MEDICAL APPOINTMENT NO-SHOW: ANALISI PREDITTIVA DEL FENOMENO

Boller Mattia, De Rosso Daniel - Università degli studi di Milano Bicocca dipartimento DISCO, Data science

Sommario
I pazienti che non si presentano agli appuntamenti programmati dai loro medici causano disguidi in termini di tempo, in quanto altri avrebbero potuto presentarsi al loro posto, e in termini di spese, per via delle remunerazioni non pervenute a medici e dottori. Sarebbe pertanto utile riuscire ad identificare i pazienti con alta probabilità di assenza prima che l'appuntamento abbia luogo.
La domanda di ricerca che ci si è posti, e a cui si è voluto rispondere, riguarda proprio la previsione della probabilità di no-show dei pazienti, sulla base di dati come: la distanza in giorni che intercorre tra la prenotazione e il giorno effettivo della visita, l'età, il sesso e altre caratteristiche fisiologiche e sociali. Per ovvia natura, il dataset risulta essere sbilanciato sulla classe positiva dell'attributo no-show, pertanto si è da subito fatto ricorso a funzioni che migliorassero lo squilibrio, per poi ricorrere a diversi modelli di classificazione che vengono infine comparati per giungere ad una soluzione ottimale nel contesto del problema.

Keywords
Classification model - Medical appointment - Machine learning

Indice

Introduzione

Si stima che annualmente la mancata presenza dei pazienti agli appuntamenti costi all'industria sanitaria 150 miliardi di dollari, questo in quanto per ogni singola persona che non dichiara la propria assenza si calcolano dai 100 ai 500 dollari di perdite di guadagno[1].
Tra le cause di questo fenomeno compaiono: la mancanza di tempo e la scarsa organizzazione degli impegni personali, i costi che potrebbero far ripensare i pazienti e la paura della visita/operazione, la quale sfocia in ansia e rivalutazioni all'ultimo momento.
Ma la causa più frequente che porta i pazienti a non presentarsi agli appuntamenti è la dimenticanza, scordarsi delle prenotazioni sembra infatti essere il principale problema che le persone riportano dai sondaggi.
Considerando tutto ciò, sarebbe utile riuscire a sviluppare un modello predittivo in grado di identificare le persone che con buona probabilità non si presenteranno all'appuntamento, e ricordare loro l'evento tramite un semplice messaggio.
Si procederà da subito alla presentazione del dataset, verranno poi esposte le tecniche di preprocessing sfruttate per la rifinitura dei dati e infine si descriveranno i modelli di classificazione utilizzati con le relative misure di performance.

1. Analisi dataset

1.1 Presentazione

Il dataset in analisi è stato scaricato dal sito Kaggle.com e si chiama: "Medical appointment no show"[2]. Il contenuto di tali dati raccoglie un totale di 110.527 record di appuntamenti, avuti luogo in Brasile nel 2016, con la relative misure di 14 variabili. L'attributo di classe è rappresentato dalla colonna "No-show".
Le variabili presenti nel dataset iniziale sono le seguenti:

- PatientId : id identificativo del paziente
- AppointmentID: id identificativo di uno specifico appuntamento
- Gender: attributo nominale che rappresenta il sesso del paziente
- ScheduledDay: il giorno effettivo dell'appuntamento
- AppointmentDay: il giorno in cui si è fatta la prenotazione dell'appuntamento
- Age: età del paziente
- Neighbourhood: luogo della visita
- Scholarship: attributo booleano che indica se la famiglia ha ricevuto aiuti finanziari per la scolarizzazione e la vaccinazione (programma welfare brasiliano)
- Hipertension: attributo booleano indicante ipertensione arteriosa nel paziente
- Diabetes: attributo booleano che indica se il paziente è affetto da diabete
- Alcoholism: attributo booleano che indica lo stato o meno di alcolismo del paziente
- Handicap: attributo booleano che riporta la disabilità del paziente
- SMS_received: attributo booleano che indica se è stato inviato un messaggio al paziente
- No-show: attributo di classe che si vuole predire, indica la presenza o meno del paziente nel momento della visita

In prima fase si è voluto svolgere un'analisi esplorativa del dataset per permettere una migliore comprensione dei dati in esame.

