

UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA
DIPARTIMENTO DI INFORMATICA, SISTEMISTICA E COMUNICAZIONE
CORSO DI LAUREA IN INFORMATICA



Fetal Health

Marco Piazza - 829588 Cazzaniga Elisa - 829914

Indice

Acronimi	6
1 Dataset	7
1.1 Descrizione Generale	7
2 Analisi Esplorativa	8
2.1 Analisi Univariata	8
2.2 Analisi Multivariata	17
3 Principal Component Analysis	20
3.1 Confronto tra analisi esplorativa tramite grafici e tramite Principal Component Analysis (PCA)	23
4 Modelli di apprendimento	24
4.1 Decision tree	24
4.2 Support Vector Machine	30
5 Dataset sbilanciato	33
5.1 Introduzione	33
5.2 Possibili approcci	34
5.3 Realizzazione	34
5.3.1 Decision tree - Apprendimento standard	35
5.3.2 Decision tree - 10-fold cross validation	37
6 Esperimento	39
6.1 Confronto tra Support Vector Machine (SVM) e decision tree	39
7 Conclusione	44
7.1 Confronto tra i modelli di albero proposti	44

7.2	Dataset normale vs dataset dopo PCA	45
7.3	Conclusioni finali	45

Elenco delle figure

2.1	Distribuzione variabile target fetal_health	8
2.2	Istogramma del fhr	14
2.3	Feature plot degli attributi	15
2.4	Feature plot degli attributi	16
2.5	Feature plot degli attributi	16
2.6	Feature plot degli attributi	17
2.7	Heatmap completa di tutti gli attributi	18
3.1	Varianza spiegata dalle varie componenti	21
3.2	Rappresentazione delle variabili nel nuovo spazio	22
3.3	Biplot di variabili e individui nel nuovo spazio	23
4.1	Plot cp	25
4.2	Decision tree	26
4.3	Decision tree e decisione tree pruned applicato a dataset risultante da PCA	27
4.4	Plot cp PCA	28
4.5	Dataset normale (destra) e dataset ottenuto da PCA (sinistra)	29
4.6	Dataset normale (destra) e dataset ottenuto da PCA (sinistra)	32
5.1	Distribuzione variabile target fetal_healt	33
5.2	Dataset sbilanciato - Originale (sinistra) - Classificazione basata su pesi (destra)	37
5.3	Dataset sbilanciato - Oversample (sinistra) - Undersample (destra)	37
6.1	Curve Receiver Operating characteristic (ROC) 10-fold con dataset prodotto con la PCA, utilizzando l'apprendimento pesato, con i modelli migliori. (sinistra - decision tree, destra - SVM)	40
6.2	Dotplot Area Under Curve (AUC)	41
6.3	Bwplot	42

Elenco delle tabelle

4.1	Confronto tra le matrici di confusione ottenute dalla previsione con albero (sinistra) e albero pruned (destra)	26
4.2	Confronto tra le misure di performance ottenute l'albero normale (sinistra) e l'albero pruned (destra)	27
4.3	Confronto tra le matrici di confusione ottenute dalla previsione con albero (sinistra) e albero pruned (destra) su dataset ottenuto da PCA	28
4.4	Confronto tra le misure di performance ottenute l'albero normale (sinistra) e l'albero pruned (destra) su dataset ottenuto da PCA	28
4.5	Valori di AUC ottenuti da modelli decision tree	29
4.6	Confronto tra le matrici di confusione ottenute dall'applicazione sul dataset originale (sinistra) e sul dataset ottenuto mediante PCA (destra)	30
4.7	Confronto tra le misure di performance ottenute con il dataset originale (sinistra) e con il dataset ottenuto mediante PCA (destra)	31
4.8	Valori di AUC ottenuti da modelli pca	32
5.1	Confronto tra le matrici di confusione ottenute dall'applicazione sul dataset originale dei vari metodi per risolvere la problematica del dataset sbilanciato (originale - oversampling - undersampling - classificazione con pesi)	35
5.2	Confronto tra le misure di performance ottenute con il dataset originale dei vari metodi per risolvere la problematica del dataset sbilanciato (originale - oversampling - undersampling - classificazione con pesi)	35
5.3	Confronto tra le matrici di confusione ottenute dall'applicazione sul dataset ottenuto mediante PCA dei vari metodi per risolvere la problematica del dataset sbilanciato (originale - oversampling - undersampling - classificazione con pesi)	36
5.4	Confronto tra le misure di performance ottenute con il dataset ottenuto mediante PCA applicando i vari metodi per risolvere la problematica del dataset sbilanciato (originale - oversampling - undersampling - classificazione con pesi)	36
5.5	Valori AUC (originale - oversampling - undersampling - classificazione con pesi)	38

6.1	Confronto tra le matrici di confusione ottenute da SVM (sinistra) e decision tree (destra)	39
6.2	Confronto tra le misure di performance ottenute un modello SVM e un decision tree.	40
6.3	Valori AUC (Decision Tree e SVM)	41
6.4	Confronto tempistiche Decision Tree e SVM	43

Acronimi

CTG Cardiotocography

bpm Battiti per minuto

FHR Fetal Heart Rate

FIGO International Federation of Gynecology and Obstetrics

SVM Support Vector Machine

ROC Receiver Operating characteristic

PCA Principal Component Analysis

AUC Area Under Curve

Capitolo 1

Dataset

1.1 Descrizione Generale

Il dataset che abbiamo scelto di utilizzare è disponibile al seguente link:

`https://www.kaggle.com/andrewmvd/fetal-health-classification`

La cardiocografia(CTG) è un'opzione semplice ed economica per valutare la salute fetale, consentendo agli operatori sanitari di agire per prevenire la mortalità infantile e materna.

Questo dataset contiene 2126 registrazioni di caratteristiche estratte da esami di cardiocografia, che sono stati poi classificati da tre ostetriche esperte in 3 classi:

- Normale
- Sospetto
- Patologico

Capitolo 2

Analisi Esplorativa

2.1 Analisi Univariata

Inizialmente, abbiamo effettuato un'analisi esplorativa del dataset per comprenderne le caratteristiche generali.

Il dataset è composto da 2126 istanze e 22 features; non sono presenti valori nulli. Attraverso la funzione `str()` visualizziamo i tipi delle variabili. La variabile `fetal_health`, utilizzata come variabile target per la classificazione, risulta essere rappresentata come intero. È quindi necessario trasformarla in dato categorico tramite la funzione `factor()`.

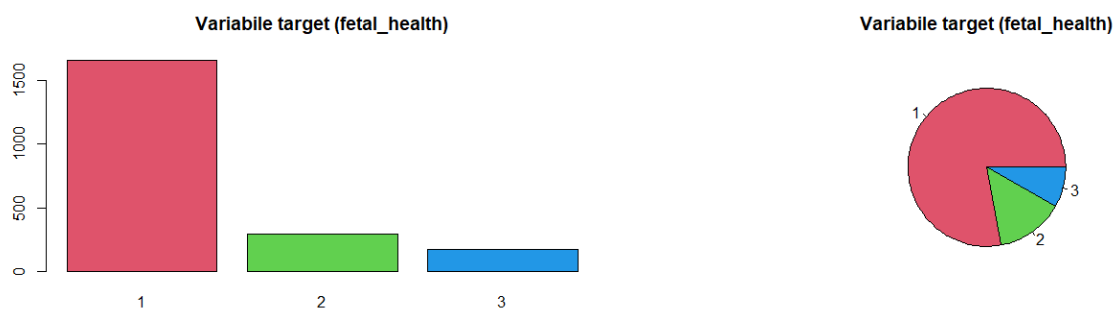


Figura 2.1: Distribuzione variabile target fetal_health

Come possiamo osservare dai grafici, il dataset è fortemente sbilanciato rispetto la variabile target:

- Normale (1): 1655 istanze

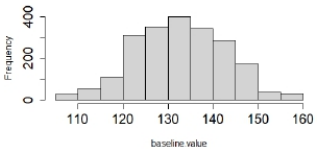
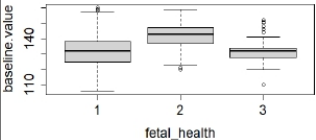
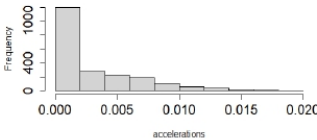
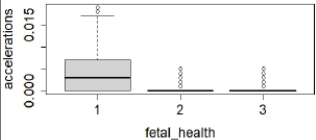
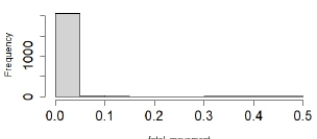
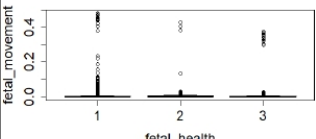
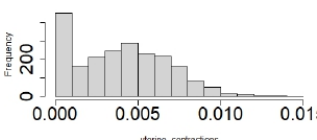
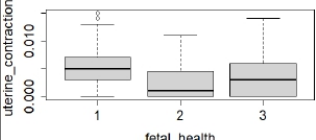
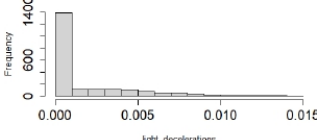
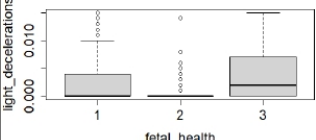
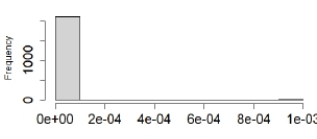
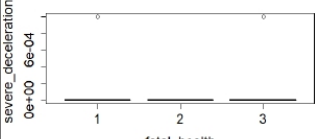
- Sospetto (2): 295 istanze
- Patologico (3): 176 istanze

Esistono studi che caratterizzano l'appartenenza ad una delle categorie sulle base dei valori osservati durante l'analisi dei risultati dell'esame. In particolare le linee guida dell'istituto International Federation of Gynecology and Obstetrics (FIGO) suggeriscono:

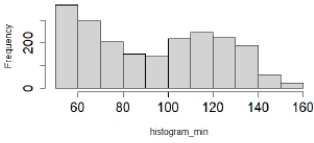
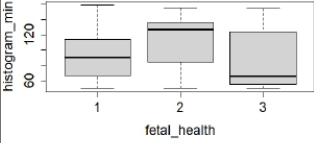
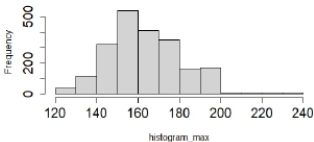
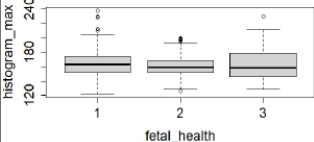
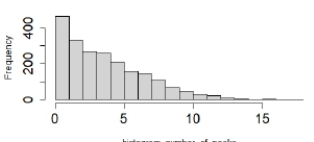
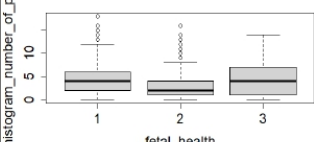
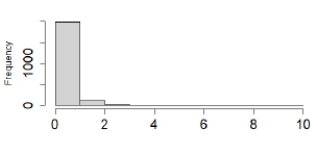
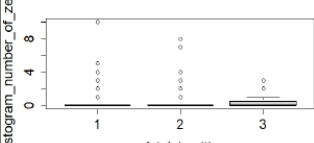
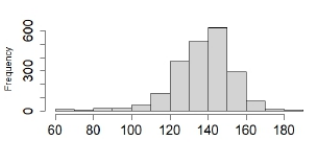
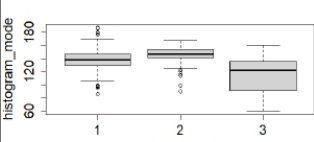
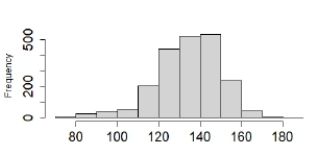
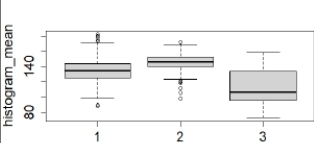
- Normale:
 - Baseline: 110-160 Battiti per minuto (bpm);
 - Variability: 5-25 bpm;
 - Nessuna presenza di decelerations;
- Sospetto:
 - Manca di una delle condizioni precedenti, senza la presenza di caratteristiche anomale.
- Patologico:
 - Baseline < 100 bpm;
 - Minore o maggiore valore di variability;
 - Presenza di Prologued Decelerations;
 - Presenza di Decelerations più lunghe di 5 minuti.

