Stile di vita e rischio ictus: analisi descrittiva e predittiva su un dataset sbilanciato

Davide Dell'Orto[1], Matteo Lanzillotti[1]

Abstract

L'ictus è una delle principali cause di morte nel mondo, seconda alle cardiopatie, stando ai dati pubblicati dall'OMS nel 2019[1]. Per prevenirlo è importante tenere sotto controllo i principali fattori di rischio. Come riporta il sito *humanitas.it,* il 10-20% delle persone colpite da ictus per la prima volta muore entro un mese ed un altro 10% entro il primo anno. Fra le restanti, circa un terzo sopravvive con un grado di disabilità spesso elevato, tanto da renderle non autonome; un terzo circa presenta un grado di disabilità lieve o moderata che gli permette spesso di tornare al proprio domicilio in modo parzialmente autonomo ed un terzo, i più fortunati o comunque coloro che sono stati colpiti in forma lieve, tornano autonomi al proprio domicilio [1]. Riconoscere i soggetti più a rischio, monitorarli ed intervenire al più presto in caso di un episodio ischemico può fermare il danneggiamento dei tessuti, migliorando nettamente la qualità della vita. L'obiettivo della seguente analisi è quello di costruire un modello di classificazione che, sulla base di alcuni parametri vitali, possa classificare correttamente i pazienti a rischio ictus, cercando di aggirare il problema dato da un dataset sbilanciato tramite le comuni tecniche di ricampionamento e di classificazione. Inoltre, ai fini di un'analisi puramente descrittiva è stata ricercata, tramite tecniche di clustering, la presenza di cluster naturali all'interno del dataset, per comprendere se a pazienti con tipologie di lavoro simili fossero associati gli stessi valori dei parametri vitali.

Indice

Introduzione **1**
1. Descrizione del dataset **1**
1.1. Preprocessing 2
2. Prima domanda di ricerca **2**
2.1. I classificatori utilizzati 2
3. Metriche per la misura delle performance **2**
4. Tecniche utilizzate **3**
4.1. Cost sensitive learning 3
4.2. Oversampling e undersampling 4
4.3. Oversampling e feature selection 4
4.4. Undersampling + oversampling 4
5. Ottimizzazione dei parametri **4**
6. Confronto con altri modelli **5**
7. Seconda domanda di ricerca **6**
7.1. Definizione dei cluster 6
7.2. Limiti del clustering 7
Conclusioni **7**
Riferimenti **7**

Introduzione[2]

L'ictus cerebrale è causato dell'improvvisa chiusura o rottura di un vaso cerebrale e dal conseguente danno alle cellule cerebrali dovuto alla mancanza di ossigeno e di nutrienti portati dal sangue (ischemia) o alla compressione dovuta al sangue uscito dal vaso (emorragia cerebrale). La caratteristica principale dell'ictus è la sua comparsa improvvisa, solitamente senza dolore. Il principale fattore di rischio è la pressione alta, seguita da fumo di tabacco, obesità, colesterolo alto, diabete mellito, un precedente TIA[3] e fibrillazione atriale. L'ictus ischemico è causato generalmente dall'ostruzione di un vaso sanguigno; l'ictus emorragico, invece, dal sanguinamento intracranico generalmente in seguito alla rottura di un aneurisma cerebrale. Il ministero della salute stima che ogni anno si verificano circa 196.000 episodi di arresto cerebrale, posizionando così l'ictus *tra le prime tre cause di morte nel mondo*. Un modo per prevenire episodi simili è quello di diminuire i fattori di rischio sopra elencati. L'obiettivo principale di questo studio è quello di creare un modello che, sulla base dei dati a disposizione, possa con una certa precisione classificare correttamente i soggetti più a rischio.

