Data Analytics

Francesco Saverio Beccafichi - 1052139
francesco.beccafichi@studio.unibo.it

Laurea Magistrale in Informatica 2022/2023

Introduzione

Obiettivo del progetto: predirre il voto medio di un film a partire dalle sue caratteristiche.

La **realizzazione** prevede l'implementazione di tutte le fasi della pipeline analitica:

- Data Acquisition
- Data Visualization
- Data pre-processing
- Modeling
- Performance Evaluation

Data Acquisition

Dataset: il dataset utilizzato e **Movielens 25m**, un sistema di raccomandazione che contiene oltre 60.000 film.

Il dataset è suddiviso in sei file *csv*, per l'utilizzo nel progetto ne sono stati utilizzati solamente quattro:

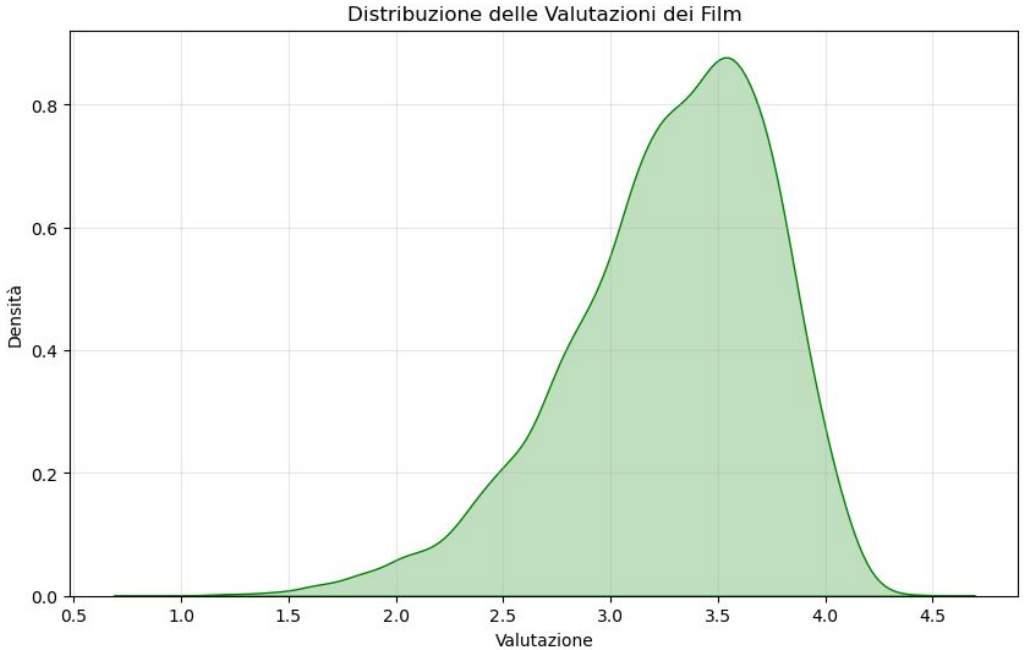
1. ***movies.csv***: contiene tutte le informazioni che riguardano il film.
2. ***genome-tag.csv***: contiene i tag utilizzati per descrivere il film.
3. ***genome-score.csv***: contiene la rilevanza del film rispetto ad un tag.
4. ***ratings.csv***: contiene le valutazioni degli utenti per i film.

Data Acquisition(2)

Il dataset finale è stato salvato in un file *csv* da utilizzare per le analisi

	movieId	title	007	007 (series)	18th century	1920s	1930s	1950s	1960s	1970s	...
0	1	Toy Story (1995)	0.02875	0.02375	0.06250	0.07575	0.14075	0.14675	0.06350	0.20375	...
1	2	Jumanji (1995)	0.04125	0.04050	0.06275	0.08275	0.09100	0.06125	0.06925	0.09600	...
2	3	Grumpier Old Men (1995)	0.04675	0.05550	0.02925	0.08700	0.04750	0.04775	0.04600	0.14275	...
3	4	Waiting to Exhale (1995)	0.03425	0.03800	0.04050	0.03100	0.06500	0.03575	0.02900	0.08650	...
4	5	Father of the Bride Part II (1995)	0.04300	0.05325	0.03800	0.04100	0.05400	0.06725	0.02775	0.07650	...

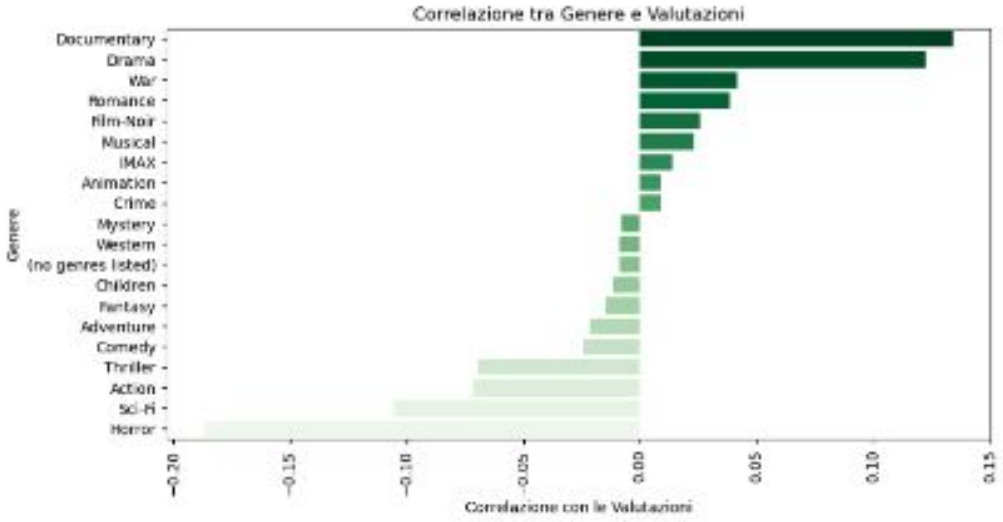
Data Visualization



Il **voto medio** per ogni film è rappresentato con un valore che varia tra 0 e 5.

La **maggiore densità ottenuta** emerge intorno al voto **3,5**.