Il nostro dataset è composto da molte covariate, 22 in tutto che corrispondono ai risultati estratti mediante l'esame della Cardiotocography (CTG). Abbiamo effettuato un'analisi univariata di tutte le features per individuarne le principali caratteristiche come la distribuzione e la presenza di eventuali outliers. Di seguito riportiamo una tabella con la spiegazione di ogni attributo del nostro dataset e due grafici:

- Grafico(1): Istogramma della distribuzione dell'attributo
- Grafico(2): Boxplot dell'attributo paragonato alla variabile target

Attributo	Spiegazione	Grafico(1)	Grafico(2)
1. Baseline Value	Rappresenta la frequenza cardiaca fetale durante il travaglio.		
2. Accelerations	Rappresenta le accelerazioni. Le accelerazioni sono aumenti a breve termine della frequenza cardiaca di almeno 15 battiti al minuto, della durata di almeno 15 secondi.		
3. Fetal Movement	Rappresenta il numero di movimenti del feto al secondo.		
4. Uterine Contractions	Rappresenta il numero di contrazioni uterine al secondo.		
5.1 Light Decelerations	Rappresenta una diminuzione nel FHR sotto il livello di baseline della durata di almeno 15 sec e con l'ampiezza minima di 15 bpm. Vengono identificati tre categorie di decelerations light (< 120 sec.), severe (>120 e <300) e prolonged (>300).		
5.2 Severe decelerations			

5.3 Prolongued Decelerations			
6. Abnormal short term variability – valore percentuale in tempo	Un caso di Abnormal short term variability si verifica quando la differenza tra due segnali di FHR consecutivi è meno di 1 bpm.		
7. Abnormal short term variability – numero medio di occorrenza			
8. Abnormal long term variability – valore percentuale in tempo	Un punto con LTV anormale viene identificato quando la differenza tra valori massimi e minimi di una finestra scorrevole di 60 sec centrato su di esso non supera i 5 bpm.		
9. Abnormal long term variability – numero medio di occorrenze			
10. Histogram width (ampiezza)	È possibile rappresentare il valore completo di FHR su un istogramma. Sono riportati alcuni valori caratteristici dell'istogramma risultante.		

11. Histogram min (minimo)		
12. Histogram max (massimo)		
13. Histogram number of peaks (numero di picchi)		
14. Histogram number of zeros (numero di zeri)		
15. Histogram mode (valore di moda)		
16. Histogram mean (valore medio)		

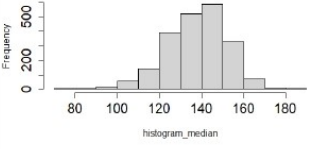
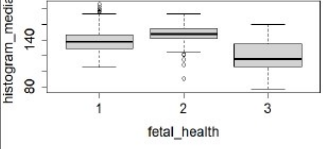
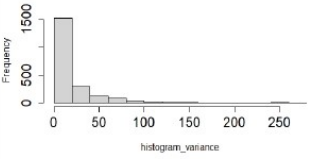
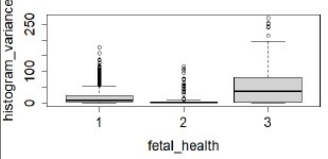
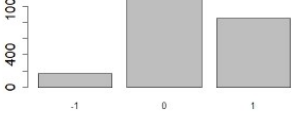
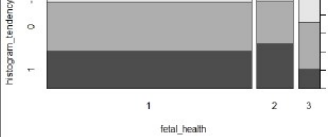
17. Histogram median (valore mediana)			
18. Histogram variance (varianza)			
19. Histogram tendency	Rappresenta la tendenza del grafico del FHR.		

Fig. 2.2 Esempio di istogramma rappresentativo del Fetal Heart Rate (FHR)

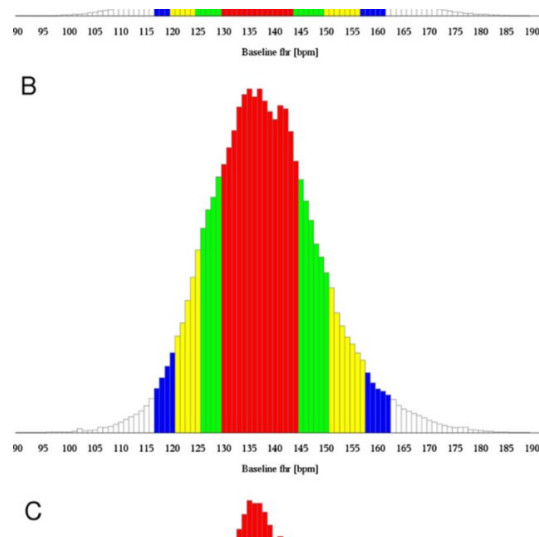


Figura 2.2: Istogramma del fhr

Analizzando il grafico (1) è possibile dividere gli attributi in due categorie sulla base della distribuzione osservata:

- Distribuzione secondo una Normale (1, 6, 10, 11, 15, 16, 17). In questo caso probabilmente i dati essendo uniformemente distribuiti avranno valori vicini alla media per la classe target normale, mentre avranno valori più vicini agli estremi per la classe target sospetto e patologico.
- Distribuzione secondo una Normale asimmetrica (2, 3, 4, 5.1, 5.2, 5.3, 7, 8, 9, 14, 18) con right skew, ovvero i valori sull'estremità alta (lato destro) sono molto più distribuiti rispetto ai valori sull'estremità bassa (lato sinistro). Sono presenti anche alcuni con left skew. Anche qui si potrebbe ipotizzare che i valori maggiormente distribuiti corrispondano a quelli appartenenti ai target sospetto e patologico.

Analizzando invece il grafico (2) non è possibile identificare degli attributi che sono molto discriminatori per nessuna tra le possibili variabili target. È invece possibile identificare la presenza di outliers, in modo particolare in alcuni casi si rileva la presenza di molti outlier, tuttavia non è chiaro da identificare se si trattino di outlier o di valori che si discostano dalla media (quindi rappresentativi di un caso sospetto o patologico).

Analisi dei feature plot

Tramite questa analisi è stato possibile indagare circa la distribuzione degli attributi e quanto questi attributi siano discriminatori rispetto alla variabile target. Inoltre, è stato possibile fare ipotesi circa la possibilità di rimuovere parte degli attributi in quanto non discriminino abbastanza tra le varie classi.