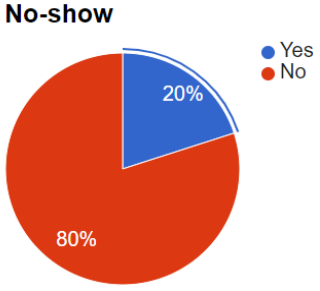

Figura 1: Pie chart variabile no-show

Come viene rappresentato dal grafico precedente (Figura 1), l'attributo "No-show" risulta essere sbilanciato maggiormente nella classe dei no. Come ci si poteva aspettare infatti, i pazienti che non si presentano agli appuntamenti sono una minoranza, tale considerazione è stata tenuta a mente nelle fasi successive per evitare fenomeni di class imbalance.
Passando alla verifica delle altre colonne si è constatato come alcune righe presentino valori che eccedono i range ammissibili dei dati, essendo tali record sporadici si è preferito effettuare una semplice cancellazione delle righe.

1.2 Preprocessing

Utilizzando il programma Knime si è lavorato per effettuare una pulizia iniziale delle righe del dataset che presentavano errori. Nel dettaglio è stato applicato un filtro agli attributi booleani per non permettere che ci siano valori che eccedono il range [0,1] ed è stato imposto un vincolo sulla colonna "age" per limitare solo numeri positivi.
Essendo il dataset molto vasto si è preferito raggruppare i record per PatientId (essendo che un paziente può effettuare più di una visita) piuttosto che rimuovere informazioni aleatoriamente. I dati del singolo paziente prendono quindi in considerazione solamente l'appuntamento più recente, mentre tutte quelle precedenti sono state sfruttate per aggiungere informazioni relative al passato del paziente. Ciò è stato fatto attraverso la creazione di due nuove colonne, "AppointmentCount" e "NoShowCount", che contano rispettivamente il numero di visite effettuate in passato dal paziente e il numero di volte in cui il paziente non si è presentato ad uno di questi appuntamenti.
Si è deciso poi di arricchire ulteriormente il dataset aggiungendo la colonna relativa ai giorni che intercorrono tra la prenotazione e la visita effettiva ("DayToWait") e la colonna che indica il giorno della settimana della visita ("DayOfWeek"); queste due colonne sono state calcolate a partire dagli attributi "ScheduledDay" e "AppointmentDay", dopo essere stati trasformati in tipo Date&Time, così da rendere le date manipolabili. Si è deciso di esplicitare questi valori per permettere una maggiore comprensione dei dati al modello.
Alla fine del processo di manipolazione e pulizia si è deciso di effettuare una normalizzazione di tipo min-max su tutti gli attributi interi.
I record risultanti da tutti questi processi si sono ridotti a 62190 e il numero delle colonne a 13.

2. Modelli utilizzati

2.1 Modelli testati

Per la risoluzione della domanda di ricerca si è deciso di utilizzare cinque diversi modelli di classificazione, appartenenti alla libreria open source Weka, suddivisi in:

- Modelli euristici: J48[3] (implementazione di un decision tree per la classificazione di variabili nominali) e Random forest (algoritmo di classificazione volto a limitare l'overfitting)
- Modelli di regressione: Regressione logica (algoritmo che modula una funzione logistica per descrivere una variabile dipendente binaria)
- Modelli di separazione: Multilayer Perceptron (rete neurale formata da una moltitudine di percettroni)
- Modelli probabilistici: Naive Bayes (classificatore basato sulla formula di Bayes per la probabilità condizionata)

2.2 Misure di performance

Per determinare le performance dei modelli testati, si è fatto uso di diverse metriche utilizzate di norma nella misurazione delle prestazioni delle tecniche di classificazione, con un occhio di riguardo però allo sbilanciamento del dataset utilizzato.
In particolare è stato dato poco conto al valore di Accuracy ottenuto dai modelli, dato che non si tratta di una metrica adatta a trattare dati sbilanciati dando poco peso alla classe minoritaria. In fase di valutazione sono stati quindi considerati i valori di Recall, Precision, F-measure e AUC (area under the curve) della curva ROC, variabili di cui ora si parlerà più in specifico.
La Recall indica la percentuale di istanze positive predette correttamente rispetto al totale, ne vediamo di seguito la formula:

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

La Precision indica invece la percentuale di vere istanze positive rispetto a tutte le istanze predette positivamente:

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

Nel caso della nostra domanda di ricerca, il nostro obiettivo è stato trovare il giusto compromesso tra Recall e Precision, cioè rilevare il maggior numero possibile di pazienti che non si sarebbero presentati alla visita ma evitare di classificare erroneamente troppi pazienti che invece all'appuntamento ci sarebbero andati, ipotizzando che il costo di un paziente che non si presenta sia di molto maggiore rispetto al ricordare ad una persona (attraverso ad esempio ulteriori messaggi o una chiamata) l'imminente arrivo della visita.
Per fare ciò è stata tenuta in considerazione la misura F-measure, che indica il valore della media armonica tra Recall e Precision:

$$F - measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

Infine si è tenuto conto anche della misura AUC, un buon indicatore di performance che rappresenta l'area del grafico sotto la curva ROC.

