

Corso di Laurea Magistrale in Ingegneria Matematica
Politecnico di Torino

Tesina Apprendimento Statistico

prof. Francesco Vaccarino

Antonio Cirigliano
s275053



Anno Accademico 2022/2023

Contents

1	Introduzione	1
1.1	Dataset	1
1.2	Linguaggi e tools	1
2	Esplorazione e visualizzazione dei dati	2
2.1	Istogramma	2
2.2	Box Plot	3
2.3	Matrice di correlazione	6
2.4	Pair Plot	7
3	Preprocessing	8
3.1	Training e test set	8
3.2	Standardizzazione	8
3.3	Riduzione della dimensionalità	9
3.3.1	Principal Component Analysis	9
4	Analisi dei dati mediante algoritmi di classificazione	11
4.1	K-Fold Cross Validation	11
4.1.1	Learning Curve	11
4.2	Misure di valutazione	12
4.3	K-Nearest Neighbor	13
4.3.1	Cenni teorici	13
4.3.2	Applicazione	13
4.4	Linear Support Vector Machine	17
4.4.1	Cenni teorici	17
4.4.2	Applicazione	18
4.5	Kernel Support Vector Machine	20
4.5.1	Cenni teorici	20
4.5.2	Applicazione	21
4.6	Decision Tree	23
4.6.1	Cenni teorici	23
4.6.2	Applicazione	24
5	Conclusioni	28
6	Complemento PCA con k=1	29
6.1	K-Nearest Neighbor	29
6.2	Linear Support Vector Machine	30
6.3	Kernel Support Vector Machine	31
6.4	Decision Tree	32

1. Introduzione

1.1 Dataset

Questa tesina si pone l'obiettivo di condurre l'analisi di un dataset utilizzando algoritmi di Machine Learning appresi durante il corso di Apprendimento Statistico. Il dataset oggetto dello studio è *Wine Data Set*, consultabile e scaricabile sul sito *UCI Machine Learning Repository* cliccando *qui*.

Il dataset è composto da misurazioni chimiche di vini prodotti in tre diverse regioni dell'Italia. Per ciascuna regione, sono fornite informazioni su 13 attributi chimici: alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, proline. In totale, il dataset contiene 178 istanze, suddivise equamente tra le tre regioni, che rappresentano le 3 classi.

L'importanza dell'analisi risiede in diversi aspetti. Innanzitutto, l'industria vinicola è un settore di grande rilevanza economica e culturale, con un vasto interesse per produttori, esperti di vino e appassionati. L'analisi dei dati può fornire informazioni utili per comprendere le caratteristiche chimiche dei vini e come influiscono sulle loro proprietà *organolettiche*.

Inoltre, è spesso utilizzato come caso di studio per problemi di classificazione e clustering nel campo del machine learning. Gli algoritmi di apprendimento possono essere applicati per predire la regione di provenienza di un vino in base alle sue caratteristiche chimiche, o per raggruppare i vini in base alle loro somiglianze. Questo tipo di analisi può essere utile per supportare decisioni di marketing, valutare la qualità dei vini o identificare anomalie nel processo di produzione.

1.2 Linguaggi e tools

L'intera analisi è stata svolta utilizzando il linguaggio di programmazione Python nelle sue principali librerie per il Machine Learning:

- *Pandas*: permette di leggere dati (come CSV o SQL database) e creare oggetti Python con righe e colonne chiamati Data Frame, molto simili alle tabelle nei software statistici.
- *Numpy*: è il pacchetto fondamentale per il calcolo scientifico con Python. Permette di creare array multidimensionali ad alte prestazioni e fornisce strumenti per lavorare con questi array. Può anche essere usato come un efficiente contenitore multidimensionale di dati generici.
- *Sklearn*: contiene molti algoritmi di apprendimento supervisionato (Decision Tree, Regressione Lineare, SVM, ecc..) e non-supervisionato (PCA, K-means, ecc..) e diversi dataset didattici di addestramento.
- *Matplotlib*: permette la creazione di grafici.
- *Seaborn*: fornisce un'API basata su Matplotlib che offre scelte per stile e colore, definisce semplici funzioni di alto livello per i tipi di grafici statistici comuni e si integra con la funzionalità fornita da Pandas DataFrames.

L'implementazione del codice *Python* è stata sviluppata utilizzando *Google Colab*, mentre la stesura del report è scritta in *Latex* sulla piattaforma *Overleaf*.

2. Esplorazione e visualizzazione dei dati

Al fine di migliorare la comprensione dei dati in analisi, di seguito sono riportate delle informazioni sul contenuto, la quantità e il tipo delle variabili del dataset.

Frammento dei primi 5 samples:

	Label	Alcohol	Malic Acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyanins	Color intensity	Hue	OD280/OD315 of diluted wines	Proline
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

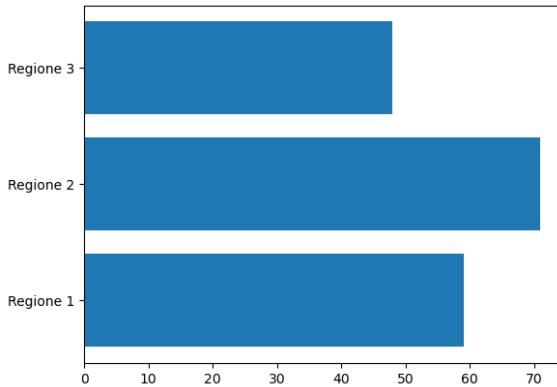
Informazioni generali:

```
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Label            178 non-null    int64  
 1   Alcohol          178 non-null    float64
 2   Malic Acid       178 non-null    float64
 3   Ash              178 non-null    float64
 4   Alcalinity of ash 178 non-null    float64
 5   Magnesium        178 non-null    int64  
 6   Total phenols    178 non-null    float64
 7   Flavanoids        178 non-null    float64
 8   Nonflavanoid phenols 178 non-null    float64
 9   Proanthocyanins  178 non-null    float64
 10  Color intensity  178 non-null    float64
 11  Hue              178 non-null    float64
 12  OD280/OD315 of diluted wines 178 non-null    float64
 13  Proline          178 non-null    int64  
dtypes: float64(11), int64(3)
memory usage: 19.6 KB
```

E' possibile osservare che il dataset è composto da 178 righe e 14 colonne, di cui 13 sono gli attributi chimici e una è la classe di appartenenza. Quest'ultima è espressa come intero su 64bit e assume valori 1, 2 o 3 che corrispondono rispettivamente alle tre regioni d'appartenenza. Rilevante anche il dato che non ci sono valori mancanti per alcuna colonna, poiché tutte hanno 178 righe.

2.1 Istogramma

Il seguente **istogramma** mostra come sono distribuiti i campioni nelle varie classi. Il dataset presenta tre classi distribuite in maniera equa, senza problemi di sbilanciamento. Questo equilibrio garantisce che il modello di machine learning riceva un numero simile di istanze per ogni classe, consentendo un'apprendimento bilanciato. La distribuzione uniforme delle classi è vantaggiosa poiché evita una predominanza di una classe rispetto alle altre, facilitando l'analisi e l'apprendimento statistico. Inoltre, una distribuzione equa permette una valutazione affidabile delle performance del modello, garantendo una rappresentazione adeguata di tutte le classi e una valutazione accurata della sua capacità di generalizzazione.



2.2 Box Plot

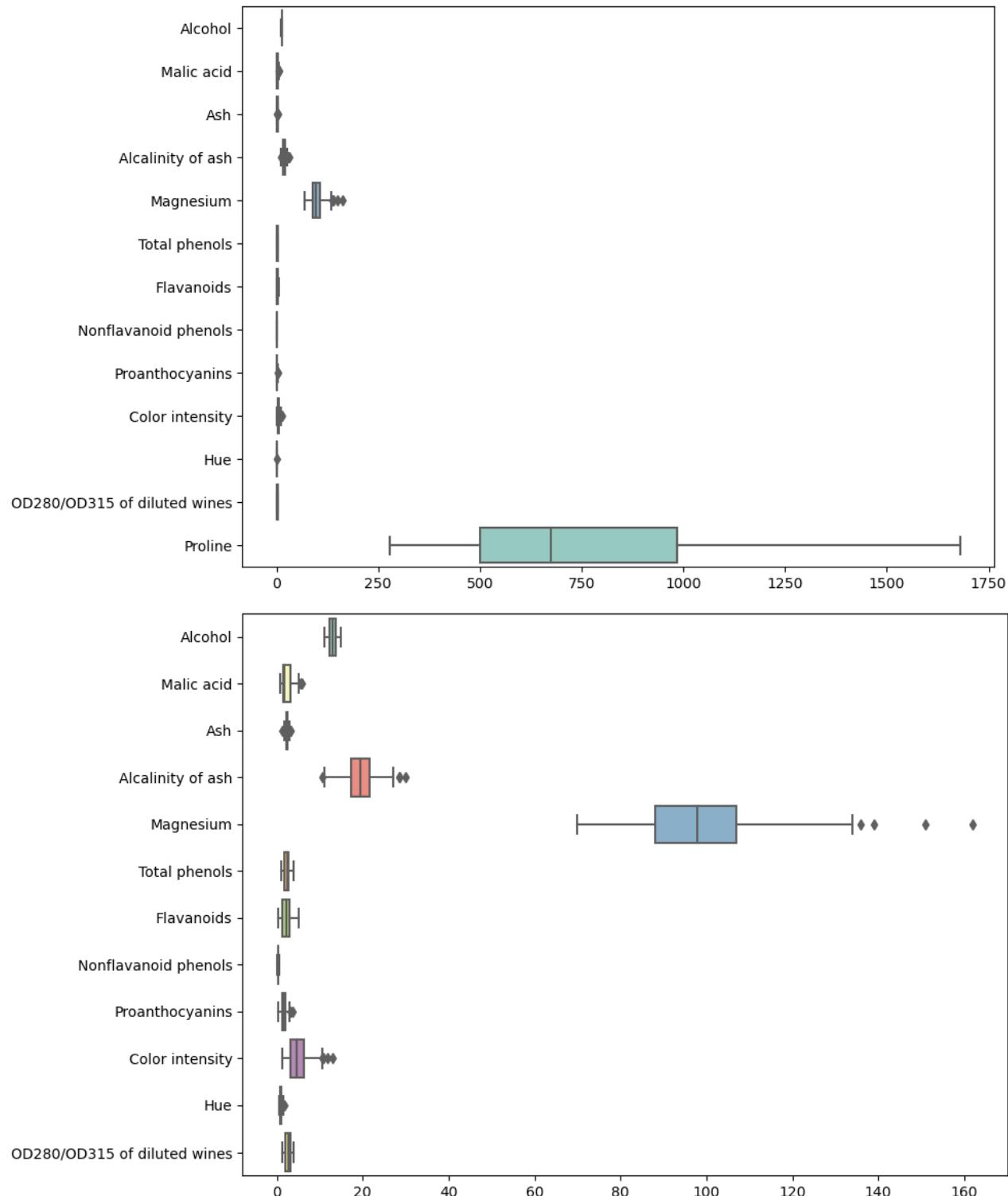
Di seguito alcune statistiche più dettagliate per ogni attributo in esame:

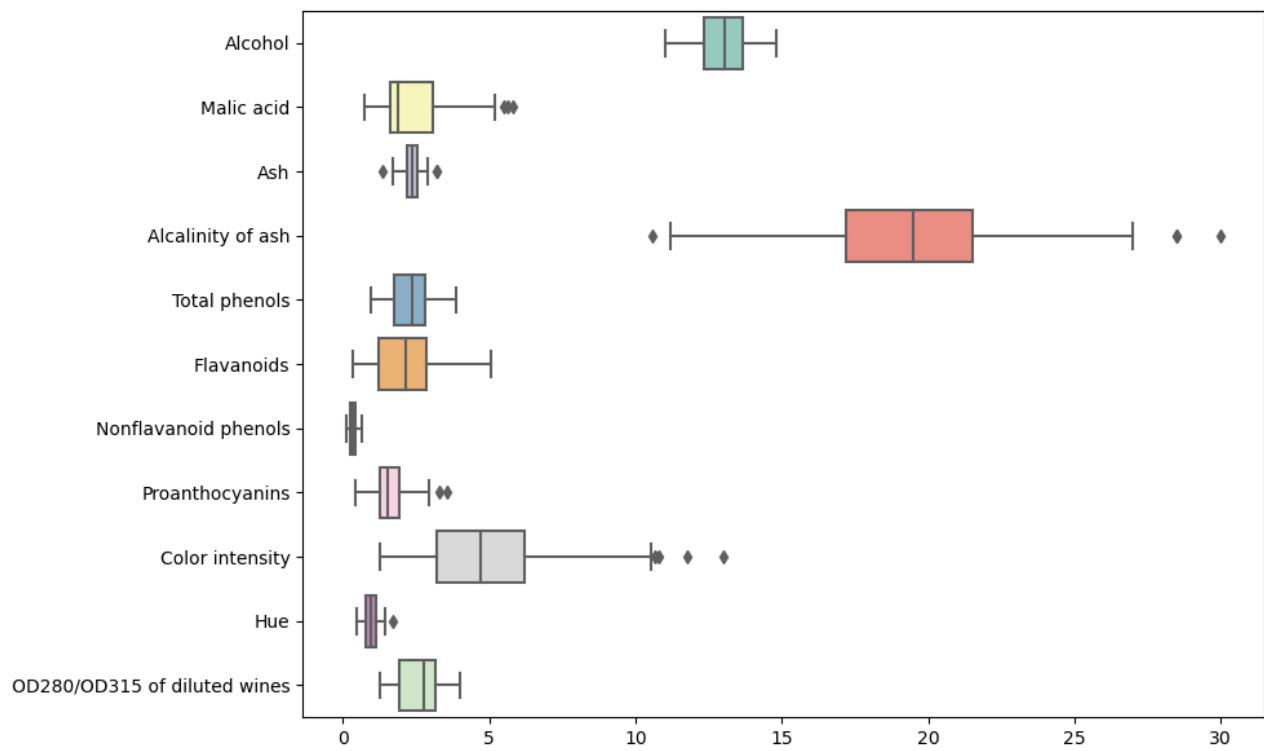
- *count*: il numero di record, cioè il numero di campioni.
- *mean*: il valore medio su tutti i campioni.
- *std*: la deviazione standard, che indica quanto i valori si scostano dal valore medio.
- *min*: il valore minimo su tutti i campioni.
- *25%, 50%, 75%*: i quartili, cioè indici di posizione non centrale. Si ottengono dividendo l'insieme di dati in 4 parti uguali e nello specifico:
 - il *primo quartile* è il valore che lascia alla sua sinistra il 25% degli elementi della distribuzione;
 - il *secondo quartile* coincide con la *mediana* dato che è quello che lascia alla sua sinistra il 50% dei dati della distribuzione;
 - il *terzo quartile* è il valore che lascia il 75% degli elementi a sinistra e il 25% a destra.
- *max*: il valore massimo su tutti i campioni.

