Data Analytics

Isabella Marasco - 1040993
isabella.marasco3@studio.unibo.it

Laurea magistrale in Informatica

2022/2023

Introduzione

Lo scopo di questo porgetto è di predire il voto medio di un film a partire dalle sue caratteristiche.

La pipeline seguita è formata da diverse fasi:

- Data Acquisition
- Data Visualization
- Data Preprocessing
- Modeling
- Evaluation

Data Acquisition

Il **dataset** utilizzato è **MovieLens**, un recommendation system per i film, contenente più di 60.000 film.

I file .csv utilizzati sono:

- movies: contiene l'ID del film, il titolo con il loro anno di uscita e il genere
- genome-scores: contiene l'ID del film, l'ID del tag e la rilevanza del tag.
- genome-tags: contiene l'ID del tag e la sua descrizione.
- ratings: contiene l'ID dell'utente, l'ID del film e la valutazione.

Data Acquisition (2)

I dati, distribuiti in diversi file, sono stati aggregati in un unico file .cvs.

movieId	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	
5	Father of the Bride Part II (1995)	

userId	movieId	rating	timestamp
1	296	5.0	1147880044
1	306	3.5	1147868817
1			
1			
1			

movieId	tagId	relevance
1	1	0.02875
1	2	0.02375
1	3	0.06250
1	4	
1	5	

tagId	tag
1	007
2	007 (series)
3	18th century
4	1920s
5	1930s

movieId	title	007	007 (series)	18th century	1920s	1930s	1950s	1960s	1970s	...	world politics	world war i
1	Toy Story (1995)	0.02875	0.02375	0.06250	0.07575	0.14075	0.14675	0.06350	0.20375	...	0.04050	0.01425
2	Jumanji (1995)	0.04125	0.04050	0.06275	0.06275	0.09100	0.06125	0.06925	0.09600	...	0.05250	0.01575
3	Grumpier Old Man (1995)	0.04675	0.05550	0.02925	0.08700	0.04750	0.04775	0.04600	0.14275	...	0.06275	0.01950
4	Waiting to Exhale (1995)	0.03425	0.03800	0.04050	0.03100	0.06500	0.03575	0.02900	0.08650	...	0.05325	0.02600
5	Father of the Bride Part II (1995)	0.04300	0.05325	0.03800	0.04100	0.05400	0.06725	0.02775	0.07650	...	0.05350	0.02050

Data Visualization

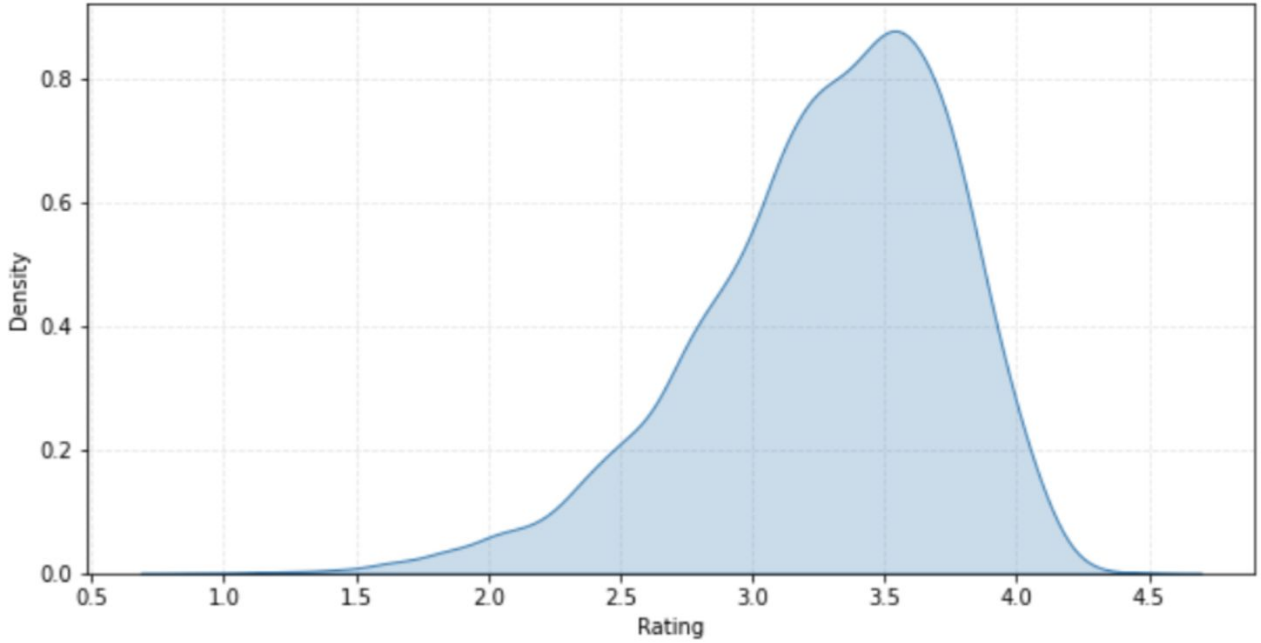

Figure 1: Distribuzione dei voti

Data Visualization (2)

