FLAVIO DI DOMENICO MATR. 851181

**NO SHOW**

1. **INTRODUZIONE**:

Lo scopo di questo studio è l’analisi del fenomeno dei “no show”, attraverso cui un cliente prenota una prestazione sanitaria e, senza preavviso, non si presenta al momento di usufruirne. In particolare, lo studio in questione si rivolge ai “no show” che si verificano presso l’Azienda socio-sanitaria della Brianza, che comprende i presidi ospedalieri di Vimercate, Carate Brianza, Giussano, Desio e Seregno.

Il fenomeno dei no show è un problema ampiamente diffuso in ambito sanitario che porta alla perdita di diverse migliaia di euro per il mancato rimborso del ticket da parte di Regione Lombardia e per la presenza di personale medico e sanitario che non effettua attività lavorativa pur essendo pagato. Il fenomeno è complesso da analizzare perché è causato da una somma di elementi, molti dei quali sono casuali oppure non sono osservabili e tracciabili, per i quali non possiamo avere dei dati statistici di rilevazione. Per fornire alcuni esempi ed entrare nel dettaglio, una delle cause di questo fenomeno può essere il lungo tempo di attesa che intercorre tra la data della prenotazione e la data della prestazione e questo aspetto può portare il paziente a dimenticarsi dell’appuntamento oppure a prenotarsi anche presso una struttura ospedaliera privata, che di solito garantisce un tempo di attesa minore, senza annullare il precedente appuntamento fissato.

I dati sono stati forniti dai Sistemi Informativi Aziendali tramite un’estrazione dalle prenotazioni registrate presso il CUP. Per questo motivo è stato necessario uno studio approfondito delle dinamiche di prenotazione e gestione degli appuntamenti che avvengono presso la struttura ospedaliera.

Gli obiettivi principali sono tre:

- descrivere ed inquadrare il fenomeno dal punto di vista quantitativo per comprendere dimensioni e impatto economico sull’azienda

- capire quali fattori influenzano maggiormente il fenomeno e con quale intensità

- sviluppare modelli di classificazione per prevedere il comportamento di nuove persone, ovvero se la persona sarà presente o assente alla visita/appuntamento e con quale probabilità ciò avverrà

Come informazione di contesto, non si sono potute analizzare le prestazioni erogate c/o pres. Osp. di Desio dato che sono registrate su un programma informatico diverso rispetto agli altri presidi ospedalieri; inoltre gli anni presi in considerazione sono il 2019 (mesi aprile-dicembre) e il 2022 (mesi gennaio-aprile) poiché gli anni 2020 e 2021 sono stati contraddistinti fortemente dalla pandemia e i dati risultano essere sicuramente alterati.

1. **METODI E MODELLI STATISTICI:**

Le principali tecniche statistiche utilizzate per questo studio sono le seguenti.

**2.1 Regressione logistica**:

Questo tipo di modello statistico (noto anche come modello logit) è spesso utilizzato per la classificazione e l'analytics predittiva. La regressione logistica1 stima la probabilità del verificarsi di un evento sulla base di uno specifico dataset di variabili indipendenti. Poiché il risultato è una probabilità, la variabile dipendente è vincolata tra 0 e 1.

Pertanto, con questo modello si vuole descrivere la relazione di dipendenza del possesso di un attributo dicotomico da una o più variabili indipendenti (X1 , X2 , ..., Xp ) = X, di natura qualsiasi (cioè, indifferentemente, quantitative o qualitative); per fare ciò è necessario costruire un modello di regressione per Y, variabile risposta dove Y è dicotomica a valori 0 e 1, corrispondenti rispettivamente all’assenza e alla presenza dell’attributo.

Nella regressione logistica, viene applicata una trasformazione logit sulla risposta media, rappresentata dalla probabilità di successo divisa per la probabilità di fallimento e questa funzione logistica è rappresentata dalla seguente formula:

Y= logit(μ) = β0 + β1\*X1 + … + βκ\*Xk

La differenza principale rispetto al modello lineare sta quindi nell’aggiunta di questa trasformazione logit che converte i valori della risposta media facendo assumere ad essa valori compresi fra 0 e 1 invece che fra -Inf e +Inf.

In questa equazione di regressione logistica, y è la variabile dipendente o di risposta e x è la variabile indipendente. Il parametro beta, o coefficiente, in questo modello è comunemente stimato servendosi della stima di massima verosimiglianza (MLE, maximum likelihood estimation). Questo metodo testa diversi valori di beta mediante molteplici iterazioni per eseguire un’ottimizzazione per il best fit della probabilità logaritmica. Per una classificazione binaria, una probabilità minore di 0,5 prevederà 0 mentre una probabilità maggiore di 0,5 prevederà 1.

I coefficienti di regressione beta rappresentano il logaritmo dell’ODDS Ratio; pertanto, per facilitare l'interpretazione dei risultati si applica l'esponenziazione delle stime beta trasformando i risultati in un OR (odds ratio, o rapporto crociato). L'OR rappresenta le probabilità del verificarsi di un risultato dato uno specifico evento, rispetto alle probabilità del verificarsi del risultato in assenza di tale evento. Se l'OR è maggiore di 1, l'evento è associato a una probabilità più elevata di generare uno specifico risultato. Viceversa, se l'OR è minore di 1, l'evento è associato a una probabilità più bassa del verificarsi di tale risultato.

Gli obiettivi della regressione logistica sono quelli di individuare tra le variabili indipendenti quelle a maggiore potere esplicativo, che vanno quindi interpretate come determinanti del possesso o meno dell’attributo: a seconda che siano correlate positivamente o negativamente con il fenomeno studiato possono essere considerate rispettivamente come fattori di rischio o come fattori di protezione; inoltre consente di ricercare la combinazione lineare delle variabili indipendenti che meglio discrimina fra il gruppo delle unità che possiedono l’attributo e quello delle unità che non lo possiedono. Questo metodo permette infine di stimare la probabilità del possesso dell’attributo per una nuova unità statistica su cui è stato osservato il vettore di variabili esplicative X e, fissato per tale probabilità un valore soglia, classificare l’unità alla categoria delle unità che possiedono l’attributo o a quello delle unità che non lo possiedono.

* 1. **Bilanciamento della variabile risposta**:

Un fenomeno molto frequente nei dataset con variabile risposta dicotomica è lo sbilanciamento delle classi poiché esse non sono rappresentate con numerosità simili nel training set.

In queste situazioni le consuete metriche di performance utilizzate, come l’accuracy o l’error rate, non sono più significative ed oggetto d’interesse.

Essendo anche questo un problema a classi sbilanciate, poiché la variabile di nostro interesse è caratterizzata da una forte prevalenza di show (93.5%) a fronte di una minoranza di no show (6.5%), tutti gli algoritmi tendono a classificare le nuove unità statistiche come show ottenendo un’accuratezza pari alla frequenza relativa della classe prevalente senza però essere in grado di riconoscere alcun no show che invece costituisce la classe di interesse dello studio.

Esistono varie tecniche per risolvere questo problema:

La prima è il campionamento retrospettivo2 o retrospective sampling in cui il training set è ottenuto come concatenazione di sottocampioni e quindi le numerosità dei gruppi sono fissate a priori.

La seconda è il ricampionamento del dataset che può avvenire in due modi: sovracampionamento aggiungendo osservazioni estratte casualmente dalla classe sottorappresentata e sottocampionamento togliendo osservazioni estratte casualmente dalla classe sovrarappresentata.

Inoltre, altre tecniche utili nel caso di unbalanced data consistono nel cambiare le metriche di riferimento, ad esempio la balanced accuracy, oppure, in maniera più complessa, inserire nella fase di stima una penalizzazione per le misclassificazioni relative alla classe sottorappresentata.

Per questo studio si è deciso di utilizzare la seconda e la terza tecnica elencate nell’ordine, effettuando un bilanciamento perfetto delle classi (50% e 50%) tramite un sovracampionamento della classe meno prevalente e sottocampionamento casuale della classe prevalente. Si è deciso di non operare solo il sottocampionamento perché altrimenti, essendoci poche u.s. della classe no show, non ci sarebbero stati sufficienti dati per l’allenamento del classificatore.

**2.3 Clustering gerarchico**:

Nei metodi gerarchici3 si individua una sequenza di partizioni nidificate: la partizione in K + 1 gruppi si ottiene dalla partizione in K gruppi facendo di due degli elementi di questa un elemento di quella (AGNES), o viceversa (DIANA). Esistono infatti due tipi di metodi gerarchici. L’ Algoritmo Agglomerativo (AGNES, AGGlomerative NESting) e l’Algoritmo Scissorio (DIANA, DIvisive ANAlysis) che verranno dettagliati nei due paragrafi seguenti.

**2.3.1 Algoritmo agglomerativo:**

Si incomincia dalla partizione in n gruppi, ciascuno singoletto, ovvero formato da una sola unità statistica; Si inizializza k = n, quindi il numero dei gruppi è pari al numero di osservazioni. (passo 1)

Poi si determina quale coppia di gruppi sia quella ‘migliore’ da unire, tra le k(k − 1)\*2 coppie di gruppi possibili (passo 2); in ultima istanza, si fonde la ‘migliore’ coppia di gruppi in un unico gruppo (passo 3); si imposta k = k − 1 e si va al passo 2 se k > 1, altrimenti STOP.

Per questo algoritmo sono previste n − 1 iterazioni dei passi 2 e 3 prima dell’arresto.

Bisogna precisare come si determina al passo 2 la ‘migliore’ coppia di gruppi da fondere in un unico gruppo. Se abbiamo k gruppi con matrice delle distanze/dissimilarità D, basta determinare quale sia la coppia di gruppi con minore distanza/dissimilarità (se più di una coppia, si sceglie una).

Legame singolo (single linkage):

d(GI , GL) = min{d(ui , ul), ui ∈ GI , ul ∈ GL}

Legame completo (complete linkage):

d(GI , GL) = max{d(ui , ul), ui ∈ GI, ul ∈ GL}

Legame medio (average linkage):

d(GI , GL) = 1/(nGi \* nGL) ΣΣ d(ui , ul)

dovenGi e nGL sono le numerosità dei gruppi GI e GL.

Una peculiarità del legame singolo è l’effetto catena (chaining):

da un lato consente di cogliere gruppi di forma particolare mentre dall’altro rischia di legare osservazioni che non appartengono a uno stesso gruppo.

Il metodo del legame completo, d’altra parte, tende a individuare gruppi molto compatti al loro interno ma di forma circolare (ipersferica, in generale) quindi si rischia di perdere gruppi di forma irregolare.

**2.3.2 Algoritmo Scissorio**:

Parte dalla considerazione di tutte le unità appartenenti ad un solo gruppo e si procede per suddivisione secondo opportuni criteri di ottimalità. Consiste in sintesi nel procedimento inverso rispetto all’algoritmo agglomerativo.

**2.4 Algoritmo delle K-medie**

4Si parte con una attribuzione iniziale per x¯1, . . . , , che rappresentano i centroidi ovvero la media delle unità statistiche (e.g. considerando K unità statistiche) e si procede iterando i passi 2 e 3 fino alla convergenza:

Si cerca di minimizzare la distanza rispetto ai gruppi: per i = 1, . . . , n, si individua il centroide più vicino (secondo d^2 ) all’unità ui e la si attribuisce al gruppo corrispondente Gk. (passo 2)

In seguito, si minimizza la distanza rispetto ai centroidi: per k = 1, . . . , K, si aggiorna il valore del k-simo centroide con la media delle unità del gruppo Gk. Si calcola W, ovvero la distanza entro i gruppi. (passo 3)

Si arresta l’algoritmo quando W non cambia rispetto al passo precedente ovvero quando si raggiunge la convergenza.

W= Σ W(Gk ) = Σ d2(Ui, X¯k) = Σ [ Σ Σ (Xij - X¯kj)2 ]

Ciò che si può notare è che la distanza entro i gruppi decresce ad ogni iterazione dell’algoritmo: Wi+1 ≤ Wi , dove Wi è W all’iterazione i-sima.

L’algoritmo converge sempre, indipendentemente dall’attribuzione iniziale dei centroidi. Ci mette al massimo Kn iterazioni. I gruppi finali dipendono dall’attribuzione iniziale dei centroidi. Tipicamente si fa girare l’algoritmo più volte inizializzando i centroidi casualmente, e si sceglie il risultato con W minimo; infatti l’algoritmo non garantisce di minimizzare globalmente W ma soltanto localmente.

Le principali caratteristiche del metodo delle k-medie risiedono nell’ottimo adattamento a scoprire gruppi di forma convessa mentre risulta inadatto per gruppi di forma concava; inoltre il risultato è sensibile alla presenza di valori anomali e non è invariate a trasformazioni di scala.

**2.5 Model-Based Clustering:**

Questo tipo di clustering è basato su un modello statistico che si chiama finite mixture. Una mixture5 è una variabile casuale la cui distribuzione è una combinazione lineare, con specifici pesi, di leggi di distribuzione, rappresentanti k cluster, ciascuna delle quali descrive la distribuzione dei valori per i membri di quel cluster.

Il caso più semplice e quello in cui si ha una sola variabile reale e due cluster con distribuzione normale. Media e varianza della distribuzione sono diverse per i due cluster. Obiettivo del clustering è quello di prendere un insieme di istanze e di trovare la media e la varianza delle due distribuzioni normali oltre alla distribuzione delle istanze nei cluster.

Ci sono due cluster A e B con medie e deviazioni standard μA , σA e μB , σB. I campioni sono presi da A con probabilità pA e da B con probabilità pB con pA+pB =1. Il problema da risolvere è, date le istanze (senza le classi A o B), stimare i cinque parametri μA , σA , μB , σB e pA (pB può essere ricavato da pA ).

Se conoscessimo i 5 parametri, potremmo trovare le probabilità che una istanza x appartenga a ciascuna distribuzione con la seguente formula Pr(A|x)=Pr(x|A)Pr(A)/ Pr(x) =f(x; μA ,σA )dx pA/ Pr(x) dove f(x; μ,σ) e’ la distribuzione normale. Pr(x) non è noto.

Il problema è che non si conoscono ne’ i 5 parametri ne’ l’appartenenza delle istanze ai cluster. Perciò si adotta un metodo chiama EM; si tratta di una procedura numerica iterativa che ha l'obiettivo di massimizzare la mixture likelihood. Per fare ciò si alternano i passi di Expectation e Maximization.

Il primo passo è l’inizializzazione, ovvero si assegna un valore dei al vettore dei parametri η(0) = ( θ1(0), θ2(0),….. θk(0), p(0)). Quindi il vettore dei parametri al passo 0 è noto.