	Alcohol	Malic acid	Ash	Alkalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	0.361854	
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	
	Proanthocyanins	Color intensity	Hue	OD280/OD315 of diluted wines		Proline			
count	178.000000	178.000000	178.000000		178.000000	178.000000			
mean	1.590899	5.058090	0.957449		2.611685	746.893258			
std	0.572359	2.318286	0.228572		0.709990	314.907474			
min	0.410000	1.280000	0.480000		1.270000	278.000000			
25%	1.250000	3.220000	0.782500		1.937500	500.500000			
50%	1.555000	4.690000	0.965000		2.780000	673.500000			
75%	1.950000	6.200000	1.120000		3.170000	985.000000			
max	3.580000	13.000000	1.710000		4.000000	1680.000000			

Queste informazioni sono visualizzabili graficamente tramite **box plot**. Il box plot o *diagramma a scatola e baffi*, è un grafico, ottenuto a partire dai 5 numeri di sintesi (*minimo*, *primo quartile* (*Q1*), *mediana*, *terzo quartile* (*Q3*)) e *massimo*).

*quartile (Q3), massimo], che descrive le caratteristiche salienti della distribuzione. Si ottiene riportando su un asse i 5 numeri di sintesi: la scatola del box plot ha come estremi inferiore e superiore rispettivamente Q1 e Q3 (primo e terzo quartile). La mediana divide la scatola in due parti. I baffi si ottengono congiungendo Q1 al minimo e Q3 al massimo. I *baffi* mettono inoltre in evidenza la presenza di eventuali outliers.*





E' possibile osservare che quasi la metà degli attributi presentano un certo numero di outliers, dato da tenere in considerazione nella valutazione degli algoritmi usati per l'analisi. A parte per Alchol e Alcalinity of ash, che risultano essere i più simmetrici, quasi tutti gli attributi hanno una distribuzione asimmetrica e sparsa rispetto al valore medio.

Da notare anche che ci sono attributi con scale differenti anche di un ordine di grandezza, osservando già soltanto il valore medio. Nella terza figura, infatti, sono mostrate più chiaramente le distribuzioni degli attributi con valori molto piccoli indistinguibili nella scala del primo e secondo grafico.

2.3 Matrice di correlazione

Date due variabili statistiche X e Y, l'indice di correlazione di *Pearson* è definito come la loro *covarianza* divisa per il prodotto delle *deviazioni standard* delle due variabili:

$$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$

dove σ_{XY} è la covarianza tra X e Y, e σ_X e σ_Y sono le due deviazioni standard.

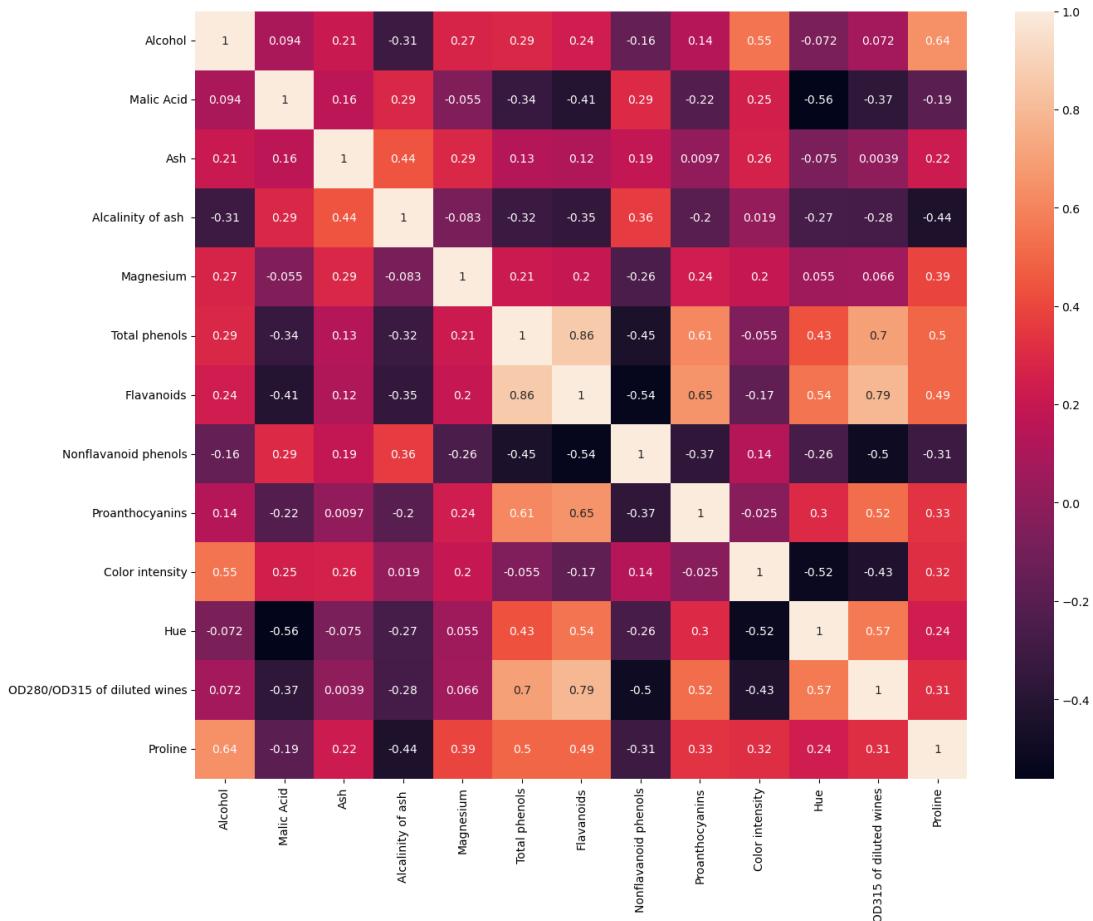
Il coefficiente assume sempre valori compresi tra -1 e 1:

- se $\rho_{XY} > 0$, le variabili X e Y si dicono *direttamente correlate*, oppure correlate positivamente;
- se $\rho_{XY} = 0$, le variabili X e Y si dicono *non correlate*;
- se $\rho_{XY} < 0$, le variabili X e Y si dicono *inversamente correlate*, oppure correlate negativamente.

Inoltre si distinguono i seguenti gradi di correlazione:

- se $0 < |\rho_{XY}| < 0.3$ si ha *correlazione debole* (es. Flavanoids e Magnesium);
- se $0.3 < |\rho_{XY}| < 0.7$ si ha *correlazione moderata* (es. Proline e Alcohol);
- se $|\rho_{XY}| > 0.7$ si ha *correlazione forte* (es. OD280/OD315 of diluted wines e Flavanoids).

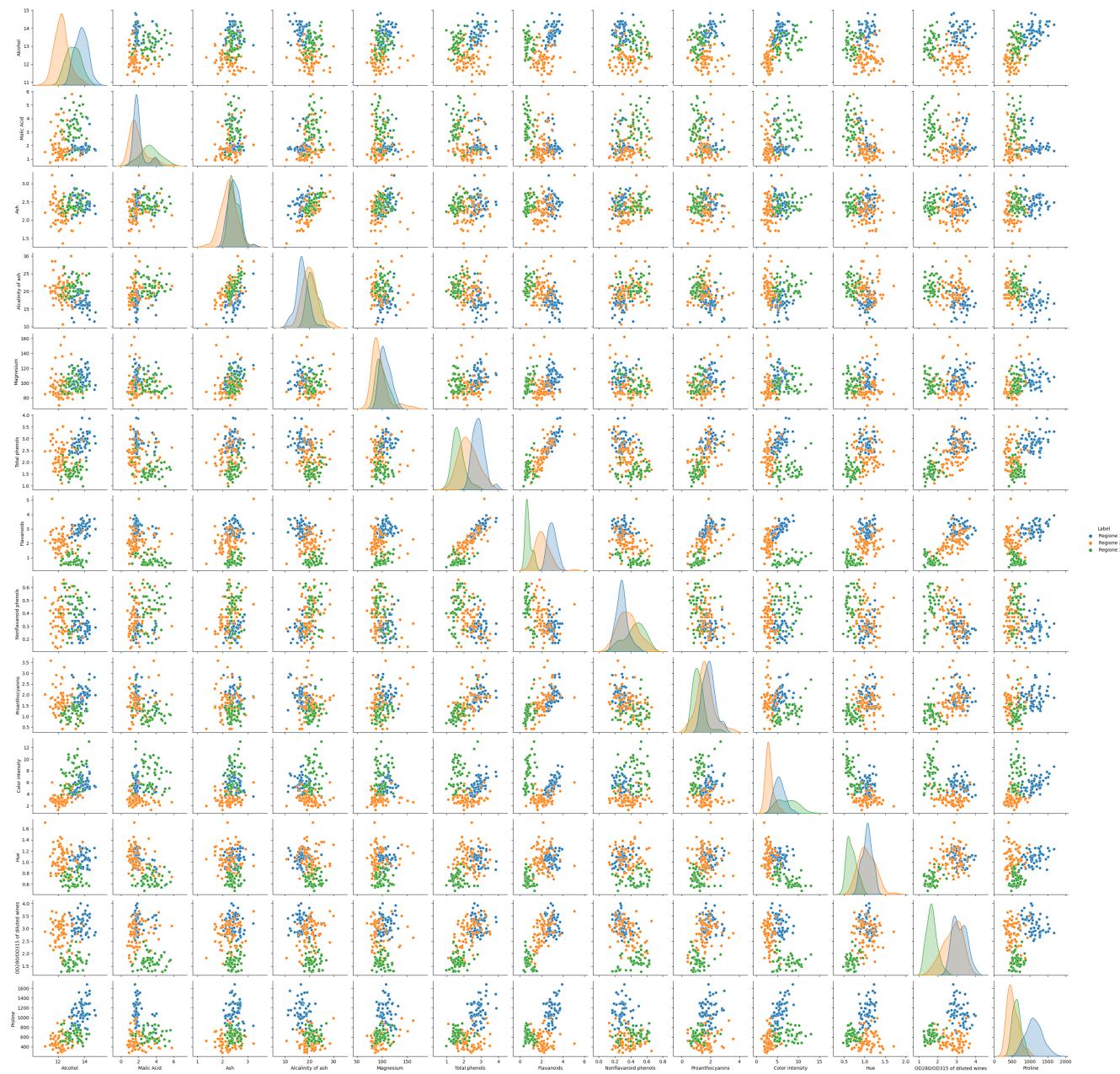
Gli indici di correlazione di n variabili possono essere presentati in una *matrice di correlazione*, che è una matrice quadrata e simmetrica di dimensione $n \times n$ in cui i coefficienti sulla diagonale valgono 1.



2.4 Pair Plot

Il **Pair Plot** permette di osservare sia la distribuzione delle singole variabili sulle tre classi (sulla diagonale) che le relazioni tra tutte le coppie di attributi. Con lo *scatter plot* si può riconoscere se i dati si concentrano attorno a qualche curva. E' possibile notare come per coppie di attributi che hanno una correlazione forte, ad esempio OD280/OD315 of diluted wines e Flavanoids con $\rho_{OD280/OD315, Flavanoids} = 0.79$, il grafico mostra un andamento quasi lineare; questo vuol dire che a un'alta concentrazione di un composto corrisponde un'altrettanto alta concentrazione dell'altro.

Per coppie con correlazione quasi nulla, ad esempio Ash e OD280/OD315 of diluted wines con $\rho_{Ash, OD280/OD315} = 0.0039$, non si riesce a delineare un modello che ne rappresenti l'andamento congiunto poiché le variabili sono appunto quasi completamente scorrelate.



