UNIVERSITÀ
DEGLI STUDI
DI PALERMO

Celiachion:
riduzione dimensionale, relative visualizzazioni e classificazione

Salvatore Calderaro Simone Contini Dario Curreri

Abstract

Abbiamo condotto uno studio empirico per determinare le relazioni presenti tra quattro diverse tecniche di visualizzazione (scatterplot 2D per dati bidimensionali, scatterplot 2D per dati tridimensionali, scatterplot 3D interattivo e plot di Draftman) e quattro diverse tecniche per la riduzione della dimensionalità (PCA, kernel PCA, MDS e t-SNE). L'analisi è stata effettuata su un dataset virtuale contenente dati biomedici utilizzati per la diagnosi della celiachia. Dalle rappresentazioni grafiche si evince come PCA e kernel PCA siano più efficienti per la visualizzazione delle due classi. Inoltre si nota che nella maggior parte dei casi gli scatterplot 2D sono già sufficienti per rilevare una buona distinzione tra le classi. Sono stati creati, dunque, degli alberi decisionali sia per il dataset non ridotto, sia per i dataset ridotti tramite le tecniche precedentemente citate. Infine i risultati ottenuti sono stati confrontati - in termini di accuratezza - con quelli del classificatore addestrato sul dataset non ridotto.

1 Introduzione

Ridurre la dimensionalità dei dati è uno dei task fondamentali per l'analisi dei dati e per le eventuali elaborazioni successive (regressione, classificazione, etc.).
In *Sezione 2* sono descritte le tecniche di riduzione dimensionale applicate al nostro dataset: PCA, kernel PCA, MDS e t-SNE.
In *Sezione 3* vengono dapprima definite le quattro tecniche di visualizzazione impiegate: scatterplot 2D per dati bidimensionali, scatterplot 2D per dati tridimensionali, scatterplot 3D interattivo e plot di Draftman; successivamente queste vengono applicate per ciascuna tecnica di riduzione.
In *Sezione 4*, invece, vengono addestrati e illustrati gli alberi decisionali sia per il dataset non ridotto, sia per i dataset ridotti attraverso le tecniche di riduzione trattate.
In *Sezione 5*, infine, vengono tratte le conclusioni riguardo le migliori visualizzazioni prodotte e confrontate le metriche dei classificatori addestrati.

1.1 Il dataset "nome dataset"

Il dataset utilizzato per gli esperimenti è il "celiachion", un insieme di dati relativi a pazienti virtuali generato per addestrare un classificatore fuzzy per la diagnosi della celiachia nei bambini siciliani e maltesi.
In particolare il dataset contiene per ogni paziente le seguenti informazioni:

- Anemia: valore booleano. Indica se il paziente soffre di anemia (0 negativo, 1 positivo)
- Osteopenia: valore booleano. Indica se il paziente soffre di osteopenia.
- Diarrea cronica: valore booleano. Indica se il paziente soffre di diarrea cronica
- Mancata crescita: valore booleano. Indica se il paziente soffre di ritardo della crescita
- Disturbi genetici: valore booleano. Indica se il paziente soffre di disturbi genetici
- Madre celiaca: valore booleano. Indica se la madre del paziente soffre di celiachia
- POCT: valore booleano. Indica se il test POCT è risultato positivo o negativo.
- IGA totali: numero reale positivo. Indica la quantità di immunoglobuline A misurata in mg/dl

- TTG IGG: numero reale positivo. Indica la quantità di anticorpi anti-transglutaminasi appartenenti alla famiglia delle immunoglobuline G, misurata in AU/ml

- TTG IGA: numero reale positivo Indica la quantità di anticorpi anti-transglutaminasi appartenenti alla famiglia delle immunoglobuline A, misurata in AU/ml

- Esami del sangue: valore booleano. Indica se gli esami del sangue svolti dal paziente sono risultati idonei alla celiachia

- Class: numero naturale. Indica se il paziente soffre di celiachia

2 Tecniche di riduzione dimensionale

L'obiettivo delle tecniche di riduzione dimensionale è quello di eseguire un mapping dallo spazio iniziale $\mathcal{R}^d$, dove d rappresenta il numero delle feature, a uno spazio dimensionale inferiore $\mathcal{R}^k$ con $k < d$. Utilizzare tali tecniche è utile principalmente per due motivi: permettono di analizzare i dati in due o più dimensioni e rendono più semplice e meno oneroso dal punto di vista computazionale l'addestramento di algoritmi di machine learning.
A seguire vengono descritte brevemente le tecniche di riduzione dimensionale usate in questo lavoro.