Il secondo passaggio consiste nell’allocazione delle unità statistiche con η(0) noto e nella stima di η(1) con allocazione nota, ovvero si divide il secondo processo in E-step ed M-step. L’obiettivo dell’E-step è quello di stimare Zij(m), ovvero la probabilità a posteriori al passo m. E’ sufficiente calcolare

E (Zij|Y) = P (Zij=1 | Y) = P ( Si= j | Y) = Pj\*f(yi ; θj) / ( Σ pj\*f(Yi; θj) )

=P ( Si= j | Y) = Pj(m-1)\*f(yi ; θj(m-1)) / ( Σ pj(m-1)\*f(Yi; θj(m-1)) ) = Zij (m)

L’obiettivo dell’M-step invece è quello di stimare il vettore dei parametri η = (θ1(m), θ2(m),….., θk(m), p(m)). Per quanto riguarda le proporzioni P, dato che Zij sono note, le Si sono note pertanto:

Pj= ( # Si = j) / n = nj(s) / n

Pj (m) = Σ Zij(m) / n = nj(m) / n

Mentre per quanto riguarda i θ1, θ2,….. θk, si applica la classica stima di massima verosimiglianza.

Il terzo passo rappresenta la convergenza: si alternano i passi E ed M fino a quando la verosimiglianza mostra una differenza fra due passi successivi arbitrariamente piccola.

L(η(m+1); y) - L(η(m); y) < ε (con ε > 0)

In EM non si ha l’appartenenza netta delle istanze ai vari cluster ma solo una probabilità. K-means si ferma quando le classi delle istanze non variano più mentre EM converge verso un punto fisso. Si può però capire quanto è vicino calcolando la verosimiglianza (likelihood) globale che i dati derivino da questo dataset, dati i valori dei 5 parametri (verosimiglianza=probabilità). La verosimiglianza globale è una misura della “bontà’” del clustering e aumenta a ogni iterazione dell’algoritmo EM. Inoltre, grazie alle proprietà della verosimiglianza, per semplificare le formule, viene calcolato il logaritmo della verosimiglianza che trasforma i prodotti in somme. Per esempio, un criterio per fermarsi potrebbe essere: ci si ferma quando la differenza tra due valori successivi della verosimiglianza/dx è inferiore a 10^-10 per dieci iterazioni successive.

Per quanto riguarda le proprietà dell’algoritmo, anche se è garantito che EM converga a un massimo, non è detto che sia un massimo globale ma potrebbe essere un massimo locale. Per questo la procedura deve essere ripetuta diverse volte, partendo da diversi valori iniziali dei parametri. La verosimiglianza globale viene poi usata per confrontare le diverse soluzioni ottenute e prendere quella con la verosimiglianza più alta.

Per la definizione dei modelli vengono posti dei vincoli sulla struttura di variabilità del modello; in particolare vengono applicati dei vincoli geometrici sulla matrice di varianze/covarianze; infatti si effettua una decomposizione spettrale della matrice Σj che la trasforma in questo modo: Σj = Dj\* Aj\*Dj’ dove Dj rappresenta la matrice ortogonale le cui colonne sono gli autovettori normalizzati ed Aj costituisce la matrice diagonale con gli autovalori ordinati in senso decrescente.

Queste matrici comportano un’interpretazione geometrica; Dj determina l’orientamento in R^d della j-esima mixture component, Aj determina la forma e λj il volume del cluster.

Il volume può essere Equal o Variable. La forma può essere Equal, Variable, Sferico mentre l’orientamento registra la possibilità di essere Equal, Variabile o Sferico. Se una di queste caratteristiche è Equal, viene denotata con la lettera iniziale E, se Variable viene rappresentata dalla V e se Sferici viene rappresentata dalla I.

Dalla combinazione di tutte queste possibilità nascono 14 possibili modelli: EEE, EII, VII ecc.. fino al modello più complesso con tutte le caratteristiche variabili ovvero VVV.

Per la parte di classificazione si usano due principali tecniche model-based: EDDA e MDA.

I modelli EDDA (Eigenvalue Decomposition Discriminant Analysis ) dove la legge di distribuzione della risposta, f(yi) corrisponde ad una mistura di variabili Normali. La matrice di varianze/coviarianze è data dalla seguente formula:

Σj = λj\* Dj\* Aj\*Dj’

Dove Aj ha a che fare con la forma, Dj con l’orientamento e λj con il volume dato che λj = |Σj | 1/d.

A seconda delle varie combinazioni si possono ottenere 14 possibili modelli come per esempio LDA (Linear Discriminant Analysis dove tutte le componenti sono Equal quindi prende il dove di EEE) oppure QDA ( Quadratic Discriminant Analysis dove tutte le componenti sono Variable quindi prende il dove di VVV).

I modelli MDA ( Mixture Discriminant Analysis) dove le componenti di ogni gruppo a loro volta corrispondono a delle misture, generando quindi delle misture di misture.

Si sostituisce la legge di distribuzione della j-esima componente con una combinazione lineare di altre leggi di distribuzione; si passa quindi da

fj (yi) = f(yi ; θj) -> Nd

a

fj (yi) = Σ pjl \*fNd ( yi; μjl ; Σjl )

dove il j delle due formule corrisponde e indicizza la singola componente.

**2.6 Variabili miste:** **indice di Gower**

Una misura della distanza tra due profili di riga di una matrice di dati con variabili di tipo misto è l’indice di distanza di Gower6; l’indice di dissimilarità di Gower assume un valore compreso tra 0, nel caso di unità identiche, e 1, che corrisponde a unità diametralmente opposte. La misura di somiglianza Sij si calcola in tre modi diversi a seconda della classe della variabile.

sG(ui , ul) = Σ δil(j)sil(j) / Σ δil(j)

dove sil(j) =

• 1 − |xij−xlj| / range j-sima variabile se j-sima variabile quantitativa

• I(xij = xlj ) se j-sima variabile binaria/nominale

• 1 − |yij − ylj | se j-sima variabile ordinale

e dove δil(j) =

• 1 se i, l confrontabili rispetto j-sima variabile

• 0 se i, l non confrontabili rispetto j-sima variabile

dove due unità sono non confrontabili rispetto alla j-sima variabile se c’è un valore mancante in almeno una delle due o se la j-sima variabile è binaria asimmetrica e si ha co-assenza 0-0.

**2.7.** **Mixture of experts model (MEM)**

Un modello a mistura finita7 è un modello statistico che presuppone la presenza di gruppi non osservati, chiamati classi latenti, all'interno di una popolazione complessiva e sono quindi un metodo popolare per modellare l’eterogeneità non osservata. Le popolazioni, infatti, sono spesso divise in gruppi o sottopopolazioni di vario tipo. I modelli di regressione e le funzioni di distribuzione differiscono tra questi gruppi. Ma a volte non si ha a disposizione una variabile che identifichi i gruppi perché è mancante, difficile da rilevare oppure non osservabile. Ogni classe latente può essere generata dal proprio modello di regressione, che può avere una funzione di risposta lineare o generalizzata. Si può confrontare modelli con diversi numeri di classi latenti e diversi insiemi di vincoli sui parametri per determinare il modello più adatto. Per un dato modello, si possono confrontare le stime dei parametri tra le classi e stimare la proporzione della popolazione in ogni classe latente, prevedendo inoltre le probabilità che le osservazioni nel nostro campione appartengano ad ogni classe latente.

La formula che rappresenta il modello è la seguente:

f(yi|xi)= Σ Pj(xi)\*f(yi;θj(xi))

Da questa formula si nota come le proporzioni Pj e i parametri θj ora sono in funzione delle covariate.

Per stimare Pj si usa la regressione multinomiale nel caso di K>2 e la regressione logistica nel caso di K=2.

La formula nel caso di K =2 è la seguente:

exp(xi’β)/(1+ exp(xi’β)) = exp(xi’β)/ exp(xi’0) + exp(xi’β)

Nel caso di k=2 e modello binomiale viene utilizzato l’algoritmo di stima IRLS come nel caso dei GLM mentre nel caso multinomiale viene utilizzato un algoritmo di stima ad hoc.

Per quanto riguarda i θj, la formula è la seguente:

θj(xi)= g^-1(xi’γj)

dove g^-1 rappresenta l’inversa della link function.

Quindi questa formula modellizza un GLM se θj rappresenta la media della j-esima mixture component.

Se invece θj non rappresenta la media della j-esima mixture component è necessario riparametrizzare.

Passando alla terminologia, f(yi;θj(xi)) rappresenta l’expert network mentre Pj(xi) prende il nome di gating network.

Per fare inferenza sui MEM si può procedere sia attraverso l’inferenza bayesiana sia attraverso il metodo della massima verosimiglianza. Si calcola la verosimiglianza della mistura e si utilizza l’algoritmo EM usufruendo così della complete Likelihood che rende più semplici i conti poiché è formata dalla somma del gating network e dell’expert network. Come di consueto, si procede all’E-step calcolando la Zij, ovvero la probabilità a posteriori al passo m; poi durante l’M-step si calcolano i βj e i γj che rappresentano rispettivamente i Pj e i θj.

Pertanto, anzichè dover calcolare i Pj, dovrò i beta, e anziché calcolare i θj dovrò stimare i γj.

In sintesi all'interno di ogni componente le variabili possono giocare un ruolo diverso implementando delle regressioni differenziate per gruppi.

1. Gareth J., Witten D., Hastie T., Tibshirani R., *An Introduction to statistical learning with application in R*, springer (2013).

2. [Geoffrey McLachlan,](https://onlinelibrary.wiley.com/action/doSearch?ContribAuthorRaw=McLachlan%2C+Geoffrey)[David Peel](https://onlinelibrary.wiley.com/action/doSearch?ContribAuthorRaw=Peel%2C+David), Finite\_Mixture\_Models, Copyright © 2000 John Wiley & Sons, Inc. (2000).

3. Johnson, Wichern, Applied Multivariate Statistical Analysis (6th Edition), cap.12.3 , Pearson Prentice Hall. (2007)

4. Johnson, Wichern, Applied Multivariate Statistical Analysis (6th Edition), cap.12.1 , Pearson Prentice Hall. (2007)

5. Ian Witten, Eibe Frank Data Mining: Practical Machine Learning Tools and Techniques (Second Edition) Morgan Kaufmann Publishers. (2005)

6. Johnson, Wichern, Applied Multivariate Statistical Analysis (6th Edition), cap.12.2-12.4 , Pearson Prentice Hall. (2007)

7. Fruhwirth-Schnatter Finite mixture and Markov switching models. (2006)

1. **ANALISI DEI DATI:**

Il Dataset iniziale era formato da circa 130000 istanze e oltre 20 colonne che caratterizzano gli individui.

Per garantire un’elevata qualità e accuratezza del dato fornito, si è deciso di prendere in considerazione soltanto le prestazioni che afferiscono alla diagnostica per immagini, dove il flusso informatico è ben regolato e organizzato. Inoltre, sono state eliminate quelle righe che facevano riferimento a prestazioni senza ricetta, per le quali non si garantiva un’esattezza del risultato, prestazioni erogate in piccoli presidi come Lissone o Arcore che contavano pochi dati ed inoltre sono stati eliminati i record di tutte quelle persone residenti al di fuori della Lombardia per i motivi che verranno indicati nel seguito. Come step successivo si è passati all’eliminazione dei record duplicati e di tutte quelle righe che presentavano anomalie nei valori; per esempio, sono state trovate situazioni in cui la data di prenotazione era successiva alla data di prestazione e di conseguenza il tempo di attesa risultava negativo.

Come passo finale sono stati uniformati tutti i dati che presentavano lo stesso “Stato” ed “Errore”; infatti, all’inizio uno stesso errore, come “Data appuntamento differente”, generava come output due valori differenti, ovvero classificava queste persone che presentavano lo stesso errore sia come show che come no show. In questa situazione specifica, per esempio, si è deciso di considerare tutti i pazienti come show, coloro i quali si sono presentati in giorni o orari diversi rispetto all’appuntamento, poiché la prestazione è stata comunque erogata ma per molteplici motivazioni è stata differita di qualche ora o giorno.

Pertanto, dopo queste varie fasi di pulizia dei dati, il dataset oggetto di analisi presenta poco più di 30000 osservazioni e numerose variabili che vengono di seguito riportate:

IdModulo: Rappresenta il codice della ricetta di prescrizione della prestazione. È quindi il codice identificativo della prestazione che poi non verrà tenuto in considerazione per le analisi.

Ricetta Digitale: Variabile dicotomica creata sulla base del numero identificativo della ricetta che indica se la ricetta è in formato digitale oppure cartaceo.

Presidio: Indica il presidio ospedaliero presso cui si è svolta la prestazione sanitaria.

Data Prenotazione: indica la data in cui la prenotazione è stata effettuata presso gli sportelli del CUP dei vari presidi ospedalieri.

Data Appuntamento: indica la data in cui si è svolta o si sarebbe dovuta svolgere la prestazione nel caso il cui la persona non si sia presentata.

Data\_Diff: variabile creata per l’analisi che rappresenta la distanza in termini di giorni fra la data di prenotazione e la data dell’appuntamento. Rappresenta in gergo sanitario il cosiddetto “Tempo di Attesa”.

Anno: Indica l’anno dell’appuntamento.

Trimestre: Indica il trimestre a cui appartiene la data dell’appuntamento.

Mese: Indica il mese a cui appartiene la data dell’appuntamento

Orario: indica l’orario dell’appuntamento in formato ora, minuti, secondi

Ora: Indica l’ora dell’appuntamento

Sezione\_giornata: Indica la sezione della giornata a cui appartiene l’ora dell’appuntamento. Può assumere i valori “Mattina”, “Pomeriggio”, “Sera”

Descrizione prestazione: Rappresenta la nomenclatura della prestazione sanitaria

Descrizione: Variabile che indica se la prestazione afferisce alla specializzazione di Radiologia o Gastroenterologia

Prezzo Prestazione: Indica la tariffa della prestazione come stabilito da tariffario regionale, che quindi non rappresenta l’entità dell’importo pagato dall’utente ma è una proxy di quest’ultima poiché il prezzo effettivo è un dato non disponibile.

Esenzione: Rappresenta l’esenzione in possesso degli utenti. Nel caso in cui un cittadino non abbia diritto a alcun’esenzione dal ticket regionale, viene restituito un campo vuoto.

Tipo di esenzione: Variabile creata ad hoc per distinguere tra i vari tipi di esenzione dal pagamento che sono stati stabiliti da Regione Lombardia

Esente: Variabile dicotomica che indica se il paziente è in possesso di esenzione dal pagamento oppure no.

Sesso: indica il sesso dei pazienti.

Data di nascita: indica la data di nascita dei pazienti. Variabile non presa in considerazione ma utile soltanto ad estrarre l’età delle varie persone

Luogo di nascita: Paese di nascita di ciascun utente

Straniero: variabile dicotomica che rileva se una persona è nata fuori dall’Italia oppure nei territori Italiani.