3. Preprocessing

3.1 Training e test set

Il dataset di partenza è stato diviso in training set e test set nel modo seguente:

- Training set: 133 record (75%)
- Test set: 45 record (25%)

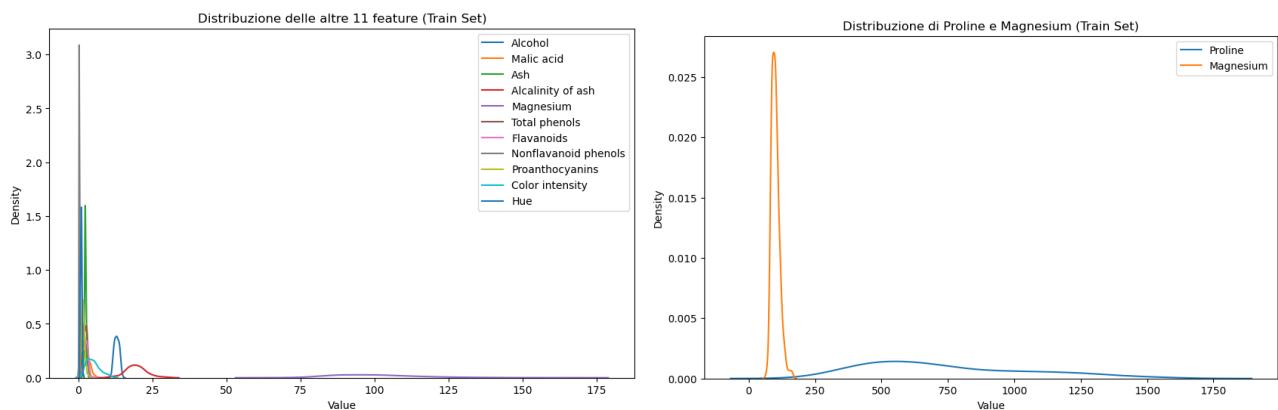
Nella successiva analisi mediante algoritmi di classificazione, per effettuare tuning dei parametri tramite *K-fold cross validation*, una parte del training è usata iterativamente come validation set.

3.2 Standardizzazione

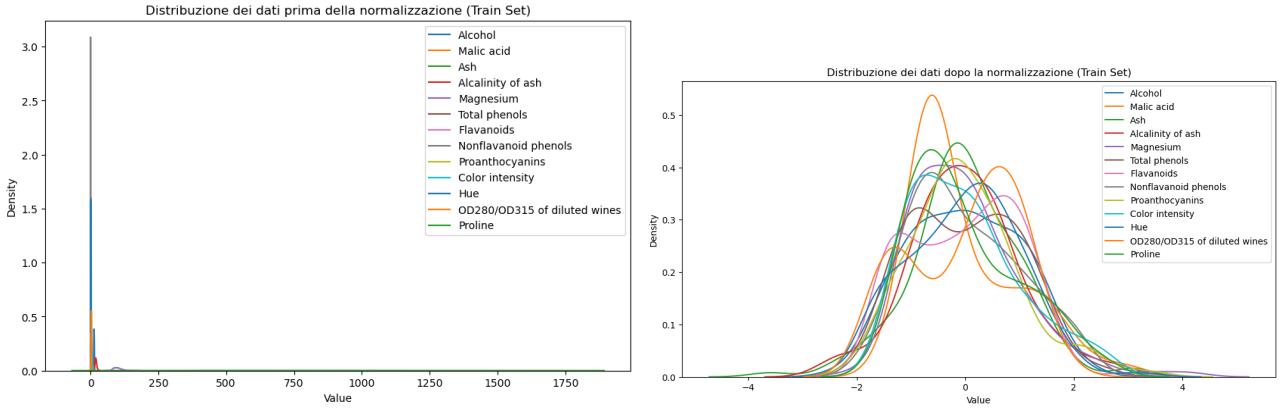
Prima di procedere con l'applicazione degli algoritmi di Machine Learning, tutti i dati sono stati normalizzati in modo da avere media 0 e varianza 1; in questo modo i valori sono riportati sulla stessa scala e quindi gli attributi contribuiscono con lo stesso peso nei calcoli avvenire. Di seguito i grafici che mostrano le distribuzioni di probabilità dei composti chimici del training set prima e dopo aver applicato la trasformazione *Standard Scaler()*. Lo *Standard Score* di ogni valore x è calcolato come:

$$Z = \frac{x - \mu}{\sigma}$$

dove μ e σ sono rispettivamente il valore atteso e la deviazione standard calcolati sul training set. Da puntualizzare il fatto che anche il test set è stato normalizzato con media e deviazione standard del training set.



Le prime due immagini sono state create per visualizzare separatamente le distribuzioni delle feature Proline e Magnesium, e delle rimanenti 11 feature. Questa scelta è stata fatta per facilitare la visualizzazione delle distribuzioni e consentire un'analisi più focalizzata su queste specifiche feature, dato che le scale sono differenti.



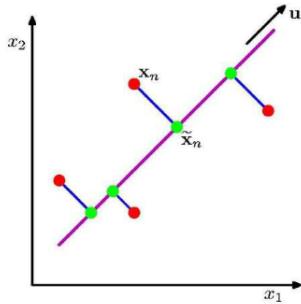
3.3 Riduzione della dimensionalità

La riduzione della dimensionalità riduce il costo computazionale degli algoritmi e può essere utile per evitare un eccesso di adattamento (*overfitting*) sui dati di train, eliminando dal dataset le informazioni ridondanti (correlate). Rappresenta inoltre una soluzione al problema della *curse of dimensionality*: quando la dimensionalità aumenta, il volume dello spazio aumenta e i dati disponibili diventano sparsi e scarsi.

3.3.1 Principal Component Analysis

PCA è un algoritmo di *unsupervised learning*, cioè basato su campioni non classificati, che dati dei punti in uno spazio a d dimensioni, li proietta in uno spazio a dimensionalità più bassa preservando il più alto contenuto informativo possibile. Il nuovo set ridotto di features non correlate è costituito dai cosiddetti *PCA vectors*. Nello specifico, osservando la figura sotto, l'obiettivo è trovare la proiezione ortogonale dei dati che:

- *massimizza la varianza* dei dati proiettati, cioè separa il più possibile i punti verdi (linea viola);
- *minimizza la distanza quadratica media* tra i dati e le proiezioni, cioè tra punti verdi e rossi (linee blu).



I *PCA vectors* hanno origine dal centro di massa e:

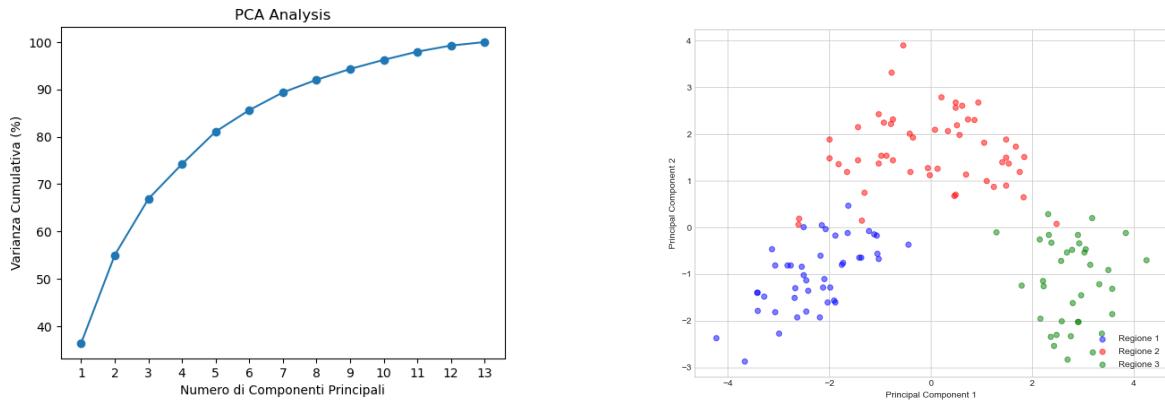
- la prima componente principale punta nella direzione della *massima varianza*;
- ogni componente successiva è *ortogonale* alle precedenti e punta nella direzione della *massima varianza* nello spazio residuo.

Il numero di componenti principali dipende da quanta varianza si riesce a coprire tramite essi.

Si può osservare come le prime due componenti racchiudano circa il 55% della varianza. Si è scelto poi di

selezionare il numero di componenti principali in modo che almeno l'80% della varianza venga preservata. Si è dovuto scegliere quindi $k = 5$, come mostrato nelle figure, dove viene rappresentata la varianza cumulata spiegata e vengono inoltre rappresentati i dati proiettati sulle prime due componenti principali.

Componente 1: 36.40%
 Componente 2: 55.01%
 Componente 3: 66.88%
 Componente 4: 74.25%
 Componente 5: 81.09%
 Componente 6: 85.63%
 Componente 7: 89.37%
 Componente 8: 92.03%
 Componente 9: 94.32%
 Componente 10: 96.24%
 Componente 11: 97.98%
 Componente 12: 99.26%
 Componente 13: 100.00%



Ai fini di un analisi affrofondita, alla conclusione di questa tesina, verrà aggiunto un complemento che includerà i risultati ottenuti applicando la PCA con $k=1$. Questa sezione avrà lo scopo di dimostrare in modo visivo che i risultati raggiunti con una sola componente principale sono in parte già soddisfacenti.

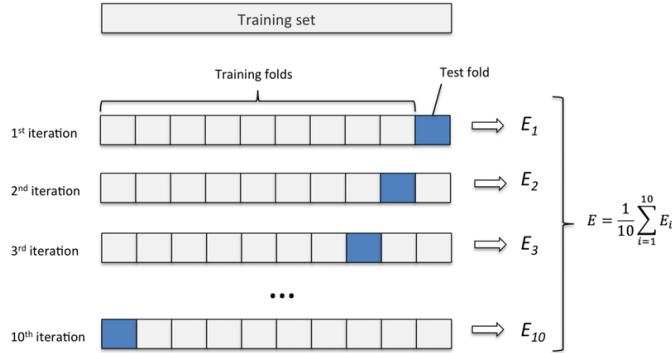
4. Analisi dei dati mediante algoritmi di classificazione

4.1 K-Fold Cross Validation

Su tutti gli algoritmi utilizzati la scelta dei parametri è stata effettuata sulla base di un *k-fold cross validation*. La validazione incrociata è una procedura di ricampionamento utilizzata per valutare i modelli di Machine Learning su un campione di dati limitato. Il fine è quello di stimare il rendimento del modello quando è utilizzato per fare previsioni su dati non utilizzati durante l'addestramento. La procedura ha un singolo parametro chiamato k che si riferisce al numero di gruppi in cui deve essere suddiviso il campione ed è composta dai seguenti step:

- Shuffle del set di dati in modo casuale.
- Divisione del set di dati in k gruppi.
- Per k volte:
 - un gruppo è usato come validation set;
 - i gruppi rimanenti costituiscono il training set per l'allenamento;
 - si addestra il modello sui dati di training e si valuta sul blocco di validation, conservando le accuratezze ottenute.
- Riassunto delle performance del modello sulla base delle accuratezze.

A ciascun campione viene data l'opportunità di essere utilizzato nel set di controllo 1 volta e utilizzato per addestrare il modello $k-1$ volte.



4.1.1 Learning Curve

Per ogni algoritmo è riportata la cosiddetta *Learning Curve* (*curva di apprendimento*) che mostra l'andamento dell'accuratezza del modello di classificazione su training e validation set (con k-fold cross validation) al crescere del numero di campioni di training. Può essere valutata sul set di dati di training per dare un'idea di quanto il modello stia "imparando", mentre la valutazione sul validation set dà un'idea di quanto il modello stia "generalizzando". È inoltre uno strumento per scoprire quanto un modello trae vantaggio dall'aggiunta di più dati di training e se soffre maggiormente di un errore di *varianza* o di un errore di *bias*. Se le accuratezze su training e validation set convergono a un valore troppo basso con l'aumentare delle dimensioni del set di addestramento, non trarranno molti benefici all'aumentare dei dati.

Il dilemma di bias-varianza consiste nel minimizzare simultaneamente queste due fonti di errore che impediscono agli algoritmi di apprendimento supervisionato di generalizzarsi oltre il loro set di addestramento:

- L'errore di *bias* è un errore derivante da ipotesi errate nell'algoritmo di apprendimento. Un alto bias può far perdere a un algoritmo relazioni rilevanti tra le features e l'output target (*underfitting*).
- La *varianza* è un errore dovuto alla sensibilità alle piccole fluttuazioni nel set di allenamento. Una varianza elevata può indurre un algoritmo a modellare il rumore casuale nei dati di allenamento, invece che gli output previsti (*overfitting*).

4.2 Misure di valutazione

Per quantificare le prestazioni del modello durante l'analisi, è generata la *matrice di confusione* multiclasse per ogni algoritmo di classificazione: nelle righe sono indicate le classi effettive, cioè le risposte corrette, mentre le colonne indicano le classi di previsione, cioè le risposte del modello.

		il modello risponde		
		CLASSE 1	CLASSE 2	CLASSE 3
la risposta corretta è	CLASSE 1	TRUE	FALSE errore	FALSE errore
	CLASSE 2	FALSE errore	TRUE	FALSE errore
	CLASSE 3	FALSE errore	FALSE errore	TRUE

A partire da questa matrice, sono calcolate e riportate le seguenti misure di valutazione per ogni classe:

- *Richiamo*: misura la percentuale delle previsioni corrette di una classe sul totale delle istanze di quella classe.

$$\text{Richiamo} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}$$

- *Precisione*: misura la percentuale delle previsioni corrette di una classe sul totale delle previsioni fatte su quella classe (giuste o sbagliate).

$$\text{Precisione} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}$$

- *F-Score*: la media armonica delle metriche *precisione* e *richiamo*.

$$F - \text{Score} = \frac{2 * \text{richiamo} * \text{precisione}}{\text{richiamo} + \text{precisione}}$$

- *Supporto*: è il numero di occorrenze della classe nel set di riferimento.

Sulla totalità delle classi è calcolata l'*accuratezza*, che misura la percentuale delle previsioni esatte sul totale delle istanze.