1. Descrizione del dataset

Il dataset "*healthcare-dataset-stroke-data*", oggetto di quest'analisi, proviene dal sito Kaggle.com [2]. Esso è composto da 5110 istanze e 12 variabili. Ogni istanza

[1] Università degli Studi di Milano-Bicocca, CdLM Data Science.
[2] Le informazioni contenute in questo paragrafo sono state estratte e riportate dal sito del Ministero della Salute "www.salute.gov.it".
[3] Attacco ischemico transitorio.

rappresenta un paziente e contiene le seguenti informazioni:
- ***id***: codice identificativo;
- ***gender***: sesso del paziente;
- ***age***: età;
- ***hypertension***: variabile binaria che indica se il paziente è soggetto ad ipertensione;
- ***heart_disease***: variabile binaria che indica se il paziente ha problemi di cuore;
- ***ever_married***: variabile binaria che indica se il paziente è o è stato sposato;
- ***work_type***: variabile che indica la tipologia di lavoro in base a 3 macrocategorie (private, self-employed o government job) ma che assume anche valori come "children" e "never worked" nel caso il soggetto, rispettivamente, è un bambino o non ha mai lavorato;
- ***Residence_type***: luogo di residenza (rurale o urbano);
- ***avg_glucose_lvl***: livello medio di glucosio nel sangue;
- ***bmi***: indice di massa corporea;
- ***smoking_status***: indica se il paziente è o è mai stato fumatore;
- ***stroke***: variabile binaria che indica se il paziente ha avuto un episodio di ictus.

1.1. Preprocessing

Si è provveduto a rimuovere la variabile *id* in quanto irrilevante ai fini dell'analisi. Inoltre, il dataset presentava 201 valori mancanti della variabile *bmi* e si è optato per una sostituzione tramite interpolazione. Tale scelta principalmente per due motivi: in primo luogo la variabilità dei valori assunti (deviazione standard pari a 7.9), in secondo luogo la necessità di mantenere le osservazioni, in quanto l'eliminazione di 201 righe avrebbe comportato un'importante perdita di dati, considerata la dimensione contenuta del dataset. Infine si è rimossa un'osservazione in cui la variabile *gender* assumeva il valore *"other"* rendendola di fatto binaria.

2. Prima domanda di ricerca

La prima domanda a cui si vuole rispondere pone come variabile target *stroke*. Si vuole infatti costruire un modello di classificazione che possa prevedere il rischio di ictus sulla base delle informazioni a disposizione.

2.1. I classificatori utilizzati

Per costruire un modello accurato si andranno a valutare cinque classificatori:
- ***Naive Bayes***: modello probabilistico che si basa sull'assunzione di indipendenza condizionale degli attributi;
- ***Support Vector Machine (SVM)***: modello che classifica le osservazioni in base ad una separazione in uno spazio n-dimensionale;
- ***Multilayer Perceptron (MLP)***: rete neurale artificiale semplice che si basa sul collegamento tra nodi (neuroni appunto) i quali, una volta ricevuto l'input pesato applicano una funzione detta "di attivazione". L'individuazione dei corretti pesi avviene, con approccio *trial and error*, tramite *backward propagation*;
- ***J48***: fa parte dei cosiddetti alberi decisionali che, sulla base di diversi indici come ad esempio l'entropia, sviluppano a cascata dei collegamenti tra nodi classificando l'istanza come la classe più frequente nell'ultimo nodo;
- ***Logistic regression***: modello di regressione non lineare che classifica le istanze misurando l'effetto di ciascun input sulla variabile target.

La scelta del miglior classificatore sarà basata sulle loro performance, che saranno confrontate sulla base di cinque diverse metriche che verranno introdotte tra poco.

3. Metriche per la misura delle performance

Introduciamo il concetto di matrice di confusione (Tabella 1) che riassume i risultati di un modello di classificazione nel modo seguente:

		Prediction	
		-1	1
Actual class	-1	TN	FP
	1	FN	TP

Tabella 1. Matrice di confusione.

L'accuracy è la misura principale utilizzata in materia per valutare la bontà di un classificatore, ed è definibile, facendo riferimento alla matrice qui sopra, con la seguente formula:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

dove il numeratore rappresenta il numero di classificazioni corrette mentre il denominatore il totale delle classificazioni.