Età: indica l’età dei pazienti al momento della prestazione programmata

Comune di residenza: Indica il comune di residenza dei pazienti

KM\_Presidio\_Residenza: variabile creata ad hoc che rappresenta la distanza in chilometri stradali fra il comune di residenza e il comune in cui è situato il presidio ospedaliero della prestazione in oggetto.

Errore: indica l’errore rilevato dal flusso informatico per cui si è rilevata una discordanza fra i dati. Variabile non presa in considerazione per l’analisi.

Stato: Stato del processo informatico riferito alla prenotazione. Assume vari valori a seconda che l’appuntamento sia stato prenotato, schedulato o accettato.

No Show: variabile target dicotomica che identifica se la persona rappresenta uno “Show” oppure un “No show”

Per favorire l’analisi e l’interpretazione del dato sono state codificate 4 variabili categoriche che vanno a suddividere in blocchi le 4 variabili continue di cui siamo a disposizione: “Range\_attesa”, “Fascia Prezzo”, “Fascià età”, “Fascia\_dist\_km” che discretizzano rispettivamente il tempo di attesa, la tariffa della prestazione, l’età e la distanza fra il presidio e il comune di residenza.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IdModulo** | **Ricetta\_**  **Digitale** | **Presidio** | **Data\_**  **Prenotazione** | **Data App** | **Data\_Diff** | **Anno** | **Trimestre** | **Mese** | **Orario** | **Ora** | **Sez\_**  **giornata** |
| 030A04375775233 | 1 | POLIAMB. OSP. VIMERCATE | 2019-10-03 | 2019-10-09 | 6 | 2019 | 4 | 10 | 101500 | 10 | mattina |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Descrizione\_**  **prestazione** | **Descrizione** | **Prezzo\_**  **prestazione** | **Esenzione** | **Tipo\_di\_**  **esenzione** | **Esente** | **Sesso** | **Data\_di\_**  **nascita** | **Luogo\_di\_**  **nascita** | **Straniero** | **Età** |
| TORACE 2  PR. RX | RADIOLOGIA | 7.90 | 048 | Malattie croniche | 1 | F | 20/06/1937 | MARIANO COMENSE | 0 | 84 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Comune di residenza** | **KM\_Presidio\_Residenza** | **Stato** | **Errore** | **No Show** |
| USMATE VELATE | 8 | 50 |  | SHOW |

Tabella 1. Esempio di una riga di dataset (per motivi di spazio è stata divisa in 3 tabelle)

* 1. **Analisi Esplorativa:**

Innanzitutto, sono state eliminate le variabili non rilevanti per le analisi ovvero: ID modulo, Data prenotazione, data appuntamento, orario, Descrizione\_prestazione, Esenzione, Data di nascita, luogo di nascita, Comune Residenza, stato, errore.

Poi sono state effettuate delle analisi sulle frequenze relative delle categorie delle molteplici variabili discrete producendo i risultati sottoriportati.

Gli aspetti interessanti e importanti tratti da questa prima analisi è che la distribuzione delle prestazioni per mese è abbastanza omogenea tranne per una prevalenza di prestazioni nel mese di Aprile e Maggio e una leggera diminuzione del numero di prestazioni nel mese di agosto e dicembre a causa delle ferie del personale sanitario e delle festività natalizie.

Per quanto riguarda la divisione delle prestazioni all’interno della giornata, si nota la prevalenza di appuntamenti al mattino, con una piccola diminuzione al pomeriggio e invece una minoranza di prestazioni effettuate la sera, ovvero dalle ore 18 alle ore 21. Inoltre, si evince una netta prevalenza delle prestazioni prescritte con ricetta digitale (74.16%) e allo stesso modo una grande maggioranza di pazienti che risultano non esenti, 74.9%, contro il 25.1% di pazienti esenti. Il sesso è equilibrato con il 53.74 % di femmine e 46.26% di maschi. La distribuzione delle prestazioni invece è sbilanciata poiché il 66.9% delle prestazioni prese in considerazione è stata erogato presso il presidio Osp. Di Vimercate, il 20.4% a Carate Brianza e solo il 9.5% a Giussano e il 3.1% a Seregno.

Per quanto riguarda la divisione delle prestazioni per specializzazione medica, si evince che l’analisi riguarda per il 77.4% prestazioni di Radiologia e per 22.57% prestazioni di Gastroenterologia.

Per quanto riguarda il tipo di esenzioni si nota come la maggior parte venga espressa da esenzioni per condizioni economiche (4886), malattie croniche (1635) e civili (717).

Il numero di persone straniere che ha usufruito delle prestazioni in oggetto è pari all’8.13% mentre la restante percentuale sono persone nate in Italia.

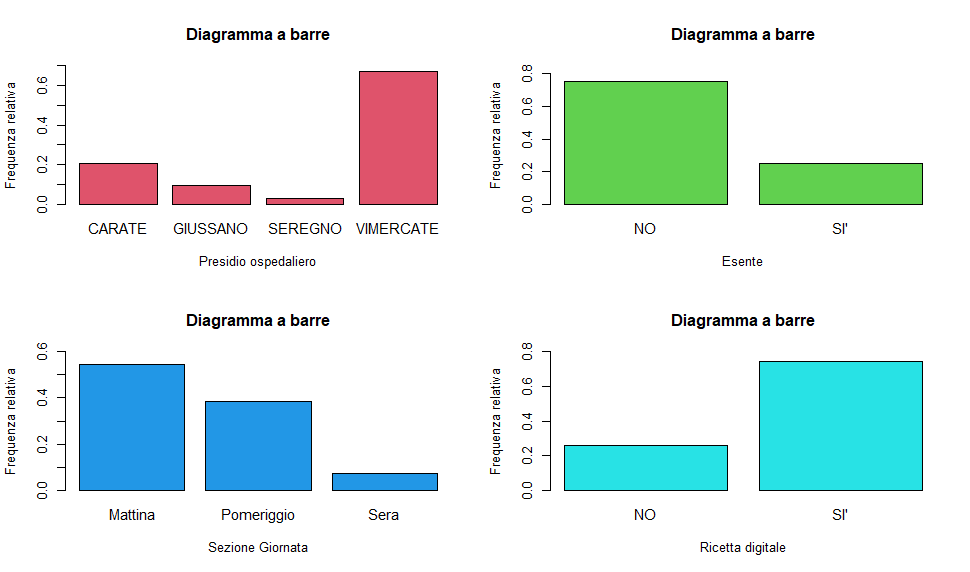


Figura 2. Diagramma a barre delle variabili presidio ospedaliero, esente, sezione giornata e ricetta digitale

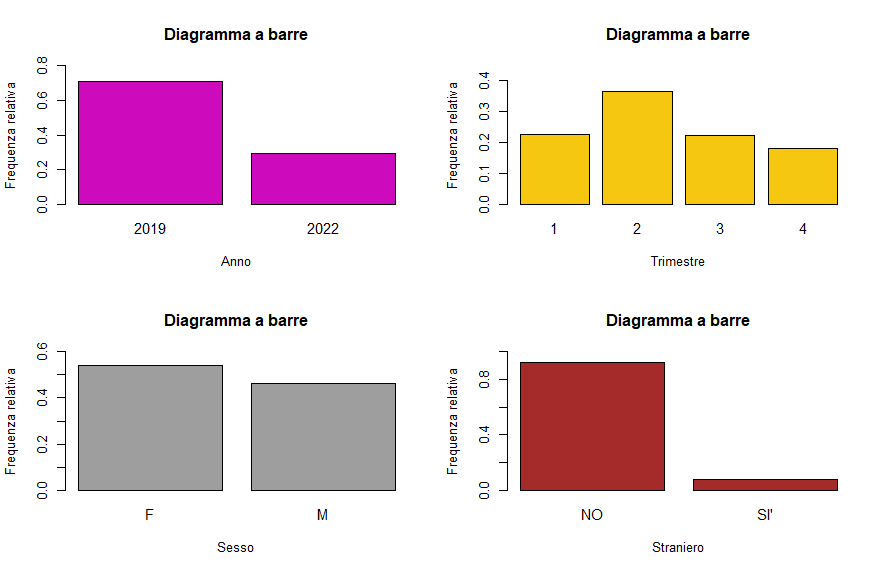


Figura 3. Diagramma a barre delle variabili anno, trimestre, sesso e straniero

Per quanto riguarda le variabili quantitative si possono fare le seguenti considerazioni:

Il tempo di attesa è una variabile fortemente asimmetrica che presenta una media di 46.59 giorni e una mediana di 14 giorni ma si possono osservare anche numerosi outliers di persone con alto tempo di attesa fino ad un massimo di 735 giorni.

La distanza media fra la residenza e il presidio è di 10.09 km con un minimo di 2km e un massimo di 100km

Il 50% delle osservazioni, che si trovano fra il primo e il terzo quartile, ha una distanza fra 5.5 e 12 km.

L’età è una variabile quantitativa più omogenea delle altre con una media di 60 anni, una mediana di 63 ma copre persone che vanno dalla nascita fino ai 101 anni. C’è comunque una prevalenza di anziani dato che lo scarto interquartile va da 48 a 75.

La tariffa della prestazione invece oscilla da un minimo di 7.9 euro ad un massimo di 350.9 euro anche se il prezzo medio è di 66 euro e la prevalenza delle prestazioni si trova fra i 17.4 euro e i 71.79 euro.

Tutte queste considerazioni vengono ben evidenziate dagli istogrammi come emerge dalla figura 2. Per il momento si decide di non applicare nessuna trasformazione alle variabili, per esempio logaritmica, dato che i piccoli miglioramenti in termini di performance del modello non sarebbero ripagati da una difficoltà interpretativa dei risultati maggiore.

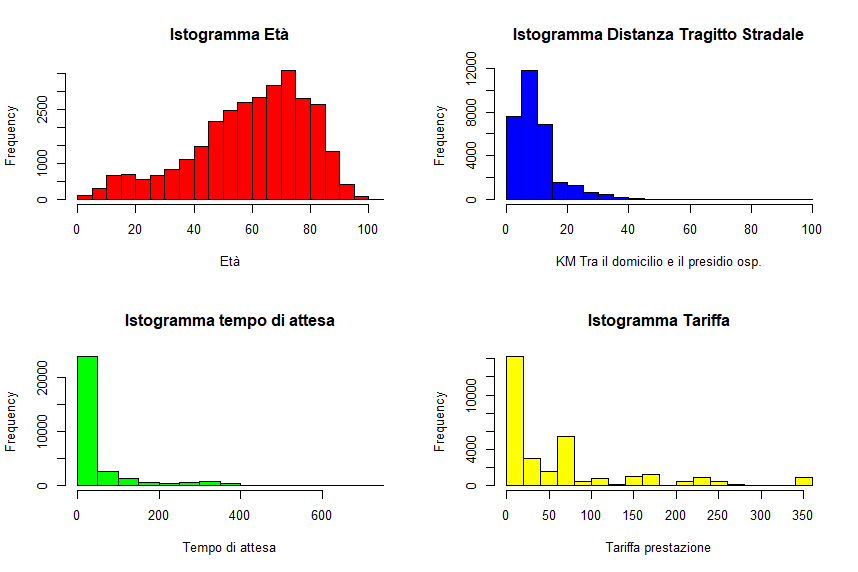


Figura 2. Istogrammi delle variabili quantitative

Gli outliers sono in percentuale molto rilevanti nel tempo di attesa (14.92%) e nella tariffa della prestazione con 13.28%. Minore invece la percentuale di outliers per la variabile “KM\_Presidio\_Resid” (8.03%) mentre l’età presenta una percentuale molto vicina allo zero di dati anomali.

Per quanto concerne la variabile etichetta di riferimento possiamo notare come si siano presentati alle visite 28742 pazienti contro invece 1875 pazienti considerati no show, rappresentando rispettivamente il 93.88 % e il 6.12% delle istanze. In pratica ogni 100 persone, 6.12 non usufruiscono poi della prestazione sanitaria per cui si erano prenotati. Dal punto di vista sanitario non è una percentuale bassa di assenti dato che le aspettative pre analisi si attestavano tra il 2 e il 4% mentre dal punto di vista statistico rappresenta un notevole sbilanciamento nelle classi.

Un aspetto molto importante e soddisfacente è il fatto che non ci siano valori mancanti e non sia quindi necessario utilizzare nessuna tecnica di gestione dei missing values, le quali di solito tendono a modificare leggermente l’informazione trasmessa dai dati.

* 1. **Analisi quantitativa:**

In questa fase vengono effettuate una serie di analisi quantitative volte a stabilire l’influenza e l’andamento delle variabili categoriche. In particolare, si applica un raggruppamento dei dati sulla base dei vari valori assunti dalle variabili e si calcola la percentuale di no show. Per esempio, si raggruppano i dati per mese e si calcola la percentuale di no show da gennaio fino a dicembre in modo da comprendere quale mese ha registrato un valore maggiore per questo fenomeno; in pratica si produce la percentuale della variabile risposta condizionatamente alle varie categorie assunte dalla variabile esplicative.

Immagine che contiene tavolo

Descrizione generata automaticamenteImmagine che contiene tavolo

Descrizione generata automaticamente

Tabella 3. % di no show per ogni mese

Tabella 4. % di no show per ogni ora

Nel mese di dicembre c'è stato qualche problema di raccolta dati o qualche emergenza sanitaria poiché il valore restituito è molto alto (41%). In generale i mesi dove la percentuale di no show è minore sono aprile e maggio mentre la percentuale maggiore si verifica a gennaio e novembre rispettivamente con 8.61% e 9.24%. Data la presenza dei risultati anomali di dicembre si decide di eliminare, per le successive analisi, tutte le osservazioni riferite a questo mese poiché modificherebbero molto i risultati ottenuti.

L’andamento delle percentuali di no show in base all’ora dell’appuntamento sembra seguire il seguente andamento: si parte con un dato basso alle 8 del mattino (4.01%) per poi risalire sempre di più fino al picco delle ore 14 con 6.74% di no show e scendere nuovamente fino al 2.39% riferito alle ore 18.

Si nota quindi come nel tardo pomeriggio o sera, la percentuale di no show sia minore.

Immagine che contiene tavolo

Descrizione generata automaticamenteImmagine che contiene tavolo

Descrizione generata automaticamente

Tabella 5. % di no show per ogni presidio ospedaliero

Tabella 6. % di no show per tipo di ricetta

Togliendo gli outliers di dicembre, il presidio con la percentuale maggiore di no show è Carate, poi Giussano poi Vimercate ed infine Seregno.

Dalla tabella 6 invece si trae il fatto che la percentuale di no show è più bassa per le persone con ricetta digitale poiché essa non è riutilizzabile più volte e di conseguenza l’utente non può prenotare la stessa prestazione in più ospedali evitando che si rechi altrove senza disdire il precedente appuntamento.