2.1 Principal Component Analysis (PCA)

Utilizza una trasformazione ortogonale per convertire una serie di osservazioni di variabili correlate in un insieme di valori linearmente non correlati, chiamate *componenti principali*. La trasformazione è effettuata in modo tale che la prima componente principale abbia la massima varianza e ogni componente principale successiva abbia a sua volta la massima varianza e sia ortogonale alle componenti principali calcolate precedentemente. Assumendo di avere n osservazioni e p feature, la matrice dei dati è una matrice $X \in \mathbb{R}^{n\times p}$, in cui i dati sono standardizzati e in cui, di conseguenza, ogni colonna ha media nulla. Allora, la prima componente principale è la combinazione lineare normalizzata di feature che ha varianza massima:

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$$

Ogni colonna è pesata sulla sua media, e si massimizza la varianza delle combinazioni:

$$\max_{\phi_{i1}\cdots\phi_{p1}} \left\{ \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \right\} \ , \ t.c. \ \sum_{j=1}^{p}\phi_{j1}^2 = 1$$

Per estrarre la componente principale si estraggono gli autovalori della matrice X^TX e si considera l'autovettore corrispondente al massimo degli autovalori ottenendo il loading vector $\phi_1 = [\phi_{11}, \phi_{21}\cdots, \phi_{p1}]$. Fatto vengono calcolati gli score $z_{i1} = \sum_{j=1}^{p}\phi_{j1}x_{ij}$. Gli score risultano dalla proiezione delle osservazioni lungo la direzione del loading vector. Per il calcolo delle componenti principali successive si utilizza lo stesso metodo sopra descritto.

2.2 Kernel Principal Component Analysis (kernelPCA)

PCA consente di effettuare una riduzione dimensionale lineare, non risulta utile invece per dati che presentano strutture complesse che non possono essere espresse da sottospazi lineari. Il kernelPCA permette di generalizzare PCA in modo da effettuare una riduzione dimensionale non lineare.
Assumiamo di avere una trasformazione non lineare ϕ definita come segue: $\phi : \mathbb{R}^D \to \mathbb{R}^M$ con $M \gg D$. Fatto ciò, ogni punto x_i viene mappato in un punto $\phi(x_i)$. Assumiamo che:

- le nuove feature proiettate hanno media 0: $\frac{1}{N}\sum_{i=1}^{N}\phi(x_i) = 0$

- la matrice di covarianza delle feature proiettate ha dimensione $M \times M$ ed è calcolata come segue: $C = \frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\phi(x_i)^T$

- gli autovalori e gli autovettori della matrice sopra definita sono dati da: $Cv_k = \lambda_k v_k$ con $k = 1, 2, \cdots, M$.

Dopo aver fatto queste premesse, si sceglie una *funzione kernel* $\kappa(x_i, x_j) = \phi(x_i)^T\phi(x_j)$. Se viene usata una notazione matriciale abbiamo che:

$$K^2a_k = \lambda_k NKa_k$$

dove $k_{i,j} = \kappa(x_i, x_j)$ e a_k è un vettore colonna di dimensione N di a_{ki}: $\begin{bmatrix} a_{k1} & a_{k2} & \cdots & a_{kN}\end{bmatrix}^T$. a_k può essere ottenuto dalla seguente equazione $Ka_k = \lambda_k NKa_k$ e le componenti principali verranno calcolate come segue:

$$y_k(n) = \phi(x)^T v_k = \sum_{i=1}^{N} a_{ki}\kappa(x, x_i)$$

Se il datset risultante dalla proiezione $\{\phi(x_i)\}$ non ha media zero, verrà utilizzata la *matrice di Gram* $\tilde{K}$ in sostituzione della matrice kernel K. La matrice di Gram viene definita cone segue:

$$\tilde{K} = K - 1_N K - K 1_N + 1_N K 1_N$$

dove 1_N è una matrice dimensione $N \times N$ con tutti gli elementi uguali a $\frac{1}{N}$. Alcuni dei kernel più comunemente utilizzati sono:

- *kernel polinomiale*: $\kappa(x, y) = (x^T y)^d$
- *kernel gaussiano*: $\kappa(x, y) = \exp\left(\frac{-||x-y||^2}{2\sigma^2}\right)$
- *kernel sigmoidale*: $\kappa(x, y) = \tanh(\alpha x^T y + c)$.