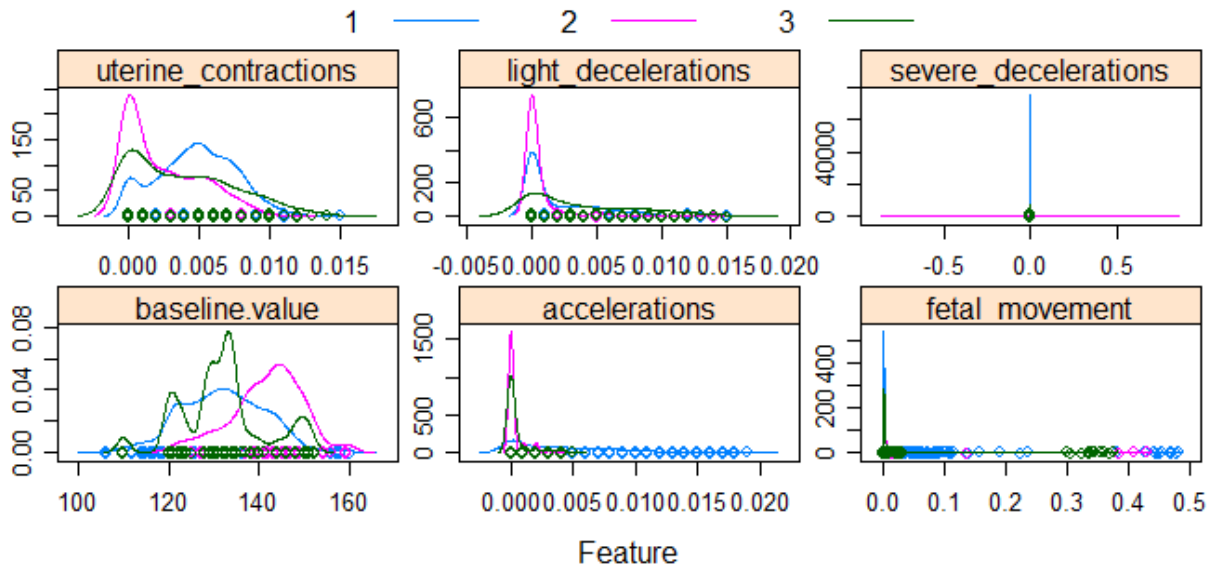


Figura 2.3: Feature plot degli attributi

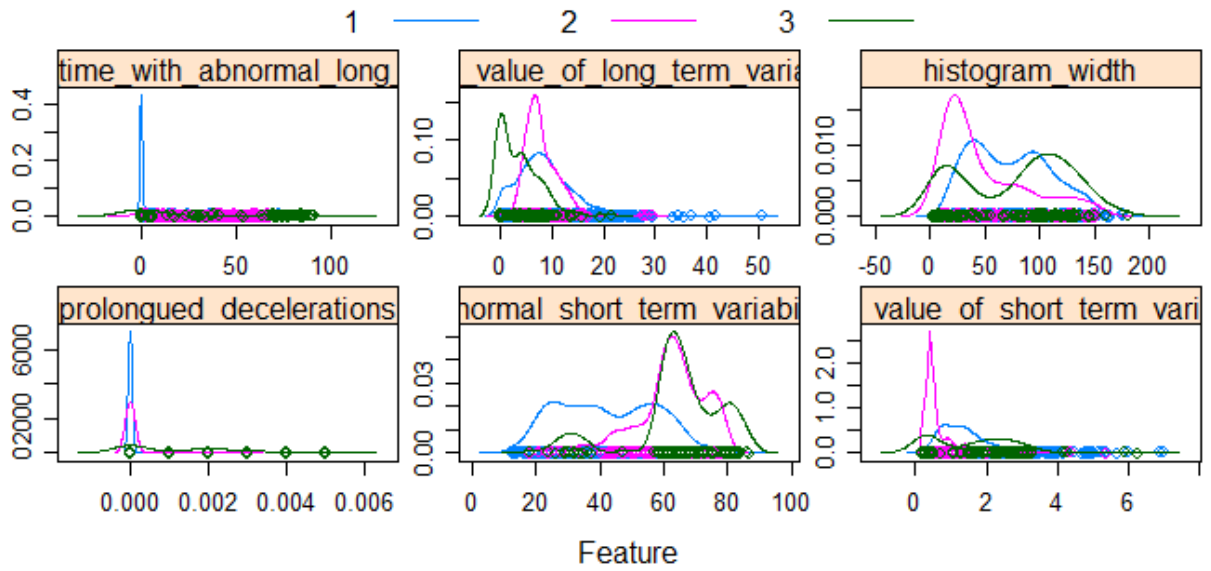


Figura 2.4: Feature plot degli attributi

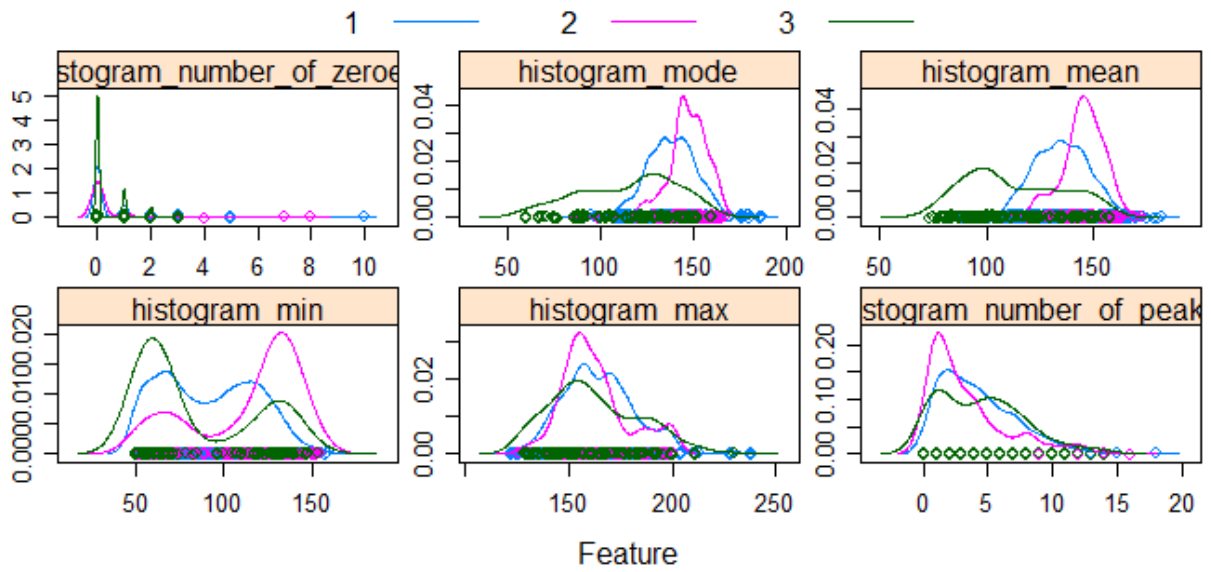


Figura 2.5: Feature plot degli attributi

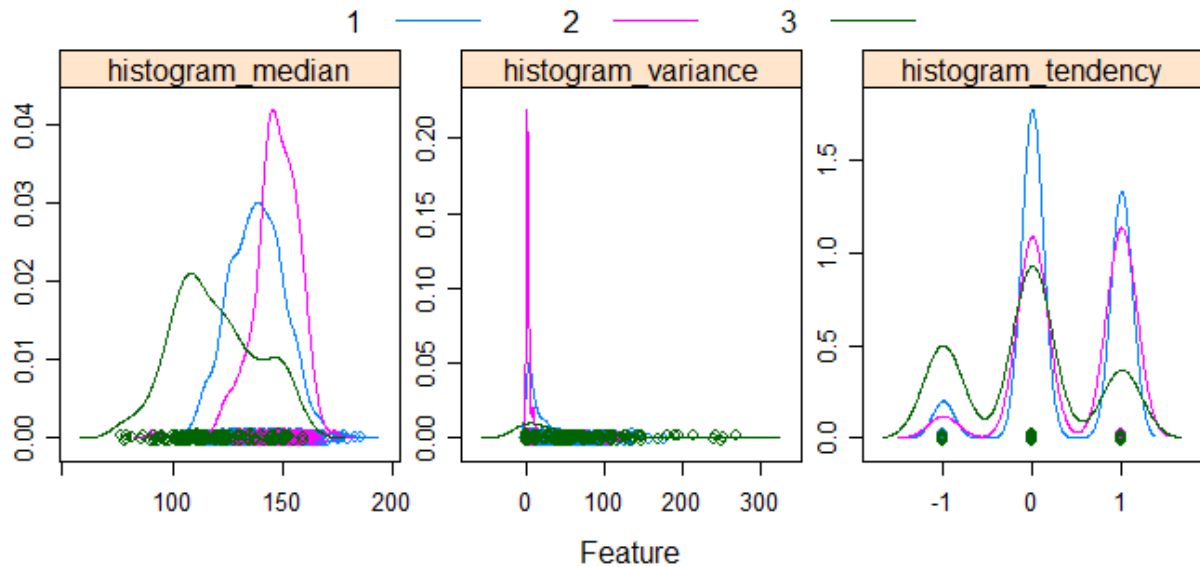


Figura 2.6: Feature plot degli attributi

Dall'analisi di questi grafici è possibile ipotizzare quanto ogni attributo possa essere efficace nel discriminare tra i valori target. È possibile ipotizzare come le variabili riferite alle dimensioni dell'istogramma della FHR aiutano molto meglio di altre variabili nell'effettuare questo tipo di scelta (2.5, 2.6). Inoltre è possibile fare delle ipotesi circa alcuni attributi che al contrario non sono utili alla discriminazione tra le varie variabili target:

- severe decelerations (2.3)
- fetal movement (2.3)
- time with abnormal long term variability (2.4)
- histogram variance (2.6)

2.2 Analisi Multivariata

Per l'analisi multivariata abbiamo creato ed analizzato una heatmap per capire la correlazioni tra i vari attributi.

Analisi correlazione tra le variabili

La correlazione è stata calcolata sfruttando l'indice di Pearson che restituisce un valore compreso tra +1 (correlazione lineare positiva), 0 (assenza di correlazione) e -1 (correlazione lineare negativa). Questa analisi ha permesso di identificare alcune relazioni tra attributi che verranno approfondite in seguito.

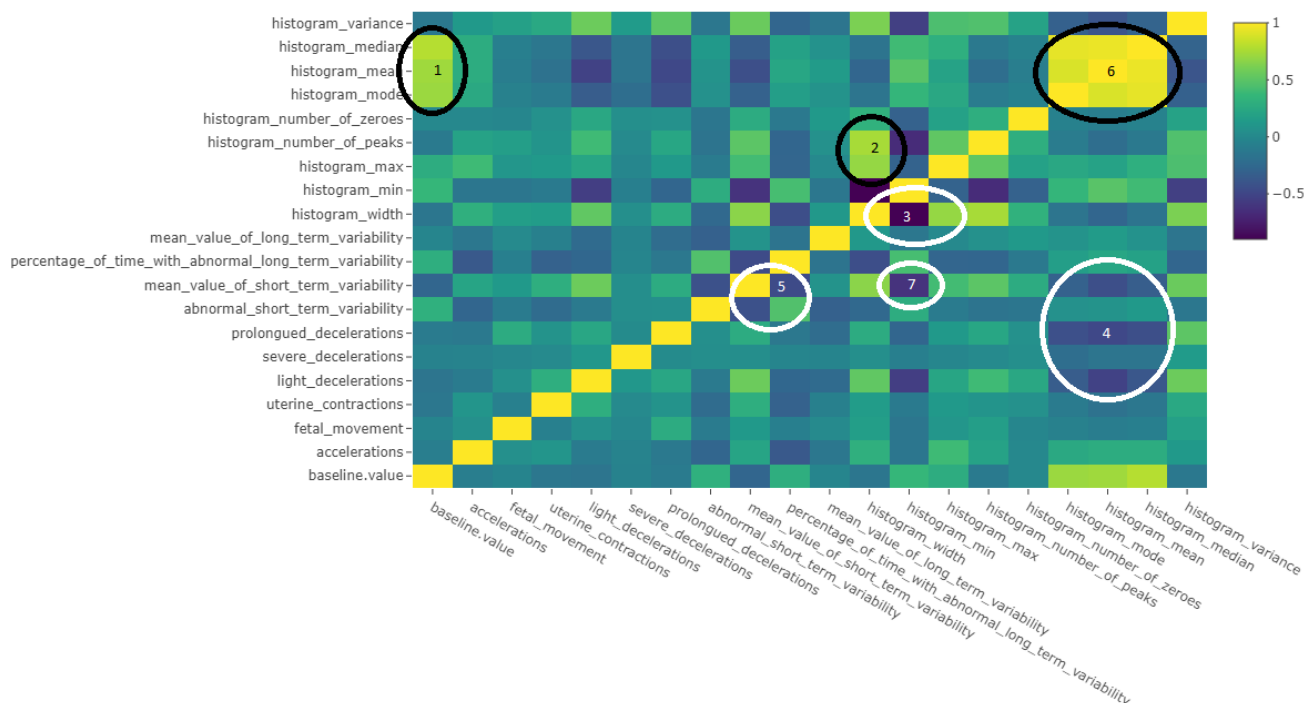


Figura 2.7: Heatmap completa di tutti gli attributi

Osservando la heatmap è possibile affermare come la maggior parte degli attributi risultino vicini al valore 0, quindi assenza di correlazione. Tuttavia è possibile identificare alcune aree all'interno della heatmap in cui i valori cambiano sia in positivo (aree prettamente verde-giallo), che in negativo (aree prettamente viola). È stato quindi deciso di procedere ad un'analisi più approfondita per valutare la correlazione tra questi attributi.

Per aiutare nella lettura della heatmap sono stati evidenziati con il colore nero quelli positivamente correlati e con il colore bianco quelli negativamente correlati. Inoltre è stato possibile ipotizzare motivazioni su questo tipo di correlazione tra gli attributi. Casi positivamente correlati:

1. valore di *baseline_value* con valori di misura della dimensione dell'istogramma (*median*, *mean* e *mode*). Probabilmente in questo caso, il valore di baseline è estratto dall'istogramma, quindi i valori più diffusi, medi e mediani saranno molto vicini a quello considerato come baseline;
2. valori di *histogram_width* con *histogram_max* e *histogram_number_of_peaks*. In questo caso questa correlazione è dovuta probabilmente al fatto che il numero di picchi dell'istogramma e il valore massimo raggiunto dalla curva sono legati all'ampiezza del grafico.
3. valori di *histogram_min* e *histogram_width*. Questi due valori sono correlati negativamente.
4. valori di *histogram_mean* con i diversi valori di *decelerations* (*severe*, *prolongued* e *light*). Sono negativamente correlati perchè ad un valore medio alto, corrisponderà l'assenza di fasi di decelerations e viceversa.
5. in questo caso sono messi a confronto il valore medio di tempo con *abnormal short term variability* e il numero di occorrenze medie, probabilmente ad un numero inferiore di occorrenze dei casi si prolunga la durata e viceversa. Inoltre viene messo a confronto il tempo di fasi di *long term variability* con il valore medio di *short term variability* e anche queste due risultano correlate negativamente.
6. in questo riquadro sono messi a confronto i valori di media, moda e mediana dell'istogramma del FHR e risultano fortemente correlati positivamente, quindi l'istogramma probabilmente sarà distribuito secondo una normale con assenza di code.
7. in questo ultimo caso vengono messi a confronti il valore minimo del grafico (*histogram_min*) con il valore di *short_term_variability*.

Capitolo 3

Principal Component Analysis

Un'altra tecnica utile per l'analisi esplorativa dei dati, soprattutto quando si hanno a disposizione un elevato numero di covariate da considerare, è la PCA. Si tratta di una tecnica di feature extraction che ci permette di effettuare una trasformazione lineare del dataset passando da uno spazio in input di n covariate ad uno con m covariate non correlate tra loro, con $m \leq n$. Analizzando il risultato ottenuto dalla PCA, possiamo stabilire quante e quali sono le componenti che ci permettono di spiegare la maggior parte della varianza nei dati. L'obiettivo è quello di ridurre la complessità, mantenendo comunque un buon valore di accuratezza.

Abbiamo utilizzato la funzione *PCA()* sul dataset scalato (senza considerare la variabile target). Viene restituita una lista che tra gli elementi contiene:

- eig: una matrice contenente tutti gli autovalori, la percentuale di varianza e la percentuale di varianza cumulata
- var: un elenco di matrici contenente tutti i risultati per le variabili attive (coordinate, correlazione tra variabili e assi, la qualità di rappresentazione delle variabili nel nuovo spazio di rappresentazione, contributo delle variabili alla componente principale)
- ind: un elenco di matrici contenente tutti i risultati per gli individui attivi (coordinate, la qualità di rappresentazione degli individui nel nuovo spazio di rappresentazione, contributo degli individui alla componente principale)

Attraverso gli autovalori possiamo visualizzare la varianza spiegata da ogni dimensione. Ogni componente principale riassume una certa percentuale della variazione totale nel dataset. Per determinare il numero di componenti principali da mantenere possiamo utilizzare una di queste due strategie:

- Scegli di utilizzare un numero di componenti pari al numero degli autovalori maggiori di 1.

- Fissata una soglia, seleziono un certo numero di componenti che mi permettono di spiegare quella determinata varianza.

Noi abbiamo scelto di utilizzare 8 componenti che spiegano circa l'81% della varianza.