4.3 K-Nearest Neighbor

4.3.1 Cenni teorici

KNN è un algoritmo di *supervised learning* con lo scopo di predire la classe di una nuova istanza a partire da un insieme di dati già classificati. L'assunzione di base su cui si sviluppa l'algoritmo è che punti simili (o vicini) abbiano la stessa etichetta. Diverse sono le metriche utilizzate per calcolare la distanza o la somiglianza dei punti, una delle più famose è la *distanza euclidea*. Oltre alla metrica, l'algoritmo fissa un parametro k che identifica il numero di punti vicini da considerare. Dato un nuovo punto, il processo di classificazione è il seguente:

- calcola la distanza del nuovo punto rispetto a tutti i punti nel training set;
- identifica i k -nearest neighbors;
- utilizza le label delle classi dei k -nearest neighbor per determinare la classe del record sconosciuto, di solito scegliendo quella che compare con maggiore frequenza.

E' anche possibile pesare il contributo dei vicini in base alla distanza, in modo che non tutti i k abbiano lo stesso peso nella classificazione.

La scelta di k è importante perché se è troppo piccolo l'appuccio è sensibile al rumore, ma se è troppo grande l'intorno può includere esempi appartenenti ad altre classi.

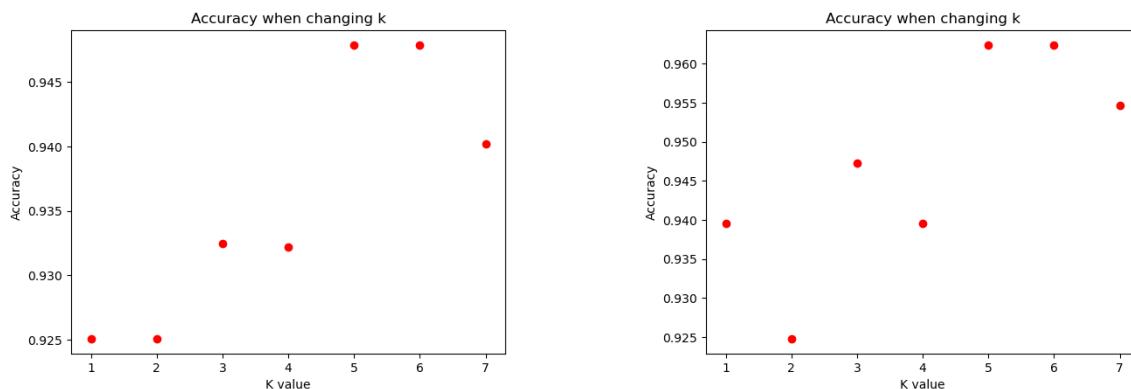
KNN è un algoritmo non parametrico, cioè non fa assunzioni sulla distribuzione dei dati, e non richiede la creazione di un modello (fase di training e di classificazione coincidono). E' necessario che gli attributi abbiano la stessa scala di valori, quindi sono normalizzati in fase di preprocessing.

4.3.2 Applicazione

Per la scelta di k , è stato effettuato un *5-fold cross validation* su un range da 1 a 7 ed è stato utilizzato il valore di k a cui corrisponde accuratezza maggiore.

A confronto i grafici sui dati originali e dopo aver applicato PCA con numero di componenti pari a 5.

Le performance con $k = 1, 2$ sono più basse rispetto al resto, e questo è attribuibile al fatto che il *Nearest Neighbor* è particolarmente sensibile agli outliers e, come osservato nel *box plot*, la metà degli attributi presentano valori anomali e distaccati.



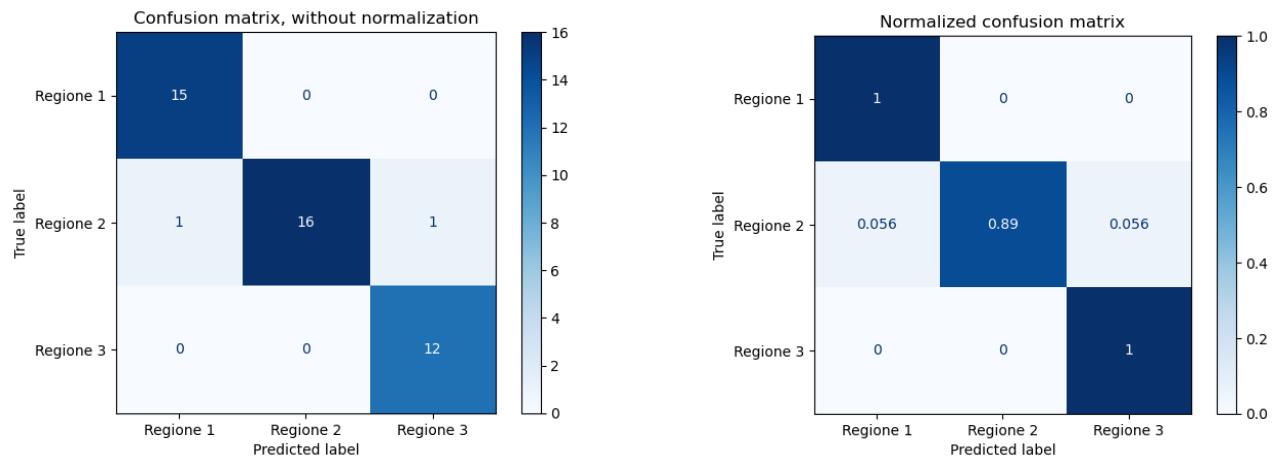
Di seguito un riassunto delle accuratezze ottenute in fase di validation e di test, con e senza PCA.

	Validation	Test	Best k
Senza PCA	94.8%	95.6%	5
Con PCA	96.2%	95.6%	5

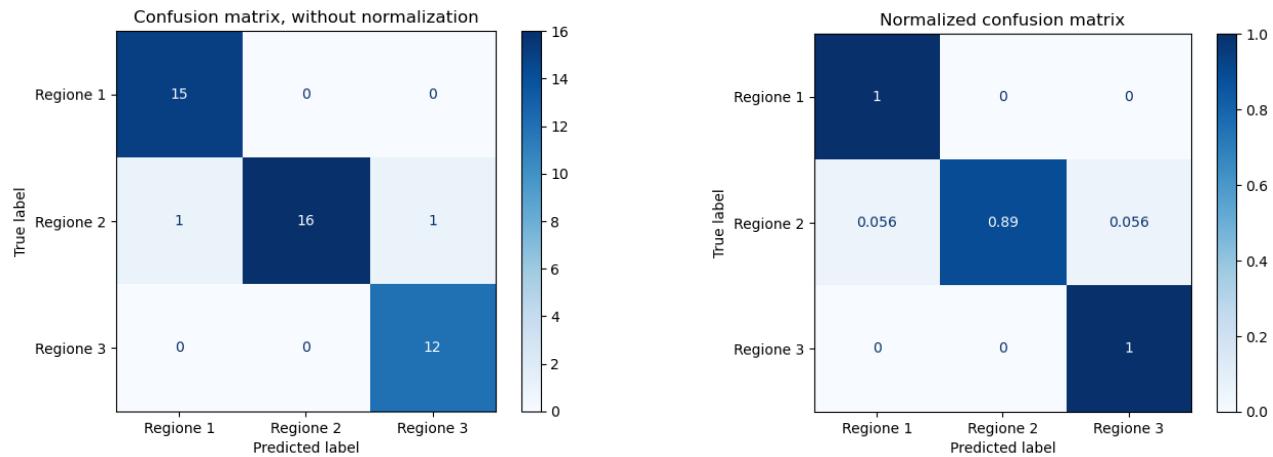
Dalla tabella si nota che con PCA si ottengono prestazioni migliori sul validation set, ma queste risultano equivalenti sul test set. L'accuratezza non è sufficiente per valutare le prestazioni del modello e il confronto tra i dati originali e ridotti nella dimensionalità, quindi di seguito a confronto un'analisi più precisa.

Matrice di confusione

Senza PCA:



Con PCA:



Dal confronto delle matrici di confusione sui dati con e senza riduzione, si nota che non ci sono differenze. Per la *Regione 1*, il modello ha correttamente classificato 15 istanze come appartenenti a quella regione e non sono stati commessi errori nel classificare istanze di altre regioni come appartenenti alla *Regione 1*. Per la *Regione 2*, il modello ha correttamente classificato 16 istanze come appartenenti a quella regione. Tuttavia, sono stati commessi due errori di classificazione. Un'istanza è stata erroneamente classificata come appartenente alla *Regione 1*, mentre un'altra alla *Regione 3*. Per la *Regione 3*, il modello ha correttamente classificato 12 istanze come appartenenti a quella regione e non sono stati commessi errori nel classificare istanze di altre regioni come appartenenti alla *Regione 3*. Nel complesso, il modello sembra avere ottime prestazioni nella *Regione 1* e nella *Regione 3*, poiché non sono stati commessi errori nella classificazione delle istanze, mentre nella *Regione 2* sono stati identificati due errori di classificazione.

Richiamo, Precisione, F-Measure, Supporto

Senza PCA:

	precision	recall	f1-score	support
Regione 1	0.94	1.00	0.97	15
Regione 2	1.00	0.89	0.94	18
Regione 3	0.92	1.00	0.96	12

Con PCA:

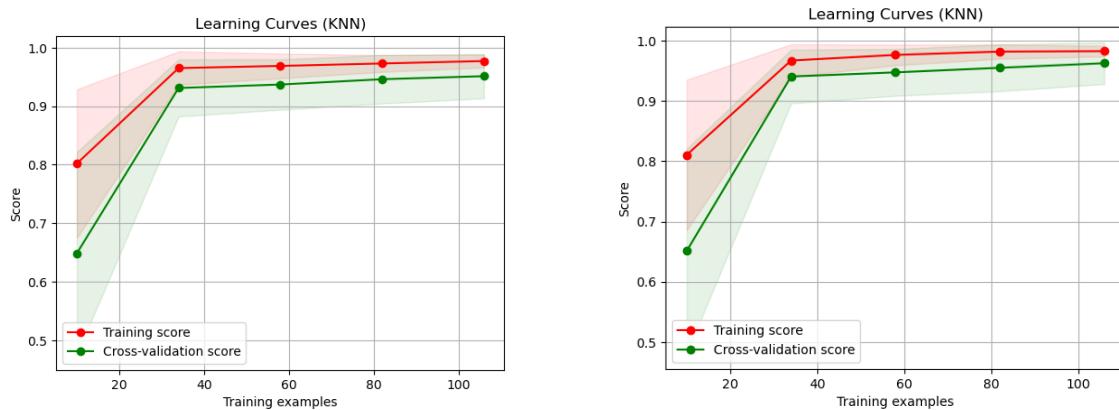
	precision	recall	f1-score	support
Regione 1	0.94	1.00	0.97	15
Regione 2	1.00	0.89	0.94	18
Regione 3	0.92	1.00	0.96	12

Il classification report riconferma quanto detto precedentemente. Per una corretta interpretazione dei dati, un esempio sulla classe *Regione 2*:

- *Precisione*: si ha una precision del 100% significa che tutte le previsioni positive fatte dal modello sono effettivamente corrette.
- *Richiamo*: l'algoritmo classifica correttamente l'89% dei casi in cui le istanze sono effettivamente appartenenti a questa classe.
- *F-score*: valore medio delle due misure precedenti.
- *Supporto*: numero di istanze del test set appartenenti a questa classe.

Complessivamente, il modello ha dimostrato un'ottima performance per tutte e tre le regioni. La *precision* e l'*F-score* sono elevati, indicando che le previsioni positive sono in gran parte corrette. Sebbene la recall della *Regione 2* sia leggermente inferiore, il modello è comunque in grado di effettuare previsioni accurate. In generale, il modello mostra una buona capacità di classificazione per le regioni, bilanciando efficacemente *precision* e *recall*. Pertanto, il modello può essere considerato affidabile nella classificazione delle regioni.

Curva di addestramento

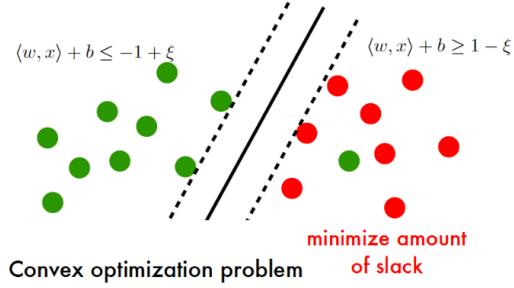


L'analisi ha evidenziato, in entrambi i casi, una convergenza significativa e un gap ridotto tra il *train score* e il *cross-validation score*. Questo indica che il modello ha raggiunto uno stato di stabilità nelle sue prestazioni, apprendendo efficacemente dai dati di addestramento e convalidando correttamente. La convergenza e il gap delle curve suggerisce che il modello è buono e non richiede ulteriori dati di addestramento per migliorare le prestazioni. L'aggiunta di nuovi dati potrebbe non apportare un ulteriore beneficio significativo al modello, poiché ha già raggiunto un livello soddisfacente.

4.4 Linear Support Vector Machine

4.4.1 Cenni teorici

L'SVM è un algoritmo di *supervised learning* che può essere utilizzato sia per scopi di classificazione che di regressione. Ottiene la massima efficacia nei problemi di classificazione binari, ma viene anche utilizzato per problemi di classificazione multiclasse, come nel caso in esame. L'SVM è basato sull'idea di trovare un iperpiano che divida al meglio un set di dati in due classi. Per un SVM multiclasse, viene utilizzato l'approccio *One-vs-One* e *One-vs-All*. Nell'approccio *One-vs-one*, si applica SVM per ogni coppia di classi.



Le linee tratteggiate prendono il nome di *support vectors* (*vettori di supporto*) e passano per i punti più vicini all'iperpiano. Tali punti dipendono dal set di dati che si sta analizzando e se vengono rimossi o modificati alterano la posizione dell'iperpiano divisorio.