Data Visualization (2)



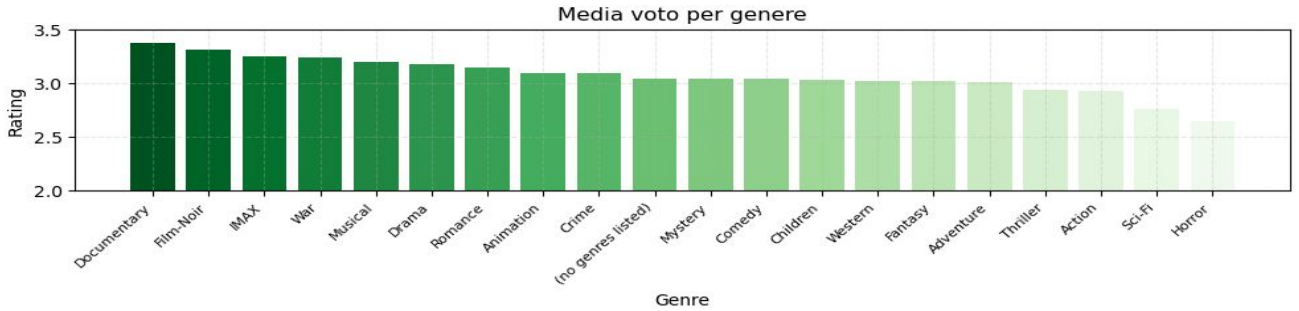
Correlazione **positiva**:

- Documentary
- Drama

Correlazione **negativa**:

- Horror
- Sci-Fi

Data Visualization (3)

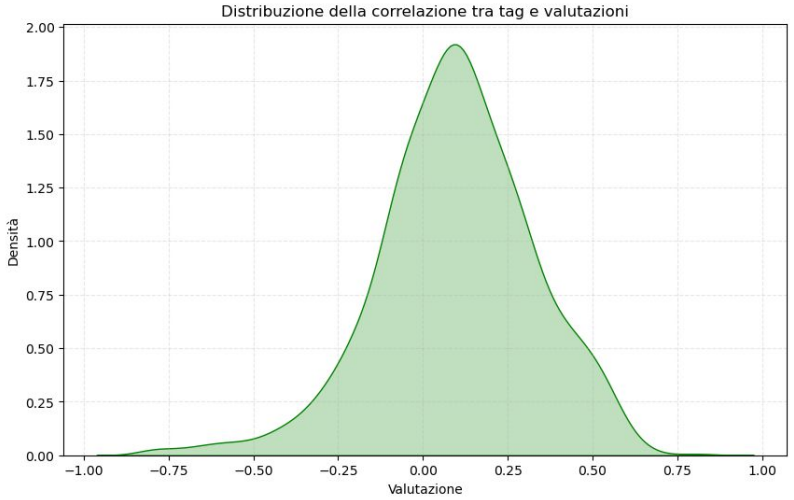


Voto medio per ogni genere di film:

La media più alta è ottenuta dai film di genere **Documentary.**

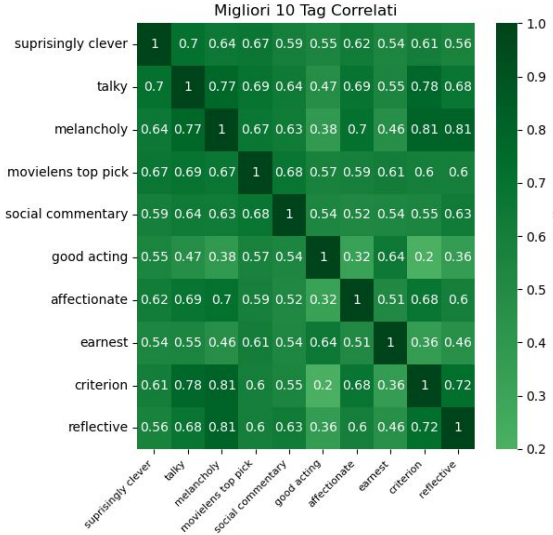
La media più bassa è ottenuta dai film di genere **Horror.**

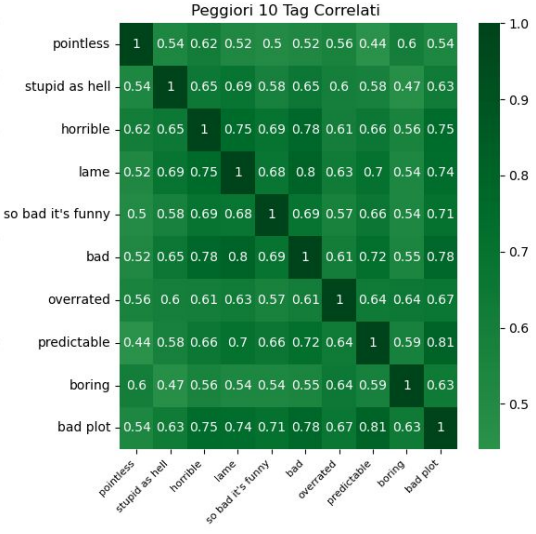
Data Visualization (4)



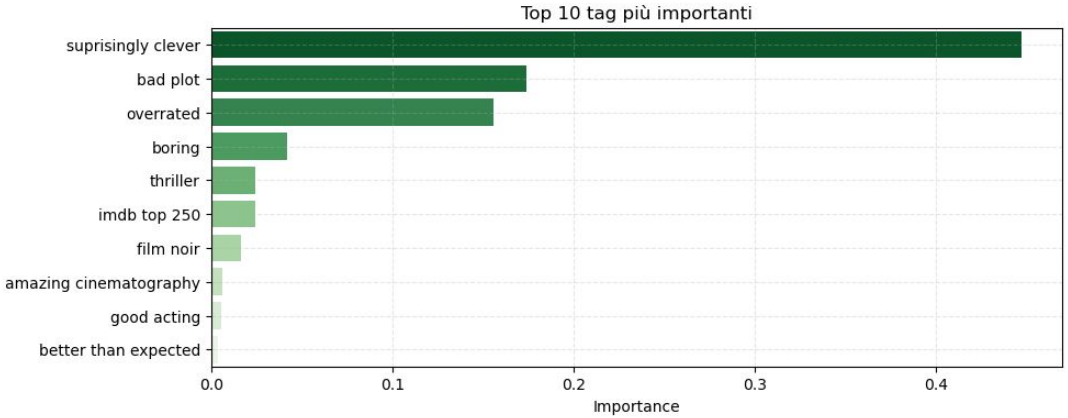
Nella distribuzione della correlazione si nota una zona di **maggiore densità** intorno allo **0.25**

Nelle matrici di correlazione sono elencati quali film possono **maggiormente influenzare** un utente in modo positivo o negativo.