3. Analisi dei modelli e risultati

3.1 Holdout

Prediligendo un'analisi progressiva volta a migliorare i vari modelli per gradi, si è deciso di analizzare le performance iniziali con una basilare tecnica di holdout semplice. Il dataset di partenza è stato suddiviso in training set (66%) e test set (34%) con stratified sampling. Essendo infatti il dataset sbilanciato nella classe no-show, il partizionamento stratificato risulta essere una soluzione scontata per trattare il problema in maniera immediata ma superficiale.
I risultati che emergono da questo primo test, riportati in Tabella 1, sono ancora molto distanti da una soluzione ottimale e soddisfacente.

Modello	Recall	Precision	F-measure	AUC	Accuracy
Random Forest	0.188	0.340	0.242	0.659	0.764
J48	0.035	0.536	0.066	0.539	0.800
Logistic	0.028	0.396	0.052	0.674	0.796
Multilayer Perceptron	0.037	0.534	0.069	0.733	0.800
Naive Bayes	0.105	0.334	0.160	0.665	0.778

Tabella 1: Performance modelli con Holdout

Come è possibile notare, si ottengono dei valori alti solo per quanto riguarda l'Accuracy, punteggi determinati dalla natura sbilanciata del dataset e dal fatto che tutti i modelli tendono a classificare la maggior parte delle istanze come negative (classe maggioritaria). Possiamo invece notare come i valori di Recall e F-Measure siano molto bassi, indicando quindi un'incapacità dei classificatori di interpretare il dataset e riconoscere casi di classe positiva.
Si è deciso quindi di adottare diverse tecniche di bilanciamento del dataset per analizzare come cambiano le performance dei vari modelli, in particolare si sono testate una tecnica di undersampling e una di oversampling, di cui si tratterà nei paragrafi successivi.

3.2 Holdout e Equal Size Sampling

La prima tecnica testata per gestire lo sbilanciamento del dataset, è stata la equal size sampling. Questo metodo si basa sull'undersampling, che consiste nell'eliminare un certo numero di istanze della classe più numerosa, fino a rendere uguale il numero di casi della classe positiva e di quella negativa. Questa tecnica si è potuta applicare grazie al fatto che il dataset contiene un numero abbastanza grande di record e lo sbilanciamento tra le due classi non è troppo pesante.
Si sono quindi testati i vari modelli applicando ancora una volta il metodo holdout per la divisione del dataset in training set e test set, questa volta con delle percentuali rispettivamente del 70% e del 30%. Al training set è stato quindi applicato il metodo equal size sampling. In Tabella 2 sono riportati i risultati ottenuti dai modelli in seguito alle precedenti operazioni.

Modello	Recall	Precision	F-measure	AUC	Accuracy
Random Forest	0.606	0.298	0.400	0.678	0.634
J48	0.765	0.319	0.451	0.704	0.626
Logistic	0.581	0.318	0.411	0.677	0.666
Multilayer Perceptron	0.654	0.341	0.449	0.727	0.677
Naive Bayes	0.496	0.322	0.391	0.665	0.689

Tabella 2: Performance modelli con Holdout e Equal Size Sampling

Come si può constatare dalla Tabella 2, sono migliorate le performance di ogni modello, in particolare possiamo vedere un notevole miglioramento per quanto riguarda i valori di Recall e F-measure ma un naturale e aspettato calo del valore di Accuracy.
I modelli più performanti sono stati:

- J48, che ha ottenuto i valori più alti di Recall (0.765) ed F-measure (0.451)
- Multilayer Perceptron, che ha ottenuto il valore di AUC più alto (0.727) e un valore di F-measure di poco inferiore rispetto a J48