2.3 Multi-Dimensional Scaling (MDS)

Tecnica esplorativa dei dati che ci consente di ottenere una rappresentazione di n oggetti in k dimensioni partendo da delle informazioni inerenti la similarità (o dissimilarità) tra ciascuna coppia di oggetti. Iniziamo con il considerare le distanze δ_{ij} tra ogni coppia degli n punti $x_1 \cdots x_n$. A partire da queste ultime viene costruita una matrice Δ $n \times n$:

$$\Delta = \begin{bmatrix} 0 & & \\ \delta_{21} & & \\ \vdots & \ddots & \\ \delta_{n1} & \dots & 0 \end{bmatrix}$$

Chiamiamo d_{ij} le distanze tra le immagini $y_1, \cdots, y_n$ nello spazio di dimensione k $d_{ij} = ||y_i - y_j||$. Le componenti di y_i nello spazio di arrivo sono $y_{11}, y_{12}, \cdots, y_{1k}$ e possono essere rappresentati dalla matrice che segue:

$$A = \begin{bmatrix} y_{11} & \cdots & y_{1k} \\ \vdots & \ddots & \vdots \\ y_{n1} & \dots & y_{nk} \end{bmatrix}$$

Adesso ci cerca la configurazione $y_1, \cdots, y_n$ per la quale le $\frac{n(n-1)}{2}$ distanze d_{ij} siano più vicine possibile alle distanze originali δ_{ij}. Di solito, non è possibile trovare una configurazione per la quale $d_{ij} = \delta_{ij}$ per tutti gli $i \neq j$. Per trovare una configurazione conveniente fra tutte quelle possibile, è necessario definire un criterio d'errore, spesso chiamato funzione di Stress definita come segue:

$$Stress^2 = \frac{\sum_{i<j} (d_{ij} - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^2}$$

Una configurazione ottima degli $y_1, \cdots, y_n$ è quella che minimizza lo Stress, e può essere trovata con procedure standard di analisi numerica. Per diminuire il tempo di calcolo, la configurazione di partenza $y_1, \cdots, y_n$ può essere scelta a caso o in modo più conveniente.

2.4 t-Distributed Stochastic Neighbor Embedding (t-SNE)

Tecnica di riduzione della dimensionalità non lineare. L'algoritmo modella i punti in modo che oggetti vicini nello spazio originale risultino vicini nello spazio a dimensionalità ridotta, e oggetti lontani risultino lontani, cercando di preservare la struttura locale. L'algoritmo consiste di due fasi. Nella prima fase viene costruita una distribuzione di probabilità che ad ogni coppia di punti nello spazio originale ad alta dimensionalità associa un valore di probabilità elevato se i due punti sono simili, basso se sono dissimili. Quindi viene definita una seconda distribuzione di probabilità analoga, nello spazio a dimensione ridotta. Nella seconda fase l'algoritmo minimizza la divergenza di Kullback-Leibler delle due distribuzioni tramite gradient descent, riorganizzando i punti nello spazio a dimensione ridotta. Dati N oggetti $x_1, \cdots x_n$ in uno spazio ad alta dimensionalità, t-SNE costruisce una distribuzione probabilistica p_{ij} simmetrica nelle due variabili e proporzionale alla similarità tra i punti x_i e x_j definita come segue:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

dove

$$p_{j|i} = \frac{\exp\left(\frac{-||x_i - x_j||^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(\frac{-||x_i - x_k||^2}{2\sigma_i^2}\right)}$$

L'ampiezza delle gaussiane σ_i è scelta in modo tale che la perplessità della distribuzione condizionale uguagli un valore di perplessità fornito come iperparametro dell'algoritmo. t-SNE cerca di costruire una mappa d-dimensionale $y_1, \cdots, y_N$ i cui punti riflettano il meglio possibile la similarità p_{ij} nello spazio di partenza. La similarità q_{ij} tra due punti y_i e y_j nello spazio a dimensionalità ridotta viene definita come segue:

$$q_{ij} = \frac{(1 + ||x_i - x_j||^2)^{-1}}{\sum_{k \neq m} 1 + ||y_k - y_m||^2)^{-1}}$$

La differenza principale è l'uso nello spazio a dimensionalità ridotta di una distribuzione t di Student con un grado di libertà al posto della gaus-

siana, le cui code pesanti consentono di modellare meglio la dissimilarità tra oggetti distanti. La posizione y_i dei punti nello spazio a dimensione ridotta è quindi calcolata minimizzando tramite gradient descent la divergenza di Kullback-Leibler della distribuzione Q rispetto a P:

$$KL(P||Q) = \sum_{i \neq j} \log \frac{p_{ij}}{q_{ij}}$$

L'uso della divergenza di Kullback-Leibler come funzione obiettivo consente di avere penalità elevate se punti vicini nello spazio originale vengono considerati lontani nello spazio a dimensionalità ridotta , mentre il viceversa ha un'influenza minore, tendendo quindi a preservare la struttura locale della distribuzione dei punti. Il risultato è una mappa a bassa dimensionalità che riflette le similarità tra i punti nello spazio ad alta dimensionalità.