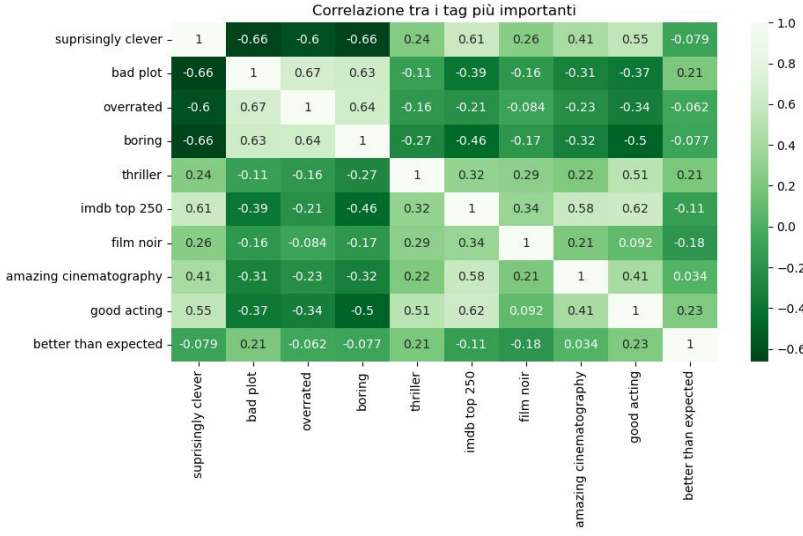


Data Visualization (5)

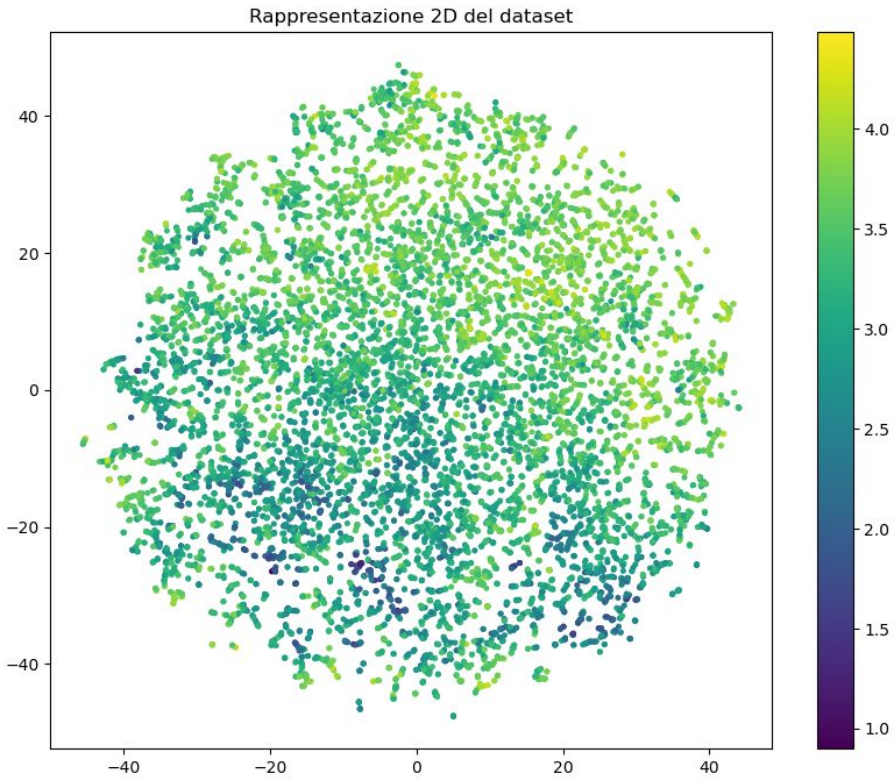


Tra i tag più influenti troviamo

- surprising clever
- bad plot
- overrated



Data Visualization (6)



Rappresentazione 2D in base al ratings dei dati attraverso t-SNE.

Si possono notare i dati che hanno una certa **similarità** tra loro.

I film vengono distribuiti gradualmente in uno spazio bi-dimensionale, separando quelli con rating basso (punti blu) e quelli con rating più alto(punti gialli)

Data Preprocessing

- Il dataset utilizzato è stato diviso in:
  - **train 70%**
  - **validation 10%**
  - **test 20%**
- **Non** sono stati trovati valori **nulli** o **duplicati**.
- **Non** vi è stato bisogno di applicare forme di **scaling**.
- E' stato fissato un seed(42) per per replicare la suddivisione del dataset
- Il test set è stato applicato a tutti i modelli per ottenere risultati coerenti

Data Preprocessing (2)

E’ stata utilizzata la **Principal Component Analysis (PCA)**.

Si punta ad avere un **nuovo sistema di coordinate** dove le componenti principali, ovvero delle variabili linearmente indipendenti precedentemente create, che andranno a formare gli assi dello spazio, in modo da proiettare i dati con **dimensioni inferiori** rispetto a quelle di partenza, ma **mantenendo** tutte l**e informazioni più significative**.

La PCA è stata applicata al trai set, con la variabile dipendente rappresentata dalla media dei voti, la percentuale di componenti mantenute è pari al 95%.

Modeling

In questa fase vengono utilizzate le seguenti **tecniche di Machine Learning:**

- Machine Learning supervisionato.
- Rete neurale
- Modello TabNet per dati tabulari

Modeling (2)

Regressioni Lineari

Sono modelli statici che mirano a trovare una funzione che meglio rappresenta la relazione tra le variabili.

L'obiettivo è di trovare i **coefficienti** che minimizzano l'errore:

- **Linear regression**.
- **Ridge regression**, alpha =[0.0001, 0.001, 0.1, 0.5, 1, 2, 3, 4, 5, 6, 10, 20].
- **Lasso regression**, alpha =[1e-1, 1e-2, 1e-3, 1e-4, 1e-5].

Modeling (3)

Random Forest Regressor

Algoritmo di apprendimento supervisionato che crea **alberi decisionali indipendenti** tra loro. Ogni albero viene allenato da un subset casuale di dati, che permetteranno di generare delle previsioni.

Gli **iperparametri** utilizzati per questo modello sono:

- **`n_estimators`**: `[10, 15, 20, 25, 30]`.
- **`criterion`**: `['squared_error', 'friedman_mse']`.

Modeling (4)

K-Nearest Neighbors

Questa regressione ha come obiettivo di trovare i **k** dati più vicini all'interno del train set.

Gli **iperparametri** utilizzati per questo modello:

- **`n_neighbors`**: `[7, 9, 11, 13, 15, 17, 19, 20]`.
- **`weights`**: `['uniform', 'distance']`.