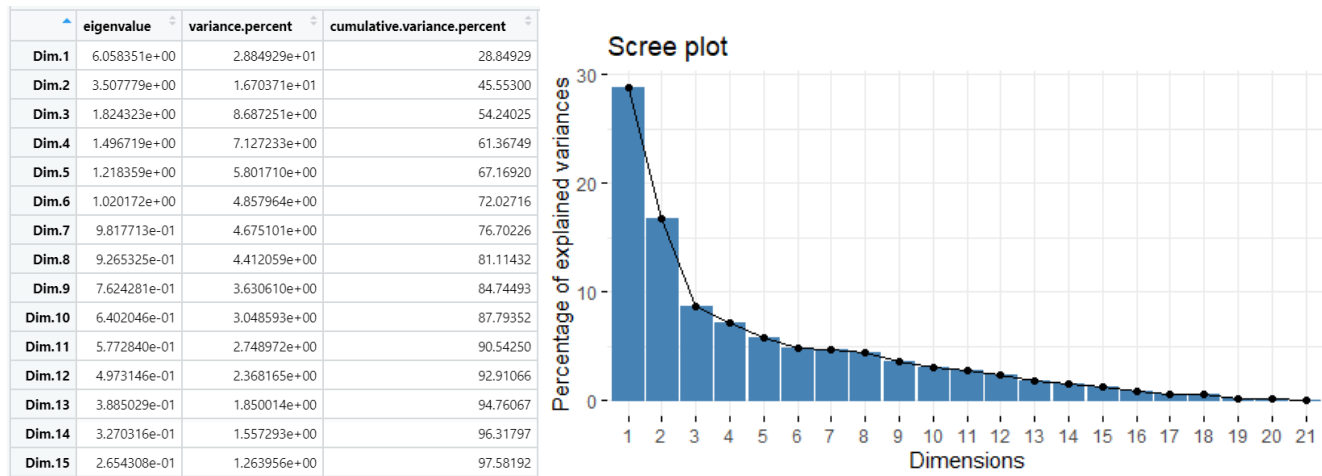


Figura 3.1: Varianza spiegata dalle varie componenti

Viene riportato di seguito il grafico delle variabili rappresentate nel nuovo spazio utilizzando come assi cartesiani le due componenti più significative. Le variabili sono colorate in base al contributo che forniscono alla PCA. Più i vettori sono vicini, più le variabili sono positivamente correlate (ad esempio: *histogram_median*, *histogram_mean*, *histogram_mode* sono correlate positivamente). Se le variabili sono correlate tra loro, contribuiranno fortemente alla stessa componente principale. Le variabili sono negativamente correlate se si trovano in quadranti opposti (ad esempio: *histogram_min* e *histogram_width* sono negativamente correlate). Le variabili sono non correlate se risultano ortogonali tra loro (es: *baseline_value* e *histogram_width* sono non correlate).

La distanza delle variabili dall'origine misura la qualità delle stesse nel nuovo spazio di rappresentazione. Le variabili che sono lontane dall'origine sono ben rappresentate nel nuovo spazio.

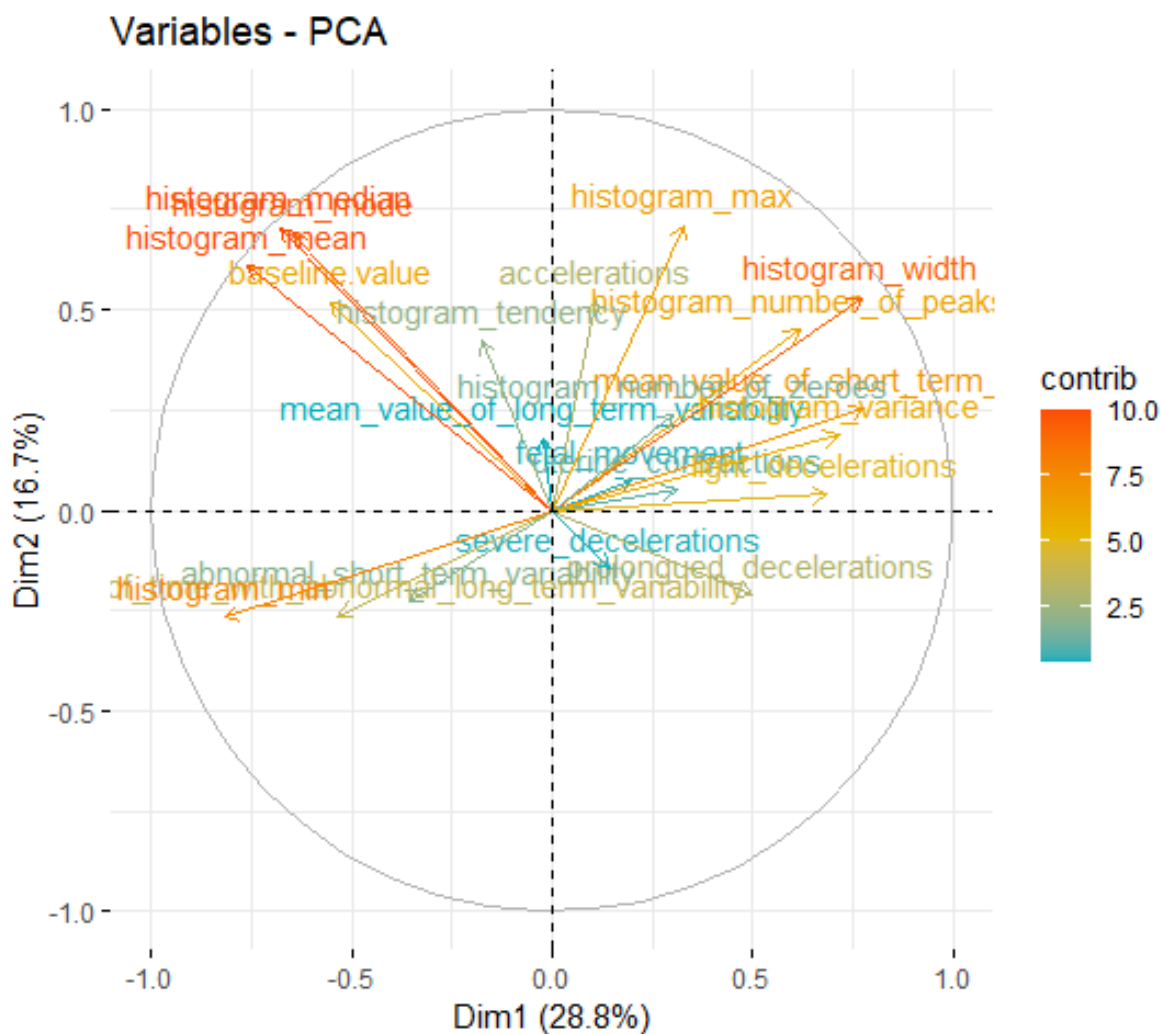


Figura 3.2: Rappresentazione delle variabili nel nuovo spazio

Viene riportato di seguito il grafico contenente le variabili e gli individui rappresentati nel nuovo spazio utilizzando come assi cartesiani le due componenti più significative. Gli individui sono raggruppati secondo i tre valori della variabile target.

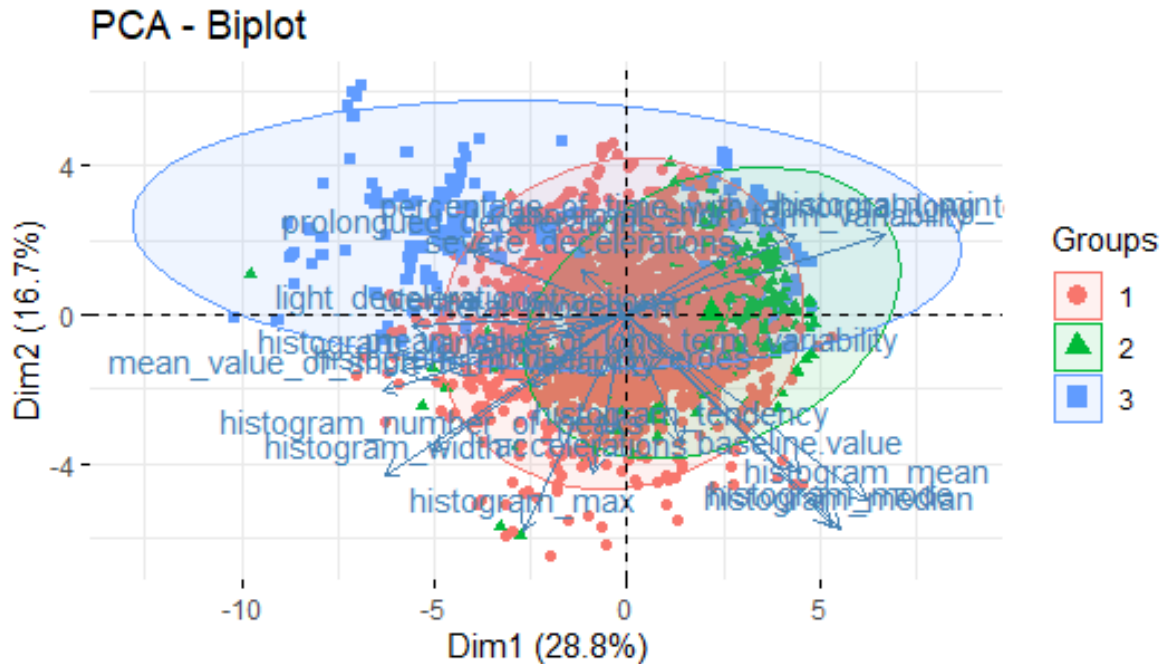


Figura 3.3: Biplot di variabili e individui nel nuovo spazio

3.1 Confronto tra analisi esplorativa tramite grafici e tramite PCA

Dopo aver svolto le varie analisi con alcune supposizioni per capire le caratteristiche delle covariate del nostro dataset, abbiamo confrontato i risultati delle due applicazioni. Seguono alcune osservazioni riscontrate:

- C'è corrispondenza tra le correlazioni delle variabili studiate attraverso la heatmap e la PCA.
- Alcune delle covariate che inizialmente avevamo classificato non particolarmente utili per la discriminazione, come *histogram_variance* e *time_with_abnormal_long_term_variability*, a seguito della PCA risultano comunque avere una media qualità. Questo può essere dovuto al fatto che ci sono numerosi outliers che corrispondono ad istanze "patologiche".
- Al contrario, due variabili che non avevamo considerato, *mean_value_of_long_term_variability* e *uterine_contractions* danno un contributo ridotto alla spiegazione della varianza del dataset.

Capitolo 4

Modelli di apprendimento

4.1 Decision tree

Abbiamo deciso di utilizzare gli alberi di decisione come metodo di apprendimento dato che tra i loro potenziali usi più frequenti sono incluse le diagnosi mediche basate su misurazioni di laboratorio, sintomi o tasso di progressione della malattia. Molti algoritmi basati sui decision tree producono la struttura risultante in un formato leggibile dall'uomo. Questo fornisce informazioni su come e perché il modello funziona o non funziona bene per una particolare attività.

Innanzitutto, abbiamo caricato il dataset impostando la variabile target come factor. Abbiamo suddiviso il dataset in modo random in due porzioni prendendo il 70% dei record per il training set e il rimanente 30% per il testset. Abbiamo impostato un valore seed in modo tale che il processo di randomizzazione segua una sequenza che può essere replicata in seguito. Questo garantisce che se l'analisi viene ripetuta in futuro, si ottiene un risultato identico.

Abbiamo addestrato il nostro modello utilizzando la funzione *rpart()*, impostando come metodo la classificazione e non la regressione. La variabile target viene fittata su tutti gli attributi.

In seguito abbiamo stimato il parametro di complessità *cp*, per determinare il minor miglioramento che il modello ha bisogno ad ogni livello, che mi permette di mantenere ridotto l'errore di classificazione, a vantaggio della riduzione del numero di split del nostro albero. A parità di accuratezza o di misura di performance si predilige un modello meno complesso. Se riuscissi a togliere dei rami decisionali pur arrivando alla stessa accuratezza, prediligo l'albero con dimensione ridotta perché riduce la probabilità di overfitting del modello. Inoltre, avere un albero molto grande, per l'esperto di dominio è complesso. Bisogna trovare un compromesso tra leggibilità del modello, complessità computazionale della decisione e bontà della predizione. Per evitare che il problema si verifichi si taglia l'albero, creando il pruned decision tree, utilizzando il parametro di

complessità stimato in modo ottimale.

Attraverso *plotcp()* possiamo visualizzare un grafico che ci mostra l'errore che viene commesso in base al valore di *cp* e alla relativa dimensione dell'albero.