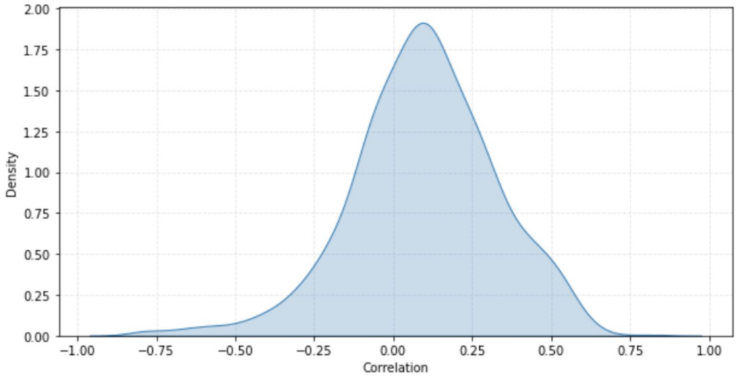

Figure 2: Distribuzione della correlazione tra tag e voti

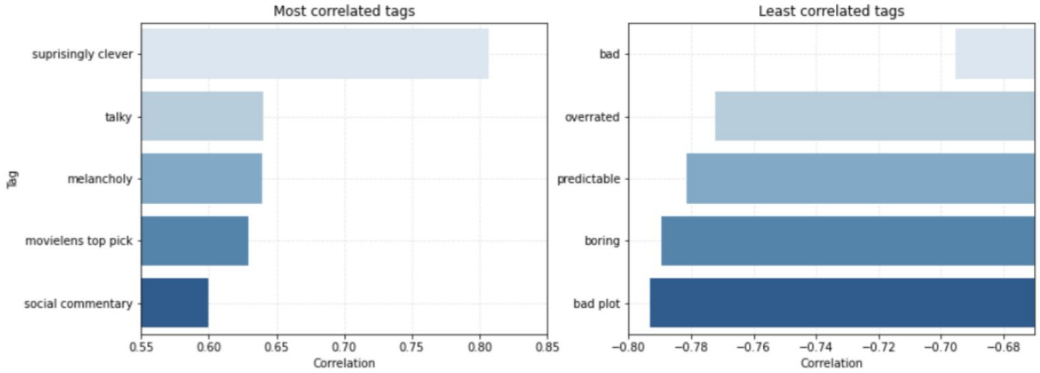

Figure 3: Correlazione tra tag e voti

Data Visualization (3)

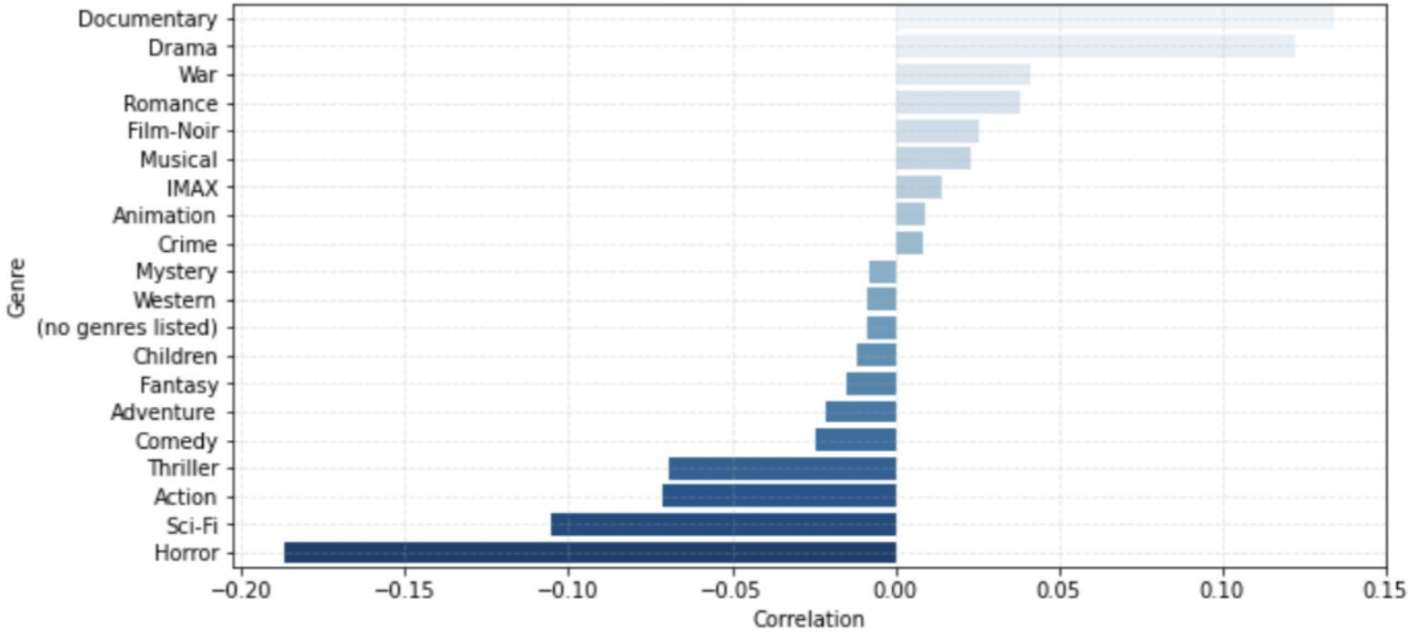

Figure 4: Correlazione tra generi e voti

Data Visualization (4)