3.3 Holdout e SMOTE

La seconda tecnica testata per il problema della class imbalance è stata la SMOTE (Synthetic Minority Oversampling Technique), basata sull'oversampling ma con alcune caratteristiche innovative. Le classiche tecniche di oversampling si basano sul duplicare semplicemente righe della classe minoritaria, in modo da renderla numerosa quanto la seconda classe, evitando così di rischiare di eliminare informazioni come potrebbe avvenire nell'undersampling. In questo modo il dataset però non viene arricchito con nuove informazioni o variazioni. SMOTE quindi cerca di risolvere questo problema adottando un sistema diverso per generare i nuovi record della classe minoritaria, basato sul clustering.
SMOTE funziona sfruttando l'algoritmo k-nearest neighbor per creare dei dati sintetici; parte selezionando casualmente un record della classe minoritaria e i k-nearest neighbor e utilizza i valori dei loro attributi per creare una nuova istanza, ripetendo il processo fino al bilanciamento del dataset[4].
Per implementare questa tecnica si è quindi fatto uso del nodo SMOTE messo a disposizione da KNIME, utilizzato sempre sul training set derivante dall'applicazione del metodo holdout sul dataset iniziale (66%-34%). A seguire, la Tabella 3 riportante i risultati ottenuti dai classificatori.

Modello	Recall	Precision	F-measure	AUC	Accuracy
Random Forest	0.270	0.347	0.304	0.672	0.751
J48	0.352	0.373	0.362	0.701	0.751
Logistic	0.598	0.319	0.416	0.668	0.663
Multilayer Perceptron	0.787	0.320	0.455	0.736	0.622
Naive Bayes	0.285	0.684	0.403	0.640	0.592

Tabella 3: Performance modelli con Holdout e SMOTE

I risultati ottenuti non sono stati soddisfacenti, in quanto quasi tutti i modelli, in seguito allo SMOTE, hanno performato peggio rispetto all'utilizzo dell'equal size sampling. Solo due modelli sembrano invece migliorare i quali sono, come si può notare dalla Tabella 2, il Multilayer Perceptron, che ha ottenuto i valori di Recall (0.787), di F-measure (0.455) e di AUC (0.736) più alti visti finora e la Logistic Regression, che ha raggiunto delle performance superiori rispetto a quando utilizzata in seguito all'equal size sampling, ma comunque non buone a sufficienza. Questi due modelli sopracitati sembrano essere quindi gli unici classificatori a riuscire a sfruttare l'arricchimento introdotto dallo SMOTE.
Il metodo SMOTE dà i risultati migliori quando il numero di feature è basso, è quindi stato testato anche in seguito ad una feature selection per verificare se le performance dei classificatori sarebbero potute migliorare ma il test non ha dato risultati positivi.

3.4 Feature selection e Parameter optimization

In seguito ai risultati ottenuti dalle due tecniche di gestione dello class imbalance problem, si è deciso di approfondire i modelli allenati attraverso l'utilizzo della equal size sampling.
Per cercare di migliorare le performance dei classificatori, sono state utilizzate le tecniche di feature selection e di automatic parameter optimization.
La feature selection si basa sul filtraggio degli attributi dati in input al modello, con l'obiettivo di selezionare le caratteristiche più significative dei record e di rendere i dati di più facile interpretazione. Questa tecnica è stata

implementata sfruttando il nodo AttributeSelectedClassifier reso disponibile da KNIME e in particolare il metodo CfsSubsetEval. Le feature che quindi sono state riconosciute come le più rilevanti dal nodo sono state: “Age”, “NoShowCount”, “DayToWait”, “DayOfWeek”.

Per automatic parameter optimization si intende la ricerca automatica dei migliori parametri con cui impostare un certo modello. Questa operazione può essere eseguita “a mano”, provando vari valori fino ad ottenere le performance migliori, ma l’automazione di tale processo può portare ad un notevole risparmio di tempo e ad una probabilità più alta di trovare i parametri migliori. Per implementare questo passaggio è stato utilizzato il nodo CVParameterSelection applicato ad alcuni dei modelli, nodo che permette di cercare il valore migliore di un certo parametro all’interno di un certo range, testando ad ogni passaggio il classificatore attraverso cross validation.

In Tabella 4 sono riportati i risultati ottenuti dai modelli in seguito alle operazioni sopra descritte.