Immagine che contiene tavolo

Descrizione generata automaticamenteImmagine che contiene tavolo

Descrizione generata automaticamente

Tabella 7. % di no show per esenzione

Tabella 8. % di no show per nazionalità

Nella tabella 7 si nota come nelle persone esenti il fenomeno dei no show è maggiore rispetto alle persone prive di esenzione mentre nella tabella 8 si vede come il fatto di essere straniero e la conseguente difficoltà linguistica incida sul fatto di essere no show poiché la percentuale risultante è pari a 7.85 contro il 5% delle persone italiane.

Immagine che contiene tavolo

Descrizione generata automaticamenteImmagine che contiene tavolo

Descrizione generata automaticamente

Tabella 9. % di no show per tipo di prestazione

Tabella 10. % di no show per anno

Dalla tabella 9 si evince che la radiologia restituisce una percentuale di no show minore anche se calcolati su un numero di prestazioni di gran lunga superiore (29237 appuntamenti contro 6552) mentre la tabella 10 racconta che nel 2019 ci sono stati meno no show in percentuale e su questa differenza probabilmente ha inciso anche il fattore Covid-19 poiché molte persone si sono trovate costrette a non presenziare all’appuntamento a causa della positività o di isolamenti domiciliari.

Immagine che contiene tavolo

Descrizione generata automaticamenteImmagine che contiene tavolo

Descrizione generata automaticamente

Tabella 11. % di no show per fascia di distanza fra domicilio e presidio

Tabella 12. % di no show per range di tariffa della prestazione

Dalla tabella 11 si nota come tendenzialmente più aumenta la distanza fra residenza e presidio, più aumenta la percentuale di no show. Questo è valido per ogni range di km passando da 4.1%, per una distanza percorsa di 2-5 km, a 8.97% tra le persone che devono fare più di 20 km di strada.

Prendendo in considerazione la tariffa della prestazione, sono stati preventivamente esclusi da questa analisi gli esenti dal ticket poiché per loro non c’è alcun esborso monetario. Dai calcoli si vede come la percentuale di no show è mediamente bassa (4.92%) quando la fascia di prezzo è bassa (7-18 euro) poi diventa la più bassa (3.88%) quando la fascia di prezzo è 18-40 euro; in seguito aumenta molto e si mantiene abbastanza costante quando la fascia di prezzo è media (40-70 euro, 70-100 euro); per poi alzarsi molto fino a toccare il picco quando la fascia di prezzo supera i 180 euro con il 7.62% di no show.

Immagine che contiene tavolo

Descrizione generata automaticamenteImmagine che contiene tavolo

Descrizione generata automaticamente

Tabella 13. % di no show per fascia di età

Tabella 14. % di no show per range di tempo di attesa

Dalla tabella 13 si nota come la maggior percentuale di no show si trova tra i 20 e i 39 anni, con una netta differenza rispetto alle altre fasce di età; invece, la fascia di età con la minor percentuale di no show è quella più anziana di 80-99 anni. A differenza di quanto si pensava le categorie anziane sono quelle che si presentano maggiormente agli appuntamenti probabilmente perché hanno meno impegni che portano a saltare l’appuntamento e sentono maggiormente la necessità della visita prenotata.

Dalla tabella 14 invece risulta che se il tempo di attesa è inferiore alla settimana la % di no show è molto bassa come previsto (2.98%). Nelle tre fasce di attesa successive la % di no show si mantiene pressoché costante attorno al 4.5/5% mentre si alza quando l'attesa supera il mese arrivando al picco di no show tra i 90 e i 180 giorni con 11.9% di no show.

L’interpretazione è la seguente: quando la prestazione viene erogata pochi giorni dopo l’appuntamento, è molto probabile che il paziente sia presente poiché diminuisce la possibilità che si prenoti altrove, che si dimentichi dell’appuntamento o che avvengano degli eventi imprevisti che proibiscano alla persona di presentarsi; questo è molto evidente per le prestazioni prenotate all’interno di una settimana. Quando invece il tempo di attesa cresce, cresce parallelamente anche la probabilità di non presentarsi all’appuntamento e questo raggiunge il suo culmine quando l’appuntamento viene dato ad una distanza di 3-6 mesi. Invece quando il tempo di attesa è molto alto, ovvero sopra i 6 mesi, la percentuale di no show diminuisce di quasi 5 punti poiché, avendo aspettato a lungo per l’appuntamento, i pazienti tendono a presentarsi e a non vanificare la lunga attesa.

* 1. **Visualizzazione grafica:**

Per realizzare una più chiara rappresentazione grafica le prestazioni sono state raggruppate per ogni data, calcolando anche la percentuale di no show per ogni data di appuntamento.

Inoltre, per ogni raggruppamento è stata calcolata la media delle variabili quantitative a disposizione in modo tale che sull’asse delle ordinate venisse rappresentata la percentuale di no show per ogni data al variare della caratteristica esplicativa presa in considerazione.

Sono stati inoltre eliminati i records outliers ovvero quelle date in cui la percentuale dei no show risultava essere oltre il 25%, dato sicuramente generato da errori informatici o da problemi sanitari di quei giorni.

La quasi totalità di queste date appartiene a dicembre 2019.

Inoltre, come metodo di regressione per interpolare i dati e spiegare la relazione tra le variabili ho usato la regressione non parametrica di tipo Loess, molto utile quando si tratta di rappresentare la relazione su un piano fra una variabile risposta e una variabile esplicativa

Tempo di attesa:

si nota che quando il tempo di attesa è vicino allo 0, ovvero di pochi giorni, la percentuale di no show è molto bassa, poi si alza quando i tempi di attesa sono tra i 15 e i 30 giorni, in seguito invece torna ad abbassarsi leggermente quando i tempi di attesa sono fra 30 e 50 giorni mentre aumenta nuovamente molto quando i tempi di attesa sono molto alti (ovvero superiori ai 50 giorni), tranne per quel valore anomalo finale che abbassa notevolmente la curva.

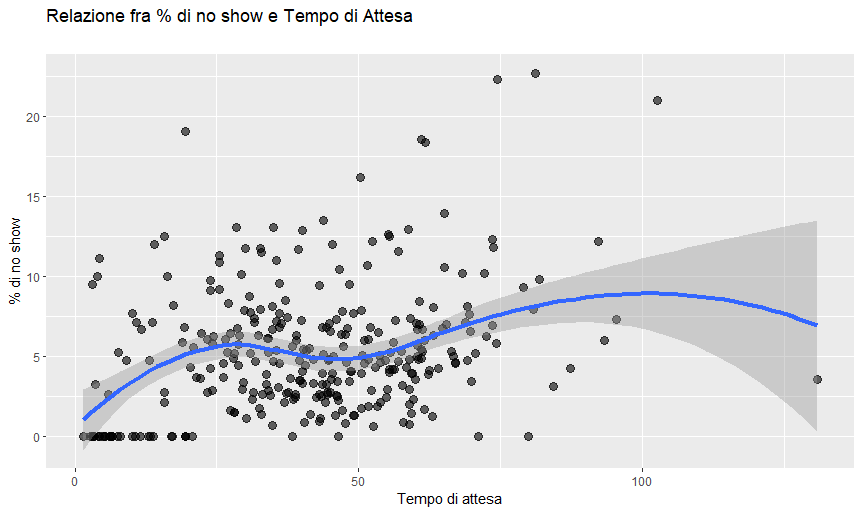


Figura 15. Scatterplot della relazione fra % di no show e tempo di attesa con modello di regressione non parametrica di tipo Loess

Distanza fra residenza e presidio:

il grafico ha poco significato perché è costituito da una nube di punti molto ravvicinati che è difficile da interpolare con qualsiasi retta; la distanza in km sembra influire poco sulla percentuale dei no show anche se si può trarre la seguente conclusione: se la distanza è bassa, la % di no show è bassa ma poi tende ad alzarsi fino a quando la distanza media raggiunge gli 11km per poi decrescere molto lentamente dagli 11km in poi.

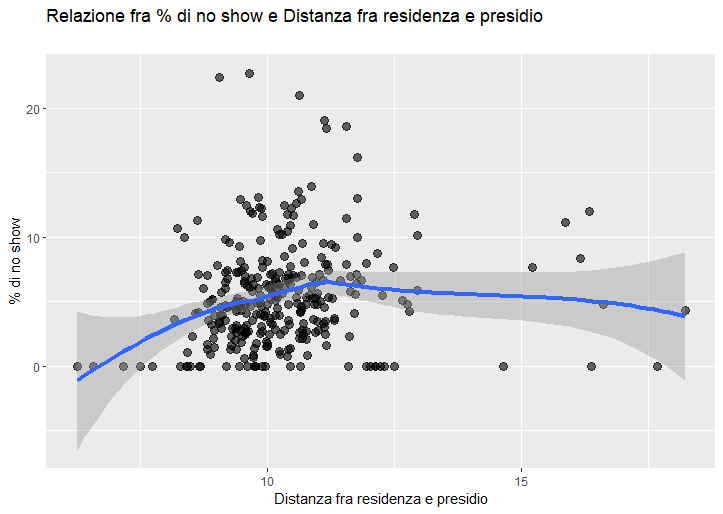


Figura 16. Scatterplot della relazione fra % di no show e distanza fra residenza e presidio con modello di regressione non parametrica di tipo Loess

Età:

L'età media sembra avere poco significato dato che la retta è pressoché costante ma si può notare una leggera diminuzione quando l'età media aumenta soprattutto in età avanzata, ovvero sopra i 68 anni di media.

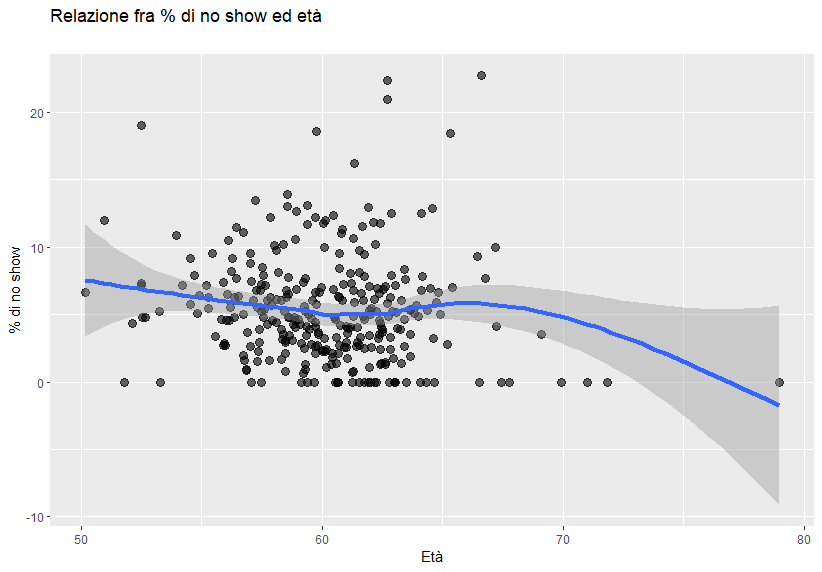


Figura 17. Scatterplot della relazione fra % di no show ed età con modello di regressione non parametrica di tipo Loess

Tariffa prestazione:

Quando la tariffa è bassa la percentuale è più alta, la quale tende ad abbassarsi quando la tariffa si trova tra 90 e 130 euro; infine la percentuale tende a rimanere abbastanza costante con una leggera decrescita dopo i 130 euro.

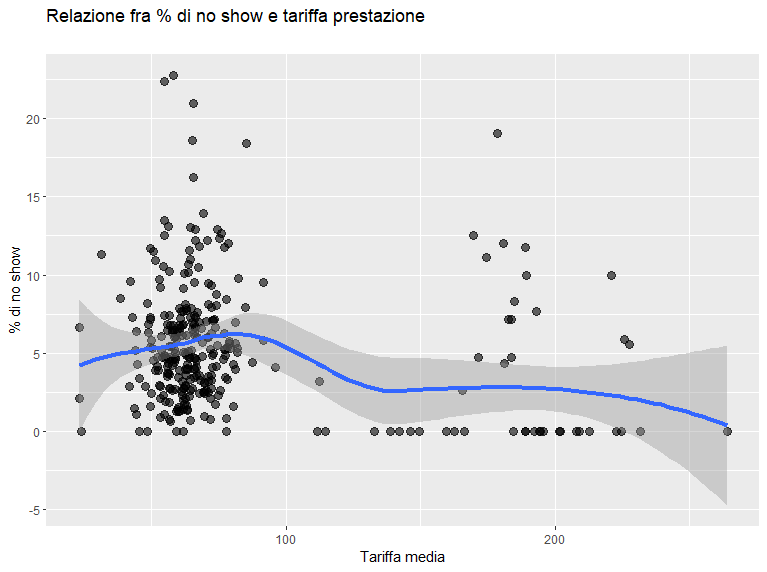


Figura 18. Scatterplot della relazione fra % di no show e tariffa prestazione con modello di regressione non parametrica di tipo Loess

**Correlazione:**

Si analizza la correlazione fra le variabili quantitative del dataset per studiare eventuali relazioni lineari fra le variabili esplicative.

Le variabili sono tutte poco correlate fra di loro e questo fenomeno è positivo per evitare problemi di multicollinearità e di dipendenza tra variabili.

Le uniche correlazioni leggermente alte sono -0.51 tra media\_tempo\_attesa e Media\_tariffa\_prestazione, ovvero all'aumentare del tempo di attesa diminuisce la tariffa della prestazione e viceversa all'aumentare della tariffa , diminuisce il tempo di attesa.

L’altra correlazione relativamente elevata è 0.36 tra Media\_tariffa\_prestazione e Media\_distanza\_km ovvero all’aumentare della tariffa della prestazione aumenta la distanza percorsa dagli utenti e viceversa, le prestazioni che costano meno sono correlate con i pazienti che abitano vicino agli ospedali.