In caso di dataset sbilanciati, che come presto vedremo è il caso in cui ci troviamo, l'accuracy risulta essere una misura poco utile per comprendere la performance dei classificatori, in quanto questi tenderanno ad assegnare a tutte le istanze la classe più frequente, ottenendo sempre valori di accuracy molto alti. In questo caso si dice che il classificatore è uno *zeroR*. Come misure alternative all'accuratezza sono state prese in considerazione *recall* e *precision*, così definite:

$$Recall = \frac{TP}{TP + FN}$$

Indica quante istanze positive sono state classificate correttamente (numeratore) rispetto al totale (denominatore).

$$Precision = \frac{TP}{TP + FP}$$

Indica quante volte il classificatore dice che un'istanza è positiva (denominatore) e quante volte questo si realizza davvero (numeratore). In particolare *recall* e *precision* sono legate da un trade-off in quanto

aumentare una significa diminuire l'altra. Sono quindi generalizzabili tramite un'unica misura, l'*F1-Measure,* ovvero la media armonica tra le due:

$$F1\text{-}Measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

Con un occhio anche all'accuratezza, l'obiettivo sarà la massimizzazione di queste misure ed in particolare la minimizzazione dei falsi negativi: in un contesto come quello medico, soprattutto se si tratta di identificare il rischio di ictus, la priorità è sicuramente quella di individuare il maggior numero possibile di classi positive anche a costo di includere più falsi positivi. A tal fine si andranno a confrontare i classificatori anche sulla base della *curva ROC*, ovvero quella curva che relaziona la percentuale di true positive con quella di false positive, e della rispettiva area sottesa *AUC (Area Under Curve)*.

4. Tecniche utilizzate

Come abbiamo anticipato, la variabile target ***stroke*** presenta un forte sbilanciamento, causando non pochi problemi per una corretta classificazione: sono infatti solamente 247 le osservazioni in cui questa assume il valore 1 (yes) contro i restanti 4863 0 (no).

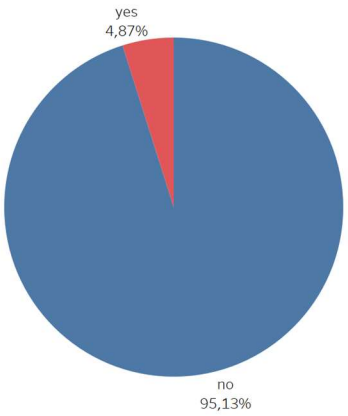

Figura 1. Distribuzione della variabile *stroke*.

Questo sbilanciamento rappresenta il principale problema di questa analisi in quanto dovranno essere adottate le giuste tecniche per fare in modo che i classificatori massimizzino la loro utilità. Ai fini dell'analisi preliminare sono stati testati i classificatori sopra nominati sul dataset partizionato, tramite campionamento stratificato, in *training* e *test set*, rispettivamente 70% e 30%, ottenendo i risultati presentati nella tabella seguente:

	Recall	Precision	F1-Measure	Accuracy
NB	0.32	0.161	0.214	0.885
SVM	0	/	/	0.951
MLP	0	0	/	0.942
J48	0	/	/	0.951
Logistic	0	/	/	0.951

Tabella 2. Performance con dataset sbilanciato e parametri di default.

Come possiamo vedere, l'accuracy ottenuta è piuttosto alta con tutti i classificatori, tuttavia, come detto, in questo caso non è un buon risultato: quattro classificatori su cinque si comportano come classificatori zeroR. I risultati ottenuti dal Naive Bayes, pur riescendo a classificare correttamente 24 positivi su 75, non sono comunque incoraggianti.

Di seguito sono quindi illustrate le metodologie testate per superare i problemi dati dallo sbilanciamento delle classi e migliorare le performance di modelli.

4.1. Cost sensitive learning

Il primo approccio utilizzato consiste nell'introduzione di una matrice di costo. Questa permette di associare ad un errore di classificazione un costo, in particolare quando il modello individua falsi negativi, con l'obiettivo di attribuire maggiore rilevanza alla classe minoritaria, minimizzando quindi il costo totale. Per fare questo si è utilizzato il nodo CostSensitiveClassifier fornito dal modulo Weka che consente di inserire i parametri della matrice di costo. Per assicurarsi della plausibilità dei risultati ottenuti è stata applicata, solo sul training set, una *10-folds Cross Validation* ottenendo così un *validation set*. Nello specifico, la Cross-Validation garantisce che ogni osservazione rientri 9 volte nel sottoinsieme di training ed una volta in quello di validation.