Modeling (5)

Support Vector Regression

Questa tecnica mira ad identificare un **iperpiano** che meglio approssima i dati in uno spazio multidimensionale. L' iperpiano deve essere il più distante possibile dai punti al di fuori della **boundary line.**

Gli **iperparametri** utilizzati per questo modello:

- **`kernel`**: `['poly', 'rbf', 'sigmoid'].`
- **`epsilon`**: `[0.001, 0.01, 0.1, 1].`
- **`C`**: `[0.001, 0.01, 0.1, 1].`

Modeling (6) - Rete neurale feed-forward

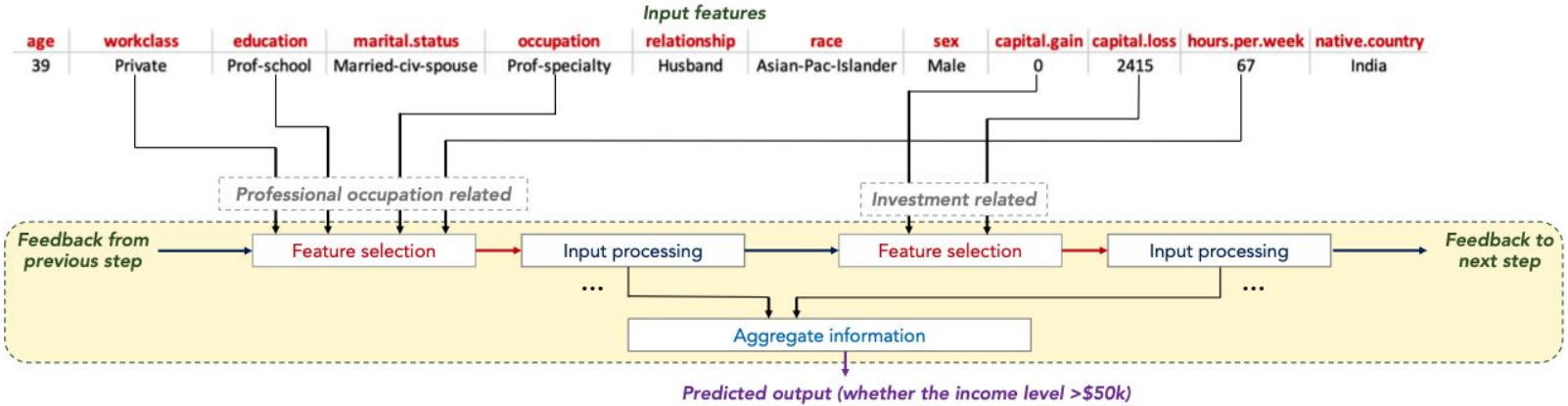
Queste reti possono essere composte da uno o più layer di neuroni, ed ogni neurone è collegato a tutti i neuroni del layer successivo. Questa struttura è molto semplice in quanto tutti i layer si muovono in una sola direzione, escludendo loop o connessioni a ritroso.

Gli **iperparametri** utilizzati per questo modello:

- **`hidden_size`**: `[64, 128, 256, 512]`.
- **`dropout_prob`**: `[0.2, 0.3]`.
- **`dept`**: `[3, 4, 5]`.
- **`batch_size`**: `[8, 16, 32]`.
- **`lr`**: `[0.001, 0.01]`.

Modeling (7) - TabNet

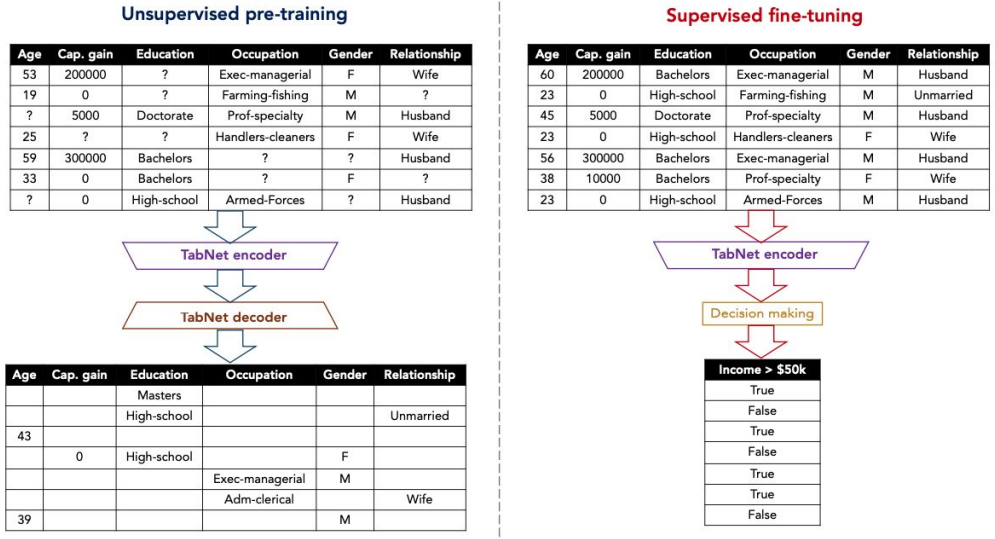
TabNet è un algoritmo di ML progettato nel 2019 da Google per l'utilizzo di dati tabulari.



Riceve in **input dati grezzi** e utilizzando l'attenzione sequenziale seleziona le **features** che vengono **utilizzate in ogni step decisionale**, ottenendo migliore interpretazione e apprendimento

Modeling (8) - TabNet

Questo modello utilizza il **pre-training non supervisionato** sui dati tabulari

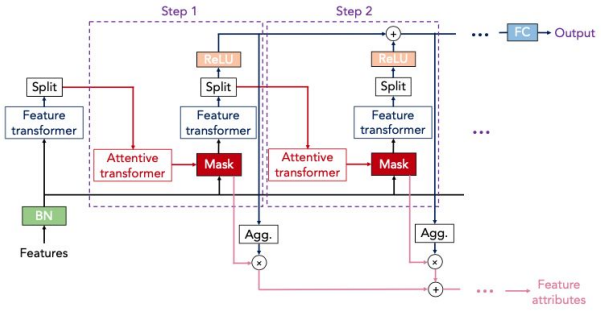


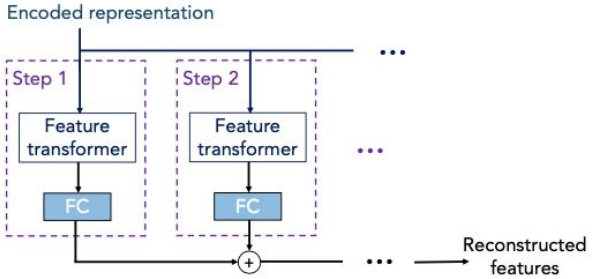
I pesi del modello vengono inizializzati utilizzando lo conoscenze apprese da **modelli pre-addestrati** su grandi dataset o su problemi simili