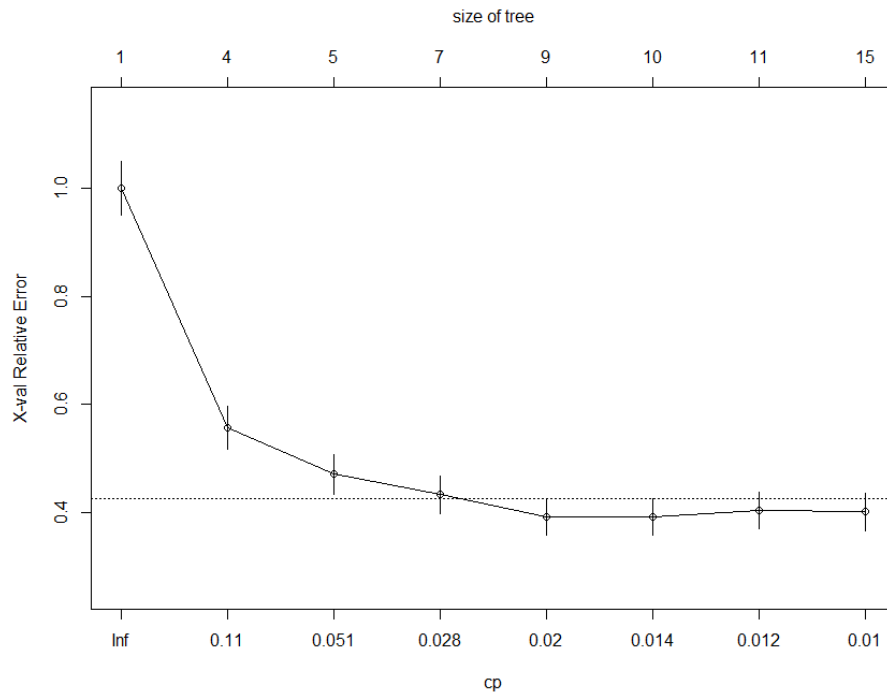


Figura 4.1: Plot cp

Per effettuare una scelta che mantiene un buon compromesso tra i tre aspetti presentati precedentemente, viene utilizzato il valore di *cp* che minimizza l'errore commesso.

Effettuiamo quindi il taglio dell'albero, specificando come valore di *cp* 0.02, che corrisponde ad avere un albero di dimensione uguale a 9. Otteniamo l'albero che segue, messo a confronto con l'albero ottenuto senza l'operazione di prune.

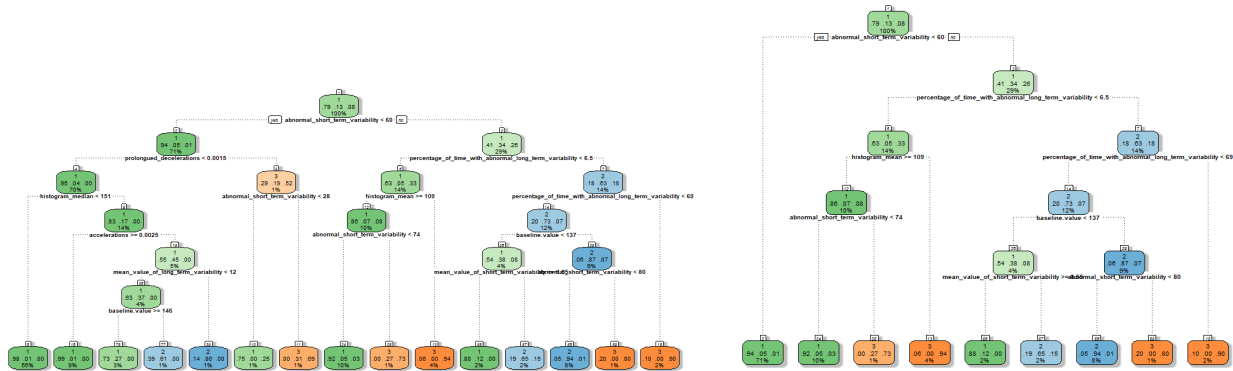


Figura 4.2: Decision tree

Effettuiamo la predizione utilizzando il testset con entrambi i modelli. Calcoliamo la matrice di confusione per visualizzare l'accuratezza del nostro modello.

	1	2	3
1	459	21	10
2	16	79	1
3	12	2	42

	1	2	3
1	472	34	16
2	7	67	1
3	8	1	36

Tabella 4.1: Confronto tra le matrici di confusione ottenute dalla previsione con albero (sinistra) e albero pruned (destra)

Osservando i risultati ottenuti si possono fare alcune osservazioni: in primo luogo non c'è molta differenza tra i risultati ottenuti dal modello normale e quelli ottenuti dall'albero dopo essere stato ridotto, quindi può essere una buona scelta l'albero meno complesso. In entrambe le matrici l'unica problematica è data da un numero abbastanza elevato di elementi di classe sospetto e patologico classificate come classe normale (riga 1). Infatti è una grande problematica se individui che presentano problematiche sono classificati come individui sani.

Inoltre abbiamo calcolato alcuni valori per stimare la performance del modello. Uno di questi parametri è l'accuratezza globale che risulta essere: 0.9034 nel caso del primo modello, scende invece di poco nel caso dell'albero pruned: 0.8956

	1	2	3		1	2	3
Accuracy	0.8713	0.8715	0.88434	Accuracy	0.8233	0.8210	0.83198
Precision	0.9367	0.8229	0.7500	Precision	0.9042	0.8933	0.8000
Recall	0.9425	0.7745	0.79245	Recall	0.9692	0.6569	0.67925
F-Measure	0.9396	0.7980	0.77064	F-Measure	0.9356	0.7571	0.73469

Tabella 4.2: Confronto tra le misure di performance ottenute l'albero normale (sinistra) e l'albero pruned (destra)

Analizzando i risultati ottenuti è possibile affermare che i valori di performance sono elevati in entrambi i modelli, per esempio i valori di accuracy bilanciata in riferimento alle singole classi della variabile target risultano simili in entrambi i modelli. I valori di precision risultano essere buone in entrambi i modelli, tuttavia peggiorano sensibilmente passando dalla prima classe alle altre, questo può essere dovuto al fatto che ci sono tante istanze di classe 1 classificate come classe 2-3. La recall è la misura che maggiormente interessa nel nostro dominio applicativo in quanto misura la percentuale di istanze realmente classificate di una classe sul totale delle istanze presenti di quella data classe. Come era ipotizzabile osservando le matrici di confusione questo dato è basso per le istanze di classi 2 - 3 e peggiora passando dall'albero normale all'albero pruned. Questo dato non può soddisfare in quanto è necessario che soprattutto per le classi sospetto e patologico abbiano un valore di recall elevato. Il dato di f-measure come ipotizzato è alto per la classe 1 ed abbastanza basso per le altre due classi, in quanto è ottenuto considerando sia precision che recall.

Abbiamo effettuato lo stesso procedimento utilizzando l'information gain come criterio di splitting, ottenendo risultati leggermente inferiori.

Abbiamo confrontato i risultati ottenuti utilizzando il dataset originale con quelli ottenuti utilizzando il dataset ottenuto attraverso la PCA. Di seguito vengono riportati i risultati ottenuti. I valori in dettaglio per l'analisi del parametro di complessità sono presenti nella parte di codice.

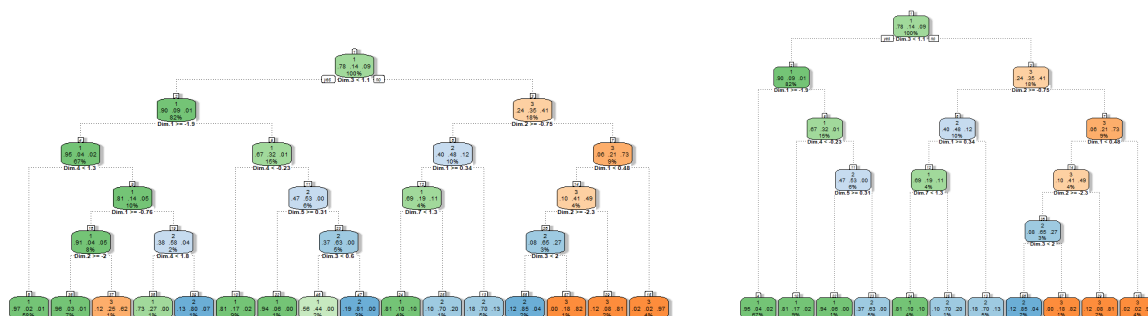


Figura 4.3: Decision tree e decision tree pruned applicato a dataset risultante da PCA

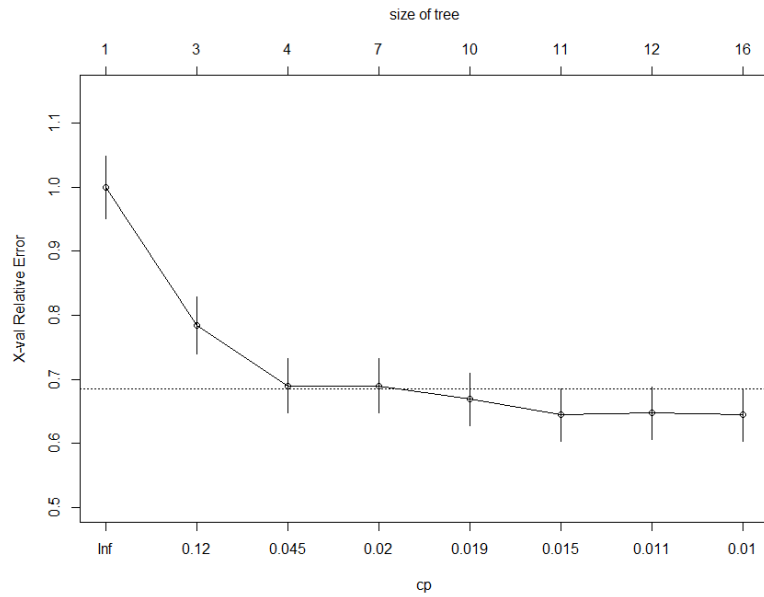


Figura 4.4: Plot cp PCA

	1	2	3
1	485	40	5
2	16	49	9
3	4	5	34

	1	2	3
1	478	40	6
2	24	49	9
3	3	5	33

Tabella 4.3: Confronto tra le matrici di confusione ottenute dalla previsione con albero (sinistra) e albero pruned (destra) su dataset ottenuto da PCA

	1	2	3
Accuracy	0.8217	0.7380	0.84665
Precision	0.9151	0.66216	0.79070
Recall	0.9604	0.52128	0.74725
F-Measure	0.9372	0.58333	0.74725

	1	2	3
Accuracy	0.8113	0.73080	0.83707
Precision	0.9122	0.59756	0.80488
Recall	0.9465	0.52128	0.68750
F-Measure	0.9291	0.55682	0.68750

Tabella 4.4: Confronto tra le misure di performance ottenute l'albero normale (sinistra) e l'albero pruned (destra) su dataset ottenuto da PCA

Le performance ottenute dal modello allenato con dataset creato dopo l'operazione di PCA sono leggermente inferiori ma non si discostano molto da quelle ottenute allenando il modello

con il dataset normale. È auspicabile trovare una strategia per migliorare alcune performance del modello, come la recall e la precision delle classi 2 - 3, perchè nel nostro dominio applicativo risultano di importanza cruciale. Tuttavia per ottenere questi risultati è necessario seguire altri approcci quali il bilanciamento del dataset presentato nel capitolo seguente.

Successivamente è stata effettuata un'allenamento utilizzando la tecnica 10-fold cross validation ed sono stati analizzati i risultati sfruttando la curca ROC e la misura di AUC.

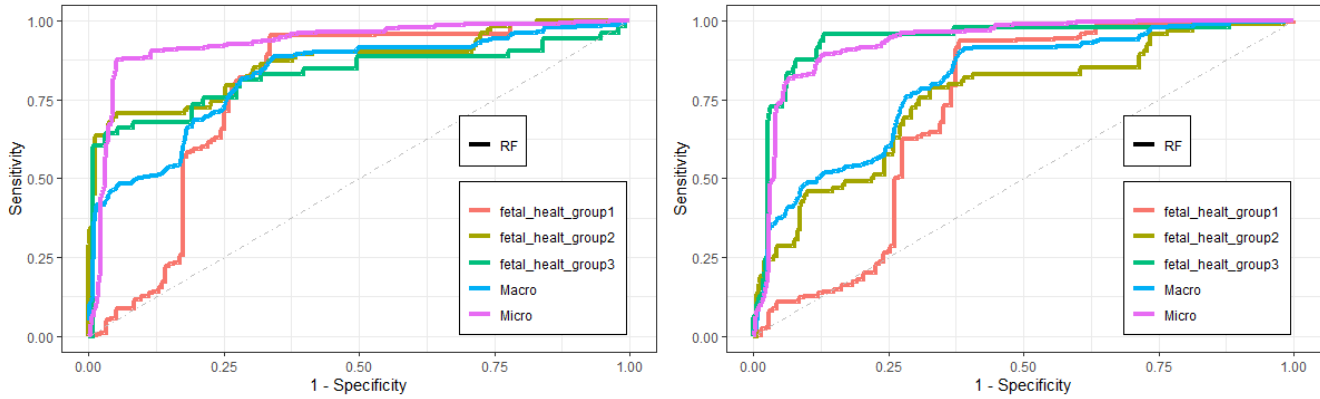


Figura 4.5: Dataset normale (destra) e dataset ottenuto da PCA (sinistra)

Come si può vedere dai due grafici le prestazioni dei modelli sono abbastanza simili, però salta all'occhio un dato molto importante: le prestazioni della classe 3 migliorano sensibilmente utilizzando il dataset modificato con PCA. Per quanto riguarda le altre due classi invece hanno delle performance simili. È possibile inoltre confrontare i valori ottenuti di AUC:

	1	2	3
Dataset normale	0.7788302	0.8665759	0.8271134
Dataset dopo PCA	0.7190629	0.7533954	0.9387869

Tabella 4.5: Valori di AUC ottenuti da modelli decision tree

La tabella con i valori di AUC conferma che le prestazioni dei due modelli sono simili e di buona qualità. Viene inoltre confermato il miglioramento nel caso della classe 3, mentre per le altre due classi si perde qualcosa. Tuttavia considerando il nostro dominio di applicazione è auspicabile avere performance più elevate nelle istanze classificate come patologico.