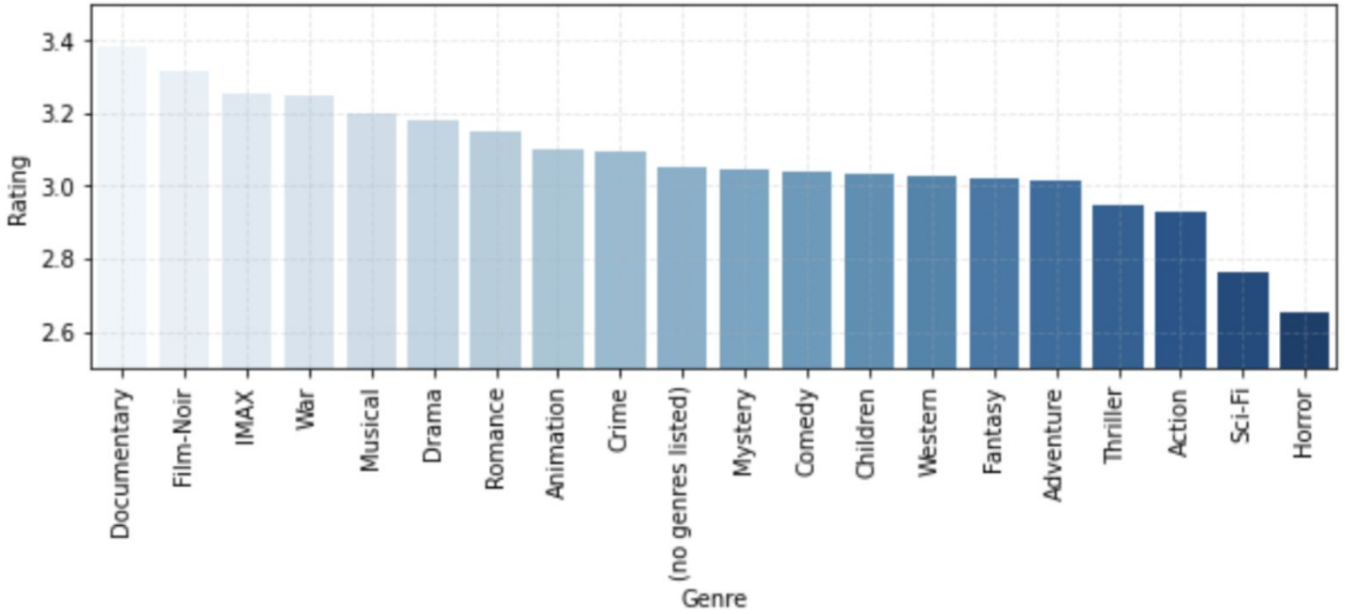

Figure 5: Voto medio per ogni genere

Data Visualization (5)

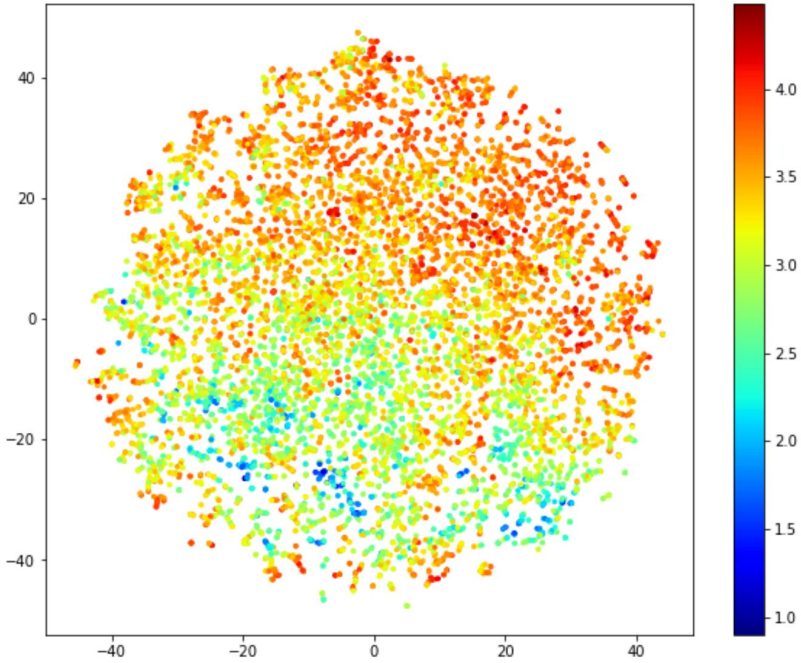

Figure 6: Rappresentazione 2D dei dati

Data Preprocessing - Dataset

- Il dataset utilizzato nel progetto è stato suddiviso in train, validation e test set.
- È stato fissato un seed per garantire che la suddivisione del dataset sia riproducibile in modo coerente in futuro.
- Lo stesso test set viene usato per valutare tutti i modelli, al fine di garantire dei risultati coerenti per tutti i modelli.

Data Preprocessing - PCA

La **Principal Component Analysis (PCA)** è una tecnica di **riduzione** della **dimensionalità** che mira a rappresentare le variazioni delle variabili del dataset utilizzando un numero inferiore di componenti principali.

L'idea è di trasformare un set di variabili correlate in un nuovo set di variabili linearmente indipendenti, chiamati **componenti principali**.

Modelling

Le **tecniche** di **Machine Learning** utilizzate sono:

- Tradizionali
- Rete neurale
- Modello deep per Tabular Data - TabNet

Modelli tradizionali - Linear Regression

I **Regressori Lineari** sono modelli costruiti utilizzando una funzione lineare che modella la relazione tra le feature del film e del voto.
L'obiettivo è trovare i **coefficienti** della funzione che **minimizzano l'errore**.

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

Modelli tradizionali - Random Forest

Il **Random Forest** è un algoritmo di apprendimento supervisionato basato sulla creazione di un insieme di alberi decisionali.

Gli **hyperparametri** e i relativi valori utilizzati sono:

- n_estimator = [10, 15, 20, 25, 30]
- criterion = [’squared_error’, ’friedman_mse’]

Modelli tradizionali - K-Nearest Neighbors

L’algoritmo **K-NN** cerca i k dati più vicini presenti all’interno del train set.

Gli **hyperparametri** e i relativi valori utilizzati sono:

- n_neighbors = [7, 8, 9, 10, 15, 20]
- weights = [’uniform’, ’distance’]

Modelli tradizionali - Support Vector Regressor

L’algoritmo **SVR** cerca il piano che meglio approssima i dati di input, ma che al contempo sia il più possibile lontano dai punti che si trovano al di fuori del boundary line, questi punti sono chiamati vettori di supporto.

Gli **hyperparametri** e i relativi valori utilizzati sono:

- kernel = [’linear’, ’poly’, ’rbf’]
- epsilon = [0.001, 0.01, 0.1, 1]

Neural Network - Feed-forward network

La **feed-forward** è una rete con un numero variabile di layer e in cui ogni layer di input ed il relativo layer di output si muovono in una sola direzione.