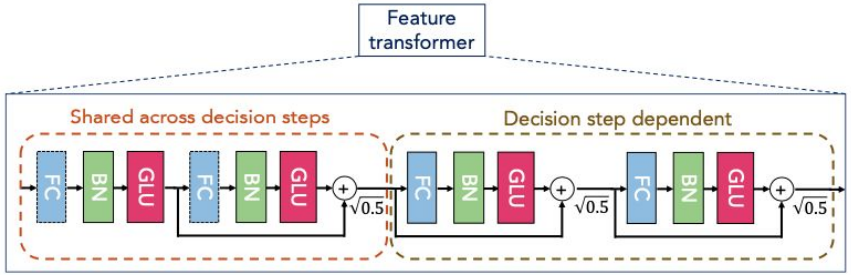
Modeling (9) - TabNet

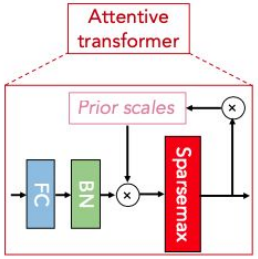
Durante il training le features vengono passate all' **encoder**, che è composto da:

- **attentive transformer:** seleziona le features più importanti da passare al mascheramento
- **mask:** impedisce al modello di utilizzare le stesse feature più volte.
- **feature transformer:** trasforma le feature della fase di mascheramento in una nuova rappresentazione









Modeling (10) - TabNet

Gli **iperparametri** utilizzati per questo modello:

- **batch_size**: [256].
- **n_epochs**: [200].
- **n_d**: [16, 32, 64].
- **n_a**: [16, 32, 64].
- **n_steps**: [3, 6, 9].
- **n_indipendent**: [2, 3].

Performance Evaluation

Le metriche utilizzate per confrontare le performance dei modelli sono:

- **Mean Squared Error:** misura l'errore quadratico medio tra le previsioni del modello e i valori osservati

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **Coefficiente di determinazione(R2):** misura quanto la varianza dei dati di output previsti dal modello si avvicina alla varianza dei dati reali.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2}$$

Performance Evaluation (2)

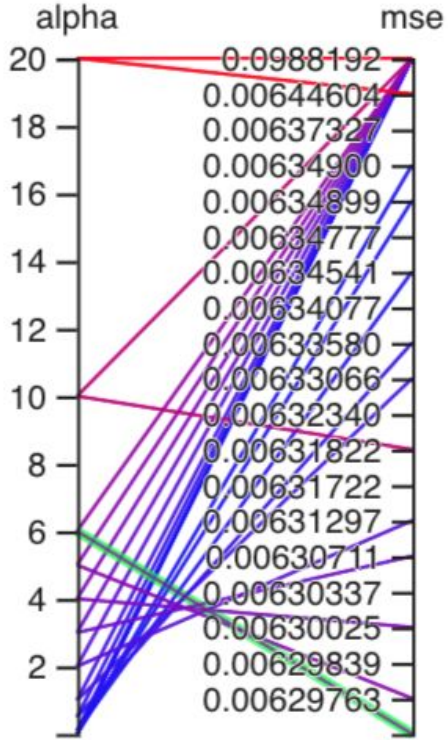
Risultati delle **linear regression**

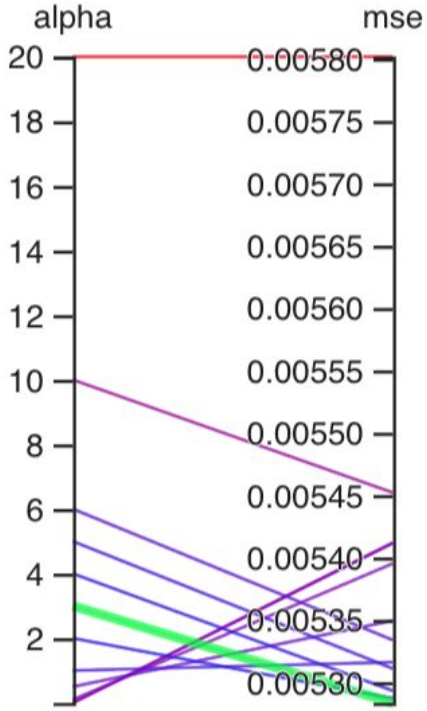
	Con PCA	Senza PCA
MSE	0.00643	0.00544
R2-score	0.97096	0.97543

Performance Evaluation (2)

Risultati delle **ridge regression**

	Con PCA	Senza PCA
MSE	0.00639	0.00531
R2-score	0.97114	0.97600
alpha	6	3

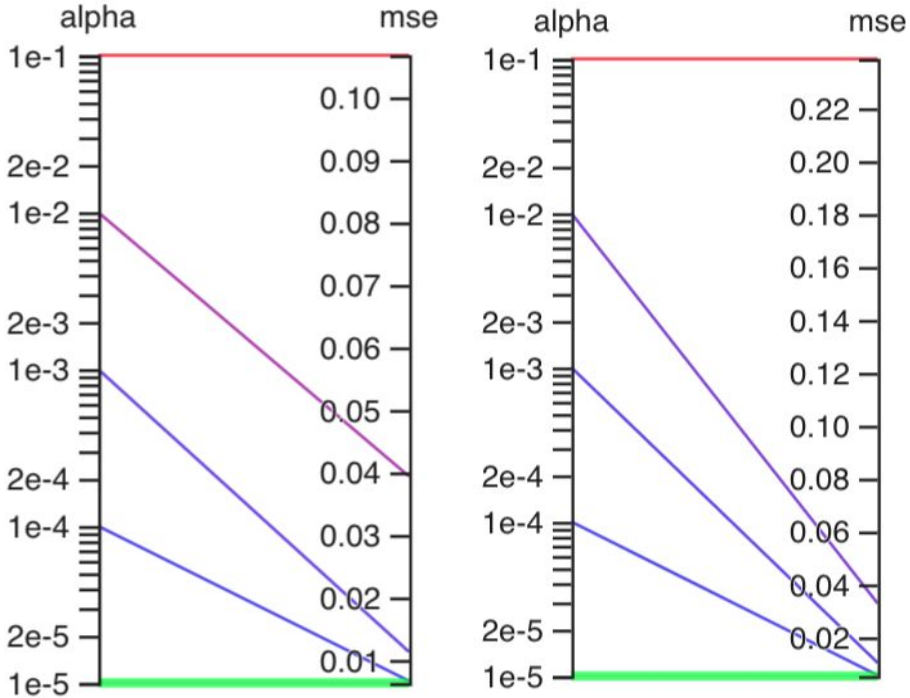




Performance Evaluation (3)

Risultati delle **lasso regression**

	Con PCA	Senza PCA
MSE	0.00641	0.00542
R2-score	0.97106	0.97553
alpha	1e-05	1e-05



Performance Evaluation (4)