Modello	Recall	Precision	F-measure	AUC	Accuracy
Random Forest	0.621	0.306	0.410	0.671	0.642
J48	0.757	0.326	0.455	0.720	0.636
Logistic	0.544	0.326	0.408	0.670	0.683
Multilayer Perceptron	0.678	0.339	0.452	0.735	0.670
Naive Bayes	0.380	0.320	0.347	0.663	0.714

Tabella 4: Performance modelli con Holdout, Feature selection e Parameter optimization

Possiamo notare come l’applicazione della feature selection su tutti i modelli e l’automatic parameter optimization su alcuni, abbia portato ad un miglioramento in termini di F-measure in tutti i classificatori, in particolare:

- J48 ha ottenuto un valore di F-measure pari a 0.455 (aumentato di 0.004) e di AUC pari a 0.720 (aumentato di 0.016)
- Multilayer Perceptron ha ottenuto un valore di F-measure pari a 0.452 (aumentato di 0.003) e di AUC pari a 0.735 (aumentato di 0.008)

3.5 Cross Validation dei migliori modelli

In ultima istanza si è deciso di applicare la tecnica di cross validation ai modelli che ottenevano risultati migliori nelle fasi precedenti. Questo ha permesso di effettuare una statistica più robusta sulle misure di performance utilizzate nelle valutazioni.

I modelli presi in considerazione sono i seguenti:

- J48 con Equal Size Sampling
- Multilayer Perceptron con Equal Size Sampling
- Multilayer Perceptron con SMOTE
- J48 con Feature selection ed Equal Size Sampling
- Multilayer Perceptron con Feature Selection ed Equal Size Sampling

Sono state usate in totale dieci fold per modello per effettuare il cross validation, questo numero è stato ottenuto come compromesso per ottenere una buona statistica ma senza risultare eccessivamente oneroso in termini computazionali.

La fase di output delle misure ha richiesto manipolazioni specifiche sui nodi scorer per memorizzare tutte le misure di performance di ogni modello per ogni singola fold; a questo punto il risultato finale viene mostrato in Tabella 5.

Modello	Recall	Precision	F-measure	AUC	Accuracy
J48 EQ	0.76	0.32	0.45	0.712	0.63
MP EQ	0.75	0.32	0.45	0.731	0.63
MP SMOTE	0.74	0.33	0.45	0.732	0.65
J48 FS	0.79	0.32	0.46	0.719	0.61
MP FS	0.74	0.32	0.44	0.723	0.62

Tabella 5: Mediana delle performance ottenute con cross validation

Le mediane delle misure di performance non si discostano di molto dalle grandezze rilevate precedentemente, di conseguenza si è proceduto a valutare il complesso della cross validation in grafici box plot per permetterne uno sguardo più dettagliato delle diverse distribuzioni.

Come referenza viene di seguito mostrato il grafico box-plot della misura F-measure. Secondo la Figura 3, il modello che apparentemente presenta la miglior distribuzione di F-measure, calcolata con cross validation, è il J48 con feature selection. Questa conclusione è stata determinata osservando il valore della mediana (maggiore rispetto a tutti gli altri modelli) e della varianza, indicando pertanto delle performance più costanti e affidabili.

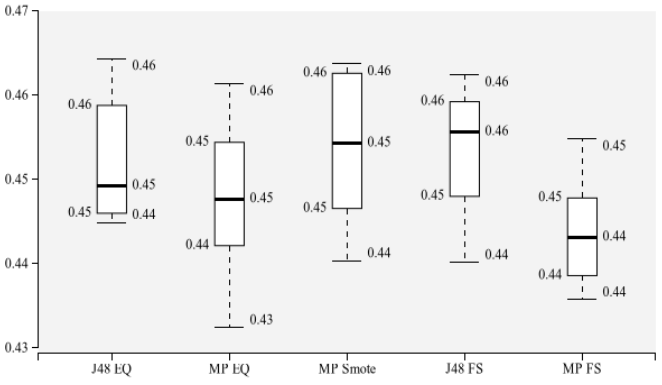

Figura 3: Box plot della F-measure dei migliori modelli

E’ importante ricordare che le misure utilizzate come valutazione dei diversi modelli, come la F-measure in questo caso, o la Recall, non determinino in modo univoco la preferenza di un modello rispetto ad un altro.

La soluzione ideale in questo specifico problema sarebbe quella di associarci una misura di costo propria, onde evitare ambiguità.

Come ultima misura grafica della bontà dei modelli viene mostrata in Figura 4 la relativa ROC curve dei metodi migliori di classificazione. Un numero della AUC prossimo a 1 è da interpretarsi come un ottimo grado di separazione del modello nelle classi corrette.