L'individuazione dei parametri ottimali della matrice di costo è avvenuta tramite forza bruta, partendo da un rapporto 1:20 (rispettivamente false positive e false negative) che rispecchia lo sbilanciamento delle classi, e facendo variare i valori osservando ogni volta la performance dei classificatori. Nella tabella seguente sono riassunte le performance ottenute con tale approccio:

	Recall	Precision	F1	Accuracy	AUC
NB	0.853	0.133	0.229	0.720	0.837
SVM	0.840	0.147	0.250	0.753	0.794
MLP	0.507	0.193	0.279	0.872	0.801
J48	0.320	0.152	0.206	0.879	0.616
Logistic	0.813	0.147	0.249	0.760	0.855

Tabella 3. Performance con cost sensitive learning.

Tutti i classificatori hanno migliorato la loro performance in tutte le misure al di fuori dell'accuratezza. Appare evidente infatti come la diminuzione di quest'ultima sia associata ad un aumento della recall, soprattutto per il Naive Bayes: quest'ultimo ha infatti individuato 64 true positive su 75, ma con poca precisione in quanto i false positive ammontano a 419. Un risultato simile è stato ottenuto dal modello logistico con una recall sì più bassa, ma una precisione maggiore: 61 true positive con 354 false positive. Questo confronto fornisce un'idea del livello di trade-off tra recall e precision.

4.2. Oversampling e undersampling

La seconda metodologia utilizzata per il bilanciamento delle classi riguarda il ricampionamento. Nello specifico sono state adottate due tecniche: l'oversampling, che consiste nella creazione di nuove osservazioni della classe minore e l'undersampling che invece elimina le osservazioni della classe più frequente. Entrambi consentono di ottenere un dataset in cui la proporzione tra le classi è esattamente 50% e 50%. A differenza dell'approccio precedente non si è fatto uso della Cross-Validation ma di un semplice Holdout che ha diviso la prima partizione in un ulteriore 63% e 37%, in modo da ridurre i tempi computazionali anche in previsione dell'ottimizzazione dei classificatori che verrà fatta in seguito.

Per quanto riguarda l'oversampling si è scelto di utilizzare l'algoritmo *SMOTE*[4], presente nativamente in Knime, mentre l'undersampling è stato effettuato tramite il nodo *equal size sampling*. Entrambi sono stati applicati sul training set. Nello specifico, l'undersampling campiona casualmente un sottoinsieme di osservazioni mentre l'algoritmo *SMOTE* ne genera di nuove sfruttando il concetto di *k-nearest neighbors*: per ogni osservazione reale, quella creata giacerà sulla linea (intesa in uno spazio n-dimensionale) che collega quest'ultima con una delle *k* osservazioni più vicine [3].

	Recall	Precision	F1	Accuracy	AUC
NB	0.787	0.137	0.233	0.746	0.843
SVM	0.840	0.149	0.253	0.757	0.796
MLP	0.387	0.136	0.201	0.849	0.755
J48	0.213	0.105	0.140	0.872	0.569
Logistic	0.800	0.144	0.244	0.757	0.855

Tabella 4. Performance con SMOTE.

	Recall	Precision	F1	Accuracy	AUC
NB	0.867	0.132	0.230	0.716	0.837
SVM	0.760	0.096	0.170	0.638	0.696
MLP	0.667	0.104	0.181	0.704	0.768
J48	0.853	0.121	0.212	0.689	0.805
Logistic	0.840	0.137	0.236	0.733	0.853

Tabella 5. Performance con equal size sampling.

Come possiamo vedere, anche questi metodi ci consentono di migliorare le performance dei classificatori. Tuttavia la perdita di dati comportata dal sottodimensionamento del training set e la generazione di osservazioni sintetiche ci portano ad esplorare altre soluzioni prima di scegliere il modello migliore.

4.3. Oversampling e feature selection

Con l'obiettivo di migliorare l'interpretabilità del modello ed eventualmente aumentare le performance si è effettuata, dopo oversampling, una selezione dei parametri tramite *wrapper*: un modello, in questo caso probabilistico (Naive Bayes), effettua una ricerca di sottoinsiemi di attributi tali da massimizzare una determinata funzione. La scelta è ricaduta sulla massimizzazione dell'AUC ed il classificatore, sulla base di questo, ha individuato 3 features rilevanti: *gender*, *age* e *heart_disease*. Questo sottoinsieme di variabili è stato filtrato dal training set ed è stato dato in pasto agli algoritmi di classificazione, che hanno restituito le seguenti performance:

	Recall	Precision	F1	Accuracy	AUC
NB	0.800	0.142	0.240	0.753	0.852
SVM	0.853	0.141	0.242	0.739	0.793
MLP	0.600	0.172	0.268	0.840	0.845
J48	0.560	0.151	0.238	0.825	0.761
Logistic	0.853	0.141	0.242	0.739	0.850

Tabella 6. SMOTE con feature selection.