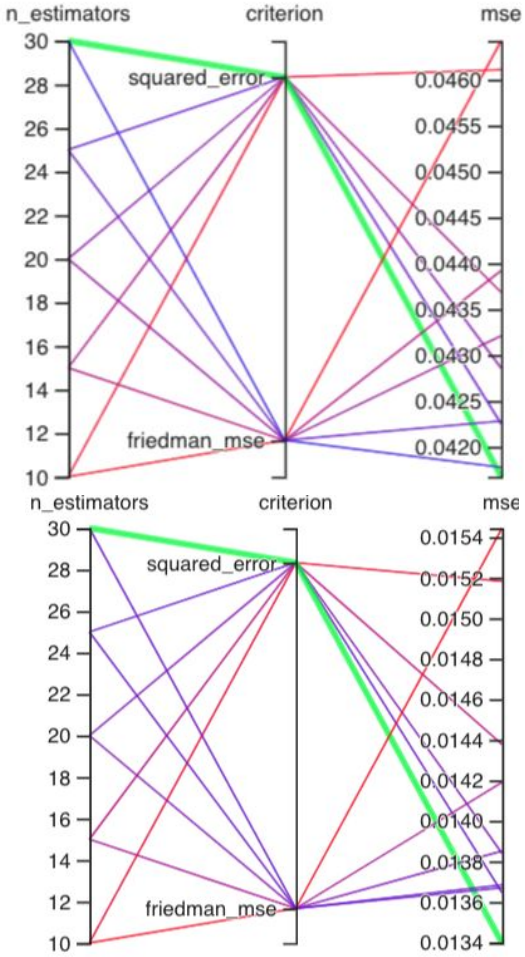
Risultati delle **lasso regression,** migliori 10 features con coefficienti a 0 e relativi voti

Features	Correlazione
007	-0.097706
1920s 3	0.379278
3d	-0.153728
aardman	0.060788
afterlife	-0.079177
alcoholism	0.206545
almodovar	0.083976
amnesia	0.062696
animation	0.024383
arms dealer	-0.073337

Performance Evaluation (5)

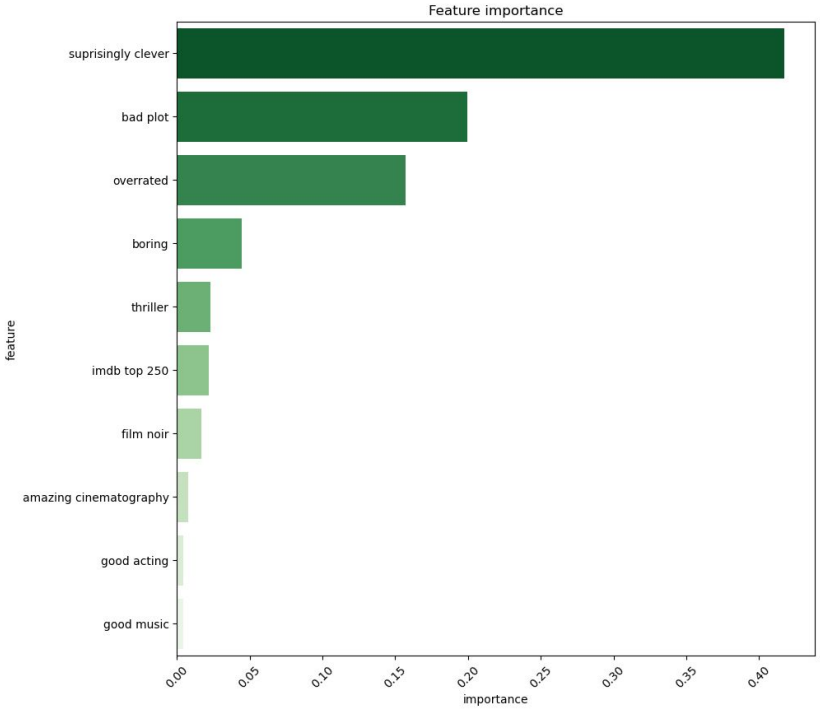
Risultati delle **Random Forest Regressor**

	Con PCA	Senza PCA
MSE	0.03801	0.01234
R2-score	0.82850	0.94431
n-estimators	30	30
criterion	squared_error	squared_error



Performance Evaluation (6)

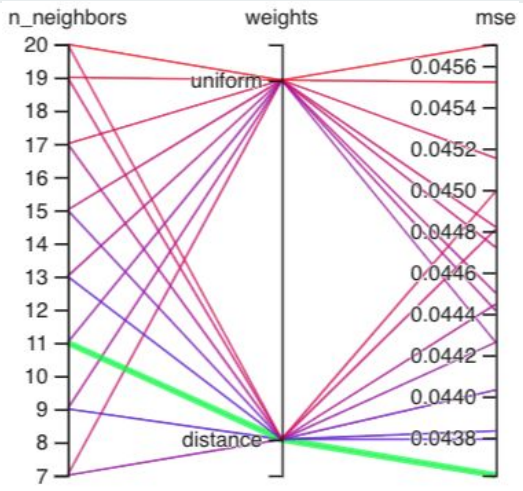
Risultati delle **random forest regressor,** prime 10 features selezionate per importanza.

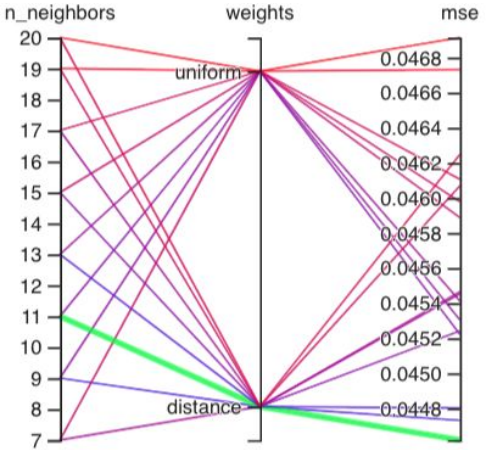


Performance Evaluation (7)

Risultati delle **KNN regression**

	Con PCA	Senza PCA
MSE	0.04015	0.04049
R2-score	0.81883	0.81729
n_neighbors	11	11
weights	distance	distance