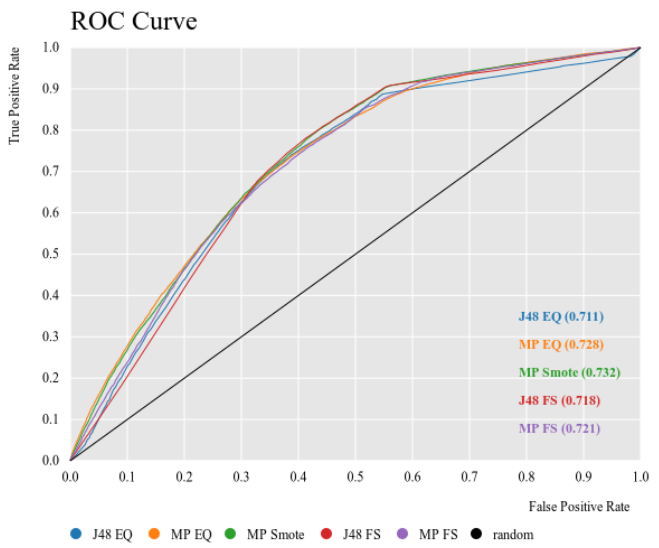

Figura 4: ROC curve con relative AUC dei migliori modelli

Come possiamo notare dalla Figura 4, tutte le curve sono molto simili tra loro, indicando quindi livelli di performance similari tra i vari modelli. Tuttie le curve ROC presentano un valore di AUC (area sotto la curva) maggiore rispetto a quello del classificatore ZeroRule (modello che si limita a classificare tutte le istanze come appartenenti alla classe più frequente e la cui curva ROC è rappresentata dalla retta di colore nero in Figura 4).

Conclusioni e sviluppi futuri

Con questo lavoro si è cercato di trovare un metodo per predire in anticipo se un certo paziente, con alta probabilità, non si sarebbe presentato ad un futuro appuntamento. Il non presentarsi delle persone alle visite prenotate, come già detto durante l'introduzione, è un problema reale, che comporta dei costi da parte della sanità. In un periodo come questo, segnato dall'avvento di una pandemia globale, abbiamo ritenuto importante cercare di fornire nuovi strumenti al settore sanitario, strumenti che possano portare a risparmio di tempo e denaro.
Attraverso alcuni dati relativi a visite varie avvenute in Brasile, abbiamo cercato di trovare il miglior modello di machine learning per individuare coloro che non si sarebbero presentati all'appuntamento.

Il primo problema che ci siamo trovati a dover risolvere è stato lo sbilanciamento (seppur lieve) delle classi da prevedere, con l'80% dei record che presentavano casi in cui il paziente è andato all'appuntamento. Fin da subito abbiamo potuto notare che i modelli che riuscivano a rilevare più frequentemente i casi di NoShow erano il J48 e il Multilayer Perceptron, classificatori che hanno continuato a sovrastare gli altri per tutta la durata del nostro lavoro.
I modelli che abbiamo ottenuto alla fine della nostra analisi hanno delle buone performance che potrebbero sicuramente portare a dei vantaggi nello scenario reale, rispetto al non utilizzare alcun metodo per cercare di rilevare i possibili assenti agli appuntamenti, ma in futuro potrebbero essere senza dubbio migliorati.
Il classificatore che potrebbe essere ritenuto come il più performante, è il Multilayer Perceptron che ha raggiunto una Recall pari a 0.74, una F-measure pari a 0.45 e una Accuracy pari a 0.65. Quest'ultimo modello sembra il migliore in termini di trade-off tra F-measure e Accuracy, ma la scelta finale dovrebbe avvenire conoscendo più a fondo l'impatto in termini di tempo e spesa che ha un paziente che non si presenta alla visita.

Lavorando a stretto contatto con un ente che si occupa di sanità, si potrebbero ottenere molti più dati, dati storici riguardanti i pazienti o informazioni su come vengono presi gli appuntamenti. Può essere fondamentale conoscere nelle specifico il costo di un paziente che non si presenta ad una visita e il costo che ha ricordare ad un individuo l'imminente arrivo dell'appuntamento, in modo da trovare il modello che possa minimizzare la spesa totale.

I risultati ottenuti sono comunque un buon punto di partenza e dimostrano come il problema del medical appointment no-show, seppur non facile da risolvere, possa essere trattato attraverso tecniche di machine learning.

Referenze

[1] Patient No-Shows: Everything Practice Managers Need to Know - www2.relatient.net/resources/patient-no-shows

[2] Medical Appointment No Shows, dataset kaggle.com - www.kaggle.com/joniarroba/noshowappointments/data

[3] J48, Weka implementation of the C4.5 algorithm - https://en.wikipedia.org/wiki/C4.5_algorithm

[4] 5 SMOTE Techniques for Oversampling your Imbalance Data - https://towardsdatascience.com/5-smote-techniques-for-oversampling-your-imbalance-data-b8155bdbe2b5