E' chiamata *margine* la distanza tra i vettori di supporto di due classi differenti. Alla metà di questa distanza viene tracciato l'iperpiano, o retta nel caso si stia lavorando a due dimensioni.

La funzione di classificazione che definisce l'iperpiano è:

$$f(x) = \sum_i \alpha_i y_i \langle x_i, x \rangle + b$$

L'SVM, a differenza del *Perceptron* ad esempio, risolve un problema di ottimizzazione e restituisce la soluzione che ha margine più grande per migliorare l'accuratezza del modello.

Problema primario di ottimizzazione:

$$\min \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \quad \text{subject to} \quad \begin{cases} y_i [\langle w, x_i \rangle + b] \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

in cui $w = \sum_i y_i \alpha_i x_i$

Dunque, w è definito come la combinazione lineare dei punti x_i , delle rispettive label y_i e dei *moltiplicatori di Lagrange* α_i , utilizzati per la risoluzione del problema di ottimizzazione.

$\sum_i \xi_i$ prendono il nome di *slack variables* e definiscono quanti errori di classificazioni sono permessi, cioè il cosiddetto *gap di tolleranza*. Per questo studio è stata scelta la versione di SVM con le variabili di *slack*, chiamata *Soft Margin SVM*, per gestire il problema degli *outliers* di cui sono affetti i dati.

Il parametro C è il *costo di misclassificazione*: rappresenta quanto si desidera evitare errori nella decisione sull'etichetta. Pertanto, per valori molto piccoli di C , il margine è elevato e il tasso di errori di classificazione è elevato. Quest'ultimo diminuisce quando C aumenta e il margine è più piccolo.

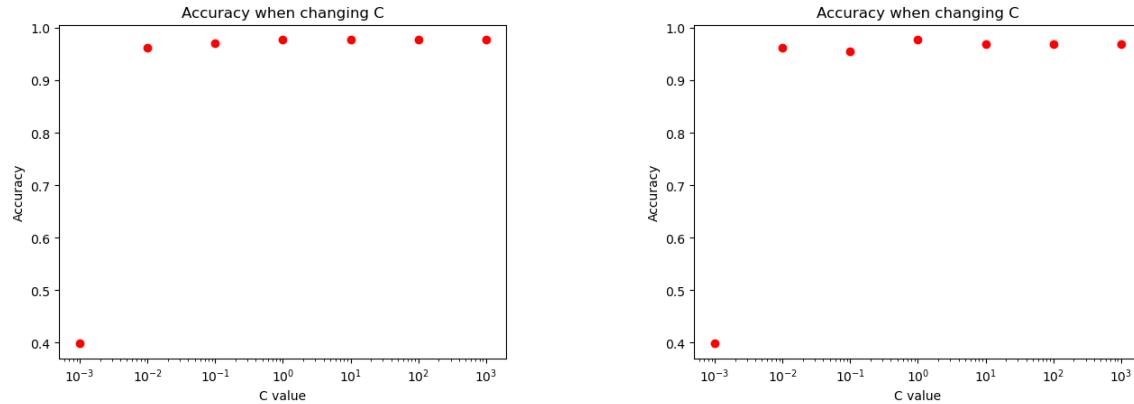
- se $\alpha_i = 0 \Rightarrow y_i [\langle w, x_i \rangle + b] \geq 1$, il punto è lontano dall'iperpiano.
- se $0 < \alpha_i < C \Rightarrow y_i [\langle w, x_i \rangle + b] = 1$, il punto è su un support vector.
- se $\alpha_i = C \Rightarrow y_i [\langle w, x_i \rangle + b] \leq 1$, il punto è all'interno del margine.

Da questo dato è rilevante il fatto che al calcolo del vettore w contribuiscono solo i punti che sono sul margine e gli eventuali errori di classificazione: l'SVM non considera tutti i punti, ma solo quelli più significativi per la classificazione.

4.4.2 Applicazione

Per la scelta del parametro C è stato effettuato un *5-fold cross validation* sull'insieme $C = [0.001, 0.01, 0.1, 1, 10, 100, 1000]$ ed è stata utilizzato il valore a cui corrisponde accuratezza maggiore.

A confronto i grafici sui dati originali e dopo aver applicato PCA con numero di componenti pari a 5.



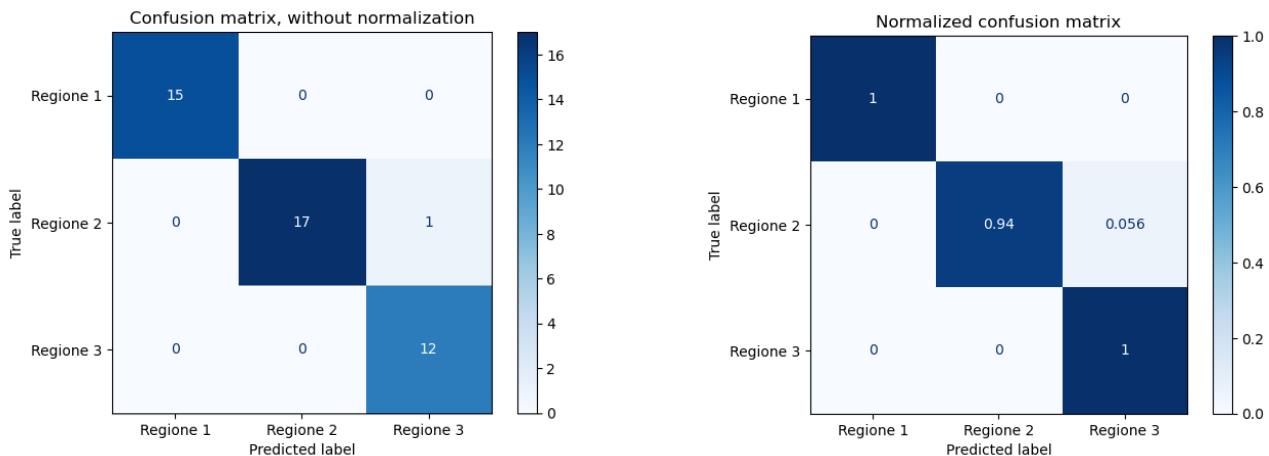
Di seguito un riassunto delle accuratezze ottenute in fase di validation e di test, con e senza PCA.

	Validation	Test	Best C
Senza PCA	97.7%	97.8%	1
Con PCA	97.7%	97.8%	1

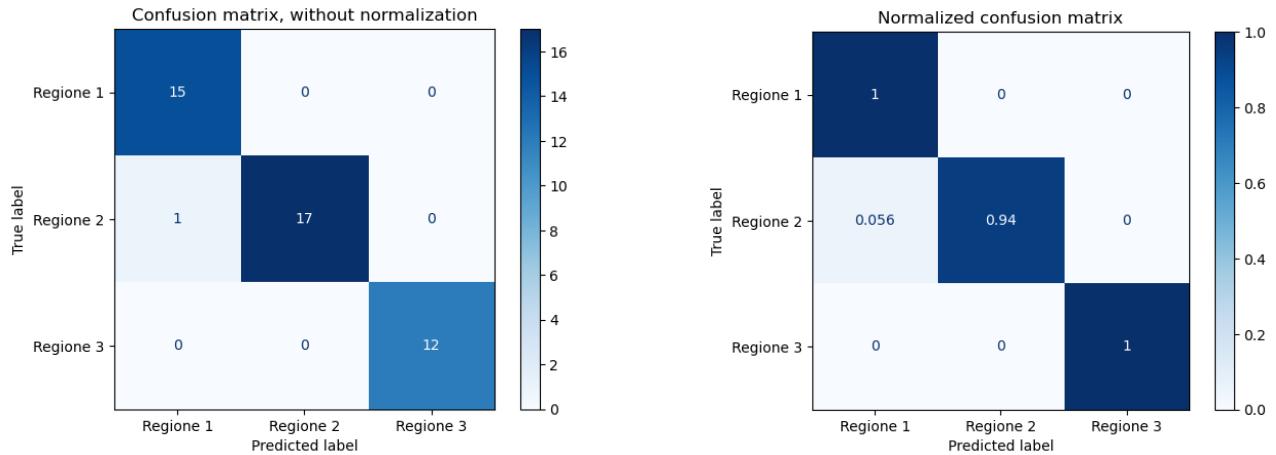
Simile al caso KNN, le accuratezze sul validation set e test set coincidono, sia con che senza PCA, ma non ci danno particolari informazioni sulle performance. Rispetto al KNN, risultano essere di qualche punto percentuale più alte ma solo da un'analisi più dettagliata è possibile stabilire se il modello sia effettivamente migliore.

Matrice di confusione

Senza PCA:



Con PCA:



Possiamo notare che sia con che senza PCA, sia per la *Regione 1* che per la *Regione 3*, tutti i campioni sono classificati correttamente. L'unica differenza riguarda la *Regione 2*, dove in entrambi i casi c'è un'istanza classificata erroneamente, ma invertita tra la *Regione 1* e la *Regione 3*.

Richiamo, Precisione, F-Measure, Supporto

Senza PCA:

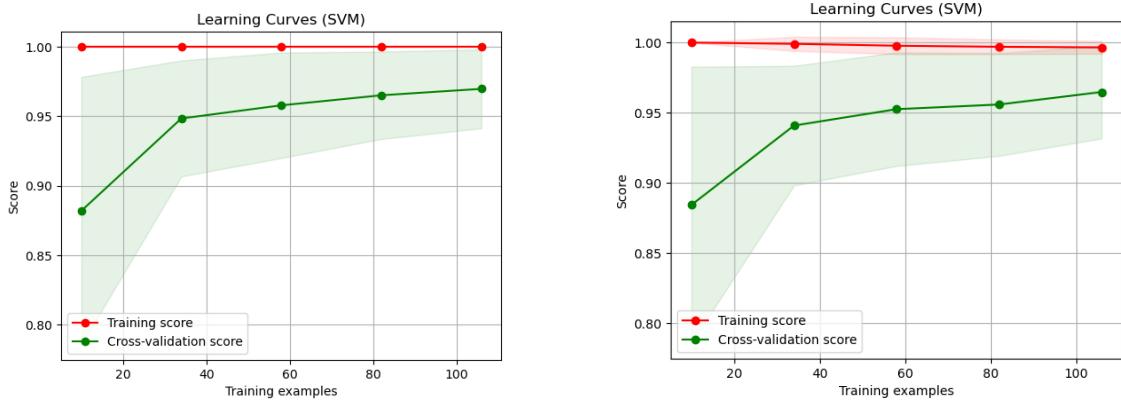
	precision	recall	f1-score	support
Regione 1	1.00	1.00	1.00	15
Regione 2	1.00	0.94	0.97	18
Regione 3	0.92	1.00	0.96	12

Con PCA:

	precision	recall	f1-score	support
Regione 1	0.94	1.00	0.97	15
Regione 2	1.00	0.94	0.97	18
Regione 3	1.00	1.00	1.00	12

Il classification report riconferma quanto detto precedentemente. Così come nel caso di KNN, il modello dimostra un'eccellente capacità di classificazione delle regioni, riuscendo a bilanciare con successo la *precisione* e il *richiamo*. Pertanto, possiamo considerare il modello come affidabile nella sua abilità di classificazione.

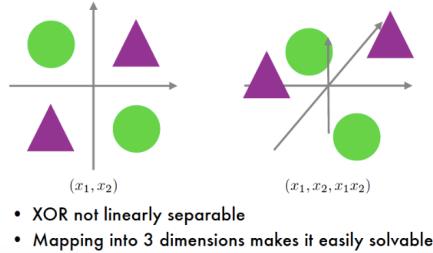
Curva di addestramento Possiamo considerare i due grafici, con e senza PCA, equivalenti. L'analisi rivela una convergenza promettente tra il *training score* e la *cross-validation score*, indicando che il modello sta apprendendo in modo efficace dai dati di addestramento e sta convalidando adeguatamente. Tuttavia, è importante notare che esiste ancora un piccolo gap tra le due curve, suggerendo la possibilità di un ulteriore miglioramento delle prestazioni mediante l'aumento del numero di istanze di addestramento. L'aggiunta di più dati potrebbe consentire al modello di apprendere ulteriori dettagli e di ridurre ulteriormente il gap tra le curve, contribuendo così a una migliore prestazione.



4.5 Kernel Support Vector Machine

4.5.1 Cenni teorici

Come già affrontato sopra, *SVM* lavora bene su dati linearmente separabili, al massimo con una tolleranza di qualche errore di classificazione. Ci sono casi in cui i punti non sono linearmente separabili e l'aggiunta di variabili di *slack* non è sufficiente per risolvere il problema. E' a questo punto che si introduce il concetto di *kernel trick*: l'idea è quella di mappare i punti in uno spazio a più alta dimensionalità in cui sono linearmente separabili. Un tipico esempio è quello dello *XOR problem* nella figura in seguito.



Questo metodo si chiama così per le funzioni kernel, che vengono usate per operare nel "nuovo" spazio senza effettivamente calcolare le coordinate dei dati nello spazio, ma solo calcolando il prodotto scalare tra le immagini di tutte le copie di dati nello spazio di partenza. Questa operazione è spesso computazionalmente più economica che l'esplicito calcolo delle coordinate.