In conclusione, è quindi possibile affermare il modello allenato con il dataset ottenuto tramite PCA risulta di qualità buona se confrontato con il modello originale, in particolare migliora le previsioni per la classe 3 che è l'obiettivo più importante del nostro progetto. Inoltre utilizzan-

do il dataset ottenuto da PCA permette di ottenere modelli più semplici da un punto di vista computazionale.

4.2 Support Vector Machine

Abbiamo deciso di utilizzare una SVM come secondo metodo di apprendimento in quanto si presta bene a problemi di classificazione, come quello a cui ci troviamo di fronte.

Come primo passo è stato caricato il dataset impostando la variabile target come factor. Il dataset è stato suddiviso in due porzioni: 70% per la fase di training e il 30% per la fase di test. Anche in questo caso è stato utilizzato un valore di seed per replicare più volte il processo di randomizzazione. Per effettuare il training è stato utilizzato il metodo *tune.svm* per stimare il valore ottimale di costo, che è stato trovato essere 100 per il modello allenato sfruttando il dataset originale e 10 per il modello allenato sfruttando il dataset ottenuto tramite PCA. Inoltre analizzando varie prove effettuate è emerso che il kernel migliore è di forma polinomiale per entrambi i modelli. La variabile target viene fittata rispetto a tutti gli attributi appartenenti al dataset.

A questo punto viene effettuata la predizione sul test set, ovvero si chiede al modello di prevedere la corretta label di una parte del dataset (il test set) sulla base di quanto appreso durante la fase di training. Questa fase è necessaria per stimare la qualità del modello creato dall'allenamento. In questo caso viene specificato il parametro *prob = true*, cosicché i risultati vengono forniti in modalità probabilistica, ovvero viene fornita una stima di probabilità di appartenenza ad una delle tre classi. Questa scelta è necessaria per poter calcolare la curva ROC in quanto il nostro problema era multiclasse. Le stesse operazioni sono state effettuate sia sul dataset originale, che sul dataset ottenuto dopo l'applicazione della PCA

Viene calcolata la matrice di confusione:

	1	2	3
1	486	49	0
2	1	52	1
3	0	1	52

	1	2	3
1	502	42	1
2	3	52	15
3	0	0	32

Tabella 4.6: Confronto tra le matrici di confusione ottenute dall'applicazione sul dataset originale (sinistra) e sul dataset ottenuto mediante PCA (destra)

Analizzando la prima matrice di confusione possiamo ipotizzare che il modello fittato ha una buona accuratezza ed è in grado di fare previsioni in maniera adeguata. È possibile tuttavia identificare una problematica: 49 istanze di classe 2 (sospetto) sono classificate come istanze di classe 1 (normale). Questa problematica può avere un peso importante nel nostro dominio in quanto individui non sani vengono classificati come sani. Non è possibile identificare altre gravi

problematiche. Analizzando la seconda matrice risulta esserci qualche problematica in più della precedente: c'è ancora un numero elevato di istanze di classe 2 predette come istanze di classe 1 (42) e inoltre c'è un numero abbastanza elevato di istanze di classe 3 predette come istanze di classe 2 (15), questa problematica è molto importante in quanto il modello confonde delle istanze etichettate come sicuramente malate con istanze etichettate come sospette.

Inoltre vengono calcolati alcuni valori per stimare la performance del modello. Uno di questi parametri è l'accuratezza globale che risulta essere: 0.919 nel caso del primo modello. Nel caso del secondo modello invece l'accuratezza globale è 0.9057.

	1	2	3		1	2	3
Accuracy	0.8409	0.75305	0.98972	Accuracy	0.8456	0.76032	0.83333
Precision	0.9084	0.96296	0.98113	Precision	0.9211	0.74286	1.00000
Recall	0.9979	0.50980	0.98113	Recall	0.9941	0.55319	0.66667
F-Measure	0.9476	0.66667	0.98113	F-Measure	0.9562	0.63415	0.80000

Tabella 4.7: Confronto tra le misure di performance ottenute con il dataset originale (sinistra) e con il dataset ottenuto mediante PCA (destra)

Analizzando i risultati ottenuti è possibile affermare che i valori di performance sono elevati in entrambi i modelli, per esempio i valori di accuracy bilanciata in riferimento alle singole classi della variabile target risultano simili in entrambi i modelli. I valori di precision risultano essere alti e simili in entrambi i modelli, a differenza della precision nella seconda classe, in quanto è più bassa nel modello PCA, la motivazione molto probabilmente è da ricercare nelle 15 istanze di classe 3 classificate come classe 2. La recall è la misura che maggiormente interessa nel nostro dominio applicativo in quanto misura la percentuale di istanze realmente classificate di una classe sul totale delle istanze presenti di quella data classe. In questo caso risulta esserci una differenza notevole tra il valore nella prima e nella seconda tabella per la classe 3, quindi risulta essere una problematica notevole da tenere in considerazione. Tuttavia la spiegazione di questo dato può essere data dalla presenza delle 15 istanze classificate come 2 invece che come 3. Comunque queste istanze non sono classificate come normali ma riamangono classificate come sospette. Infine la f-measure viene utilizzata per stimare il valore ottimale considerando sia precision che recall. Vediamo una leggera differenza solo nel valore della classe 3, questo può essere ricondotto alla diversa misurazione di recall registrata.

Successivamente abbiamo condotto un'indagine tramite 10-fold cross validation su entrambi i modelli per verificare se questa migliora le prestazioni e per effettuare un'analisi della curva ROC e del valore di AUC.

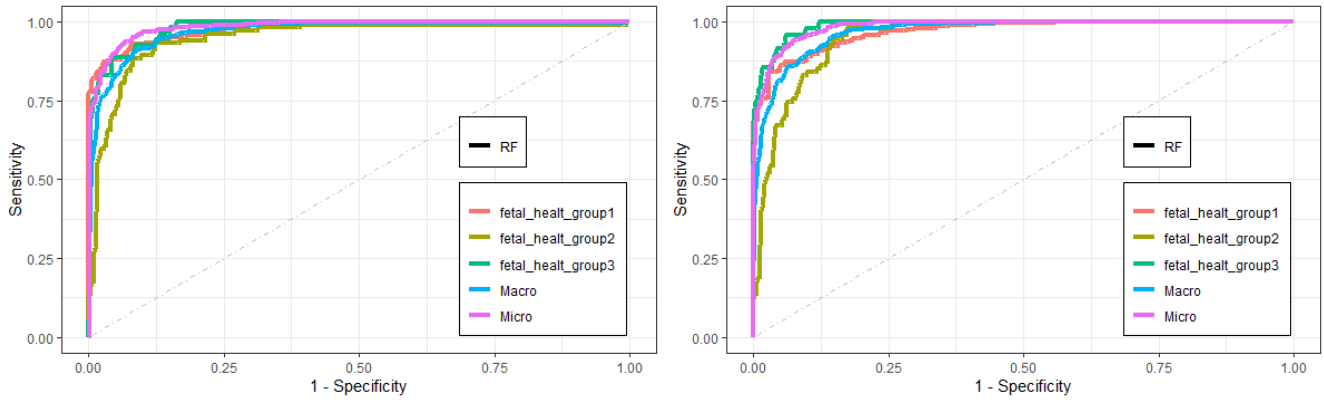


Figura 4.6: Dataset normale (destra) e dataset ottenuto da PCA (sinistra)

Come si può vedere dai due grafici le prestazioni dei modelli sono quasi sovrapponibili e in entrambi i casi sono di buona qualità. È possibile inoltre confrontare i valori ottenuti di AUC:

	1	2	3
Dataset normale	0.9796913	0.9482208	0.9807156
Dataset dopo PCA	0.9694743	0.9513101	0.9891138

Tabella 4.8: Valori di AUC ottenuti da modelli pca

Anche la tabella con i valori di AUC conferma che le prestazioni dei due modelli sono simili e di buona qualità.

In conclusione è quindi possibile affermare che il modello allenato con il dataset ottenuto tramite PCA risulta essere di qualità buona se confrontato con il modello originale ed è quindi preferibile perchè riduce la complessità computazionale.

Capitolo 5

Dataset sbilanciato

5.1 Introduzione

Analizzando la variabile target *fetal_health*, notiamo che abbiamo un dataset fortemente sbilanciato.

- Normale (1): 1655 istanze
- Sospetto (2): 295 istanze
- Patologico (3): 176 istanze

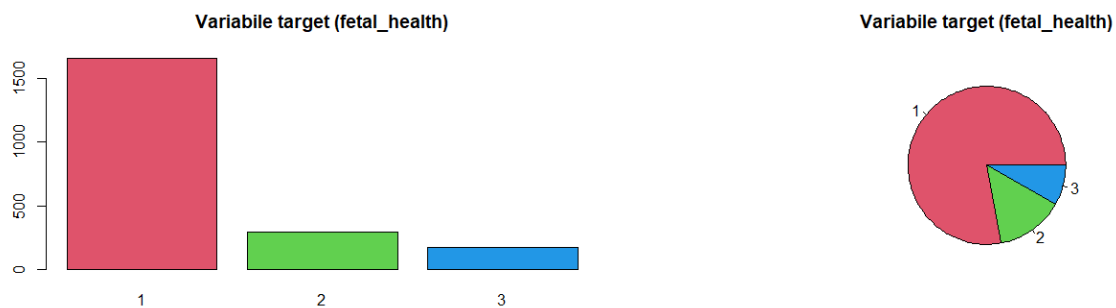


Figura 5.1: Distribuzione variabile target fetal_healt

Addestrando dei modelli di Machine Learning utilizzando un dataset molto sbilanciato si rischia di ottenere delle buone prestazioni determinate dal fatto che la maggior parte delle istanze della classe predominante verranno classificate correttamente. Questo non vuol dire però che le istanze delle classi di minoranza siano predette correttamente.

Dato che il nostro dataset riguarda la previsione di patologie mediche, riteniamo molto importante che vengano rilevate quasi la totalità delle anomalie. Questo a scapito del fatto che ci potranno essere delle istanze appartenenti alla classe "sana" che verranno classificate come "sospette" o "patologiche".

5.2 Possibili approcci

Esistono alcuni metodi per affrontare il problema del dataset sbilanciato che agiscono sul dataset stesso. Tra questi troviamo:

- *undersampling*: consiste nel campionamento dalla classe maggioritaria al fine di conservare solo una parte di questi punti
- *oversampling*: consiste nel replicare casualmente alcuni punti della classe di minoranza al fine di aumentarne la cardinalità
- *generating synthetic data*: consiste nel creare nuovi punti sintetici dalla classe di minoranza per aumentarne la cardinalità (smote)

Esiste anche un'altra tecnica che effettua una *classificazione basata sui costi*. Finora abbiamo assunto che entrambi i tipi di errore ("falso positivo" e "falso negativo") abbiano lo stesso costo. Ovvero abbiamo assunto che prevedere "sano" quando l'etichetta vera è "patologico" e prevedere "patologico" quando l'etichetta vera è "sano" abbiano la stessa gravità. Gli errori sono quindi simmetrici. Come abbiamo detto prima, per noi è più grave classificare un soggetto come "sano" quando nella realtà è "patologico". La gravità degli errori che commettiamo non deve essere simmetrica.