Gli **hyperparametri** e i relativi valori utilizzati sono:

- hidden_size = [128, 256, 512]
- dropout_prob = [0.2, 0.3, 0.4]
- dept = [3, 4, 5]
- batch_size = [8, 16, 32]
- learning_rate = [0.001, 0.01]
- epochs = [200]

Tabular Data - TabNet

TabNet è un algoritmo di ML sviluppato nel 2019 da diversi ricercatori di Google Research, è nato in **risposta** alle **limitate prestazioni** delle **reti neurali deep** applicate **ai dati tabulati.**

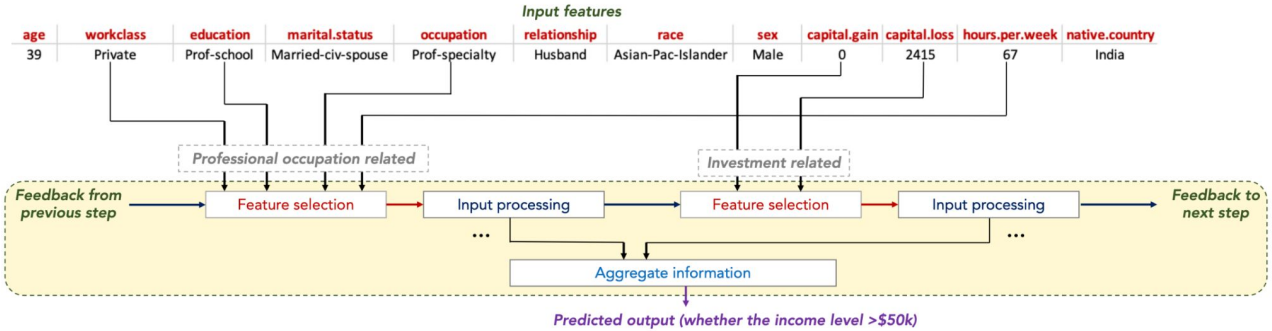

Figure 8: Selezione feature

Prende in input **dati grezzi** e utilizza l'**attenzione sequenziale** per **selezionare** le **feature** da utilizzare in ogni step decisionale.
Migliore interpretazione e apprendimento.

Tabular Data - TabNet (2)

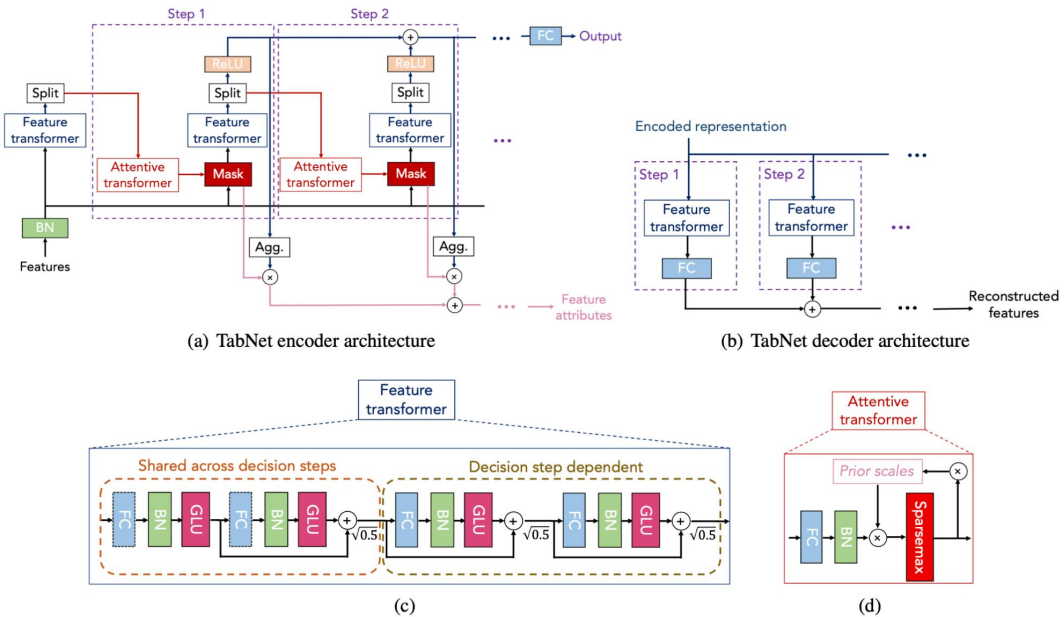

Figure 10: Architettura TabNet

Durante il training, le feature vengono passate all'**encoder** composto da:

- **attentive transformer** seleziona le **feature più rilevanti**.
- **mask** utilizza le informazioni precedenti per **aggregare** quanto ogni **feature** è stata **utilizzata** nelle fasi **decisionali precedenti.**
- **feature transformer** che trasforma le feature in una nuova rappresentazione, utilizzata per prendere decisioni, utilizzando una serie di layer.

Il **decoder** formato da un **blocco di feature trasformer per ogni step**.

Tabular Data - TabNet (3)

Gli **hyperparametri** e i relativi valori utilizzati sono:

- batch_size = [256]: numero di sample contenuti in ogni batch.
- n_d = [8, 16, 32]: determina la dimensione del layer di predizione.
- n_a = [8, 16, 32] : determina la dimensione dello spazio di output dell'attentive transformer.
- n_steps = [3, 5, 7]: numero di step nell'architettura.
- n_independent = [2, 3] : numero di layer GLU indipendenti per ogni blocco GLU.
- epochs = [200]: indica il numero dei cicli completi di addestramento della rete.

Performance Evalution

Le misure utilizzate per valutare tutti i modelli sono:

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

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

- **Coefficient of Determination (R2):** misura quanto bene il modello si adatta ai dati.

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

Per ogni modelli sono state valutate le performance con e senza l'applicazione della PCA.