3 Tecniche di visualizzazione

L'analisi esplorativa dei dati è un'operazione finalizzata ad ottenere, per mezzo di rappresentazioni visuali, quanta più informazione possibile riguardo al problema in esame in un dataset. Per stabilire l'efficienza delle quattro tecniche di riduzione dimensionale - descritte in *Sezione 2* - nell'individuare le correlazioni tra le componenti del dataset "nome dataset", sono state utilizzate quattro diverse tecniche di visualizzazione:

- scatterplot 2D per dati bidimensionali
- scatterplot 2D per dati tridimensionali
- scatterplot 3D interattivo
- plot di Draftman

Lo scatterplot 2D per dati bidimensionali è una tecnica di visualizzazione per la rappresentazione su un piano cartesiano dei record di un dataset. In particolare, negli assi X ed Y vengono rappresentate le due componenti del dataset con la più alta varianza, calcolate dalla tecnica di riduzione dimensionale di riferimento.
Per l'aggiunta di una terza dimensione nello scatterplot 2D, viene associato ad ogni data point del piano un attributo - lo spazio - il cui valore è in relazione con la terza componente.
Lo scatterplot 3D interattivo è un'altra tecnica per la rappresentazione dei data point del dataset ridotto a tre componenti su un piano cartesiano. In questa tipologia di visualizzazione, la terza componente è rappresentata dall'asse Z.
Infine, Il plot di Draftman è una soluzione utile per la visualizzazione grafica di attributi multivariati. Tale tecnica consiste nel rappresentare coppie di attributi mediante scatterplot bidimensionali in una matrice NxN, dove N è il numero di attributi da rappresentare.

Per stabilire il numero di componenti ottimale del nostro dataset ridotto è stato creato uno scree-plot, un grafico i cui valori dell'asse delle ascisse corrispondono al numero delle componenti ed i valori dell'asse delle ordinate agli autovalori.

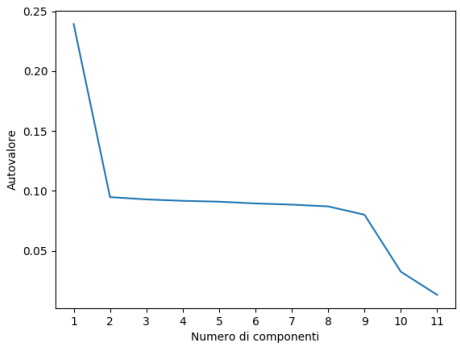

Figura 1: scree plot del dataset Celiachion

La *Figura 1* mostra lo scree-plot applicato al nostro dataset, che indicherebbe come migliore scelta un numero di componenti pari a 2 o 9, dal momento che è presente un "gomito netto" in corrispondenza di questi valori.

3.1 PCA

La *Figura 2* mostra lo scatterplot 2D bidimensionale dopo aver ridotto il nostro dataset con due componenti per mezzo della tecnica di riduzione PCA.

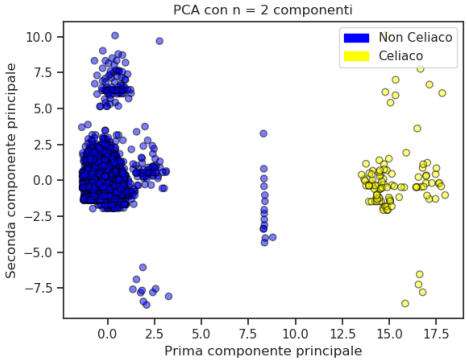

Figura 2: scatterplot 2D bidimensionale applicato su PCA

L'uso dei colori - blu per la classe *non celiaco* e giallo per la classe *celiaco* - è un'ulteriore informazione che abbiamo voluto aggiungere per verificare la veridicità della separabilità di ciascuna tecnica di riduzione.
Dalla *Figura 2* si evince che il PCA - con un numero di componenti principali pari a due - riesce a mantenere una buona separabilità dei data point, ad eccezione fatta per quelli centrali, i quali potrebbero essere facilmente fraintesi.
Una situazione simile a quella di *Figura 2* è evidente in *Figura 3*, che mostra lo scatterplot 2D a seguito della riduzione della dimensionalità del dataset a tre componenti per mezzo della PCA.