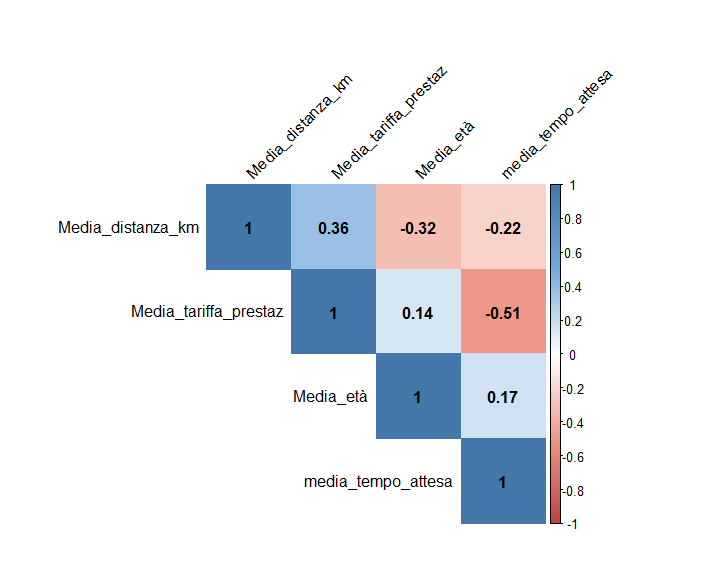


Figura 19. Matrice di correlazione fra le 4 variabili quantitative

**Regressione logistica:**

Inizialmente viene stimato un modello di regressione logistica (rif. par. 2.1) con tutte le variabili a disposizione, tranne le variabili range, e tramite il link canonico ovvero il link logit. La devianza ottenuta, ovvero la distanza dal modello saturo che costituisce il miglior modello possibile, è bassa perché risulta essere 12883 su 30573 gradi di libertà.

Si calcola il test per la bontà del modello con il relativo p\_value equivalente a 1; si accetta, quindi, l’ipotesi nulla che il modello sia adeguato, ovvero sia sufficientemente vicino al modello saturo in termini di estremo superiore della log-verosimiglianza.

Il numero di iterazioni effettuate dall’algoritmo IRLS è pari a 12, un numero non basso rispetto al solito dato l’alto numero di variabili e di unità statistiche. Finora, quindi, il modello calcolato è complesso ma corretto.

Si passa poi all’analisi delle variabili significative, le quali risultano essere le seguenti:

Per ogni variabile viene restituito il valore del coefficiente di regressione. Di esso poi si esegue la trasformazione esponenziale, si sottrae uno (e si moltiplica per 100) in modo tale da ottenere l’odds ratio, ovvero un rapporto fra ragioni di scommessa. Per esempio, se la ragione di scommessa è pari a 4, la persona è disposta a scommettere 4 euro per ottenerne 1 in caso di vincita, mentre in caso negativo si perde tutto.

L’odds ratio se è maggiore di 1, oppure se è positivo in percentuale, significa che è presente un’associazione positiva fra la variabile esplicativa aumentata di un’unita e la variabile risposta.

•Ricetta digitale:

(exp(0.1650041)-1)\*100= 17.94 rappresenta l’odds ratio; ciò significa che passando da ricetta cartacea a ricetta digitale l'odds ratio relativo alla probabilità di essere SHOW aumenta del 17.94%; sussiste quindi un’associazione positiva fra ricetta digitale ed essere show.

•Tariffa\_prestazione:

(exp(-1.507e-03)-1)\*100= -0.150 rappresenta l’odds ratio; aumentando di un euro la tariffa prestazione l'odds ratio relativo alla probabilità di essere SHOW diminuisce dello 0.15%; è presente quindi un’associazione negativa fra tariffa prestazione ed essere show.

•Distanza\_Residenza\_presidio:

(exp(-1.777e-02)-1)\*100= -1.76% rappresenta l’odds ratio; aumentando di un km la distanza fra residenza e presidio l'odds ratio relativo alla probabilità di essere SHOW diminuisce dell'1.76%; associazione negativa fra tragitto stradale ed essere show.

•Esente:

(exp(-2.581e+00)-1)\*100= -92.43017% rappresenta l’odds ratio; passando da essere non esente a esente l'odds ratio relativo alla probabilità di essere SHOW diminuisce del 92.43%; associazione fortemente negativa fra essere esente ed essere show.

•Sesso:

(exp(2.890e-01)-1)\*100= 33.098% rappresenta l’odds ratio; passando da maschio a femmina l'odds ratio relativo alla probabilità di essere SHOW aumenta del 33.5%; associazione positiva fra essere femmina ed essere show

•Tempo di attesa:

(exp(-3.165e-03)-1)\*100= -0.32% rappresenta l’odds ratio; aumentando di un giorno il tempo di attesa l'odds ratio relativo alla probabilità di essere SHOW diminuisce dello 0.32%; associazione negativa fra tempo di attesa ed essere show.

•Ora:

(exp(-5.643e-02)-1)\*100= -5.49% rappresenta l’odds ratio; aumentando di un'ora l'orario della visita l'odds ratio relativo alla probabilità di essere SHOW diminuisce dello 5.49%; associazione negativa fra ora ed essere show.

•Sezione Giornata:

(exp(6.590e-01)-1)\*100= 93.28585 rappresenta l’odds ratio; passando da mattina a sera l'odds ratio relativo alla probabilità di essere SHOW aumenta del 93.29%; associazione positiva fra appuntamenti serali ed essere show.

•Anno:

(exp(7.133e-01)-1)\*100= 104.0715 rappresenta l’odds ratio; passando dall'anno 2022 al 2019 l'odds ratio relativo alla probabilità di essere SHOW aumenta del 104%; associazione positiva fra anno 2019 ed essere show.

•Fascia d’età:

(exp(-4.487e-01)-1)\*100= -36.15424 rappresenta l’odds ratio; passando dalla fascia di età 0-20 anni alla fascia 20-39 l'odds ratio relativo alla probabilità di essere SHOW diminuisce del 36%; associazione negativa fra le persone con età fra 20 e 39 ed essere show.

•Straniero:

(exp(-3.850e-01)-1)\*100= -31.95494 rappresenta l’odds ratio; passando da persone italiane a persone straniere l'odds ratio relativo alla probabilità di essere SHOW diminuisce del 32%; associazione negativa fra persone straniere ed essere show.

|  |  |
| --- | --- |
| **Variabili con associazione positiva** | **Variabili con associazione negativa** |
| Ricetta digitale (17.94%) | Tariffa\_prestazione (-0.15%) |
| Sezione Giornata Serale (93.29%) | Distanza\_Residenza\_presidio (-1.76%) |
| Sesso Femmina (33.5%) | Esente (-92.43%) |
| Anno 2019 (104%) | Tempo di attesa (-0.32%) |
|  | Ora (-5.49%) |
|  | Straniero (-32%) |
|  | Fascia d’età 20-39 anni (-36%) |

Diagnostica:

Viene esaminata la diagnostica del modello logistico che però come dimostrato dalla figura 20 risulta essere problematica; tuttavia, trattandosi di una situazione di dati sparsi, ovvero non esistendo molte persone con le medesime caratteristiche per tutte le variabili, la diagnostica non è significativa e da tenere in considerazione per interpretazioni.

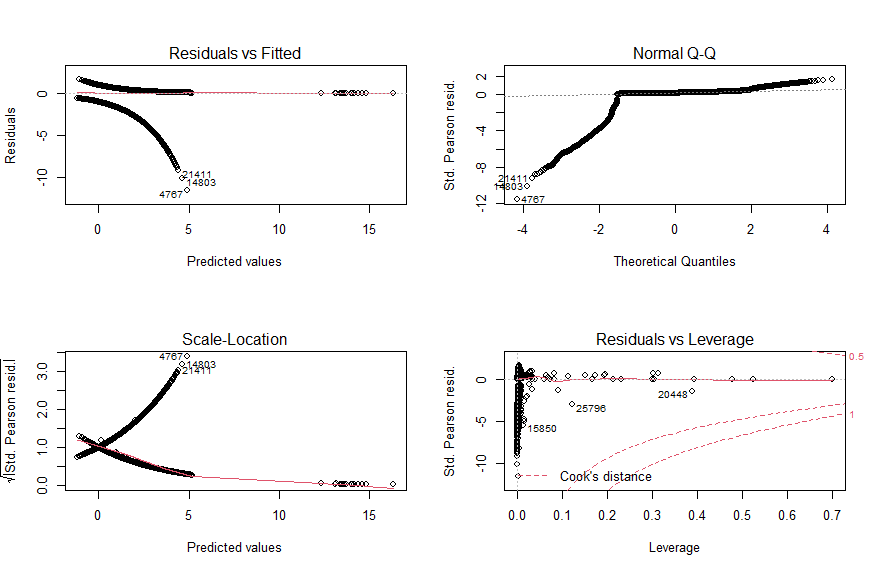


Figura 20. Quattro grafici principali per la diagnostica del modello di regressione logistica

Si verifica la presenza di outliers nel modello attraverso il test di Bonferroni. Le unità statistiche con un p\_value al di sotto del 5% vengono considerate valori anomali. In questo caso non vengono riscontrati valori anomali tranne per la riga 4767 che però presenta soltanto il p\_value non aggiustato inferiore al 5%.

Per quanto riguarda i punti influenti se ne trovano soltanto sei che corrispondono ai seguenti record:

4767 , 9039 , 14803, 20448, 24592 , 25796.

Essendo pochi e non provocando problemi di dati anomali, si decide comunque di mantenere tali punti influenti nel modello.

Infine, si è pensato di analizzare il modello logistico usando anche gli altri link principali: probit, cauchy e cloglog. Stimando i vari modelli con i nuovi tre link e confrontando anche i risultati con il modello col link canonico, si nota che l’AIC minore e la devianza minore si ottengono con il link cloglog. L’AIC è una delle principali misure per sintetizzare la bontà del modello; AIC bassi significano una migliore performance del modello.

In seguito, si cerca di diminuire il numero di variabili esplicative usate nel modello e preservare soltanto quelle significative ma applicando l’algoritmo step con direzione backward, l’AIC e la devianza non migliorano in maniera rilevante.

Provando invece ad inserire qualche interazione, l’AIC peggiora e la devianza diminuisce anche se a favore di un aumento dei parametri stimati e quindi del numero di gradi di libertà; si decide pertanto di non inserire le interazioni nel modello dato che aumenterebbe esponenzialmente il numero di parametri da stimare rendendo il modello troppo complesso ed inoltre non garantendo una chiara interpretazione dei risultati a fronte di un piccolo miglioramento del fit.

Utilizzando la visualizzazione grafica (figura 21.) si è rappresentato un plot contenente sull’asse delle ascisse la distanza fra domicilio e presidio ospedaliero che risulta essere una variabile quantitativa significativa e sull’asse delle ordinate l’esito della classificazione come show oppure no show.

Inoltre è stato stimato un modello di regressione univariato, quindi con un unico regressore, e si è proiettato in verde la stima della previsione delle unità statistiche al variare della distanza dove il valore 1 rappresenta essere show e 0 significa essere no show. Come si mostrato precedentemente, secondo il modello logistico la distanza influisce negativamente sul fatto di presentarsi all’appuntamento; infatti la linea verde spiega come all’aumentare della distanza, ci si avvicina sempre di più verso lo 0 che significa essere no show.

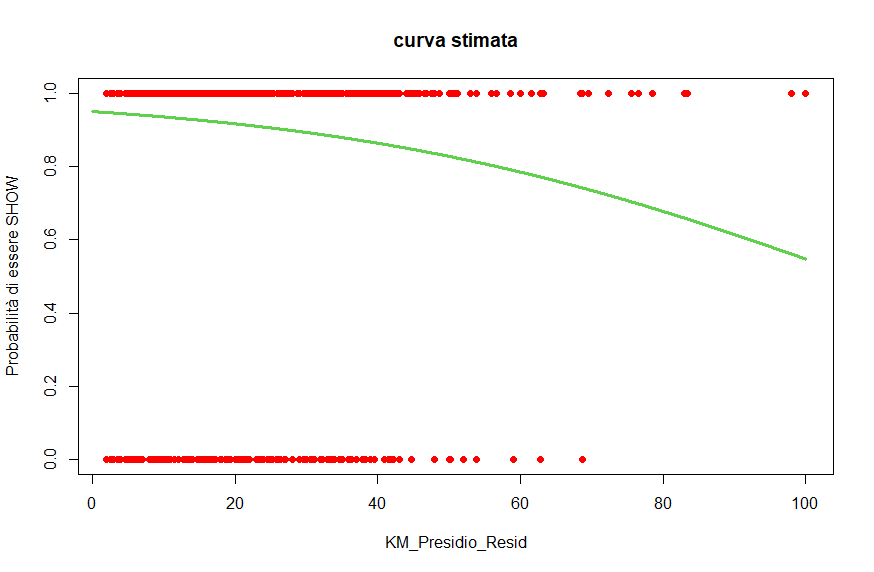


Figura 21. Grafico della suddivisione fra show e no show in base alla distanza fra residenza e presidio ospedaliero con l’aggiunta della curva verde che rappresenta la curva stimata

**Regressione logistica per la classificazione:**

Per iniziare si suddivide il dataset in training, validation e test set.

Il training set servirà per l’allenamento del classificatore e la stima dei parametri, il validation set per l’ottimizzazione degli iperparametri e il test set per verificare e testare i risultati ottenuti in un insieme di dati in cui non si conosce la variabile target. Infatti mentre nel training set si conosce la classificazione della variabile target per far imparare al modello quando un’istanza deve essere classificata come show oppure no show, nel test set si calcola la bontà dell’algoritmo allenato misurando varie metriche.

Il dataset viene suddiviso attribuendo l’80% delle osservazioni al training set e il 20% al test set; a sua volta il training set viene diviso ponendo il 65% delle istanze al sub\_training set e il 35% al validation set.

Alla fine di queste divisioni l’allocazione delle unità statistiche sarà la seguente: 52% training set (sub training) 28% validation set e 20 % test set.

Un aspetto importante da verificare è che la ripartizione in classe della variabile target sia simile per tutti e 3 i sottodataset creati; in effetti questo si verifica perché nel sub training e nel validation la ripartizione è 7% no show contrapposto ad un 93% di show e nel test set queste percentuali sono 6% e 94%.

Per la validità della regressione logistica non è necessaria nessuna assunzione a priori sulla distribuzione delle variabili quindi l’unico aspetto che si va a controllare, per assicurarsi un modello senza problemi di multicollinearità, è la correlazione fra le variabili numeriche del sub\_training.

Come per il dataset originale, anche per questo sottoinsieme le variabili risultano molto poco correlate portando a non compiere ulteriori operazioni preliminari sui dati.

Il modello costruito ha le stesse caratteristiche di quello precedentemente analizzato poiché si sceglie di applicare il link cloglog e di utilizzare tutte le variabili esplicative utilizzate nella regressione del paragrafo sopra.

Sulla base del modello appena costruito si calcolano le previsioni sulle osservazioni del validation set.

Il modello logistico, restituisce sempre delle probabilità per la variabile target, ovvero dei numeri compresi fra 0 e 1. Tutte le previsioni maggiori di 0.5 portano alla classificazione dell’utente come show mentre tutte le previsioni minori di 0.5 classificano l’utente come no show.

Utilizzando questa soglia di default si ottiene un’accuracy del 93.39% ma non è la metrica rilevante con classi sbilanciate. La metrica più interessante da analizzare è la balanced accuracy, ovvero la media fra sensitivity e specificity, che a loro volta indicano rispettivamente il true positive rate e il true negative rate. Con la linea di confine fissata a 0.5 la balanced accuracy misura 0.78 mentre provando a far variare il cutoff, sia aumentandolo che diminuendolo, la metrica in esame peggiora in performance.