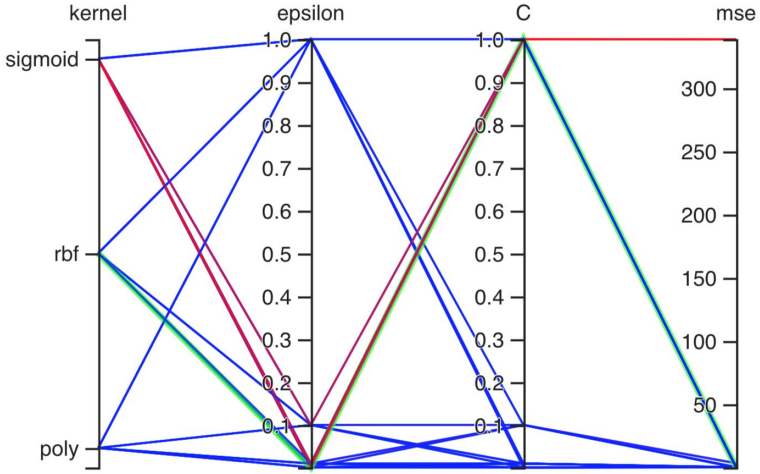


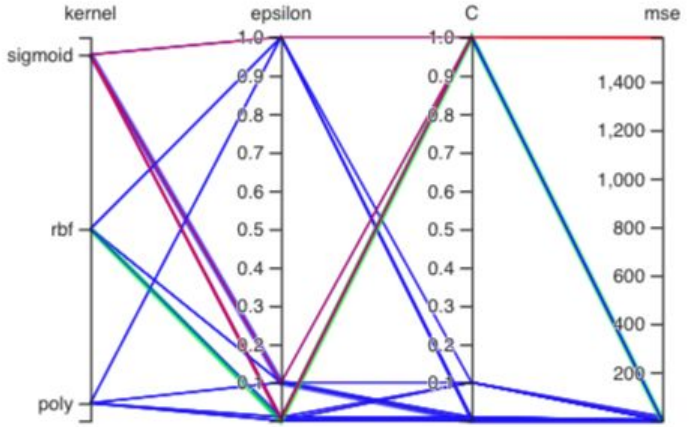


Performance Evaluation (8)

Risultati delle **SVR**

	Con PCA	Senza PCA
MSE	0.00659	0.00521
R2-score	0.97022	0.97647
kernel	rbf	rbf
epsilon	0.001	0.001
C	1	1



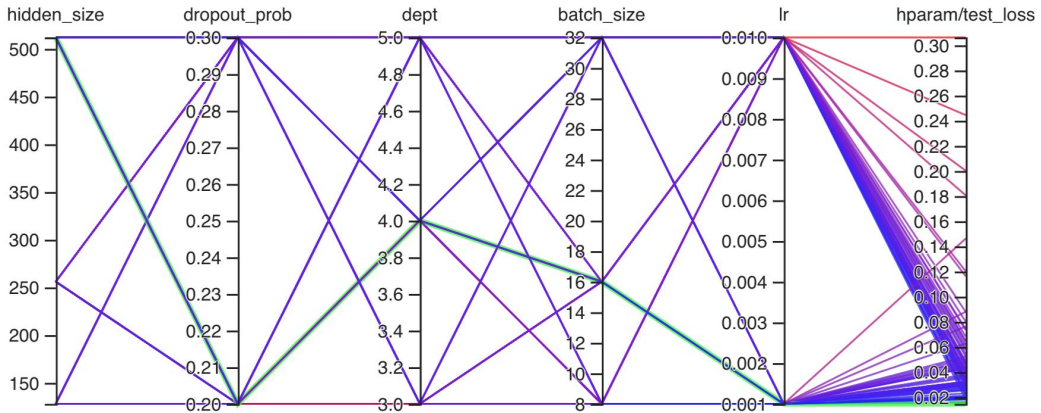


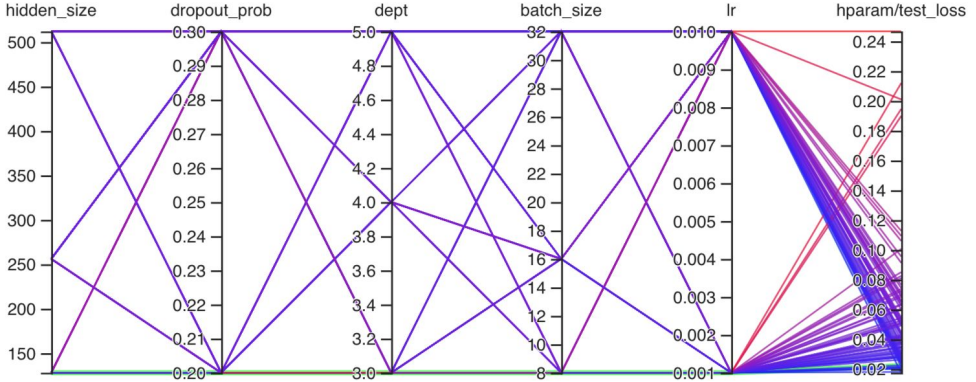
Performance Evaluation (9)

Risultati delle **rete neurale feed-forward**

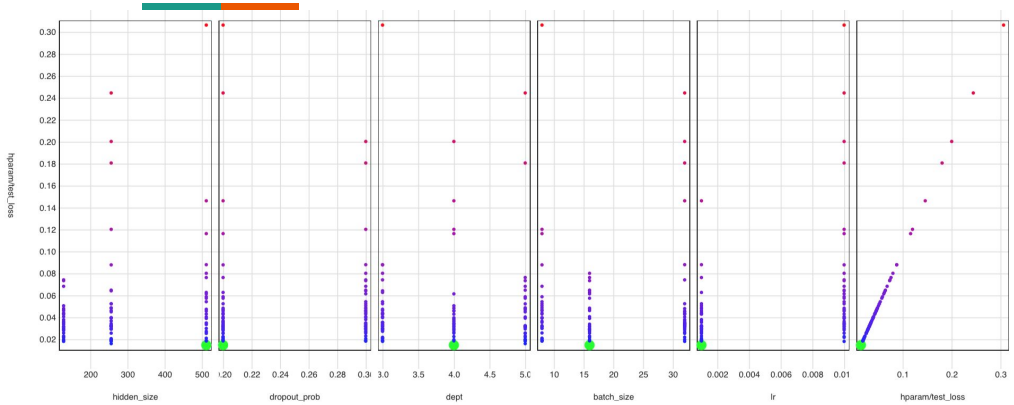
	Con PCA	Senza PCA
MSE	0.01471	0.01758
R2-score	0.93248	0.91234
hidden_size	512	128
dropout_prob	0.2	0.2
dept	4	3
batch_size	16	8
lr	0.001	0.001

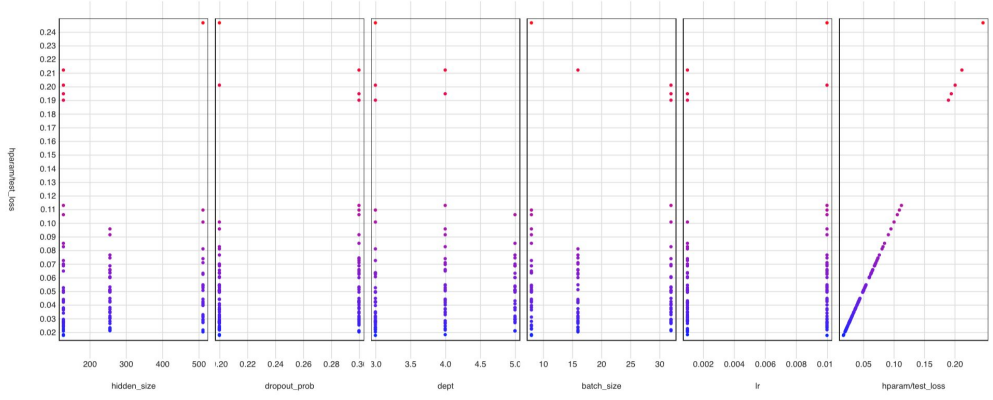
Parallel coordinates view della rete neurale feed-forward