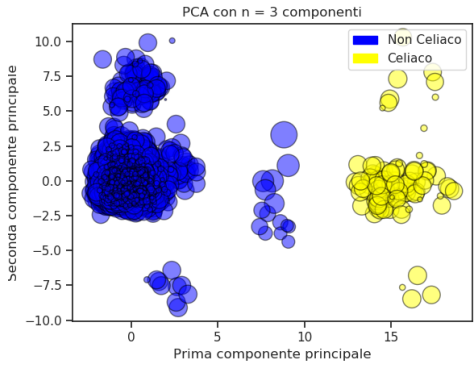

Figura 3: scatterplot 2D tridimensionale applicato su PCA

Ben evidente è la separabilità dei data point presenti nello scatterplot 3D interattivo, come visibile in *Figura 4*.

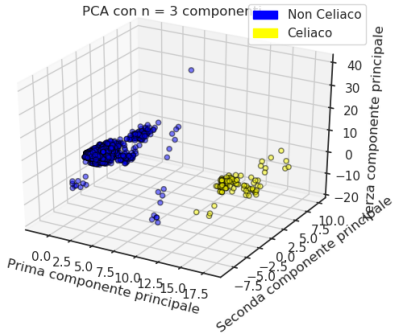

Figura 4: scatterplot 3D interattivo applicato su PCA

Infine, la *Figura 5* mostra il plot di Draftman applicato al dataset ridotto a 9 componenti per mezzo della tecnica PCA.

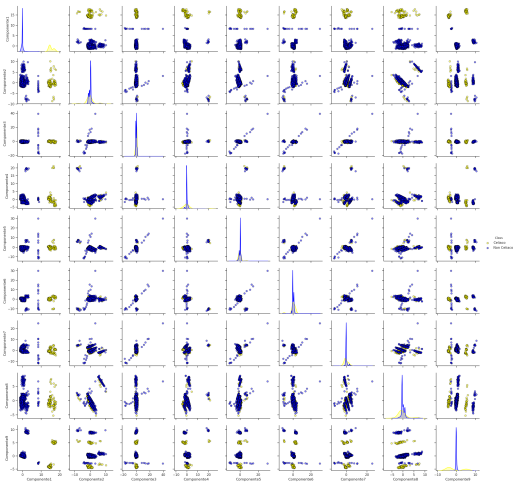

Figura 5: plot di Draftman applicato su PCA

Da tale rappresentazione è possibile notare come la Componente 1, se messo in relazione con tutte le altre componenti della matrice, riesca a mantenere la migliore separabilità.

3.2 kernelPCA

In *Figura 6* viene raffigurato lo scatterplot 2D bidimensionale a seguito della riduzione del dataset ridotto a due componenti per mezzo del kernelPCA.

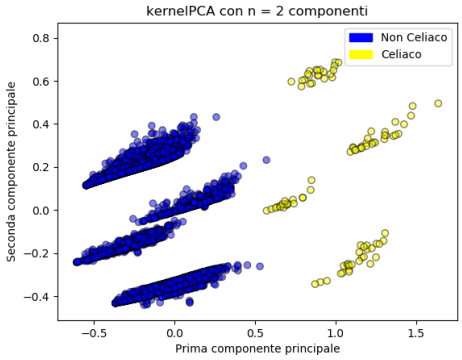

Figura 6: scatterplot 2D bidimensionale applicato su kernelPCA

Da tale rappresentazione - anche se non in maniera netta, data la formazione di diverse agglomerazioni - si evince comunque una buona separabilità dei data point.
La percezione di separabilità dei data point diminuisce drasticamente con il plot 2D tridimensionale, come ben visibile in *Figura 7*.

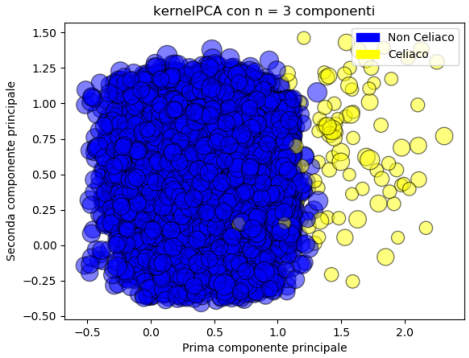

Figura 7: scatterplot 2D tridimensionale applicato su kernelPCA

In *Figura 8*, invece, é possibile notare come i data point siano ben distinti con una rappresentazione grafica 3D, risultando di fatto migliore delle precedenti per questa tecnica di riduzione.