Le funzioni di kernel sono diverse, ma nel caso in analisi è stato scelto un RBF Kernel:

$$k(x, x') = \exp(-\lambda \|x - x'\|^2)$$

Il problema dell'SVM lineare, diventa:

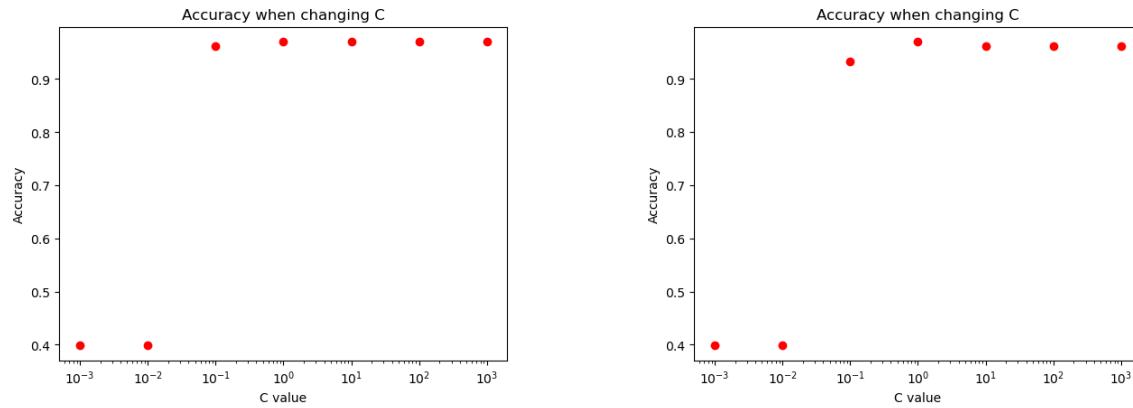
$$\min \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \quad \text{subject to} \begin{cases} y_i [\langle w, \phi(x_i) \rangle + b] \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

Con la seguente funzione di classificazione:

$$f(x) = \sum_i \alpha_i y_i k(x_i, x) + b$$

4.5.2 Applicazione

Come per il *Linear SVM*, la scelta del parametro C è stata fatta sulla base di un *5-fold cross validation* sull'insieme $C = [0.001, 0.01, 0.1, 1, 10, 100, 1000]$ ed è stata utilizzato il valore a cui corrisponde accuratezza maggiore. A confronto i grafici sui dati originali e dopo aver applicato PCA.



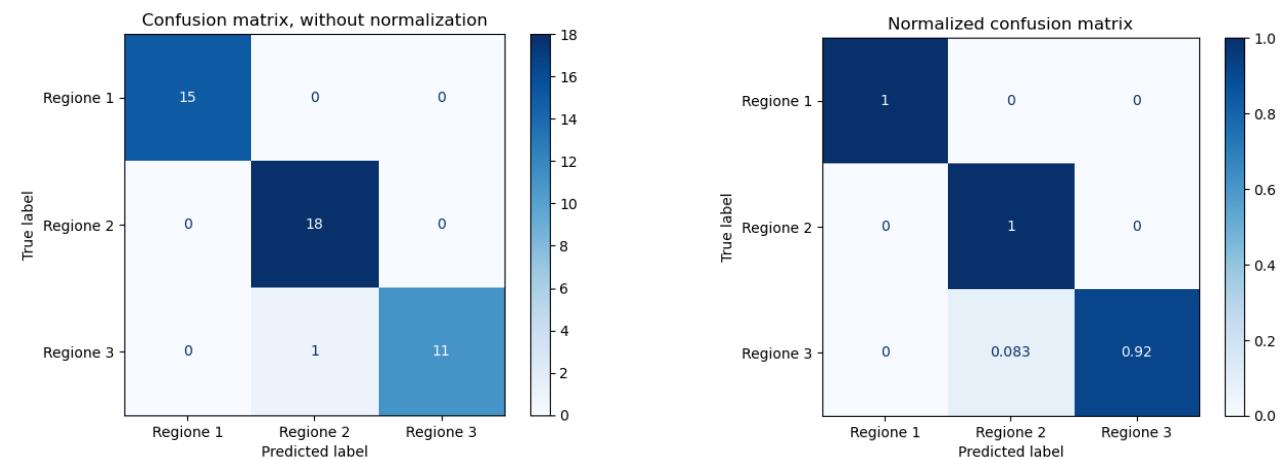
Di seguito un riassunto delle accuratezze ottenute in fase di validation e di test, con e senza PCA.

	Validation	Test	Best C
Senza PCA	97.0%	97.8%	1
Con PCA	97.0%	100%	1

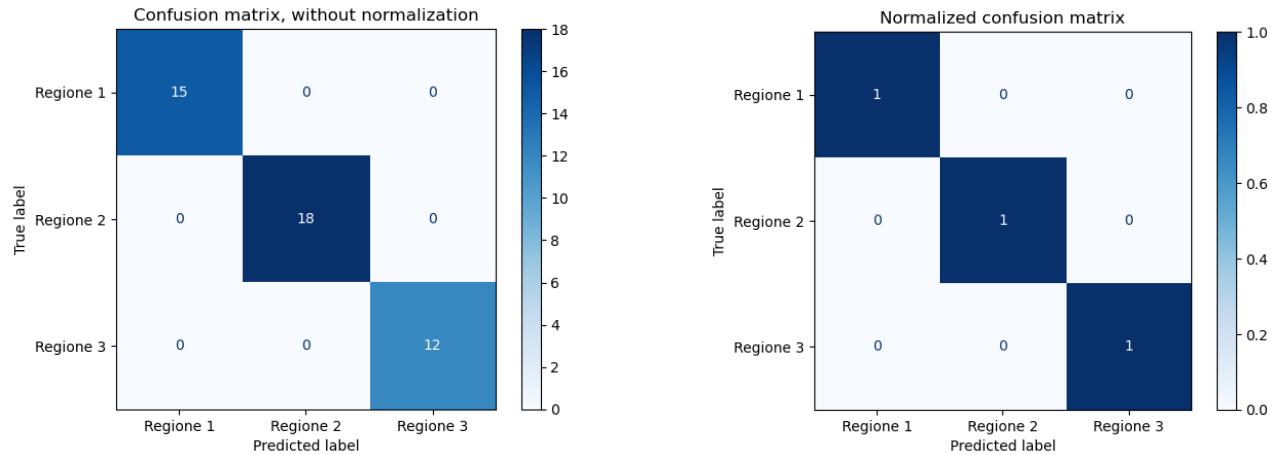
Come per gli algoritmi precedenti, l'accuratezza non fornisce differenze significative tra i due casi e dettagli sulle previsioni corrette e sbagliate per classe. Quello che è possibile notare è che è in linea con quella raggiunta con *SVM lineare*, tranne per quanto riguarda il test con pca (100%).

Matrice di confusione

Senza PCA:



Con PCA:



Senza l'utilizzo di PCA, la matrice presenta un unico errore di classificazione nella *Regione 3*, in cui una istanza è stata erroneamente assegnata alla *Regione 2*. Al contrario, con l'utilizzo di PCA, la matrice mostra una classificazione corretta per tutte le istanze delle 3 *classi*. Entrambe le matrici confermano una buona capacità di classificazione, sia nel caso con PCA che senza.

Richiamo, Precisione, F-Measure, Supporto

Senza PCA:

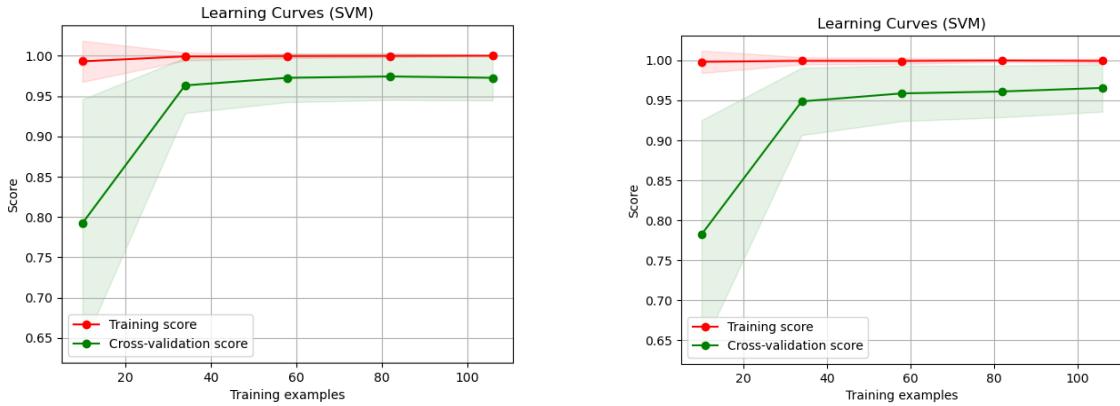
	precision	recall	f1-score	support
Regione 1	1.00	1.00	1.00	15
Regione 2	0.95	1.00	0.97	18
Regione 3	1.00	0.92	0.96	12

Con PCA:

	precision	recall	f1-score	support
Regione 1	1.00	1.00	1.00	15
Regione 2	1.00	1.00	1.00	18
Regione 3	1.00	1.00	1.00	12

I risultati ottenuti dal classification report confermano quanto precedentemente osservato, ovvero che non vi è alcuna differenza significativa tra l'utilizzo di *SVM lineare* e altri *kernel SVM*. Ciò può essere attribuito al fatto che il modello di classificazione SVM lineare iniziale era già molto efficace per il problema in questione. Pertanto, l'assenza di miglioramenti sostanziali con altri *kernel SVM* può essere considerata una diretta conseguenza dell'efficacia del modello *SVM lineare* precedentemente sviluppato.

Curva di addestramento Ci ritroviamo nello stesso caso di KNN, infatti, entrambi i modelli mostrano una convergenza significativa e un gap ridotto tra il training score e il cross-validation score. Ciò indica che entrambi sono buoni, apprendono efficacemente e convalidano correttamente, suggerendo che l'aggiunta di ulteriori dati potrebbe non portare a un miglioramento significativo delle prestazioni.



4.6 Decision Tree

4.6.1 Cenni teorici

Un **albero di decisione** è un metodo molto semplice ed efficace per fare classificazione e regressione. Ogni nodo interno è associato ad una particolare “domanda” su una caratteristica. Dal nodo dipartono tanti archi quanti sono i possibili valori che la feature può assumere, fino a raggiungere le foglie che indicano la categoria associata alla decisione. Una buona “domanda” divide i campioni di classi eterogenee in dei sottoinsiemi con etichette abbastanza omogenee. Per permettere questo è necessario definire una metrica che misuri questa *impurità*. La metrica usata per l’analisi è il *GINI index*. Il *GINI* al nodo t è definito come:

$$GINI(t) = 1 - \sum_j [p(j|t)]^2$$

dove $p(j|t)$ è la frequenza relativa della classe j al nodo t .

- Il valore *massimo*, che corrisponde al più alto grado di impurità è: $1 - \frac{1}{n_c}$, in cui n_c è il numero di classi in uscita.
- Il valore *minimo*, che corrisponde al minimo valore di impurità, è 0.

Per valori bassi di GINI il nodo tende ad essere più "puro", cioè i samples tendono ad essere classificati su un solo ramo (informazione più interessante), mentre per valori più alti c’è disordine maggiore e i record tendono ad essere distribuiti sui diversi rami figli.

Data la metrica che misuri l’impurità, il guadagno $GINI_{split}$ è un criterio che può essere usato per determinare la bontà della divisione:

$$GINI_{split} = \sum_{i=1}^k \frac{n_i}{n} GINI(t)$$

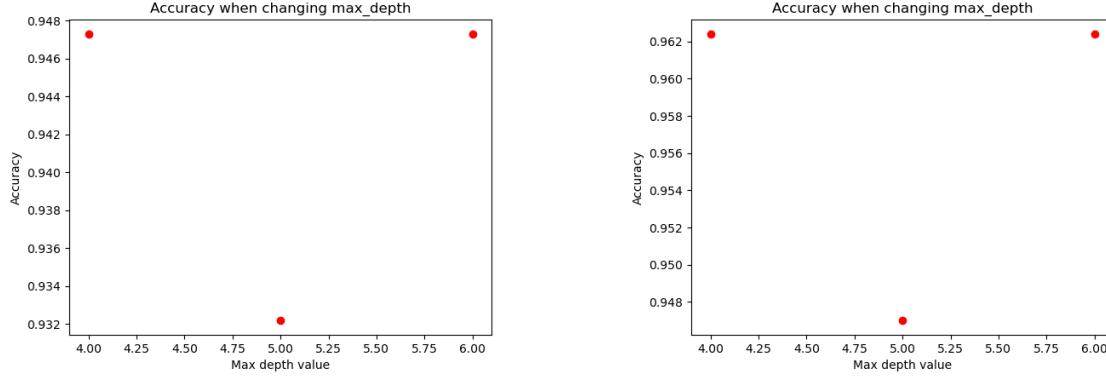
dove:

- n è il numero di record in ingresso al nodo p su cui si sta calcolando il $GINI_{split}$.
- n_i è il numero di record appartenenti al ramo figlio i .

Un vantaggio importante di questo algoritmo è l’*interpretabilità*: a differenza di altri, infatti, seguendo l’albero di decisione è possibile capire perché è stata scelta una determinata classe.

4.6.2 Applicazione

E' stato effettuato un *5-fold cross validation* per trovare il valore migliore di *max depth*. Sulla sinistra il grafico sui dati originali, sulla destra su quelli con dimensionalità ridotta a 5 dal PCA.