Dopo il bilanciamento del dataset (rif. par. 2.2) si ripetono i calcoli effettuati in precedenza stimando lo stesso modello logistico per la classificazione e si ottengono i seguenti risultati:

l’accuracy ora è la metrica ora da tenere in considerazione per la performance, la quale però cambia al variare dell’estrazione casuale del campione. Per questo motivo si effettua una moltitudine di estrazioni casuali e alla fine si fa una media delle accuracy misurate sui vari campioni. Il risultato finale mostra un’accuratezza del 65.1%.

Cercando di modificare il cutoff anche per i dati bilanciati si ottiene un gran miglioramento dopo aver posto la soglia a 0.15; in questo modo infatti l’accuratezza misura in media 92.75%, ovvero viene classificato in modo errato soltanto il 7.25% delle unità statistiche. Il validation set non è stato bilanciato perchè rappresenta la situazione reale, ovvero caratterizzata da una forte prevalenza di show rispetto ai no show.

Ripercorrendo le medesime tecniche anche sul test set, si ottiene la stessa performance (attorno al 92%) mostrando come il modello non soffra di overfitting, ovvero la situazione in cui un modello si comporta molto bene sui dati di training grazie all’allenamento con una grande mole di dati ma poi quando deve prevedere il comportamento di nuove istanze performa in maniera decisamente peggiore.

**Clustering gerarchico:**

Si applica il clustering gerarchico (rif. par. 2.3) sulle 4 variabili numeriche a disposizione provando con tutti e 3 i tipi di legame: singolo, medio e completo. Vengono impostati 2 gruppi in modo tale che si possa elaborare una classificazione rispetto alla realtà.

Il single linkage fornisce un error rate di solo 6.59 % poiché la classificazione è a classi sbilanciate. Analizzando la ripartizione della classificazione delle etichette si nota come il metodo classifichi solo 7 u.s. come no show e 30610 come show, non discriminando quindi accuratamente tra le classi. Prendendo in considerazione come metrica l’ARI, risulta negativo e questo porta a pensare ad una classificazione non corretta. Il risultato evidente è dato da una Specificity pari 99.9% contrapposta ad una Sensitivity pari allo 0%, come si evince dalla tabella 22, poiché vengono previsti 0 no show di quanti effettivamente non si sono presentati.

|  |  |  |
| --- | --- | --- |
|  | **Predicted** | |
| **Class** | No show | Show |
| No show | 0 | 7 |
| Show | 2019 | 28591 |

Tabella 22. Confusion matrix per clustering gerarchico e legame singolo con dati sbilanciati

Con il legame medio i risultati non cambiano rispetto al caso precedente con l’error rate e l’ARI che restano identici in valore; anche per la ripartizione della variabile target i risultati sono i medesimi così come per la matrice di confusione che viene prodotta con gli stessi dati. (rif. Tab. 22.)

|  |  |  |
| --- | --- | --- |
|  | **Predicted** | |
| **Class** | No show | Show |
| No show | 0 | 7 |
| Show | 2019 | 28591 |

Tabella 23. Confusion matrix per clustering gerarchico e legame medio con dati sbilanciati

Passando al legame completo, i risultati migliorano rispetto ai due casi sopracitati poiché vengono classificati come no show ben 1143 records anche se con una sensitivity ancora molto bassa (3.1%) e con una specificity molto alta (96.2%) come riportato dalla tabella 24.

|  |  |  |
| --- | --- | --- |
|  | **Predicted** | |
| **Class** | No show | Show |
| No show | 64 | 1079 |
| Show | 1955 | 27519 |

Tabella 24. Confusion matrix per clustering gerarchico e legame medio con dati sbilanciati

**K-means clustering:**

Anche per questo metodo (rif par. 2.4), si prendono in considerazione le variabili numeriche, essendo un algoritmo basato sulla matrice di distanze fra osservazioni. Poiché l’obiettivo rimane quello di classificare rispetto alla realtà formata da due gruppi, si fissa il numero di centroidi pari a 2 mentre il numero di iterazioni dell’algoritmo viene fatto variare fra 10 e 50 anche se non produce sostanziali differenze nei risultati.

I risultati ottenuti sono diversi in confronto al clustering gerarchico con un netto miglioramento della sensitivity che sale al 29.3% e ad una piccola diminuzione della specificity (80.5%). In sintesi la classificazione migliora e ciò è confermato anche dall’ARI che assume valori positivi nonostante rimanga un valore molto basso.

Il metodo divide le u.s. in 2 gruppi con le seguenti dimensioni: 6147 istanze per il gruppo 1 e 24470 istanze per il gruppo 2; una partizione sbilanciata che ricalca la situazione reale di forte prevalenza di show.

Analizzando i 2 centroidi si evince come il primo gruppo sia caratterizzato da un alto tempo di attesa, una tariffa molto costosa, un’età media avanzata e una distanza da percorrere più elevata.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gruppo** | **data\_diff** | **PrezzoPrestazione** | **Età** | **KM\_Presidio\_Resid** |
| 1 | 154.71938 | 167.84144 | 61.98145 | 11.779746 |
| 2 | 19.42987 | 40.47832 | 59.49620 | 9.664548 |

Tabella 25. Centroidi ottenuti con il metodo delle k medie sul dataset sbilanciato

Tramite il metodo della silhouette si stima la bontà dei clusters che si sono venuti a creare.

Il primo gruppo riporta un valore basso pari 0.13 che denota un cluster formato da istanze che sono poco simili fra di loro mentre il secondo gruppo porta una silhouette di 0.74 che è indice di un cluster ben formato con poca distanza fra i punti all’interno del gruppo.

Inoltre calcolando l’algoritmo delle k medie facendo variare il numero dei gruppi k e generando in seguito la silhouette media il risultato è che il valore maggiore si ottiene con 3 gruppi (silhouette = 0.66) mentre la silhouette media con 2 gruppi è poco inferiore (0.61).

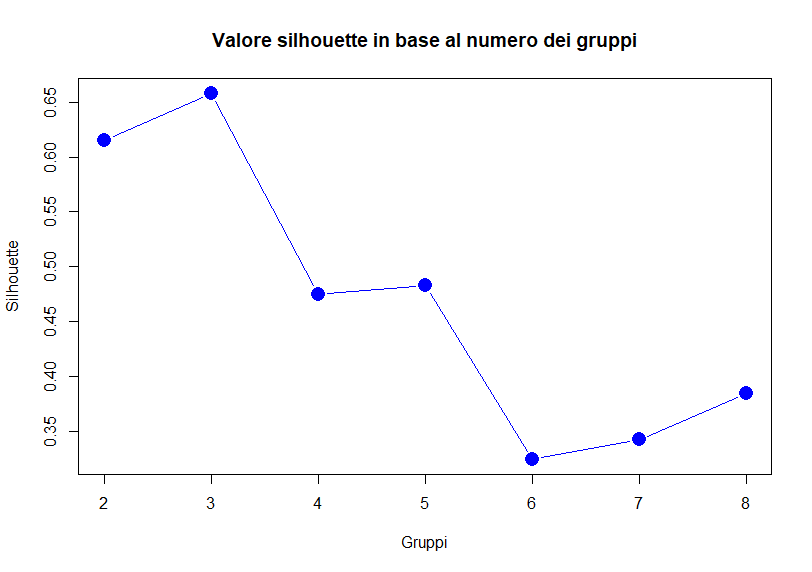


Figura 26. Silhouette al variare del numero dei gruppi.

**Dati bilanciati:**

Le stesse analisi effettuate nei paragrafi precedenti vengono ripetute con le stesse caratteristiche anche sul dataset bilanciato con variabile target distribuita 50 e 50. (rif. par. 2.2)

Calcolando l’error rate con il legame singolo si ottiene il 49.9% il quale rappresenta un pessimo risultato perchè equivale a lanciare casualmente una moneta. L'errore così alto è dato dal fatto che l'algoritmo considera 8074 osservazioni su 8076 appartenenti ad un solo gruppo e le restanti due u.s. appartenenti all’altro gruppo.

Anche in questo caso il legame medio produce gli stessi risultati del legame singolo confermando la somiglianza di questi due metodi in rapporto al tipo di dati oggetto di studio.

Con il legame completo l’outcome cambia leggermente portando ad un error rate minore (49.4%).

Il problema di questo metodo come per gli altri casi consiste nel fatto che 3900 u.s. su 8076 vengono classificate come show quando in realtà sono no show restituendo una sensitivity pessima (3.4%).

Con il k-means clustering (rif. par. 2.4) sul dataset bilanciato si ottengono risultati leggermente migliori in termini di accuracy che questa volta rappresenta la metrica rilevante ma i fattori che incidono su questo valore sono opposti rispetto al clustering gerarchico. Infatti la sensitivity migliora radicalmente arrivando al 72% in contrapposizione con una bassa specificity (30.2%).

Il metodo non gerarchico suddivide il dataset in due gruppi da 6057 e 2019 istanze e restituisce dei centroidi distanti in termini di valori delle variabili come accadeva nel caso sbilanciato; infatti il primo gruppo presenta un tempo di attesa di gran lunga inferiore al secondo gruppo (27 giorni contro 139), una tariffa molto più economica (38 euro rispetto a 170), un’età media abbastanza simile (59 anni e 57 anni), e una distanza da percorrere maggiore per il gruppo 2 (13 km contro 10).

**MODEL BASED CLUSTERING:**

Viene applicato il model based clustering (rif. par. 2.5) soltanto sulle variabili quantitative poiché questo metodo non è applicabile con variabili categoriali. Inizialmente viene imposto un numero di gruppi pari a 2 per le motivazioni spiegate nei paragrafi precedenti e il dataset viene ristretto al sub training.

Il modello basato su BIC, ovvero una delle misure di performance del modello, restituisce un VVV in cui ci sono 5882 osservazioni appartenenti ad un gruppo e 10038 appartenenti all’ altro gruppo.

Pertanto viene stimato un modello molto complesso che corrisponde all’analisi discriminante quadratica in cui le linee di confine sono appunto quadratiche.

Vengono infatti stimati 29 parametri e il BIC corrispondente è 286212, valore che in senso assoluto non ha alcun significato ma va paragonato con altri modelli.

Prendendo in considerazione le medie, i gruppi presentano valori molto diversi fra di loro soprattutto per la variabile tempo di attesa e Prezzo\_prestazione. In generale il gruppo 1 è caratterizzato da medie più elevate per tutte le variabili e in particolare il tempo di attesa è 103 giorni contro 13 giorni mentre la tariffa è 125.18 euro di media contro 30.8 euro.

Cambiando la metrica di riferimento, anche secondo l'ICL il miglior modello con due gruppi è un VVV seguito da un VVI e un VVE ovvero tutti modelli con shape e volume variabili. Questo ci indica che con due gruppi è necessario mantenere una complessità elevata per quanto riguarda i vincoli sulla matrice di varianze/covarianze.

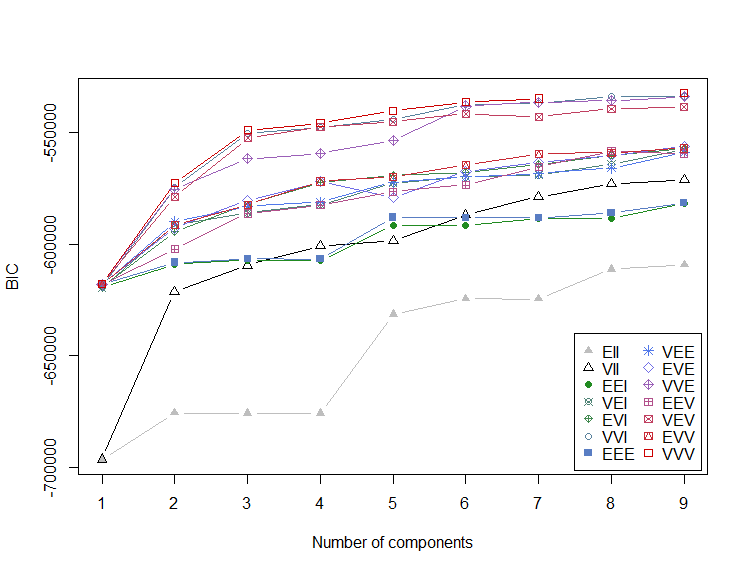


Figura 26. Grafico che mostra il BIC al variare del numero delle componenti

Provando invece a far variare il numero dei gruppi del modello si nota come il BIC (rif. fig. 26) migliora più il numero dei gruppi aumenta e questo testimonia come nella realtà non si individuino due gruppi ben definiti ma un unico gruppo molto condensato e per effettuare una miglior ripartizione è necessario utilizzare tanti piccoli gruppi; a conferma di ciò si vede come aumentando il numero di gruppi aumenta l'ICL come accadeva per il BIC.

Prendendo in considerazione l’incertezza delle u.s. ai gruppi si evince come ci siano molti punti con incertezza circa 0 e una percentuale minore di punti con un'incertezza maggiore. In particolare ci sono 160 osservazioni che superano il 45% di incertezza, il che vuol dire che c'è grande incertezza dato che con 2 gruppi l'incertezza massima che si può raggiungere è del 50%.

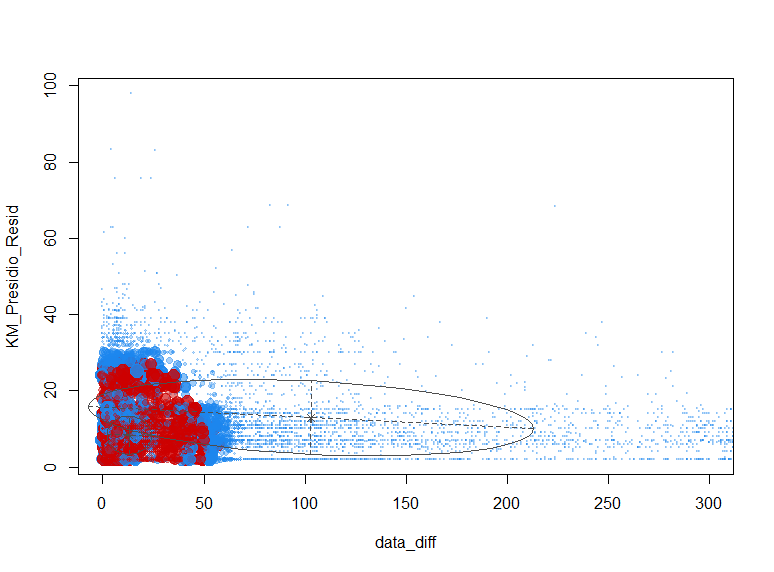


Figura 27. Scatterplot della classificazione con la relativa incertezza. Variabili prese in considerazione: tempo di attesa e distanza fra residenza e presidio

Selezionando distanza km\_presidio\_resid e data\_diff come variabili si notano bene i due gruppi che si basano su modelli di forma diversa, volume diverso e orientamento diverso. (rif. Figura 27. )

Si vede inoltre come tutti i punti con elevato tempo di attesa o elevata distanza fra presidio e residenza vengono classificati con poca incertezza mentre i punti con maggiore incertezza sono quelli il cui tempo di attesa si trova fra 0 e 50 e la distanza fra residenza e presidio assume valori tra 0 e 30.