5.3 Realizzazione

Inizialmente abbiamo svolto dei test utilizzando un apprendimento "tradizionale". Per brevità è stato riportato solamente l'esecuzione e l'analisi svolta sui modelli basic senza miglioramenti, dato che l'obiettivo di questa parte è comprendere il funzionamento delle diverse tecniche per affrontare il problema di dataset sbilanciato.

5.3.1 Decision tree - Apprendimento standard

	1	2	3
1	459	21	10
2	16	79	1
3	12	2	42

	1	2	3
1	414	12	3
2	57	86	1
3	16	4	49

	1	2	3
1	418	13	6
2	40	82	2
3	29	7	45

	1	2	3
1	300	5	0
2	104	92	3
3	83	5	50

Tabella 5.1: Confronto tra le matrici di confusione ottenute dall'applicazione sul dataset originale dei vari metodi per risolvere la problematica del dataset sbilanciato (originale - oversampling - undersampling - classificazione con pesi)

	1	2	3
Accuracy	0.8713	0.8715	0.88434
Precision	0.9367	0.8229	0.75000
Recall	0.9425	0.7745	0.79245
F-Measure	0.9396	0.7980	0.77064

	1	2	3
Accuracy	0.8767	0.8679	0.94529
Precision	0.9650	0.5972	0.71014
Recall	0.8501	0.8431	0.92453
F-Measure	0.9039	0.6992	0.80328

	1	2	3
Accuracy	0.8679	0.8631	0.89397
Precision	0.9565	0.6613	0.55556
Recall	0.8583	0.8039	0.84906
F-Measure	0.9048	0.7257	0.67164

	1	2	3
Accuracy	0.7919	0.8519	0.89700
Precision	0.9836	0.4623	0.36232
Recall	0.6160	0.9020	0.94340
F-Measure	0.7576	0.6113	0.52356

Tabella 5.2: Confronto tra le misure di performance ottenute con il dataset originale dei vari metodi per risolvere la problematica del dataset sbilanciato (originale - oversampling - undersampling - classificazione con pesi)

Possiamo osservare tramite le matrici di confusione che tutte le tecniche utilizzate ci consentono di ottenere dei miglioramenti. Infatti, le istanze realmente appartenenti alla classe 3 che vengono erroneamente predette appartenenti ad altre classi diminuiscono. Mettendo a confronto le misure di performance possiamo dire che:

- per le classi 2 e 3: la precision diminuisce, mentre la recall aumenta. Questo significa che le classi sono meglio rilevate rispetto a prima, ma aumentano i punti delle altre classi che contengono al loro interno. Ovvero, diminuiscono i falsi negativi e aumentano i falsi positivi.
- per la classe 1: la precision aumenta e la recall diminuisce. Questo significa che il modello diventa meno in grado di rilevare bene la classe, ma aumenta l'affidabilità di quando lo fa. Ovvero, aumentano i falsi negativi e diminuiscono i falsi positivi.

Riassumendo, ci saranno più istanze appartenenti nella realtà alla classe 1 che vengono sbagliate, ma lo consideriamo come un errore meno grave. Tra tutte le misure, quella più adatta al nostro obiettivo è risultata essere la classificazione basata sui costi. Ci permette calibrare l'apprendimento in modo tale che vengano considerati più gravi gli errori commessi sbagliando la previsione di un'istanza appartenente alla classe 3. Bisogna prestare attenzione però a scegliere in modo appropriato i costi, in modo da avere un buon compromesso tra il miglioramento sulla classe 3, senza peggiorare eccessivamente sulla 1.

Lo stesso procedimento è stato svolto anche sul dataset ottenuto mediante la PCA. I risultati generali rispecchiano quanto appena detto, anche se con un'accentuazione minore.

	1	2	3
1	471	48	9
2	14	52	16
3	2	2	28

	1	2	3
1	411	22	2
2	58	72	14
3	18	8	37

	1	2	3
1	390	12	2
2	74	74	11
3	23	16	40

	1	2	3
1	309	3	1
2	115	83	11
3	63	16	41

Tabella 5.3: Confronto tra le matrici di confusione ottenute dall'applicazione sul dataset ottenuto mediante PCA dei vari metodi per risolvere la problematica del dataset sbilanciato (originale - oversampling - undersampling - classificazione con pesi)

	1	2	3
Accuracy	0.7997	0.7271	0.76076
Precision	0.8920	0.6341	0.87500
Recall	0.9671	0.5098	0.52830
F-Measure	0.9281	0.5652	0.65882

	1	2	3
Accuracy	0.8552	0.7840	0.84425
Precision	0.9653	0.4654	0.50633
Recall	0.8008	0.7255	0.75472
F-Measure	0.8754	0.5670	0.60606

	1	2	3
Accuracy	0.8446	0.7863	0.82699
Precision	0.9448	0.5000	0.58730
Recall	0.8439	0.7059	0.69811
F-Measure	0.8915	0.5854	0.63793

	1	2	3
Accuracy	0.8043	0.7902	0.81973
Precision	0.9872	0.3971	0.34167
Recall	0.6345	0.8137	0.77358
F-Measure	0.7725	0.5338	0.47399

Tabella 5.4: Confronto tra le misure di performance ottenute con il dataset ottenuto mediante PCA applicando i vari metodi per risolvere la problematica del dataset sbilanciato (originale - oversampling - undersampling - classificazione con pesi)

5.3.2 Decision tree - 10-fold cross validation

Abbiamo testato anche le varie tecniche di risoluzione per il dataset sbilanciato, utilizzando la 10-fold corss validation. Abbiamo fatto un controllo sul training applicando una 10 fold cross validation, facendo cinque ripetizioni. Di seguito vengono riportati i grafici delle curve ROC ottenute applicando i vari metodi.

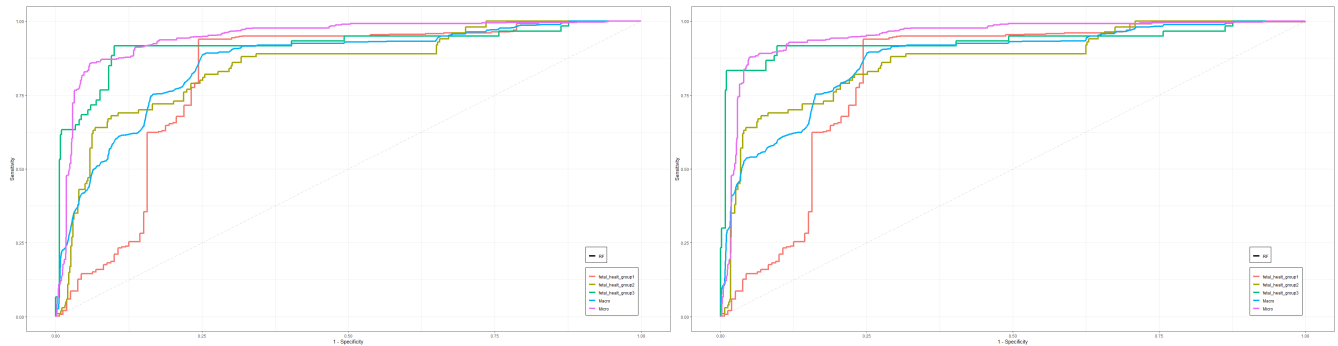


Figura 5.2: Dataset sbilanciato - Originale (sinistra) - Classificazione basata su pesi (destra)

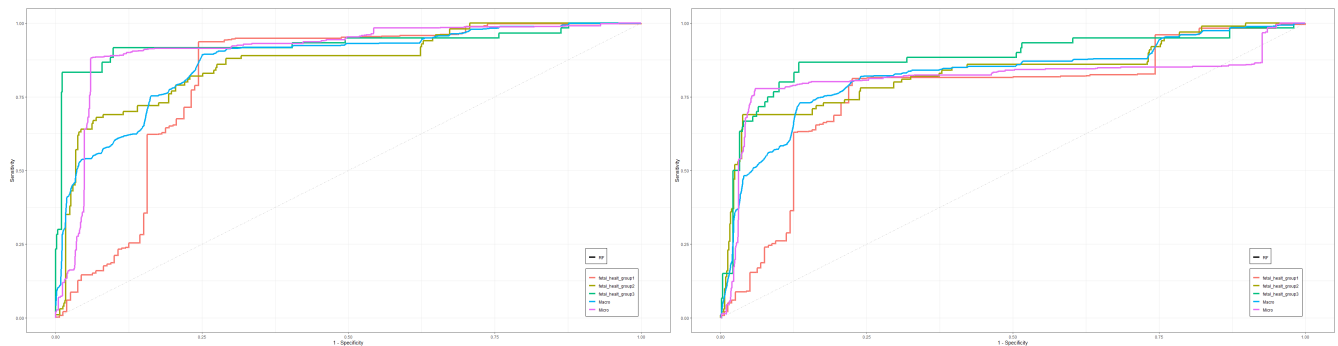
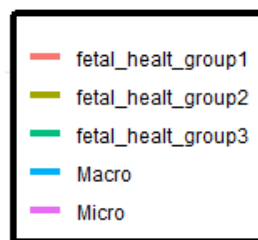


Figura 5.3: Dataset sbilanciato - Oversample (sinistra) - Undersample (destra)



Osservando i vari grafici, possiamo fare alcune osservazioni:

- La classificazione con i pesi è leggermente migliore rispetto a quella originale. Soprattutto per quanto riguarda la classe 3, quindi porta il miglioramento che cercavamo.
- Anche l'oversample porta dei miglioramenti, però bisogna prestare attenzione quando si utilizza questa tecnica, perché replicando molte istanze delle classi minoritarie il modello potrebbe andare in overfitting.
- L'undersample porta ad un peggioramento. Questo è dovuto al fatto che, avendo un dataset molto sbilanciato, viene ridotto parecchio il numero delle istanze su cui viene fatto il train del modello.

	1	2	3
Originale	0.815	0.848	0.918
Weight	0.819	0.866	0.932
Oversampling	0.818	0.866	0.931
Undersampling	0.763	0.832	0.881

Tabella 5.5: Valori AUC (originale - oversampling - undersampling - classificazione con pesi)

Capitolo 6

Esperimento

6.1 Confronto tra SVM e decision tree

Per capire qual è il miglior modello che si può fittare su un particolare tipo di dataset, bisogna fare una comparazione tra le varie misure di performance, l'accuratezza, fmeasure, precision e recall tra i vari classificatori. Abbiamo fatto un controllo sul training applicando una 10 fold cross validation, facendo tre ripetizioni, sia con decision tree che con SVM, utilizzando il dataset ottenuto tramite la PCA, utilizzando l'apprendimento pesato. Successivamente abbiamo effettuato la previsione sul testset dei modelli allenati e abbiamo calcolato la matrice di confusione di entrambi i modelli e le rispettive misure di performance.

	1	2	3
1	492	28	4
2	16	57	6
3	5	6	48

	1	2	3
1	483	56	7
2	26	22	10
3	4	13	41

Tabella 6.1: Confronto tra le matrici di confusione ottenute da SVM (sinistra) e decision tree (destra)

Dall'analisi delle matrici di confusione non si segnalano grosse problematiche, se non la previsione di un buon numero (28 nella prima e 56 nella seconda) di elementi di classe 2 come istanze di classe 1. Nella seconda inoltre c'è un numero elevato di istanze di classe 3 classificate come istanze di classe 2, che tuttavia non creano grossi problemi, in quanto rimangono istanze catalogate come malate.

Di seguito sono riportate le misure di performance calcolate:

	1	2	3		1	2	3
Accuracy	0.8721	0.7939	0.90469	Accuracy	0.7594	0.58936	0.83938
Precision	0.9389	0.7215	0.81356	Precision	0.8846	0.37931	0.70690
Recall	0.9591	0.6264	0.82759	Recall	0.9415	0.24176	0.70690
F-Measure	0.9489	0.6706	0.82051	F-Measure	0.9122	0.29530	0.70690

Tabella 6.2: Confronto tra le misure di performance ottenute un modello SVM e un decision tree.

La prima cosa che balza all'occhio dall'osservazione di queste misure sono i valori bassi di recall e precision per la classe 2 nella previsione utilizzando decision tree. Questo dato è riconducibile al gran numero di istanze di classe 2 sbagliate (come segnalato dalla matrice di confusione). Inoltre sarebbe auspicabile aumentare i valori di precision e recall di entrambi i modelli per la classe, anche sacrificando qualità della previsione della classe 1.