Modelli tradizionali - Linear Regression

Metrica	PCA	No PCA
mse	0.00643	0.00544
R2	0.97096	0.97543

Table 1: Risultati linear regression

Modelli tradizionali - Ridge Regression

Metrica	PCA	No PCA
mse	0.00639	0.00532
R2	0.97113	0.97598

Table 2: Risultati ridge regression

La **configurazione migliore** è con alpha = 5

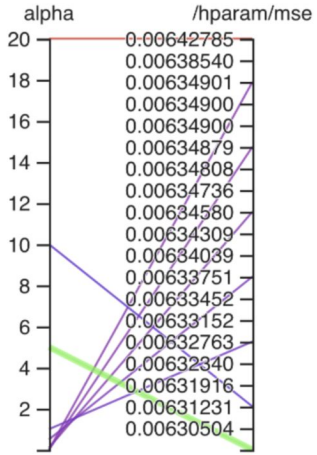

Figure 11: Parallel coordinates view ridge regression con PCA

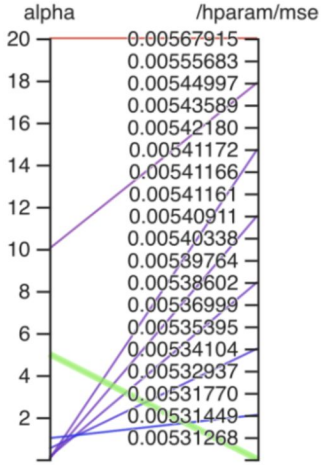

Figure 12: Parallel coordinates view ridge regression senza PCA

Modelli tradizionali - Lasso Regression

Metrica	PCA	No PCA
mse	0.00641	0.00542
R2	0.97106	0.97553

Table 3: Risultati lasso regression

La **configurazione migliore** è con alpha = 1-e05

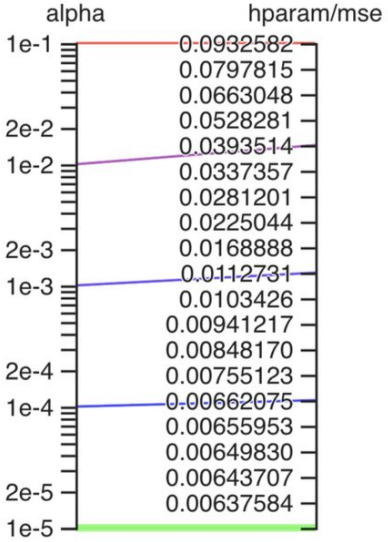

Figure 13: Parallel coordinates view lasso regression con PCA

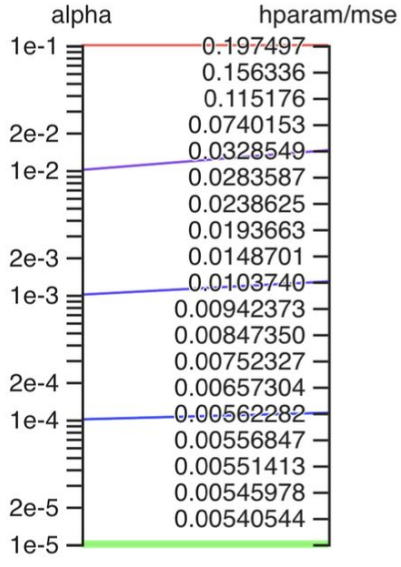

Figure 14: Parallel coordinates view lasso regression senza PCA

Modelli tradizionali - Lasso Regression (2)

Feature	Correlazione
007	-0.097706
1920s	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

Table 4: Correlazione delle prime 10 feature con i coefficienti settati a 0 e i voti

Modelli tradizionali - Random Forest

Parametro	Migliore configurazione (PCA)	Migliore configurazione (senza PCA)
n_estimators:	30	30
criterion	squared_error	friedman_mse

Table 5: Configurazione migliore Random forest

Metrica	PCA	No PCA
mse	0.03735	0.01255
R2	0.83150	0.94334

Table 6: Risultati Random forest

Modelli tradizionali - Random Forest (2)