Facendo lo zoom per capire vicino all'origine degli assi qual è la classificazione il grafico mostra un gran mescolamento e sovrapposizione di punti e ciò è indice del fatto che non si riesce a discriminare bene con solo due variabili.

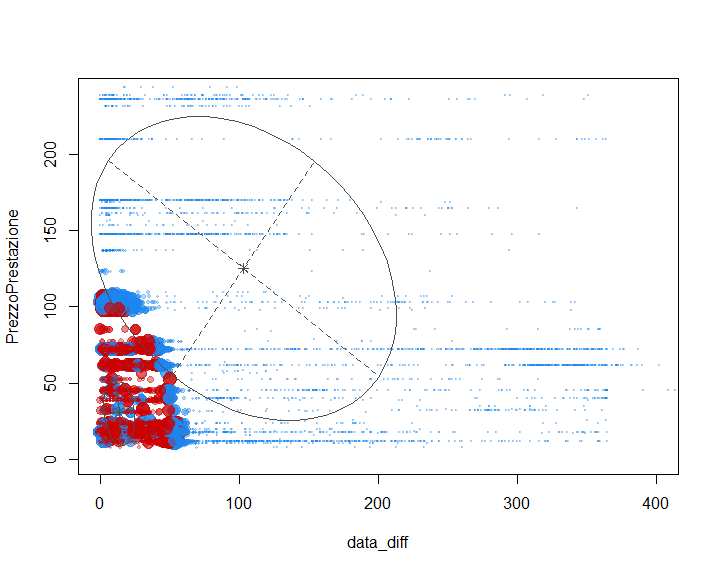


Figura 28. Scatterplot della classificazione con la relativa incertezza. Variabili prese in considerazione: tempo di attesa e tariffa prestazione

Dalla figura 28 si nota come le osservazioni con alto tempo di attesa o con una alta tariffa sono tutte classificate con poca incertezza e nel gruppo blu mentre la maggior parte delle unità statistiche con bassa tariffa e basso tempo di attesa sono classificate con più incertezza e prevalentemente nell’altro gruppo colorato di rosso.

Calcolando la distanza di Kullback Leibler il risultato ottenuto è pari a 111.25 ma anche questo valore assoluto non è significativo ma bisogna confrontarlo con un altro modello.

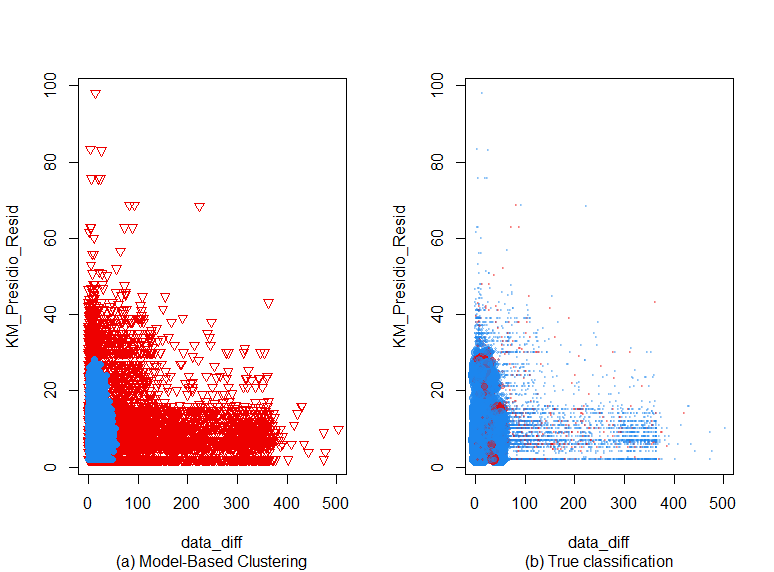


Figura 29. Scatterplot della classificazione tramite model-based clustering. Variabili prese in considerazione: tempo di attesa e distanza fra residenza e presidio

Figura 30. Scatterplot della vera classificazione con relativa incertezza. Variabili prese in considerazione: tempo di attesa e distanza fra residenza e presidio

La classificazione mostra come tutti i punti con basso tempo di attesa e bassa distanza fra residenza e presidio vengono classificati tutti in un gruppo, quello degli show mentre tutti le altre istanze vengono classificate come no show. In particolare la linea di confine sembra quando il tempo di attesa supera i 50 giorni e la distanza supera i 30km. Per quanto concerne invece la vera classificazione il grafico mostra ovviamente la grande prevalenza di punti azzurri che rappresentano gli show e come nella realtà la distinzione fra le due classi non è netta nelle due dimensioni e si evince una grande incertezza quando i punti si trovano vicino all'origine, ovvero con basso tempo di attesa e bassa distanza fra presidio e km.

Calcolando l’error rate, esso risulta essere pari al 36.25%, un risultato ancora non ottimo ma sicuramente migliore rispetto ai metodi non model-based e ciò testimonia inoltre come sia difficile classificare i dati con solo le variabili quantitative; inoltre la balanced accuracy fornisce come risultato 59.2, sempre in miglioramento rispetto ai modelli di clustering adottati fino ad ora.

**Classificazione con EDDA model:**

Si inizia lanciando un modello che prenda in considerazione tutti i modelli possibili. In automatico l’algoritmo sceglie il numero di gruppi pari a 2 come nella vera classificazione.

Viene modellato un Gaussian\_pk\_L\_C ovvero proporzioni variabili mentre volume, forma e orientamento fissi ovvero un modello molto semplice come EEE a differenza del clustering dove veniva modellato un VVV.

Estraendo le medie dei gruppi si nota come esse non siano così distanti come invece accadeva nel model based clustering.

Il primo gruppo presenta osservazioni con tempo di attesa maggiore, prezzo prestazione maggiore, età minore e distanza fra presidio e km maggiore.

Le proporzioni sono variabili e le osservazioni vengono allocate per il 6.6% al primo gruppo e per il 93.4% al secondo gruppo.

In seguito si prova a far girare l'EDDA model altre volte dato che, essendoci dietro un algoritmo E-M di estrazione casuale, ogni volta può restituire un risultato diverso. Invece questa volta il miglior modello EDDA è sempre lo stesso e questo può essere positivo perché significa che non c'è molta incertezza nei risultati.

Provando a costruire il miglior modello in base al BIC, esso è rappresentano da un Gaussian\_pk\_Lk\_D\_Ak\_D ovvero VVE (Volume variabile, shape variabile e orientation fisso).

Addentrandosi nel modello si estraggono le proporzioni che rimangono sempre uguali così come le medie sono identiche al modello EEE stimato come migliore modello tramite il criterio della CV.

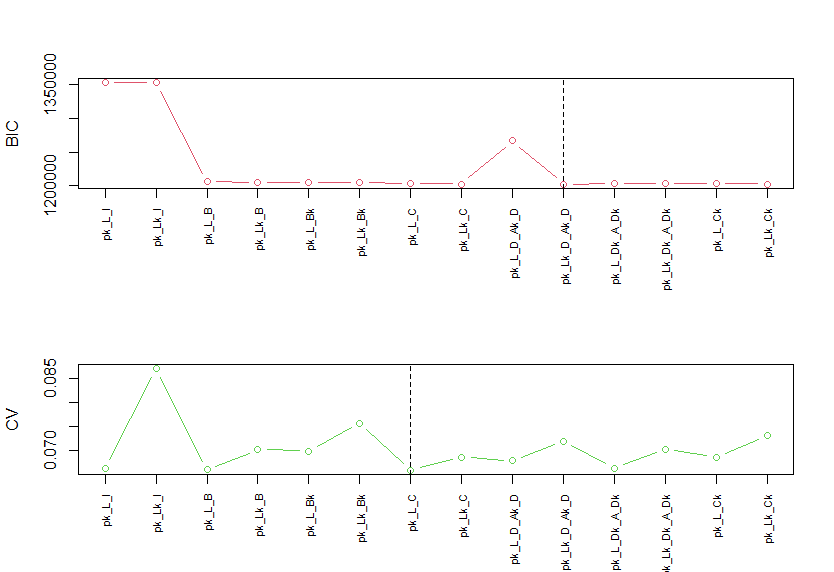


Figura 31. Andamento del BIC e del CV in base ai vari EDDA Models

Dal grafico 27 si nota come l'andamento delle due metriche sia molto volatile e se si aumenta la complessità, non necessariamente aumenta il BIC o la CV.

A questo punto inizia la fase di previsione separando come di consueto il dataset in training e test.

Dato che non ci sono molti iperparametri da massimizzare si decide comunque di non utilizzare un dataset di validation.

Facendo girare il modello di classificazione il miglior modello ottenuto in base al criterio della cross-validation è un Gaussian\_pk\_L\_C ovvero EEE che rappresenta un modello molto semplice.

Rappresentando su un grafico l’incertezza della classificazione, si vede come le istanze con maggior incertezza sono quelle con tempi di attesa bassi, (fra 0 e 18) e distanza fra presidio e residenza bassi (fra 0 e 32) come riportato anche dalla tabella n.28

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N.OBS | Data\_diff | Prezzo\_Prestazione | Età | KM\_Presidio\_Resid |
| 1453 | 0 | 99.23 | 19 | 32 |
| 2380 | 1 | 23.80 | 74 | 4 |
| 3791 | 2 | 71.79 | 73 | 10 |
| 4473 | 3 | 23.80 | 73 | 2 |
| 6045 | 4 | 350.91 | 37 | 19 |
| 6972 | 5 | 169.97 | 19 | 2 |
| 8383 | 18 | 16.80 | 56 | 4 |
| 9065 | 7 | 71.79 | 48 | 2 |

Tabella 32. Matrice che riporta le u.s. con la maggior incertezza e i valori delle relative variabili sul dataset sbilanciato

Tuttavia, la classificazione non è soddisfacente poiché il modello classifica tutte le nuove istanze come show portando quindi ad ottenere un error rate pari al 6.5% e quindi pari alla percentuale di osservazioni che nella realtà sono dei no show.

Il passo successivo è quindi quello di ripetere le analisi svolte su dataset bilanciato. Questa volta il miglior modello di classificazione è un Gaussian\_pk\_L\_Ck ovvero EVV, sicuramente più complesso del modello con la miglior performance ottenuto nel caso di dati bilanciati.

Le proporzioni di attribuzione delle unità statistiche sono molto vicine ovvero 0.5023 e 0.4977.

Considerando invece l’incertezza delle unità statistiche quelle con maggior incertezza sono riportate nella tabella n.

Queste istanze sono davvero molto incerte perché la misura di incertezza è oltre 0.499 sapendo che l'incertezza massima con due gruppi è 0.5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N.OBS | Data\_diff | Prezzo\_Prestazione | Età | KM\_Presidio\_Resid |
| 60 | 0 | 61.76 | 69 | 9.0 |
| 301 | 14 | 19.75 | 51 | 6.4 |
| 312 | 147 | 71.79 | 47 | 11.0 |
| 328 | 124 | 350.91 | 45 | 18.0 |
| 332 | 5 | 169.97 | 60 | 7.0 |
| 587 | 21 | 19.75 | 56 | 10.0 |
| 1271 | 8 | 19.75 | 95 | 8.0 |
| 1512 | 4 | 18.45 | 48 | 2.0 |

Tabella 33. Matrice che riporta le u.s. con la maggior incertezza e i valori delle relative variabili sul dataset bilanciato

Rispetto al caso sbilanciato si nota come mediamente le u.s. abbiano un'incertezza maggiore.

Provando a confrontare la classificazione del modello con la realtà tramite una misura di errore che risulta essere pari al 42.7% perchè i modelli EDDA si basano soltanto su variabili quantitative e la maggior parte delle variabili che sono discrete non si possono considerare. In realtà, essendo un'estrazione casuale, i risultati variano molto in base al campione e per questo motivo si prova ad iterare l'algoritmo molte volte prendendo una media degli errori. In conclusione il classificatore misclassifica mediamente il 40,5% delle istanze, un risultato comunque migliore dei metodi tradizionali di clustering.

**MDA MODELS PER LA CLASSIFICAZIONE:**

Come prima prova viene impostato che il numero di componenti di ogni gruppo sia pari a due in modo tale da iniziare a valutare la bontà del modello.

Ora con MDA i due gruppi in cui vengono divise le u.s. contengono al loro interno altri gruppi

Il modello ha stabilito che le due componenti all'interno di ogni gruppo siano tutte VVV, ovvero il modello più complesso in assoluto.

Df è pari a 58, ovvero sono stati stimati 58 parametri

BIC=-934820.7

Il classification error risulta essere 0.0742 misclassificando quindi solo il 7.42% delle osservazioni, ma non è una metrica di riferimento perché siamo in presenza di dati sbilanciati.

La matrice di confusione mostra come la maggior parte degli errori deriva dal fatto che 1692 osservazioni vengono classificate come show quando invece si tratta di no show

Lo step successivo è quello di provare a non fissare un numero G di gruppi all'interno di ogni componente per vedere in automatico l'algoritmo quale miglior numero di componenti restituisce.

Lasciando G libero di variare, esso di default cicla tutte le possibilità con G che va da 1 a 5.

In questo caso il miglior modello restituito è formato da 5 componenti all'interno di ogni gruppo, il che significa che all'aumentare delle componenti aumenta la precisione e la bontà del modello in termini di BIC.

Inoltre, si nota come nel gruppo dei no show il miglior modello è un VEV mostrando come all'aumentare del numero di componenti, si può diminuire la complessità e allentare leggermente le restrizioni passando a modelli con qualche parametro in meno.

Ora l'errore di classificazione è pari a 6.9%, diminuito rispetto al caso precedente.

Anche il BIC migliora (881376.7 ) e questa volta il numero dei parametri stimati aumenta a 136.

Il terzo caso è quando il numero delle componenti viene fatto variare da 1 a 10. Questa volta all'interno dei no show vengono scelte 10 componenti con un modello VEV e invece nelle osservazioni show vengono scelte 9 componenti stimando in totale 256 parametri.

Classification error = 0.0722

l'errore di classificazione è 7.22%, maggiore rispetto al caso con 5 componenti.

