

# Challenge 2 ML

Matteo Vicenzino

April 2025

Si vuole studiare l'effetto dei metodi di Kernel sugli algoritmi di Ridge Regression e Principal Component Analysis.

## Kernel Ridge Regression

Per questa parte della challenge si utilizza un dataset creato dalla seguente funzione, che produce i dati raffigurati in Figura 1.

$$y = (X + 4)(X + 1)(\cos(X) - 1)(X - 3) + \epsilon, \text{ con } \epsilon \in \mathcal{N}(0, 1)$$

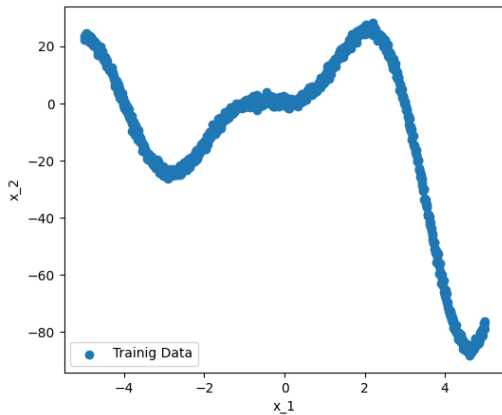


Figure 1: Dataset di train

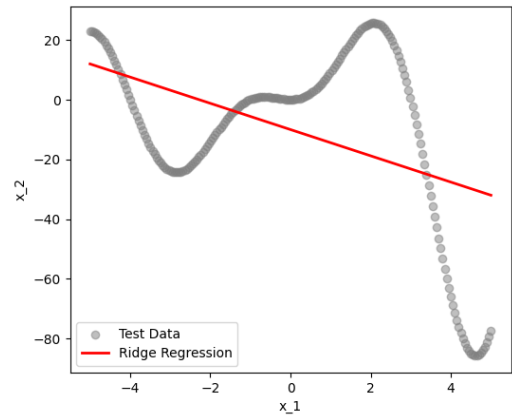
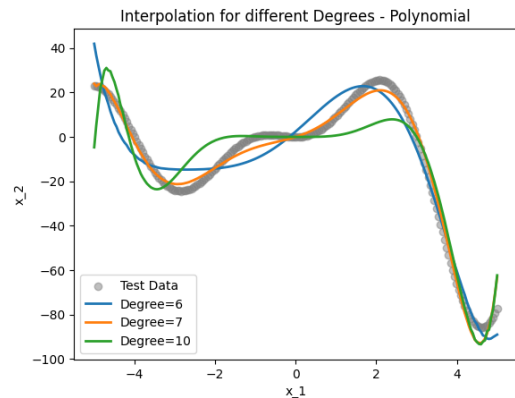
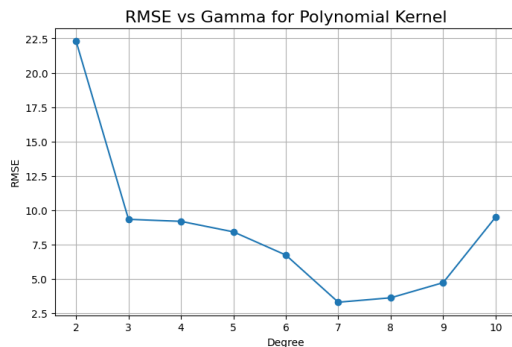


Figure 2: Ridge Regression

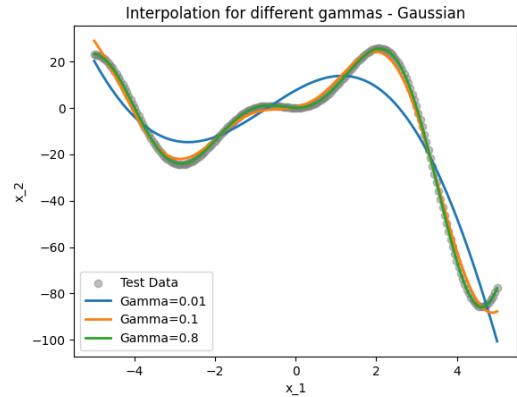
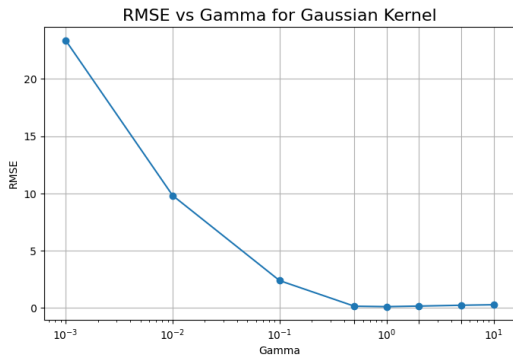
Il dataset di train contiene 1000 punti, mentre quello di test 200 e ha  $\epsilon = 0$ .