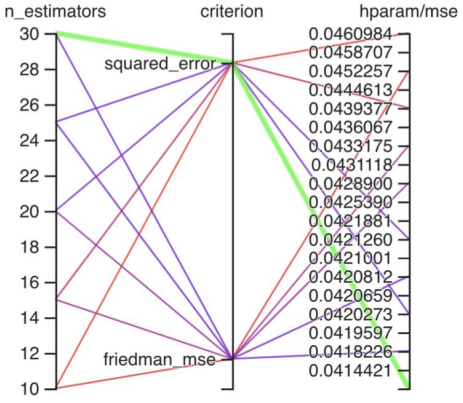

Figure 15: Parallel coordinates view random forest con PCA

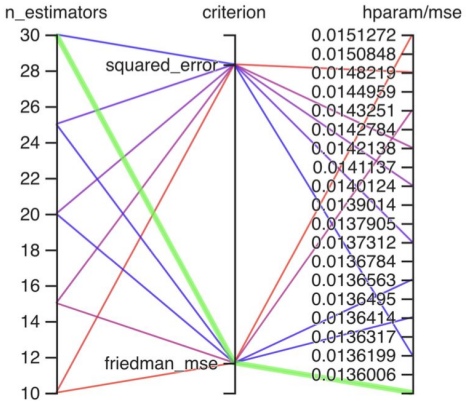

Figure 16: Parallel coordinates view random forest senza PCA

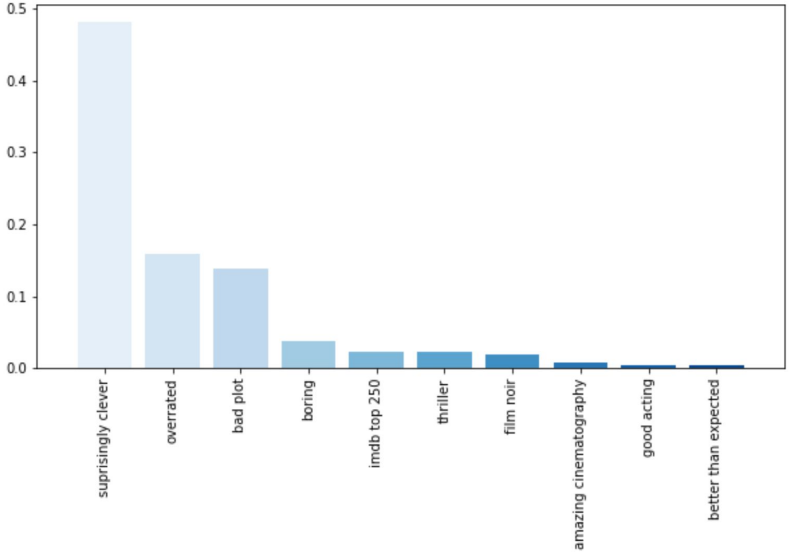

Figure 17: Le 10 feature più importanti

Modelli tradizionali - K-Nearest Neighbors

Parametro	Migliore configurazione (PCA)	Migliore configurazione (senza PCA)
n_neighbors	9	10
weights	distance	distance

Table 7: Configurazione migliore KNN

Metrica	PCA	No PCA
mse	0.04028	0.04050
R2	0.81828	0.81727

Table 8: Risultati KNN

Modelli tradizionali - K-Nearest Neighbors (2)

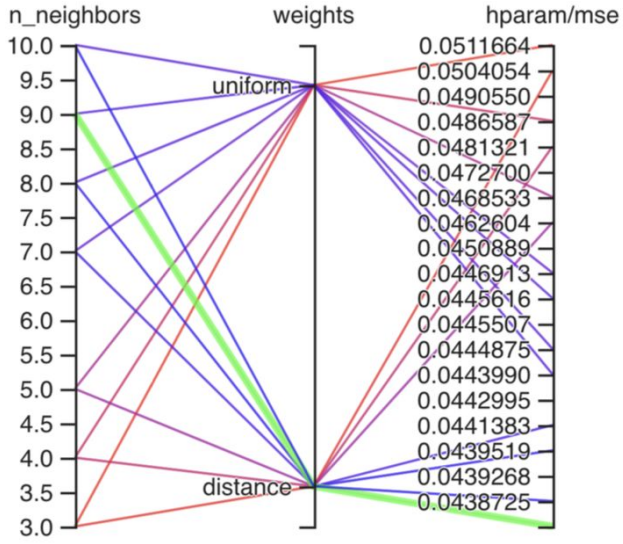

Figure 18: Parallel coordinates view KNN con PCA

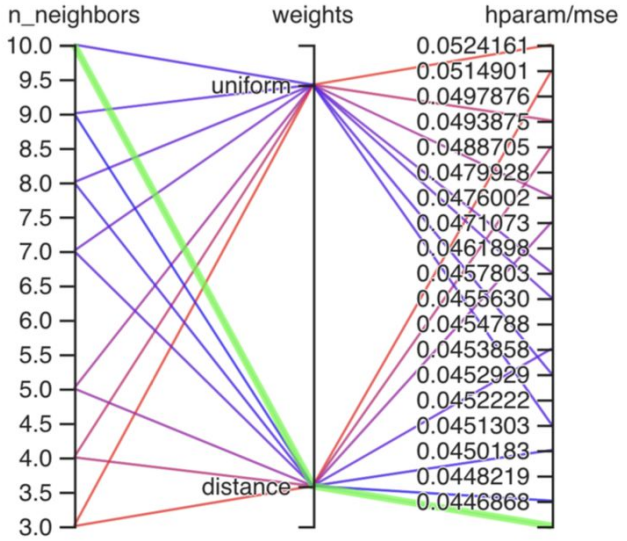

Figure 19: Parallel coordinates view KNN senza PCA

Modelli tradizionali - Support Vector Regressor

Parametro	Migliore configurazione (PCA)	Migliore configurazione (senza PCA)
epsilon	0.01	0.001
kernel	linear	rbf

Table 9: Configurazione migliore SVR

Metrica	PCA	No PCA
mse	0.00645	0.00521
R2	0.97086	0.97647

Table 10: Risultati SVR

Modelli tradizionali - Support Vector Regressor (2)

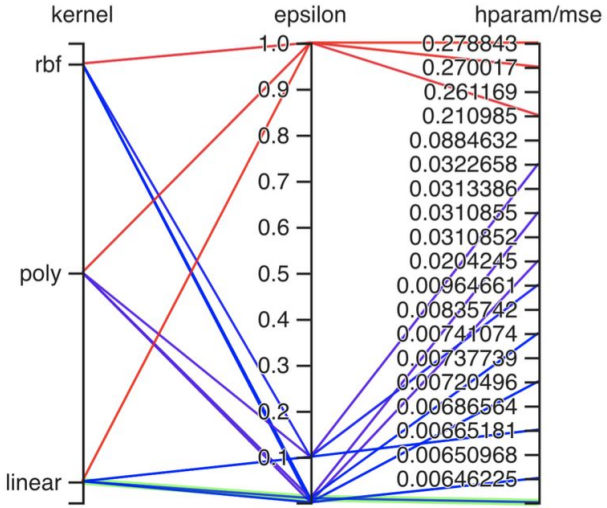

Figure 20: Parallel coordinates view SVR con PCA

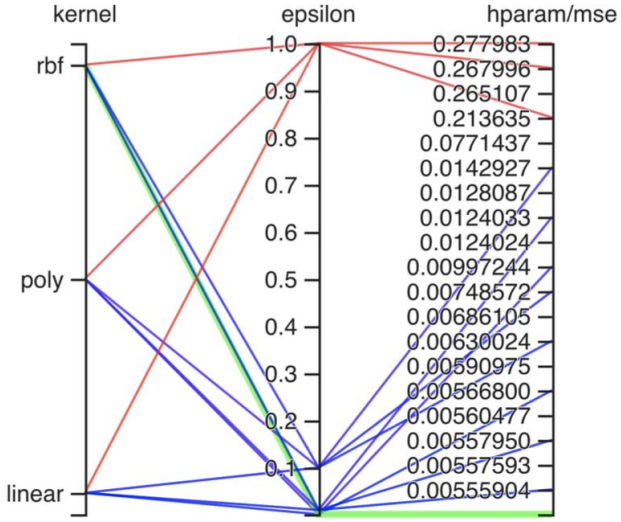

Figure 21: Parallel coordinates view SVR senza PCA

Neural Network - Feed-forward network

Parametro	Migliore configurazione PCA	Migliore configurazione senza PCA
hidden_size	512	256
dropout_prob	0.3	0.4
dept	3	5
batch_size	8	8
learning_rate	0.001	0.001