Si può notare come le prestazioni siano in generale migliori per tutti i classificatori. Nello specifico la feature selection ha comportato un aumento della recall in tutti i modelli. Aumento che però, non essendo necessariamente accompagnato da una precisione maggiore, può spiegare una diminuzione complessiva dell'accuratezza.

4.4. Undersampling + oversampling

Come abbiamo detto, le tecniche di ricampionamento potrebbero comportare una perdita di informazioni, nel caso di undersampling, o un rischio di overfitting nel caso si utilizzi lo SMOTE in quanto le osservazioni ottenute sono appunto sintetiche. Si è deciso quindi, per cercare di mitigare questi due effetti, di allenare i modelli su un dataset ottenuto tramite entrambi gli approcci e di valutare i risultati per confrontarli con i precedenti: nello specifico le osservazioni della classe prevalente sono state dimezzate con un partizionamento casuale, dopodiché quelle della classe rara sono state oggetto di oversampling tramite SMOTE. Questo con l'obiettivo di ridurre al minimo le problematiche viste sopra ed ottenere quindi un dataset che, seppur ricampionato, fosse più simile a quello reale.

	Recall	Precision	F1	Accuracy	AUC
NB	0.787	0.138	0.234	0.748	0.845
SVM	0.827	0.146	0.248	0.755	0.789
MLP	0.613	0.138	0.225	0.793	0.773
J48	0.280	0.095	0.142	0.835	0.642
Logistic	0.800	0.145	0.245	0.759	0.853

Tabella 7. Performance con undersampling + SMOTE.

5. Ottimizzazione dei parametri

Data la volontà di minimizzare i rischi derivanti da un undersampling o un oversampling si è ritenuto, anche in base alle piccole differenze di performance, che i modelli sviluppati adottando entrambe le tecniche di ricampionamento potessero essere i più adeguati a

[4] Synthetic Minority Oversampling Technique.

rispondere alle necessità di questo studio. Inoltre, nonostante per alcuni classificatori l'approccio cost sensitive avesse restituito risultati leggermente migliori, non si è optato per quest'ultimi in quanto si è ritenuto che, in un contesto delicato come quello medico, non fossero sufficientemente affidabili: quando si introduce un metodo basato sul costo, infatti, la priorità non è più la massimizzazione dell'accuratezza ma la minimizzazione del costo, che deve essere imputato nella maniera più corretta possibile. Infatti, pur essendo evidente che i falsi positivi pesano molto di più dei falsi negativi, che in questo caso significa che sbagliare a predire un soggetto a rischio è più grave che includere un soggetto sano, è difficile quantificare questo errore in termini di costo.

I classificatori presentati in Tabella 7 (ad eccezione del Naive Bayes) sono stati quindi oggetto di ottimizzazione: sono stati testati diversi parametri per ogni algoritmo tramite l'utilizzo di un ciclo, con l'obiettivo di individuare una combinazione che potesse massimizzare l'AUC. Per fare questo è stato fornito come valore di riferimento dell'AUC il rapporto tra le due componenti della curva ROC ovvero true positive rate e false positive rate, definito in questo caso come:

$$\frac{recall}{1 - specificity}$$

dove *specificity* è $\frac{TN}{TN+FP}$.

	Recall	Precision	F1	Accuracy	AUC
SVM	0.720	0.126	0.214	0.742	0.731
MLP	0.720	0.132	0.224	0.755	0.820
J48	0.520	0.151	0.234	0.834	0.806
Logistic	0.800	0.145	0.245	0.759	0.853

Tabella 8. Classificatori con ottimizzazione dei parametri.