Si inizia l'analisi con una Ridge Regression semplice, con parametro  $\alpha = 0.1$ . Si nota che i dati non sono ben interpolati da una funzione lineare, e l'errore RMSE è molto elevato (26.78) si veda Figura 2.

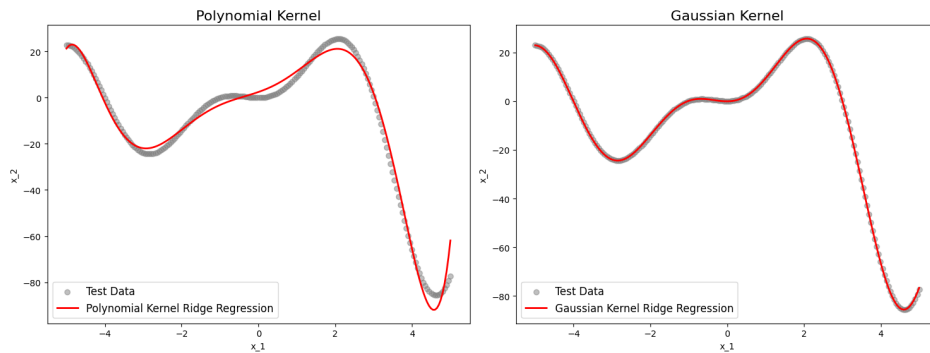
Aggiungendo una funzione di Kernel Polinomiale i risultati migliorano parzialmente: dopo aver studiato il modello con vari parametri di *degree* si decide di sviluppare un modello con i seguenti parametri:  $\alpha = 0.1$ ,  $degree = 7$ ,  $\gamma = 1$  dopo aver applicato l'algoritmo di Grid Search.



Procedendo in maniera analoga per il Kernel Gaussiano e valutando la radice dell'errore quadratico medio si osserva un'ottima interpolazione dei dati di test con parametri  $\alpha = 0.1$ ,  $\gamma = 0.8$  che portano ad un errore RMSE di appena 0.1207 nel dataset di test.



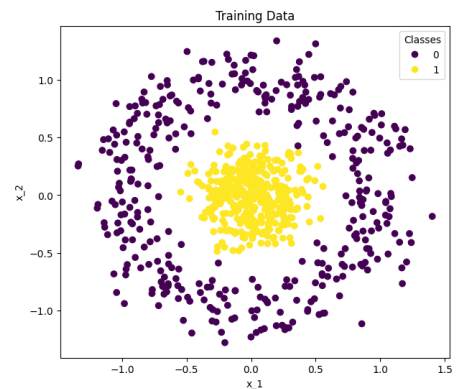
Si conclude l'analisi della Ridge Regression mostrando come il Kernel Gaussiano riesca ad interpolare perfettamente i dati del test set anche grazie alla sua capacità di modellare relazioni non lineari tra le variabili, adattandosi in modo flessibile alla struttura dei dati e garantendo una buona generalizzazione senza incorrere in overfitting.



## Kernel PCA

Il dataset per questo punto è stato generato attraverso la funzione built-in *make\_circles* di sklearn, con 750 punti di train e 250 di test, dove i punti di classe 0 sono disposti in un cluster centrato in (0,0), mentre quelli di classe 1 sono disposti attorno come in figura. Applicando una riduzione di dimensionalità PCA si nota subito che i risultati ottenuti sono insufficienti; nel dataset di train infatti si ottiene un'accuratezza di 0.6320 nella classificazione (figura 3).