Table 11: Configurazione migliore feed-forward network

Metrica	PCA	No PCA
mse	0.00519	0.00501
R2	0.97655	0.97735

Table 12: Risultati feed-forward network

Neural Network - Feed-forward network (2)

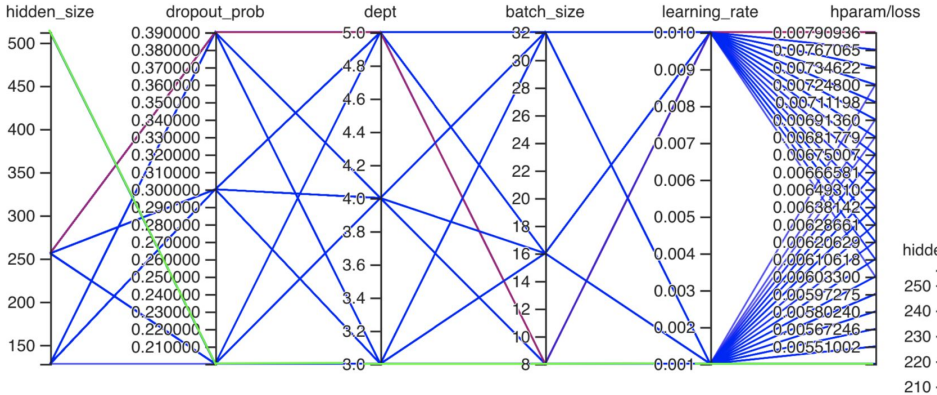

Figure 22: Parallel coordinates view feed-forward network con PCA

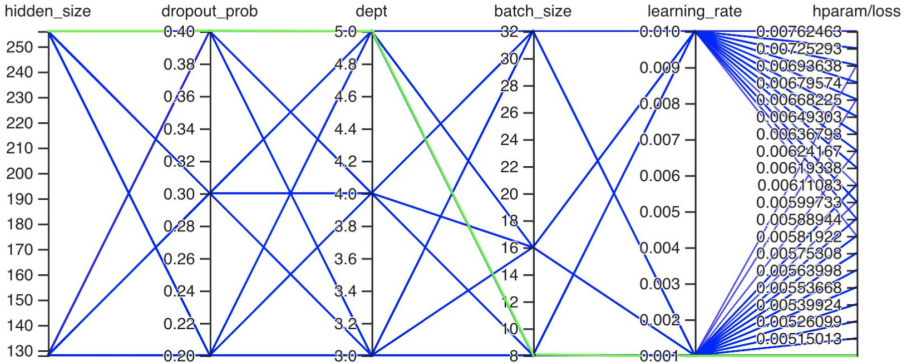

Figure 24: Parallel coordinates view feed-forward network senza PCA

Neural Network - Feed-forward network (3)

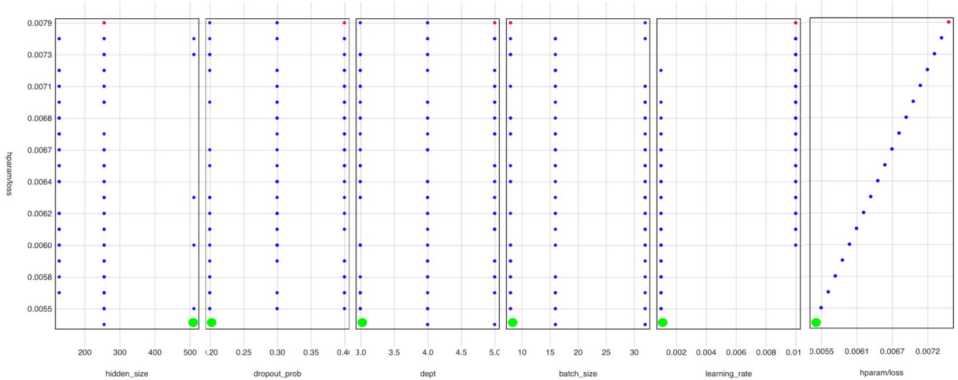

Figure 23: Scatterplot matrix view feed-forward network con PCA

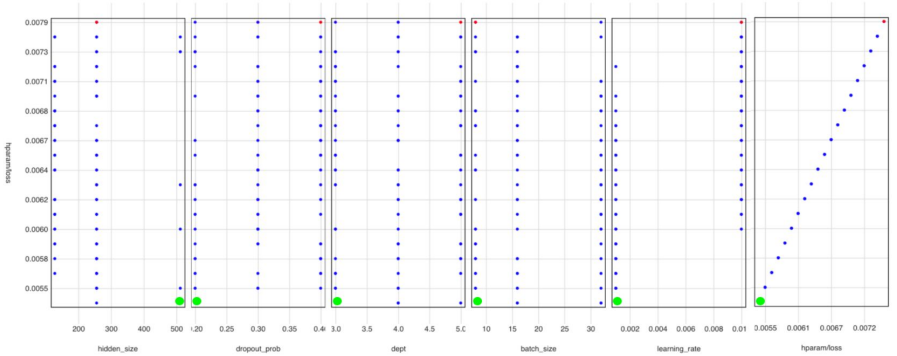

Figure 25: Scatterplot matrix view feed-forward network senza PCA

Tabular Data - TabNet

Parametro	Migliore configurazione con PCA	Migliore configurazione senza PCA
batch_size	256	256
n_d	8	8
n_a	16	16
n_steps	5	3
n_indipendent	2	2