Scatter plot matrix della rete neurale feed-forward



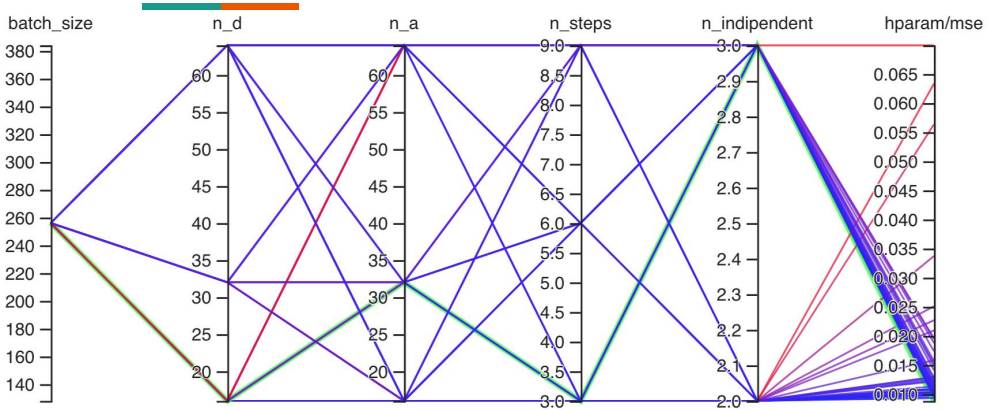


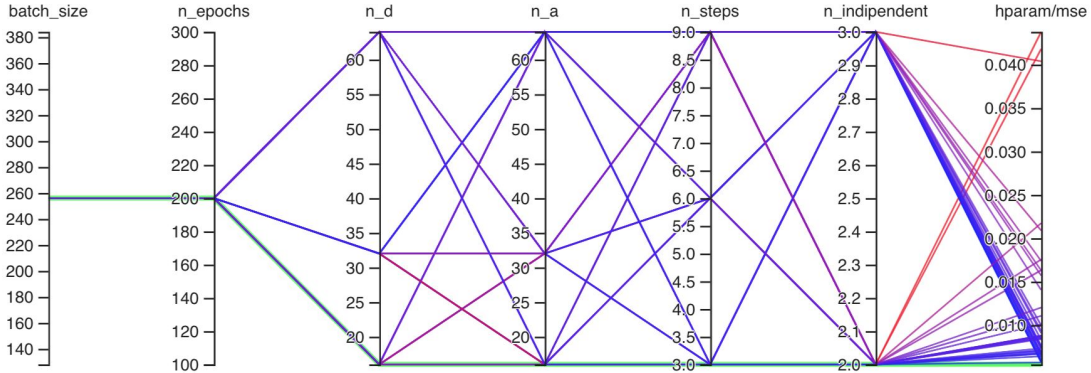
Performance Evaluation (10)

Risultati delle **TabNet**

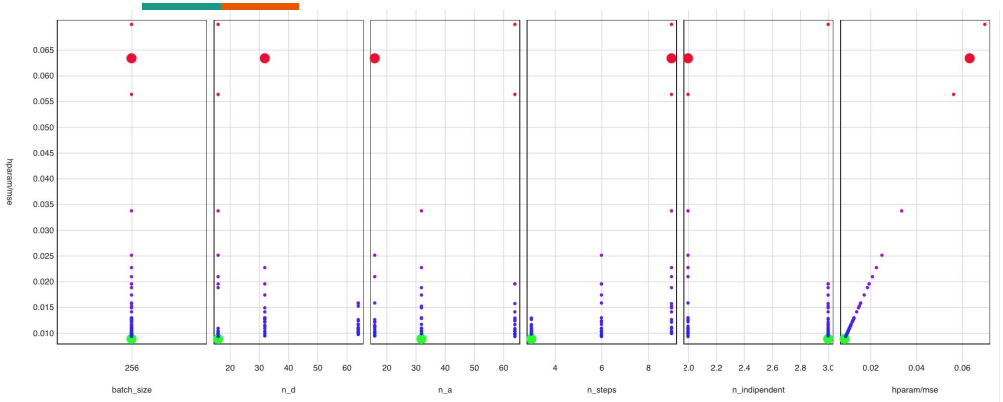
	Con PCA	Senza PCA
MSE	0.00882	0.00546
R2-score	0.96019	0.97537
batch_size	256	256
n_d	16	16
n_a	32	3
n_steps	3	3
n_indipendent	3	2

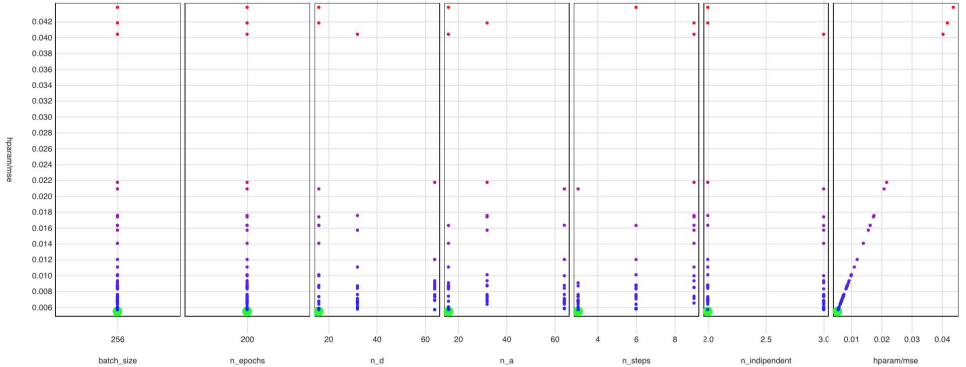
Parallel coordinates view TabNet





Scatter plot matrix TabNet





Conclusioni

Tutti i modelli hanno ottenuto **performance migliori** quando **non è stata applicata la PCA**

Ad eccezione di KNN e Random Forest, **tutti i modelli** hanno prodotto **ottime performance**.

La **rete neurale feed-forward** ha ottenuto **performance discrete**, mentre **TabNet** ha registrato **ottime performance** a discapito di lunghi tempi di training

Grazie per l’attenzione