Dopo aver valutato diversi parametri di  $\gamma$ , si sceglie quindi di applicare un kernel di tipo Gaussiano con parametro  $\gamma = 3$ , in quanto particolarmente adatto a catturare la struttura concentrica del dataset. Si ottiene un'accuratezza del 0.9920, e dal grafico si nota come i punti di test siano separabili linearmente in modo chiaro (figura 4).



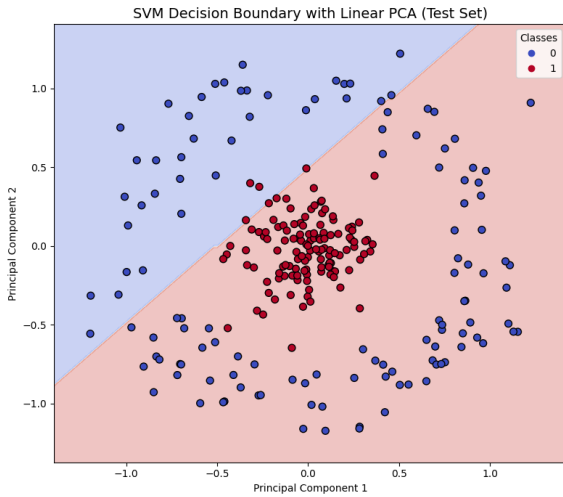


Figure 3: PCA e SVM applicato al dataset. Accuratezza bassa del 63%

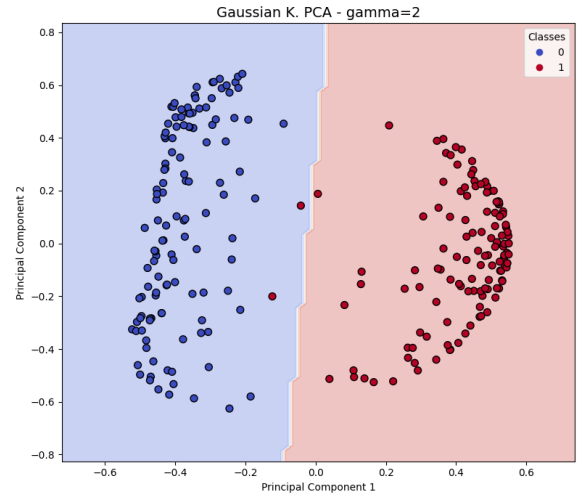


Figure 4: Aggiunta di una funzione di kernel Gaussiano. i punti sono ora divisibili linearmente. Accuratezza del 99%

### Kernel PCA con dataset make\_classification

Per la parte finale della challenge si vuole valutare un dataset sviluppato con la funzione `make_classification`, e con i seguenti parametri scelti: `n_samples=1000`, `n_features=10`, `n_classes=2`, `n_clusters_per_class=1`, `class_sep=1.2`, `flip_y=0.2`, `random_state=0`. Con questi parametri si vede che l'accuratezza della classificazione SVM dopo aver applicato la riduzione a 2 componenti PCA è di 0.8520. Si prova ad applicare altri metodi di kernel (gaussiano, polinomiale e sigmoide) con diversi parametri e si ottiene i seguenti risultati:

	Gamma	Degree	Accuracy
<b>Lineare</b>	-	-	85.2%
<b>Kernel Gaussiano</b>	0.001	-	85.6%
<b>Kernel Polinomiale</b>	0.01	3	85.6%
<b>Kernel Sigmoidale</b>	2	-	86.4%

Il miglioramento aggiungendo una funzione di kernel è minimo, con il kernel sigmoide che ottiene una performance di 86.4%. In generale possiamo affermare che la funzione di kernel migliore è fortemente dipendente dalla forma del dataset, e non sempre una scelta di kernel complessa porta a performance significativamente superiori. Spesso la scelta del kernel dipende dalla natura intrinseca dei dati e va verificata sperimentalmente, perché aspetti quali distribuzione, scala e non-linearità influenzano notevolmente l'efficacia della trasformazione.