Table 13: Configurazione migliore TabNet

Metrica	PCA	No PCA
mse	0.00664	0.00535
R2	0.97001	0.97583

Table 14: Risultati TabNet

Tabular Data - TabNet (2)

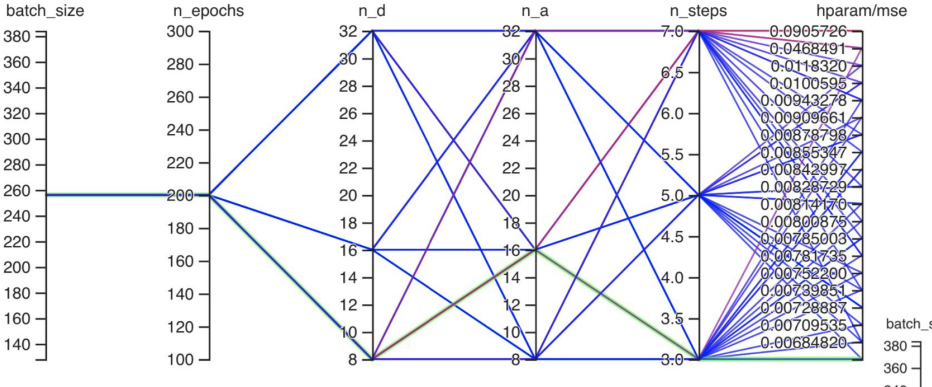

Figure 26: Parallel coordinates view TabNet con PCA

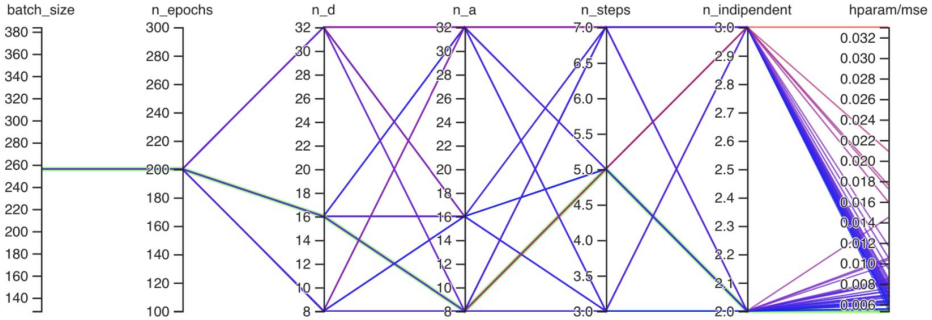

Figure 28: Parallel coordinates view TabNet senza PCA

Tabular Data - TabNet (3)

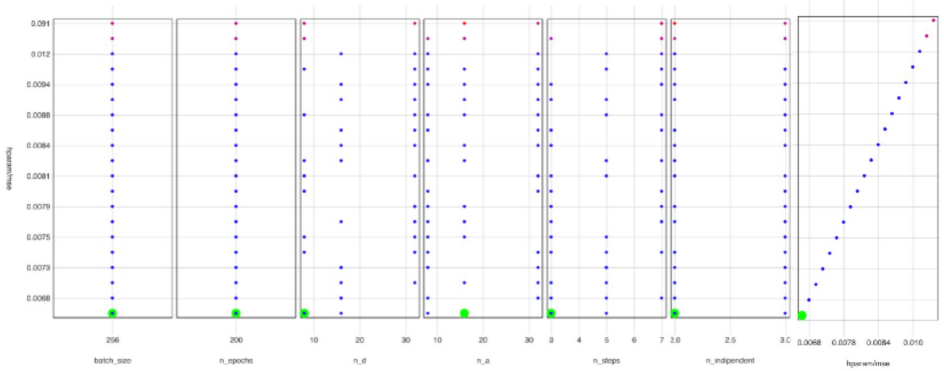

Figure 27: Scatterplot matrix view TabNet con PCA

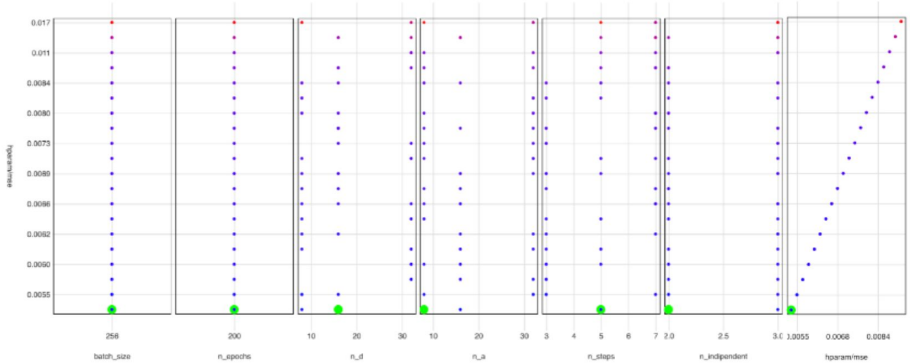

Figure 29: Scatterplot matrix view TabNet senza PCA

Conclusioni

Tutti i modelli hanno **prestazioni migliori** utilizzando i dati a cui **non** è stata **applicata** la **PCA**.

Quasi tutti i **modelli tradizionali**, ad eccezione di KNN e Random Forest, hanno ottenuto **ottime performance**.

La rete **Feed-Forward** e il **TabNet** hanno ottenuto **risultati** leggermente **migliori** rispetto ai modelli non deep.

Grazie per l’attenzione!