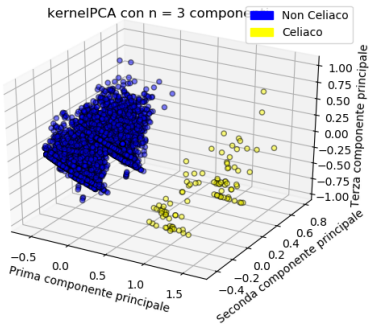

Figura 8: scatterplot 3D interattivo applicato su kernelPCA

Infine, la *Figura 9* mostra il plot di Draftman applicato su kernelPCA per un numero di componenti pari a 9, del tutto simile al plot di Draftman applicato su PCA per lo stesso numero di componenti.

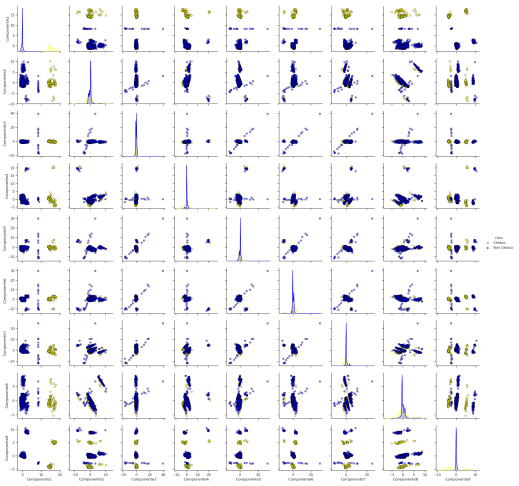

Figura 9: plot di Draftman applicato su kernelPCA

3.3 MDS

La *Figura 10* mostra lo scatterplot 2D bidimensionale del dataset ridotto a due componenti per mezzo del Multi-Dimensional Scaling (MDS).

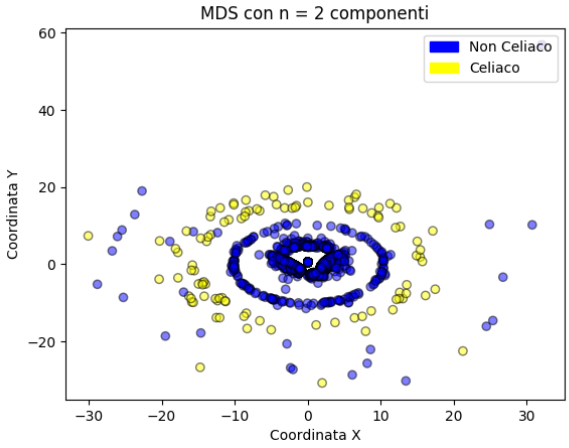

Figura 10: scatterplot 2D bidimensionale applicato su MDS

È possibile notare come MDS riesca, nonostante i numerosi outliers, a distinguere i punti, seppur in maniera decisamente peggiore delle due tecniche precedentemente descritte.

3.4 t-SNE

L'ultima tecnica di riduzione applicata è il t-SNE, rivelatosi poco efficace per la separazione dei data point del dataset, come è possibile evincere in *Figura 14*, *Figura 15*, *Figura 16* e *Figura 17*.