Si può notare come tutti i modelli, ad esclusione del SVM che ha peggiorato la propria performance, presentino ora valori di AUC uguali o superiori, stando a significare che tramite il ciclo è stato possibile individuare delle buone combinazioni di parametri. Tra tutti i risultati ottenuti, la scelta finale del modello è stata effettuata confrontando l'andamento delle curve ROC del Multilayer Perceptron, del J48 e della regressione logistica con ottimizzazione dell'AUC, e quelle del Naive Bayes e del SVM ottenute applicando sia undersampling che oversampling. Per praticità riassumiamo graficamente i risultati:

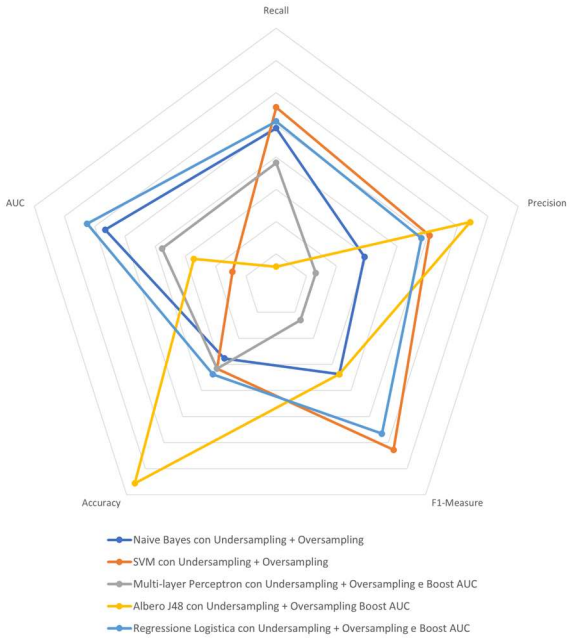

Figura 2. Confronto tra le performance dei classificatori scelti, con dati normalizzati.

A questo punto i classificatori sono stati confrontati anche osservando le rispettive curve ROC. In linea di massima sarebbe preferibile un andamento il più ripido possibile in quanto significherebbe che il modello identifica un alto numero di true positive mantenendo i false positive relativamente contenuti.

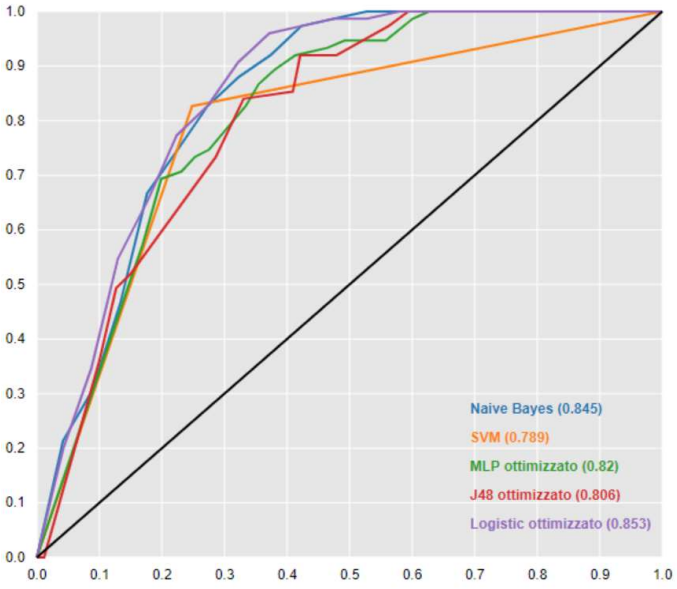

Figura 3. Confronto curve ROC.

Si ritiene migliore il modello che utilizza la regressione logistica in quanto restituisce, per quasi tutto l'asse *x*, valori di *y* superiori agli altri classificatori.

6. Confronto con altri modelli

Per verificare il livello della performance del modello logistico sviluppato in questa sede si sono considerati i risultati ottenuti, sul medesimo insieme di dati, da Sailasya e Kumari in uno studio pubblicato sull'*IJACSA (International Journal of Advanced Computer Science and Applications)* [4].