Successivamente abbiamo generato le curve ROC per approfondire l'analisi delle prestazioni dei modelli.

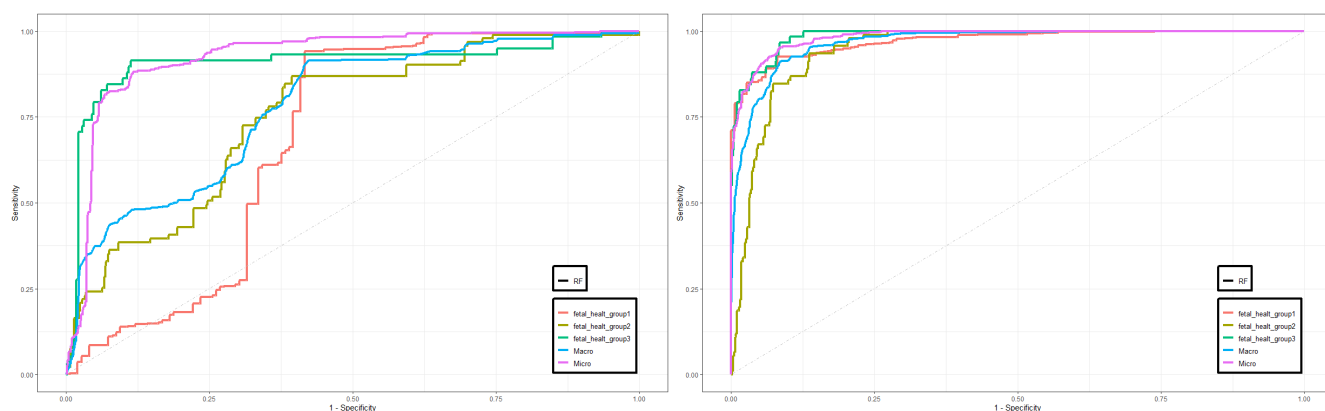


Figura 6.1: Curve ROC 10-fold con dataset prodotto con la PCA, utilizzando l'apprendimento pesato, con i modelli migliori. (sinistra - decision tree, destra - SVM)

Osservando i grafici delle curve ROC, possiamo fare le seguenti affermazioni:

- In entrambi la curva riguardante la classe 3, risulta essere molto buona.
- Nel decision tree, le classi 1 e 2 non sono molto buone. Possiamo derivare che allenare gli alberi di decisione sul dataset prodotto dalla PCA in combinazione con i pesi, porta un peggioramento delle prestazioni per le classi alle quali diamo meno importanza.
- In SVM, tutte le classi presentano in generale un buon risultato. Otteniamo come desiderato, prestazioni superiori sulla classe 3.

Di seguito è riportata una tabella contenente le misure AUC per i due modelli. Queste confermano quanto scritto sopra.

	1	2	3
Decision tree	0.683	0.753	0.908
SVM	0.972	0.948	0.986

Tabella 6.3: Valori AUC (Decision Tree e SVM)

Infine abbiamo comparato le statistiche generate dalle misure di performance.

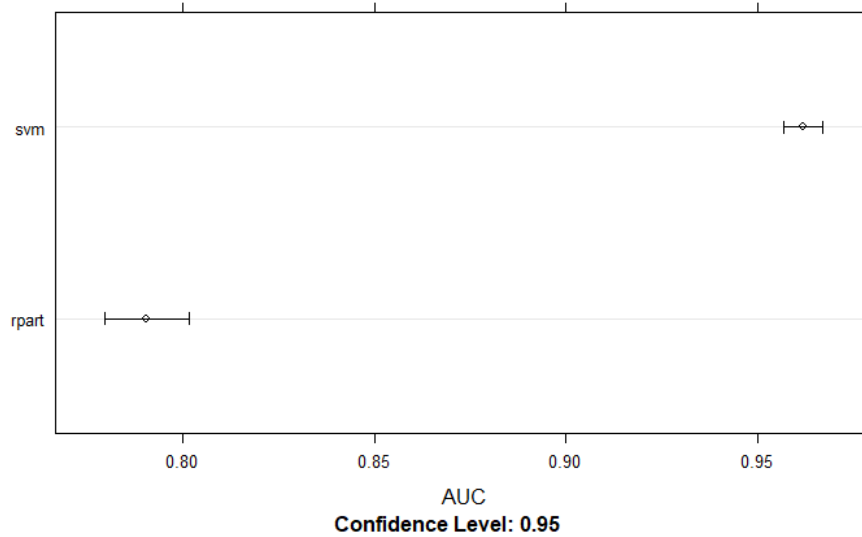


Figura 6.2: Dotplot AUC

Come emergeva anche dalle curve ROC, abbiamo una misura di AUC più elevata in SVM rispetto che in decision tree, la differenza tra la misura nei due modelli è abbastanza elevata.

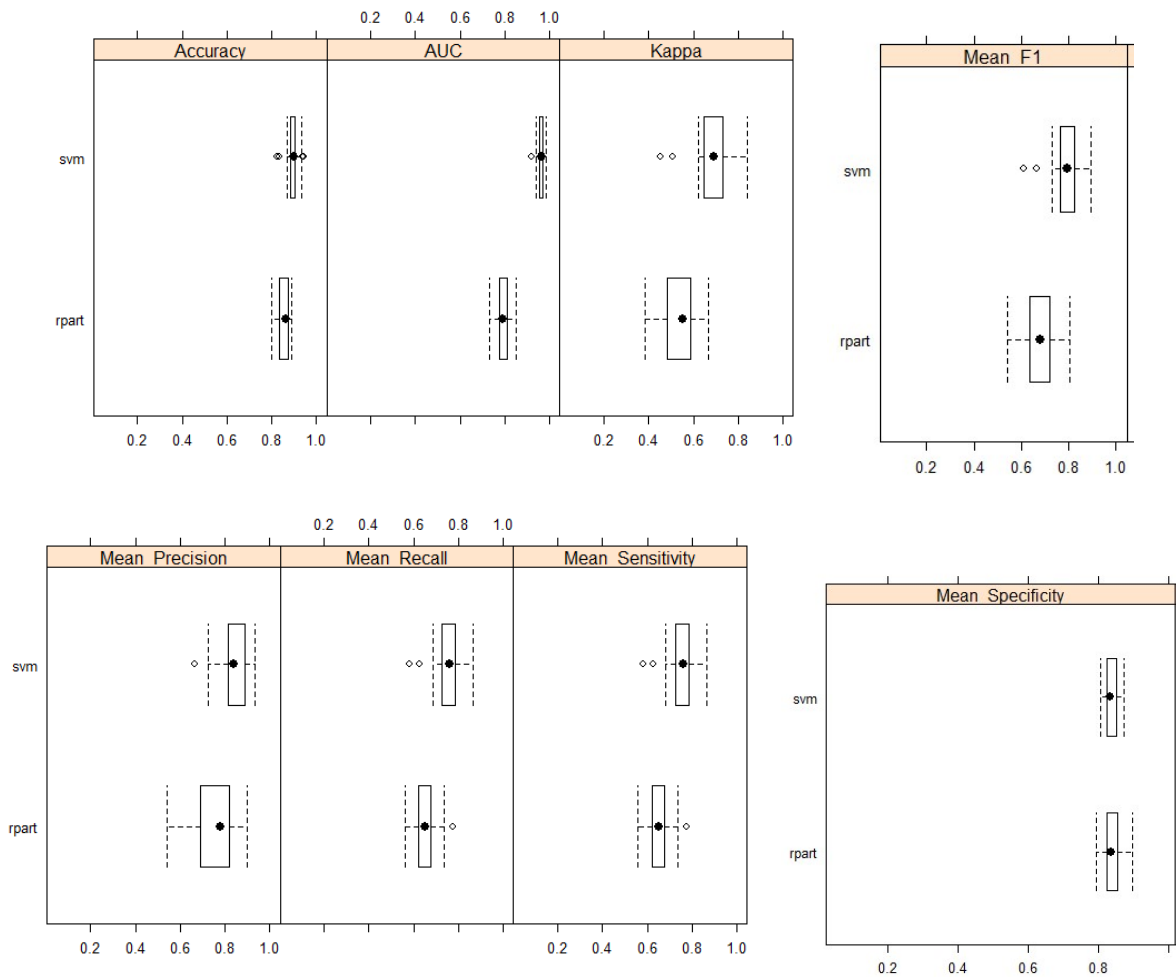


Figura 6.3: Bwplot

L'analisi individuale delle analisi di performance conferma quanto emerso dalle analisi precedenti: il modello creato sfruttando SVM è migliore in questi termini del modello creato sfruttando decision tree.

Dobbiamo sempre considerare il costo computazionale per indurre un modello di classificazione. Riportiamo di seguito l'analisi delle tempistiche:

	Everything	FinalModel
svm	176.48	0.11
rpart	8.19	0.02

Tabella 6.4: Confronto tempistiche Decision Tree e SVM

Per SVM otteniamo un costo più alto, perché deve determinare quelli che sono gli iperpiani ottimali. La capacità di apprendimento dell'albero di decisione è molto più semplice. Questo aspetto è molto importante da tenere in considerazione, soprattutto in condizione di una distanza così notevole tra i due valori.

Capitolo 7

Conclusione

7.1 Confronto tra i modelli di albero proposti

Riassumiamo ora alcuni risultati ottenuti facendo dei confronti tra i vari modelli di apprendimento.

Albero normale vs albero pruned (Gini index)

- L'albero normale ottiene complessivamente delle buone performance.
- L'albero pruned diminuisce la complessità del modello rispetto al precedente, ma riscontra una leggera diminuzione delle performance. Questo riguarda anche la terza classe, infatti, i falsi negativi (che noi vogliamo cercare di diminuire) aumentano.
- Abbiamo visto successivamente delle tecniche che ci permettono di assegnare delle pesistiche ai vari errori commessi. Quindi il problema dei falsi negativi per l'albero pruned verrebbe migliorato.

Albero normale vs albero pruned (information gain)

Anche utilizzando come criterio di splitting l'information gain possiamo fare le considerazioni precedenti. Inoltre, abbiamo riscontrato che otteniamo risultati migliori utilizzando Gini Index rispetto ad Information Gain.

7.2 Dataset normale vs dataset dopo PCA

Confrontando i risultati ottenuti utilizzando il dataset originario e quello prodotto a seguito della PCA, possiamo evidenziare che:

- I modelli allenati con il dataset originario hanno in generale buone performance.
- I modelli allenati con il dataset prodotto dalla PCA mantengono comunque delle buone performance, anche se inferiori rispetto ai precedenti.
- Addestrando i modelli con il dataset prodotto dalla PCA otteniamo sicuramente modelli meno complessi, quindi che tendono ad andare meno in overfitting. Per questo, se le performance rimangono buone è preferibile utilizzare un modello più semplice.
- In entrambi i casi si riscontrano problemi che derivano dallo sbilanciamento del dataset. In particolare, sono presenti falsi negativi. Per soddisfare gli obiettivi che ci siamo posti, abbiamo analizzato le varie tecniche per la risoluzione del problema e abbiamo osservato che entrambi reagiscono bene, in particolare, con la classificazione basata sui costi.

7.3 Conclusioni finali

Considerando i risultati ottenuti dai vari modelli e l'esperimento condotto nel capitolo 6 è possibile trarre alcune conclusioni circa il modello più efficace per lo scopo di questo progetto. Come ampiamente sottolineato l'utilizzo della tecnica della PCA aiuta tutti i modelli testati perchè riduce la complessità computazionale e riduce il rischio di incorrere nell'overfitting. Dal confronto tra i due modelli proposti nell'esperimento non si nota una grande differenza di performance. Anche se di poco, le performance ottenute dall'utilizzo della SVM superano quelle ottenute dall'albero di decisione. Di contro però, la complessità della SVM potrebbe essere maggiore, come testimoniato dall'analisi dei timing dei due modelli (Tab. 6.4). E' possibile trascurare queste differenze, in quanto il tempo di SVM nonostante sia maggiore di quello del decision tree rimane accettabile ed è necessario per ottenere performance migliori. Considerando il campo di applicazione del nostro modello, ovvero delle previsioni mediche che possono essere fondamentali per il rilevamento di casi patologici, si potrebbe preferire un modello più preciso anche se leggermente più complesso. In conclusione è possibile affermare che il modello migliore testato per questo scopo è la SVM.