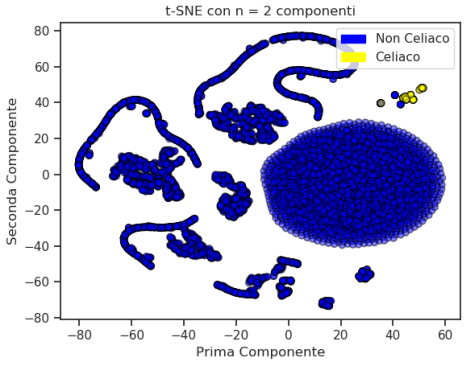

Figura 11: scatterplot 2D bidimensionale applicato su t-SNE

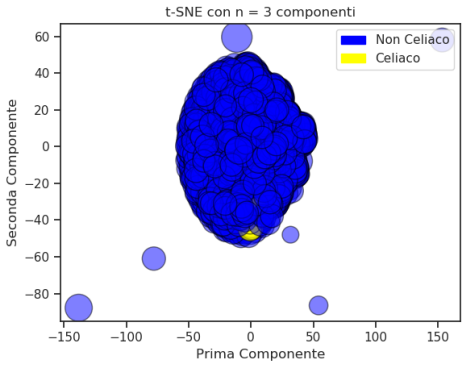

Figura 12: scatterplot 2D tridimensionale applicato su t-SNE

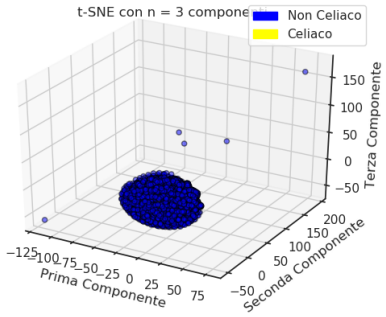

Figura 13: scatterplot 3D interattivo applicato su t-SNE

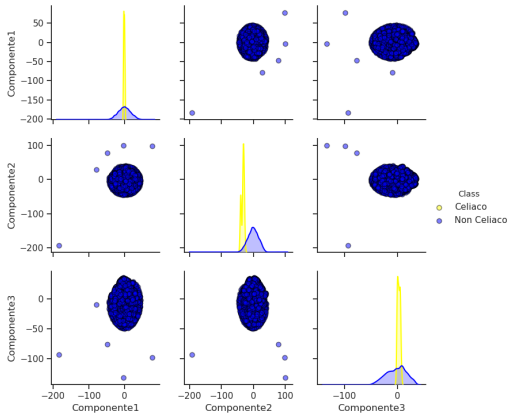

Figura 14: plot di Draftman applicato su t-SNE

4 Alberi decisionali

In machine learing, un albero decisionale è la rappresentazione di un modello predittivo progettato al fine di classificare le istanze di grandi quantità di dati (per questo motivo viene anche chiamato albero di classificazione), contenente solamente strutture selettive.
In un albero decisionale:

- ogni nodo interno effettua un test su un attributo. Questo test può risultare dunque vero o falso
- i nodi interni sono collegati attraverso rami corrispondenti ai risultati dei test
- ogni foglia rappresenta il valore predetto
- ad ogni nodo interno viene spesso associato un *tasso d'errore*, un *indice di Gini* oppure una *variazione di entropia*

Allora, dato un albero decisionale generato addestrando il modello predittivo tramite un training set, per classificare una istanza è necessario e sufficiente esaminare l'unico cammino radice-foglia compiuto da tale istanza.
In particolare, in questo studio, ad ogni nodo interno dell'albero è stato associato l'*indice di Gini*, indice di eterogeneità per variabili qualitative. Più precisamente, l'indice di Gini offre una misura della eterogeneità di una distribuzione statistica a partire dai valori delle frequenze relative associate alle k modalità di una generica variabile X.

4.1 Classificazione "nome dataset" non ridotto

In *Figura 18* viene mostrato l'albero decisionale generato a partire dal dataset non ridotto. In par-

ticolare, in questo albero, il 98% dei pazienti viene classificato attraverso il percorso radice-foglia più a sinistra.

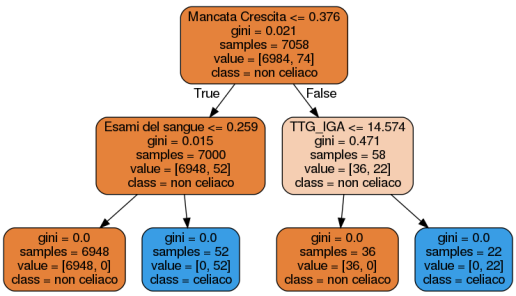

Figura 15: albero decisionale prodotto a partire dal dataset non ridotto.

A partire da questo albero, è stato possibile successivamente generare anche un *treemap*, come mostrato in *Figura 19*, utile per visualizzare graficamente i percorsi radice-foglie più rilevanti coinvolti nella classificazione.

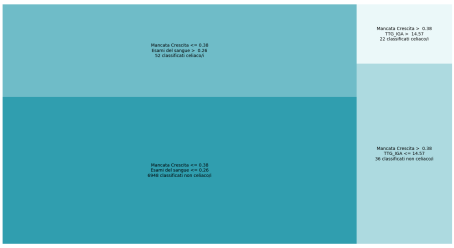

Figura 16: treemap corrispondente all'albero decisionale. Evidenzia quale siano le foglie più rilevanti.

4.2 Classificazione "nome dataset" ridotto

In *Figura 20* viene mostrato invece l'albero decisionale generato a partire dal dataset ridotto a 3 componenti tramite la tecnica PCA.

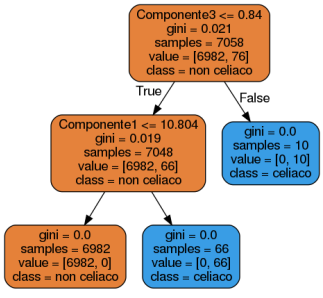

Figura 17: albero decisionale prodotto per il dataset ridotto tramite la tecnica PCA.

Di seguito vengono mostrati anche in *Figura 21*, *Figura 22* e *Figura 23*, gli alberi decisionali rispettivamente per il dataset ridotto a 3 componenti mediante le tecniche di kernel PCA, MDS e t-SNE.