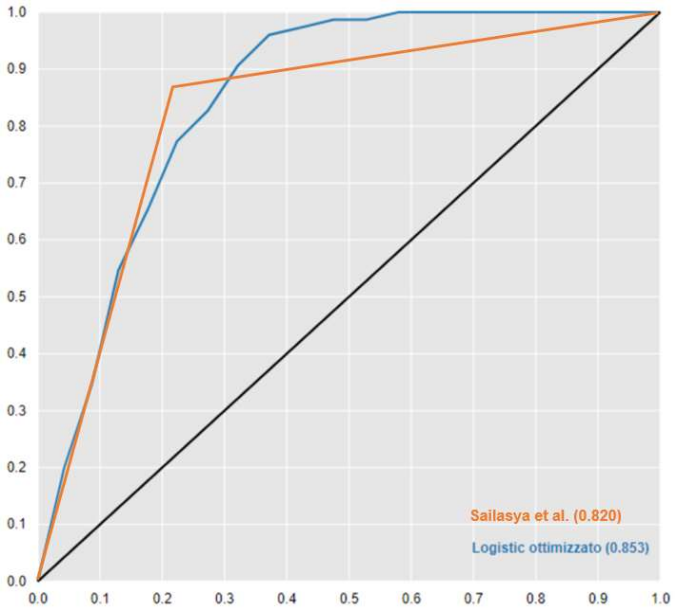

Figura 4. Modelli a confronto.

Si può notare come il modello qui presentato abbia performance migliori quando si tratta di individuare il maggior numero di true positive possibile: nello specifico, se si accettasse di includere il 40% di falsi positivi in linea teorica si potrebbe arrivare ad individuare oltre il 95% dei soggetti a rischio ictus.

7. Seconda domanda di ricerca

Si vuole ora vedere se la tipologia di lavoro che un soggetto svolge sia in qualche modo associata a determinati valori di, ad esempio, bmi o livello medio di glucosio. Infatti, non è del tutto sbagliato pensare che un lavoro ad esempio a livello governativo possa comportare uno stile di vita, e quindi "parametri vitali", diversi da tutti gli altri ma comuni tra loro. Tecnicamente rispondere a questa domanda si traduce nel verificare l'esistenza di cluster naturali della variabile *work_type* dove per cluster si intende gruppi di osservazioni con caratteristiche simili tra di loro.

In questo caso i valori mancanti della variabile bmi non sono stati sostituiti tramite interpolazione ma sono stati semplicemente eliminati. Sono state poi eliminate le righe in cui la variabile *work*_type assumeva il valore "children" e "never_worked" in quanto ritenute irrilevanti in questo contesto. Infine sono state selezionate 4 variabili considerate le più utili ai fini della domanda di ricerca: *hypertension, heart_disease, bmi* e *avg_glucose_level*, campionando poi casualmente solo 2000 osservazioni in modo da ridurre i tempi computazionali.

7.1. Definizione dei cluster

Il problema principale di questa domanda di ricerca riguarda la tipologia mista delle variabili esplicative scelte: due sono binarie mentre altre due sono continue. Le misure di distanza solitamente utilizzate, come banalmente l'Euclidea, non sono quindi adatte in quanto possono trattare dati esclusivamente continui o esclusivamente binari o nominali. Si è quindi optato per la *distanza di Gower*, specifica proprio per queste situazioni: vengono computate le distanze *per ogni variabile* tenendone in considerazione la tipologia (che porta all'uso di diverse metriche), per poi definire la distanza tra le osservazioni come la media (eventualmente pesata) delle distanze precedentemente calcolate. Nello specifico il risultato di questo procedimento non è una matrice delle distanze ma una matrice di dissimilarità. Quest'ultima è stata fornita in input all'algoritmo *PAM*[5], una variante del *k-medoids*[6] che, basandosi sul concetto di medoide, raggruppa le osservazioni in *k* cluster. Volendo verificare l'esistenza di cluster che rispecchino le macrocategorie lavorative, si è definito $k = 3$ in quanto la variabile *work_type* assume ora 3 valori nominali.

L'implementazione del metodo di Gower e dell'algoritmo PAM è stata effettuata su Knime tramite due script R, grazie all'utilizzo della libreria *clusters*. L'algoritmo utilizzato ha prima individuato i medoidi poi ha assegnato ciascuna osservazione del dataframe ad un cluster, restituendo un vettore come output. Grazie all'uso del secondo script R è stato quindi possibile convertire quest'output in un dataframe.
A questo punto, per verificare che le osservazioni fossero state raggruppate in modo tale da condividere la tipologia di lavoro, al dataframe ottenuto è stata aggiunta la colonna della variabile di interesse *work_type*.