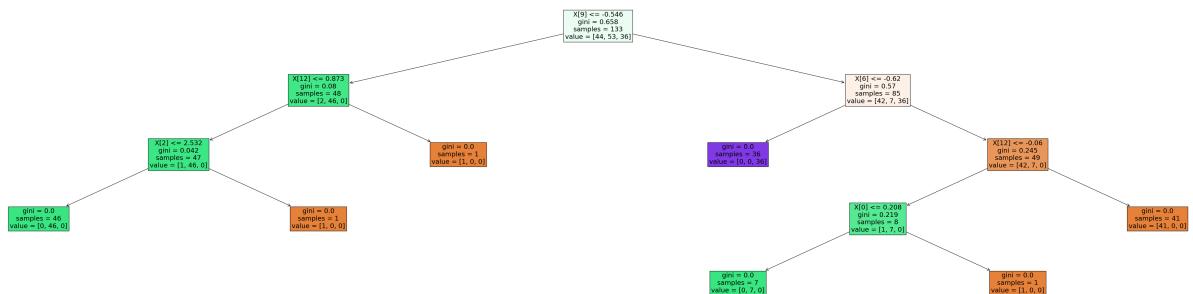


Di seguito a confronto i due alberi di decisione generati a partire dai 133 campioni di training. Per ogni nodo sono fornite le seguenti informazioni:

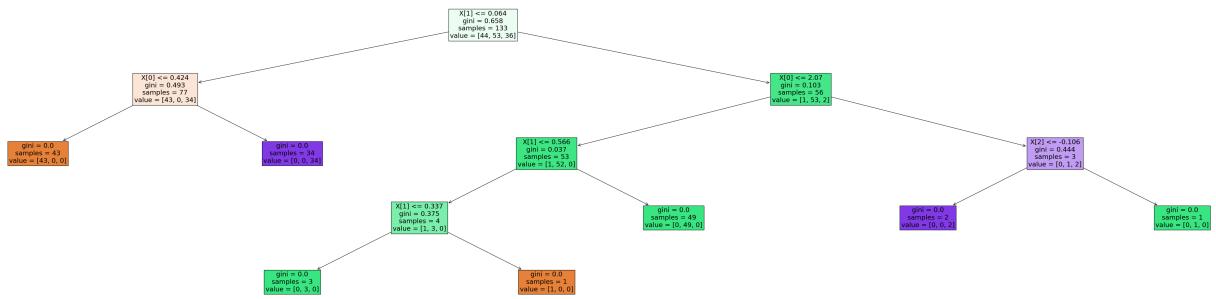
- Criterio di suddivisione, cioè la "domanda" che ci si pone.
- *GINI*: grado di impurità misurato con GINI.
- *Samples*: numero di samples in input nel nodo.
- *Value*: numero di samples appartenenti a ciascuna delle tre classi.

Nel grafico l'intensità del colore mostra il grado di impurità: maggiore è l'intensità, minore è il *GINI index* sul nodo (il nodo è quindi più "puro").

Senza PCA:



Con PCA:



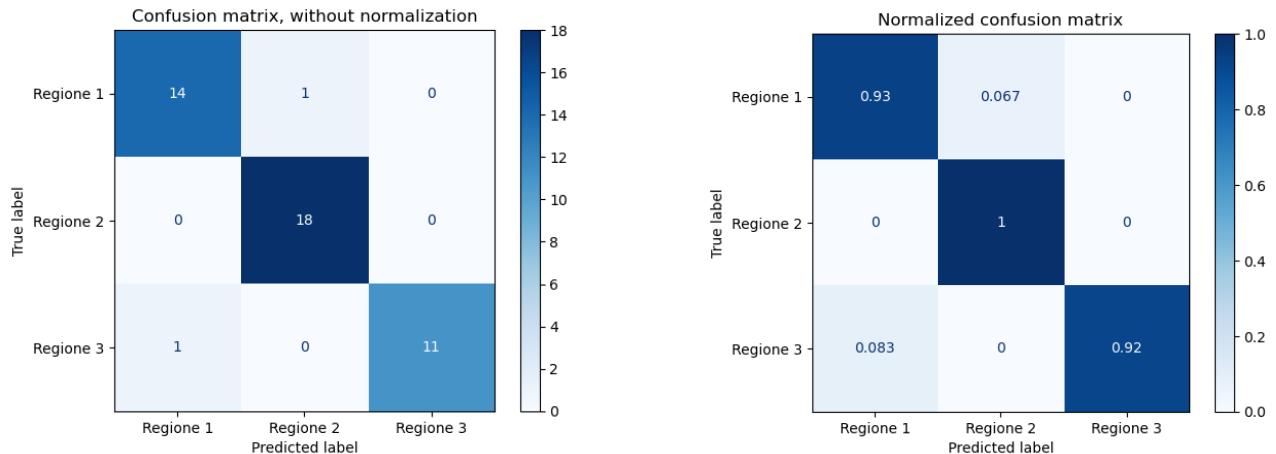
Di seguito un riassunto delle accuratezze ottenute in fase di validation e di test, con e senza PCA.

	Validation	Test	Max depth
Senza PCA	94.7%	95.6%	4
Con PCA	96.2%	95.6%	4

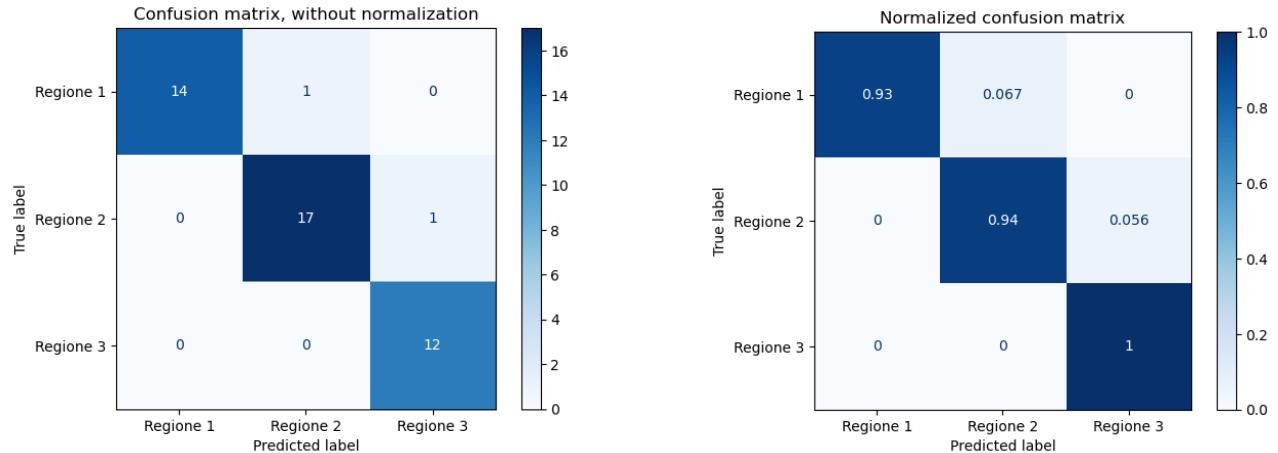
Si nota che con l'albero di decisione si raggiunge un'accuratezza identica al caso KNN. L'accuratezza sul validation set con PCA risulta essere maggiore al caso senza, ma come già visto sopra, questa misura da sola non fornisce abbastanza informazioni sulle performance.

Matrice di confusione

Senza PCA:



Con PCA:



La principale differenza tra le due matrici di confusione è che con l'utilizzo di PCA, la *Regione 2* presenta una riduzione nella classificazione corretta e un aumento degli errori di classificazione. D'altra parte, nella *Regione 3* si osserva un miglioramento nella classificazione corretta con l'utilizzo di PCA, eliminando gli errori di classificazione presenti nella matrice senza PCA. La *Regione 1* non mostra differenze nella classificazione corretta tra le due matrici. Complessivamente, l'introduzione di PCA ha un impatto variabile e ininfluente sulle prestazioni di classificazione per le diverse regioni.

Richiamo, Precisione, F-Measure, Supporto

Senza PCA:

	precision	recall	f1-score	support
Regione 1	0.93	0.93	0.93	15
Regione 2	0.95	1.00	0.97	18
Regione 3	1.00	0.92	0.96	12

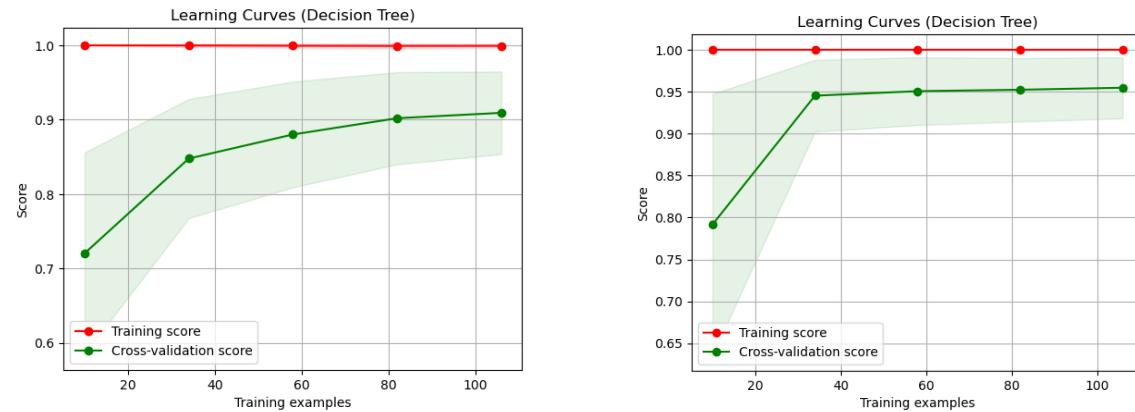
Con PCA:

	precision	recall	f1-score	support
Regione 1	1.00	0.93	0.97	15
Regione 2	0.94	0.94	0.94	18
Regione 3	0.92	1.00	0.96	12

Con PCA, la *precisione* migliora nella *Regione 1*, raggiungendo il valore massimo (1.00), mentre nella *Regione 2* e *3* si osservano leggere diminuzioni. Il *recall* nella *Regione 2* diminuisce con PCA, mentre nella *Regione 3* aumenta al massimo valore (1.00). L'*F1-score* migliora nella *Regione 1* con PCA, mentre rimane stabile nella *Regione 2* e *3*.

Curva di addestramento Nel caso in cui l'analisi sia stata condotta senza l'utilizzo di PCA, è emerso un gap tra il *training score* e il *cross-validation score*. Tuttavia, è importante notare che il *cross-validation score* è crescente, suggerendo che potrebbe essere vantaggioso aumentare la quantità dei dati di addestramento per migliorare ulteriormente le prestazioni del modello. D'altra parte, nel caso in cui sia stata applicata la PCA, si

è osservata una convergenza tra le due curve, indicando che il modello ha raggiunto un livello di performance ottimale. In questa situazione, l'aggiunta di ulteriori dati potrebbe non apportare un miglioramento significativo poiché il modello è già in grado di apprendere in modo efficace dai dati esistenti.



5. Conclusioni

Dal confronto sui quattro algoritmi utilizzati emerge che tutti hanno mostrato prestazioni promettenti. In particolare, sia KNN che Decision Tree hanno ottenuto un'accuratezza del 95,6% sia senza che con PCA, mentre Linear SVM e Kernel SVM hanno mostrato un'accuratezza del 97,8% senza PCA e si differenziano nel caso con PCA, dove il secondo raggiunge un'accuratezza del 100%. Questi risultati indicano che tutti gli algoritmi sono stati in grado di apprendere efficacemente i pattern presenti nel dataset.

Inoltre, l'uso della PCA ha portato a un aumento dell'accuratezza nel caso Kernel SVM, mentre negli altri casi le accuratezze sono identiche al caso senza PCA.

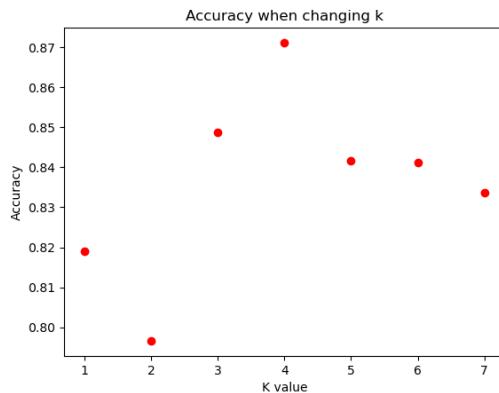
Questo può essere attribuito al fenomeno noto come *curse of dimensionality*. La PCA riduce la dimensionalità del dataset, eliminando le correlazioni tra le variabili e mantenendo al contempo la maggior parte delle informazioni rilevanti. Questo può aiutare gli algoritmi a migliorare la loro capacità di generalizzazione, evitando il sovradattamento (*overfitting*) ai dati di addestramento, soprattutto quando si affrontano problemi di dimensionalità elevata.

Un fattore che contribuisce all'ottima performance degli algoritmi è il bilanciamento delle tre classi nel dataset. Inoltre, l'analisi della matrice di correlazione, evidenzia la presenza di una correlazione moderata tra la maggior parte degli attributi e, nel pair plot, è possibile notare come nello spazio, nella maggior parte dei casi, si formino dei raggruppamenti quasi ben definiti per ciascuna classe.