Figura 18: albero decisionale prodotto per il dataset ridotto tramite la tecnica kernel PCA.

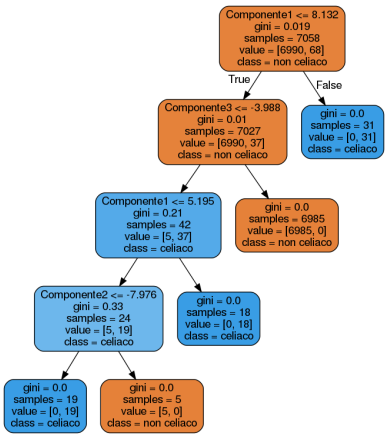

Figura 19: albero decisionale prodotto per il dataset ridotto tramite la tecnica MDS.

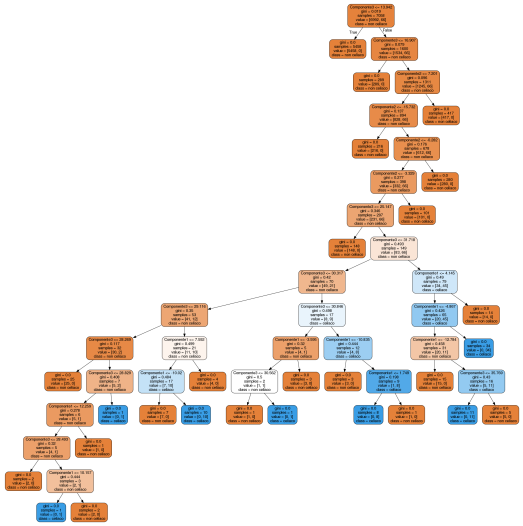

Figura 20: albero decisionale prodotto per il dataset ridotto tramite la tecnica t-SNE.

5 Conclusioni

Le tecniche di riduzione dimensionale più efficaci tra quelle discusse in questo studio sono risultate essere la PCA e la kernelPCA, come dimostrato in *Figura 2*, *Figura 3*, *Figura 4*, *Figura 5* per la PCA, e in *Figura 6*, in *Figura 8* e in *Figura 9* per il kernelPCA.

In particolare per la PCA: - interpretazione delle componenti - interpretazione dei cluster

In particolare, per la PCA è stato condotto anche uno studio volto ad interpretare le componenti.

	PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7	PCA8	PCA9	PCA10	PCA11
Anemia	0.02	-0.38	0.31	0.23	-0.68	0.33	-0.10	0.35	0.04	-0.05	-0.02
Osteopenia	-0.02	-0.12	-0.69	0.26	0.15	0.62	0.16	0.07	-0.10	-0.07	-0.02
Diarrea Cronica	0.11	-0.13	-0.20	-0.40	-0.21	-0.24	0.75	0.30	0.11	0.01	-0.01
Mancata Crescita	0.24	0.07	-0.06	-0.23	-0.03	0.27	-0.14	-0.18	0.87	-0.01	0.05
Disturbi Genetici	0.09	-0.27	0.01	0.73	0.24	-0.40	0.19	0.06	0.36	-0.00	0.02
Madre Celiaca	0.06	0.65	-0.08	0.13	0.03	-0.04	-0.15	0.72	0.09	-0.00	0.00
POCT	0.54	-0.05	-0.01	0.03	0.01	0.05	-0.04	0.01	-0.17	0.73	0.37
IGA totali	-0.01	0.56	0.03	0.35	-0.44	0.06	0.37	-0.47	-0.02	0.04	0.02
TTG IGG	-0.00	0.07	0.62	0.01	0.46	0.47	0.42	0.10	-0.00	-0.00	-0.00
TTG_IGA	0.54	0.02	0.03	0.00	0.01	-0.03	-0.01	-0.04	-0.18	-0.67	0.47
Esami del sangue	0.58	0.01	0.02	0.02	0.02	-0.01	-0.03	-0.05	-0.13	-0.04	-0.80

Figura 21:

Tutti gli alberi decisionali generati sono risultati molto efficaci nel classificare i pazienti virtuali. Infatti, l'albero decisionale per il dataset:

- non ridotto è accurato al 100%
- ridotto tramite PCA è accurato al 100%
- ridotto tramite kernelPCA è accurato al 99%
- ridotto tramite MDS è accurato al 99%
- ridotto tramite t-SNE è accurato al 99%

Quindi, l'albero decisionale corrispondente alla tecnica PCA risulta il più efficace, oltre che il più semplice in termini di profondità. Di contro, gli alberi decisionali risultanti dalle applicazioni delle tecniche kernelPCA e t-SNE risultano i più complessi.