In questo modo è stato possibile dimostrare che *i cluster non coincidono con i valori assunti dalla variabile work_type*, come mostrato in Figura 5 che presenta la distribuzione all'interno dei singoli cluster:

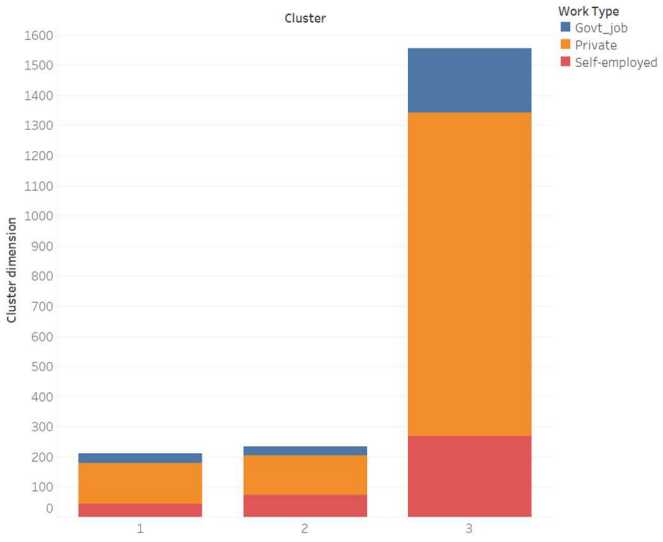

Figura 5. Clustering con PAM.

È possibile concludere dicendo che svolgere un certo lavoro e quindi, semplificando, avere un determinato

[5] Partitioning Around Medoids.

[6] A sua volta un perfezionamento dell'algoritmo *k-means*.

stile di vita, non comporta la similarità tra i principali parametri vitali.

7.2. Limiti del clustering

La breve analisi qui effettuata è chiaramente affetta da limiti e problematiche che, se aggirati, potrebbero condurre a risultati e conclusioni più veritiere ed affidabili.

Tra i problemi riscontrati vi è su tutti l'incapacità dell'algoritmo PAM di individuare cluster che non hanno forma sferica: questo limite potrebbe essere risolto andando a definire *k* superiore a 3 in modo da ottenere una divisione in più cluster per poi raggruppare quelli più vicini tra loro. Inoltre un diverso risultato sarebbe stato possibile ottenerlo considerando non più 4 variabili ma solamente 2 (bmi e livello di glucosio, verosimilmente più adatte a risolvere il problema posto in questa sede) potendo così utilizzare misure di distanza differenti ed algoritmi che, preferibilmente, siano in grado di garantire un raggruppamento non sferico.

Conclusioni

I risultati ottenuti in questo studio, soprattutto nella prima domanda di ricerca, sono ritenuti soddisfacenti anche alla luce del limitato numero di dati a disposizione. Nonostante ciò è evidente come i modelli sviluppati in questa sede siano solamente un punto di partenza, essendo quello medico un contesto estremamente più complesso, sul quale la ricerca si concentra tutt'oggi e che non può essere semplificato in poco più di 5000 osservazioni.

Questo lavoro si pone quindi come principale obiettivo quello di mostrare come sia possibile migliorare le performance aggirando un problema di sbilanciamento delle classi, e come diverse metodologie possano portare a risultati molto eterogenei tra loro. Il tutto considerando che lo sbilanciamento delle classi è un problema proprio della vita reale, a differenza di quanto può avvenire in un contesto puramente accademico dove spesso si hanno dati più completi e pronti all'uso.

Riferimenti

[1] "Ictus cerebrale", Humanitas.it, *https://www.humanitas.it/malattie/ictus-cerebrale*

[2] "Stroke Prediction Dataset", Kaggle, 2021, *https://www.kaggle.com/fedesoriano/stroke-prediction-dataset*

[3] F. Mauriello, "*Tecniche di ricampionamento per dataset con classi di risposta sbilanciate. Una proposta metodologica per dataset con predittori di natura numerica e categorica.*", Università degli Studi di Napoli Federico II, Tesi di Dottorato

[4] G. Sailasya, G. L. A. Kumari, "*Analyzing the Performance of Stroke Prediction using ML Classification Algorithms.*", IJACSA, Vol. 12, No. 6, 2021, 539-545