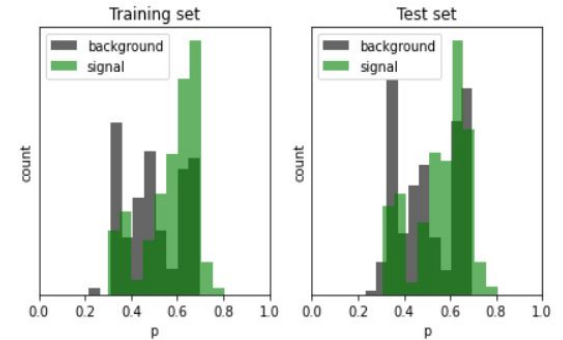
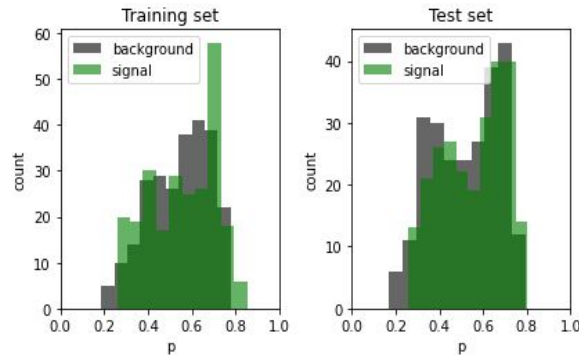
Confronto con 16 features:

-**AUC RBF** classico

-**AUC U2** quantistico



**Confusion matrix** sul set di test,  
**distribuzioni** delle probabilità  
assegnate





# Considerazioni:

- Avendo fatto più run dello stesso modello su dati estratti randomicamente si conclude che:
  - I risultati classici rimangono simili a quelli quantistici
  - I risultati dello stesso modello rimangono simili tra le run
- I risultati rimangono simili utilizzando 8 qubit o 16 nei circuiti quantistici
- Il tempo di calcolo della cpu è di circa 4 ordini di grandezza maggiore (110 s xy contro 0.05 s rbf ) tra algoritmo quantistico e classico

# Bibliografia:

- <https://qiskit.org/documentation/machine-learning/tutorials/index.html>
- Belis, Vasilis & González-Castillo, Samuel & Reissel, Christina & Vallecorsa, Sofia & Combarro, Elias & Dissertori, Günther & Reiter, Florentin. (2021). Higgs analysis with quantum classifiers. EPJ Web of Conferences. 251. 03070. 10.1051/epjconf/202125103070. <https://arxiv.org/abs/2104.07692>
- Wu, Sau Lan, et al. "Application of Quantum Machine Learning using the Quantum Kernel Algorithm on High Energy Physics Analysis at the LHC." arXiv preprint <https://arxiv.org/abs/2104.05059> (2021).