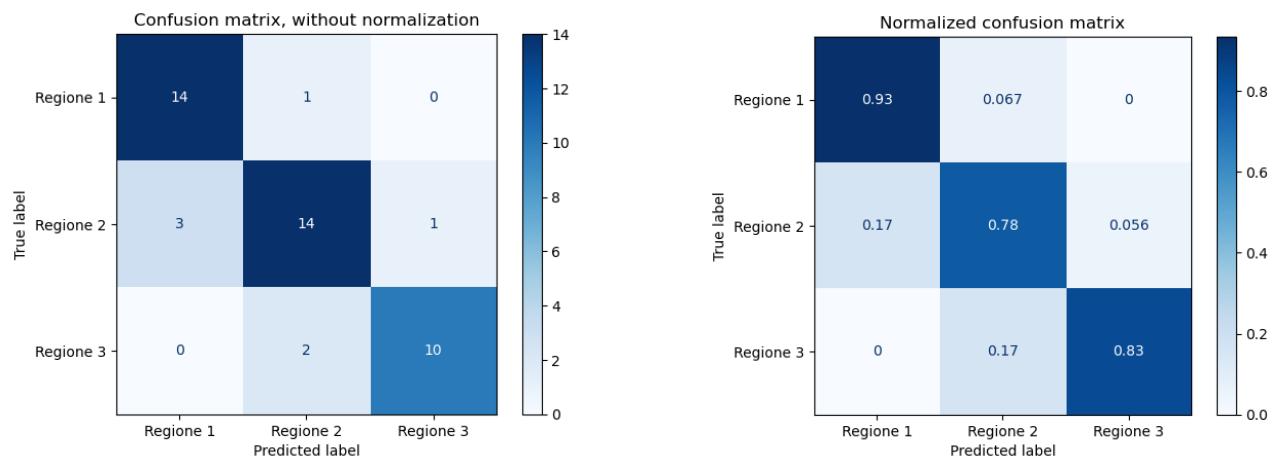
	Senza PCA	Con PCA
KNN	95.6%	95.6%
Linear SVM	97.8%	97.8%
Kernel SVM	97.8%	100%
Decision Tree	95.6%	95.6%

6. Complemento PCA con k=1

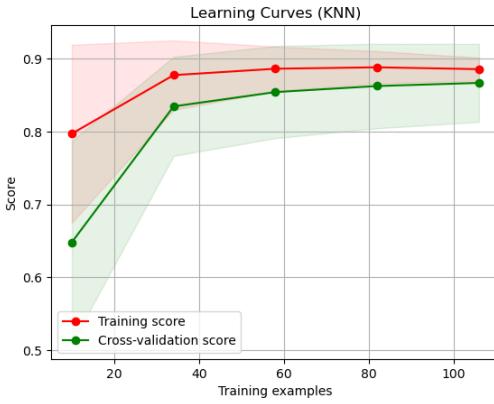
6.1 K-Nearest Neighbor



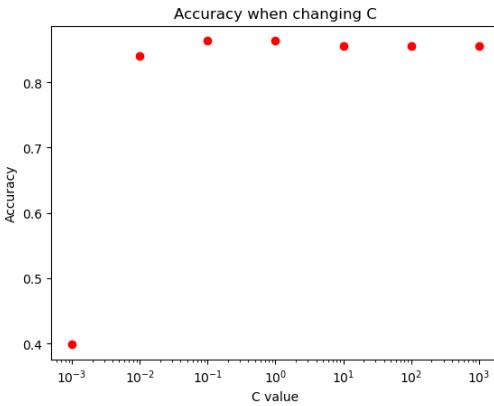
	Validation	Test	Best k
Con PCA (k=1)	87.1%	84.4%	4



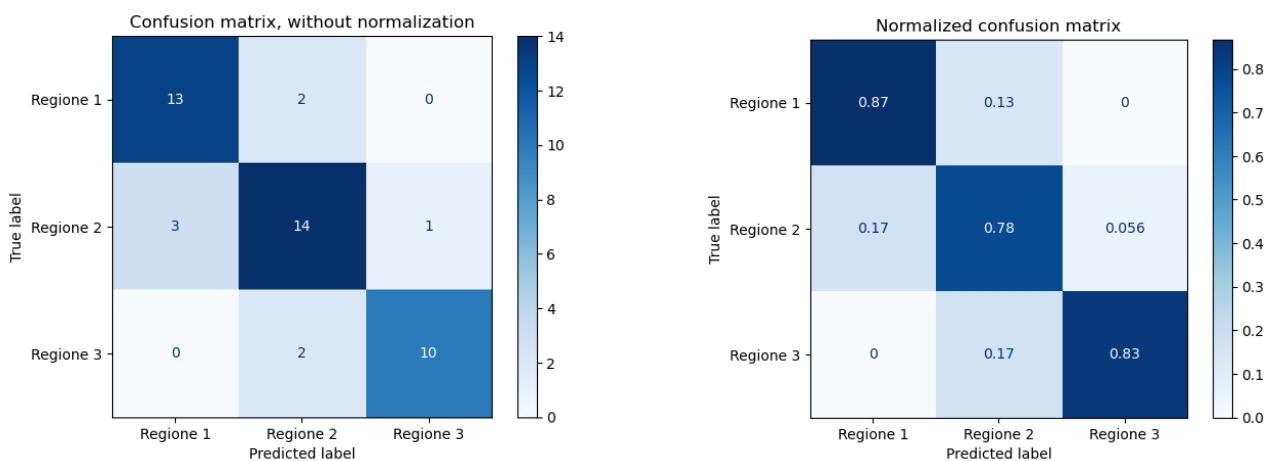
	precision	recall	f1-score	support
Regione 1	0.82	0.93	0.87	15
Regione 2	0.82	0.78	0.80	18
Regione 3	0.91	0.83	0.87	12



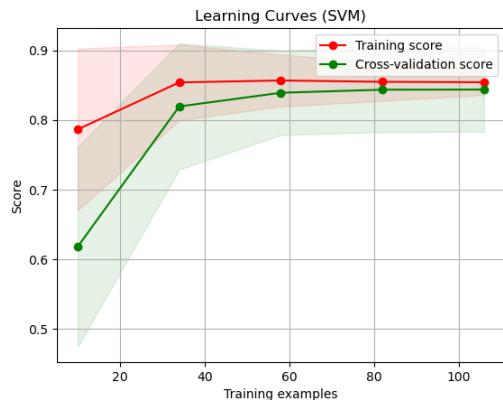
6.2 Linear Support Vector Machine



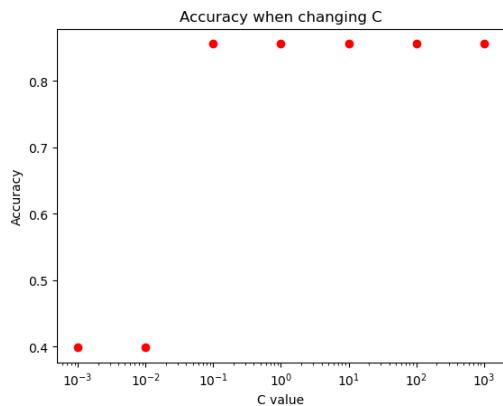
	Validation	Test	Best C
Con PCA (k=1)	86.4%	82.2%	10^{-1}



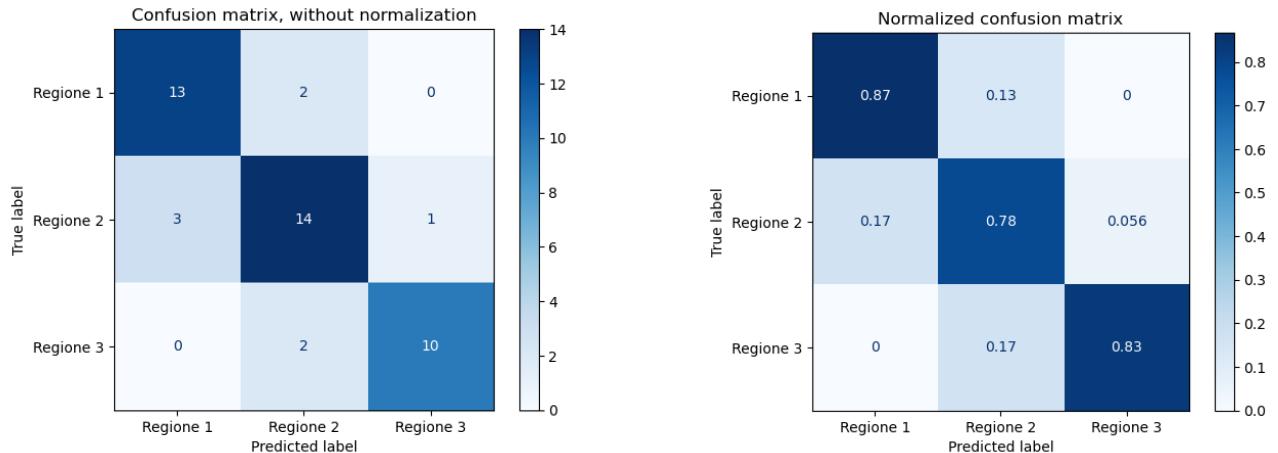
	precision	recall	f1-score	support
Regione 1	0.81	0.87	0.84	15
Regione 2	0.78	0.78	0.78	18
Regione 3	0.91	0.83	0.87	12



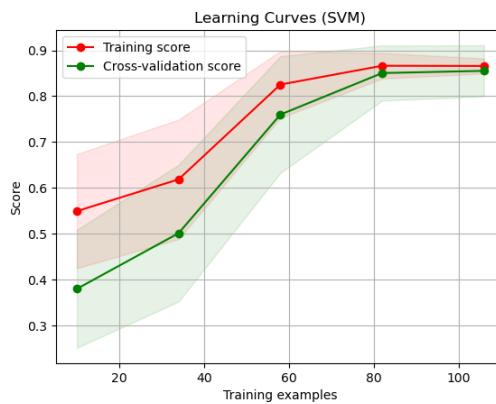
6.3 Kernel Support Vector Machine



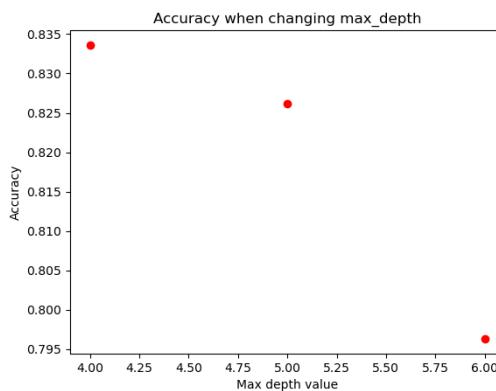
	Validation	Test	Best C
Con PCA (k=1)	85.6%	82.2%	10^{-1}

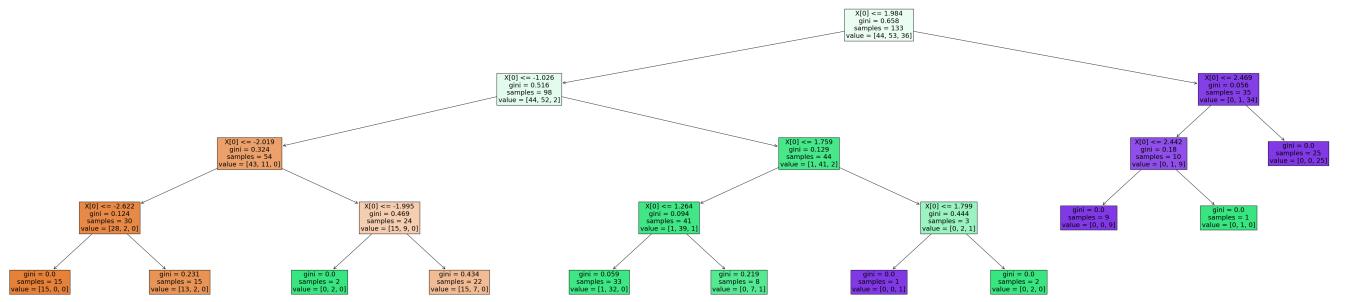


	precision	recall	f1-score	support
Regione 1	0.81	0.87	0.84	15
Regione 2	0.78	0.78	0.78	18
Regione 3	0.91	0.83	0.87	12

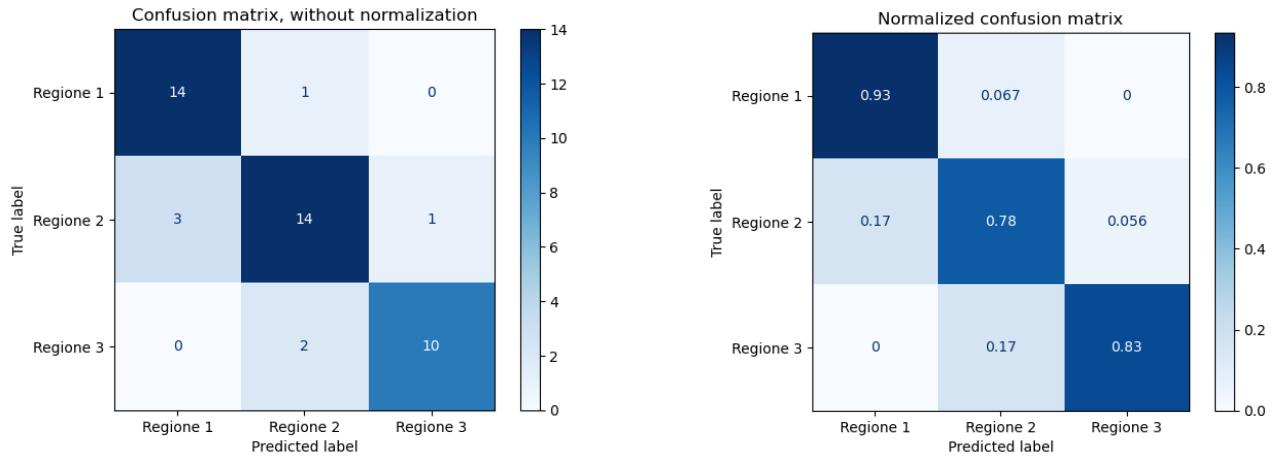


6.4 Decision Tree





	Validation	Test	Max depth
Con PCA (k=1)	83.4%	84.4%	4



	precision	recall	f1-score	support
Regione 1	0.82	0.93	0.87	15
Regione 2	0.82	0.78	0.80	18
Regione 3	0.91	0.83	0.87	12