Per massimizzare il BIC l'algoritmo sceglie sempre un numero maggiori di componenti che però non portano alla minimizzazione dell'errore; infatti rispetto ai casi precedenti aumentano le persone classificate correttamente no show ma parallelamente aumenta il numero di utenti previsti come no show quando invece sono show.

Si passa quindi ad analizzare nel dettaglio il modello che porta al minore errore di classificazione sul training set.

All'interno della classe dei no show traggo le seguenti considerazioni sui 5 gruppi formati dall'algoritmo grazie alla matrice delle medie.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variabile** | **Gruppo1** | **Gruppo2** | **Gruppo3** | **Gruppo4** | **Gruppo5** |
| Data\_diff | 256.47728 | 64.38793 | 17.052806 | 97.543071 | 12.21419 |
| PrezzoPrestazione | 50.11314 | 176.27222 | 17.736674 | 25.809732 | 65.92028 |
| Età | 59.05643 | 51.44933 | 55.839790 | 68.189151 | 55.57861 |
| KM\_Presidio\_Resid | 9.84738 | 18.18815 | 8.774795 | 8.140026 | 10.82203 |

Tabella 34. Matrice delle medie per la classe dei no show tramite modello MDA con 5 componenti per gruppo

Il primo gruppo è quello con media maggiore dei tempi di attesa (gruppo di quelli con alti tempi di attesa) mentre il resto delle variabili assumono valori nella media.

Il secondo gruppo è quello con le prestazioni più costose con gli utenti che abitano a distanza maggiore dai presidi

Il terzo gruppo è caratterizzato dalle prestazioni con prezzo minore e anche con distanza ridotta fra presidio e residenza

Il quarto gruppo rappresenta le persone con un tempo di attesa mediamente elevato (97 giorni di attesa), con l'età media maggiore e con utenti che abitano vicino ai presidi in cui vengono svolte le prestazioni

Il quinto gruppo invece rappresenta le persone con minor tempo di attesa e invece con le altre variabili nella media.

La stessa analisi viene ripetuta all’interno della classe degli show

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variabile** | **Gruppo1** | **Gruppo2** | **Gruppo3** | **Gruppo4** | **Gruppo5** |
| Data\_diff | 10.549515 | 13.955359 | 7.067012 | 80.03170 | 192.78736 |
| PrezzoPrestazione | 61.031898 | 18.520107 | 119.250088 | 195.54624 | 41.33770 |
| Età | 57.879641 | 60.428193 | 57.353759 | 58.87166 | 66.36339 |
| KM\_Presidio\_Resid | 8.010322 | 7.754818 | 15.737679 | 15.78734 | 7.90996 |

Tabella 35. Matrice delle medie per la classe degli show tramite modello MDA con 5 componenti per gruppo

Il primo gruppo è caratterizzato da persone con basso tempo di attesa e le altre variabili nella media.

Il secondo gruppo rappresenta le persone che hanno usufruito delle prestazioni meno costose e che vivono vicino ai presidi ospedalieri.

Il terzo gruppo è formato da persone con bassissimo tempo di attesa, prestazioni con tariffe alte e mediamente più giovani degli altri gruppi e più lontani dai presidi.

Il quarto è formato dalle persone che hanno usufruito delle prestazioni più costose e quelli che abitano più lontano dai presidi.

Il quinto gruppo è formato dalle persone con più alto tempo di attesa e con l'età media più alta.

Si considera ora l’incertezza della classificazione all’interno dei due gruppi.

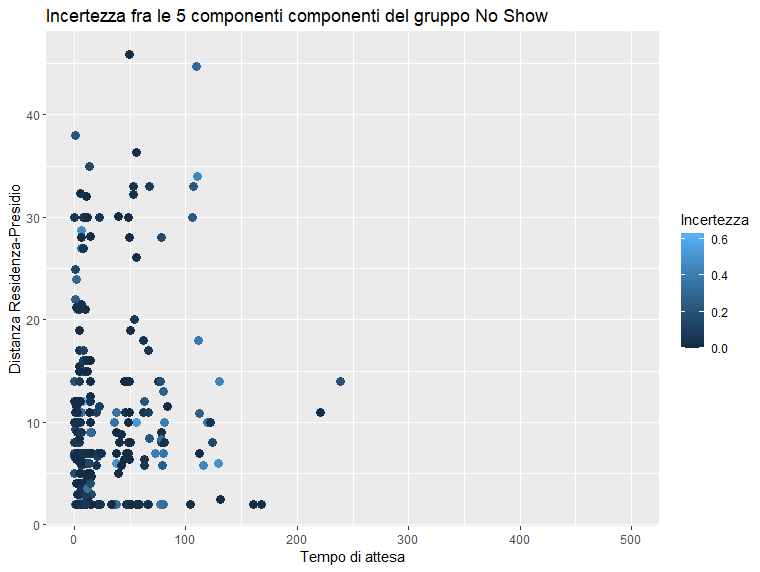


Figura 36. Scatterplot dell’incertezza all’interno della classe dei no show. Variabili prese in considerazione: tempo di attesa e distanza fra residenza e presidio.

Si nota come ci siano poche osservazioni con alta incertezza nel gruppo dei no show; infatti, soltanto lo 0.64% delle osservazioni ha un'incertezza maggiore di 0.5.

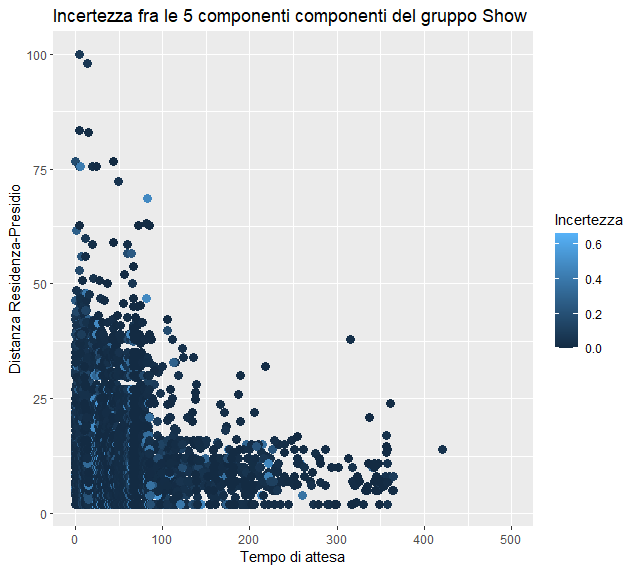


Figura 37. Scatterplot dell’incertezza all’interno della classe degli show. Variabili prese in considerazione: tempo di attesa e distanza fra residenza e presidio, dati sbilanciati.

Il grafico, avendo al suo interno tante osservazioni risulta meno leggibile ma si nota comunque come la maggior parte dei punti sia di colore scuro ovvero caratterizzata da minor incertezza; infatti solo lo 0.58% delle istanze ha un'incertezza maggiore di 0.5; complessivamente dunque nel gruppo degli show si evince un’incertezza leggermente minore.

**MDA SUI DATI BILANCIATI:**

Si parte sempre inizializzando l’algoritmo tramite la scelta di un numero di componenti all’interno di ogni gruppo pari a 2.

Anche in questo caso il modello ha stabilito che le due componenti all'interno di ogni gruppo siano tutte VVV, ovvero il modello con più parametri e meno restrizioni. I parametri stimati sono 58, il BIC è pari a -253147.3 mentre il Classification error fornisce il valore 0.3953, ossia vengono classificati in modo errato il 39.53% degli utenti, molti di più, come previsto, rispetto al caso sbilanciato.

|  |  |  |
| --- | --- | --- |
|  | **Predicted** | |
| **Class** | No show | Show |
| No show | 2017 | 1415 |
| Show | 1312 | 2155 |

Tabella 38. Confusion matrix ottenuta da modello MDA con due componenti per gruppo

La matrice di confusione mostra come vengono commessi molti errori sia quando gli show vengono classificati come no show sia quando i no show vengono classificati come show

La fase successiva è quella di non fissare un numero G di gruppi all'interno di ogni componente per vedere in automatico l'algoritmo quale miglior numero di componenti all’interno di ogni gruppo restituisce.

Lasciando G libero di variare, esso di default prova tutte le possibilità con G che va da 1 a 5

In questo caso il miglior modello restituito è formato da 5 componenti all'interno di ogni gruppo fornendo la stessa considerazione e interpretazione del caso di dati sbilanciati.

Inoltre si nota come negli show il miglior modello è un VVI mostrando come all'aumentare del numero di componenti, si può diminuire la complessità e allentare leggermente le restrizioni sulla matrice di var/cov passando a modelli con qualche parametro in meno.

|  |  |  |
| --- | --- | --- |
|  | **Predicted** | |
| **Class** | No show | Show |
| No show | 2040 | 1392 |
| Show | 1261 | 2206 |

Tabella 39. Confusion matrix ottenuta da modello MDA con cinque componenti per gruppo

Ora l'errore di classificazione è 38.45%, minore rispetto al modello con G=2.

Anche il BIC migliora (241124) e questa volta il numero dei parametri stimati aumenta a 118.

L’ultimo step è sempre quello di far variare G da 1 a 10:

Per massimizzare il BIC l'algoritmo sceglie sempre un numero maggiori di componenti che questa volta conducono anche alla minimizzazione dell'errore, a differenza dei dati unbalanced.

Questa volta all'interno dei no show vengono scelte 10 componenti con un modello VEV e invece nel gruppo degli show vengono scelte 9 componenti con un modello VVI stimando in totale 202 parametri.

Il BIC scende ancora arrivando a -238515.6.

|  |  |  |
| --- | --- | --- |
|  | **Predicted** | |
| **Class** | No show | Show |
| No show | 1906 | 1526 |
| Show | 1065 | 2402 |

Tabella 40. Confusion matrix ottenuta da modello MDA con nove/dieci componenti per gruppo

Classification error = 0.3756

L'errore di classificazione è 37.56 e migliora ancora rispetto al caso in cui ci limitavamo a 5 componenti per gruppo.

Rispetto al caso con 5 gruppi si nota come diminuiscono il numero di persone classificate erroneamente come no show quando in realtà sono show mentre invece aumenta il numero di errori nella classificazione di persone che vengono previste show quando invece sono no show.

Si passa ora ad analizzare il modello con 5 componenti perché è di più chiara analisi e interpretazione e produce risultati molto vicini in termini di performance al modello con 9/10 componenti per ogni classe.

All'interno del gruppo nei no show traggo le si possono analizzare le componenti grazie alla matrice delle medie:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variabile** | **Gruppo1** | **Gruppo2** | **Gruppo3** | **Gruppo4** | **Gruppo5** |
| Data\_diff | 16.570626 | 77.774049 | 170.174811 | 12.00751 | 53.67561 |
| PrezzoPrestazione | 17.779930 | 173.917662 | 29.586531 | 55.53568 | 176.99802 |
| Età | 56.166934 | 55.070396 | 63.810169 | 54.16202 | 53.12867 |
| KM\_Presidio\_Resid | 8.210348 | 9.451479 | 8.897289 | 15.35861 | 23.30603 |

Tabella 41. Matrice delle medie per la classe dei no show tramite modello MDA cinque componenti per gruppo

Il primo gruppo è quello con le tariffe per le prestazioni minori di tutte e con all’interno utenti che abitano vicini ai presidi

Il secondo gruppo presenta prestazioni molto costose e con le altre caratteristiche nella media

Il terzo gruppo è caratterizzato dalle persone con maggior tempo di attesa e con l'età media più alta

Il quarto gruppo rappresenta invece le persone con tempo di attesa minore e con un'età media bassa

Il quinto gruppo è invece formato da persone che hanno usufruito delle prestazioni più costose, con l'età media più bassa e la distanza fra presidio e residenza maggiore

Stessa analisi ripetuta sul gruppo degli show:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variabile** | **Gruppo1** | **Gruppo2** | **Gruppo3** | **Gruppo4** | **Gruppo5** |
| Data\_diff | 75.45338 | 7.302047 | 17.61910 | 13.402649 | 175.679292 |
| PrezzoPrestazione | 196.80166 | 90.492975 | 23.34312 | 18.027544 | 47.008795 |
| Età | 56.62373 | 58.883519 | 51.44804 | 62.242705 | 68.018857 |
| KM\_Presidio\_Resid | 16.51647 | 10.367595 | 12.15974 | 7.716309 | 7.933079 |

Tabella 42. Matrice delle medie per la classe degli show tramite modello MDA cinque componenti per gruppo

Il primo gruppo è caratterizzato dalle persone che hanno usufruito delle prestazioni più costose e con le altre caratteristiche nella media

Il secondo gruppo rappresenta le persone con tempi di attesa minori e tariffe delle prestazioni mediamente elevate

Il terzo gruppo è formato dalle persone più giovani

Il quarto gruppo è caratterizzato da persone che hanno usufruito delle prestazioni più economiche e sono anche quelli che abitano più vicino ai presidi

Il quinto gruppo è formato dalle persone con più alto tempo di attesa, più anziani e mediamente vicini alle strutture ospedaliere

Analisi dell’incertezza distinta per le due classi:

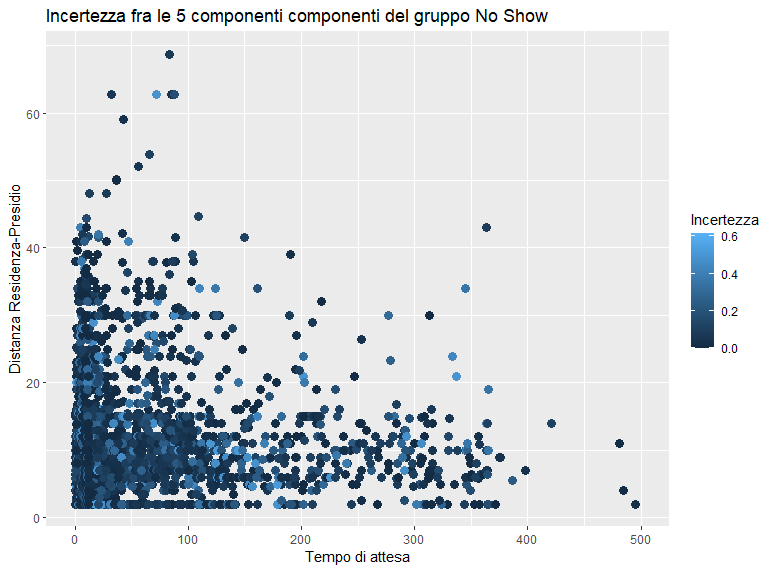


Figura 43. Scatterplot dell’incertezza all’interno della classe dei no show. Variabili prese in considerazione: tempo di attesa e distanza fra residenza e presidio, dati bilanciati.

L'1.39% delle istanze della classe no show ha un'incertezza maggiore di 0.5; si nota come ci siano poche osservazioni con bassa incertezza anche se sembra un numero maggiore rispetto al caso sbilanciato.

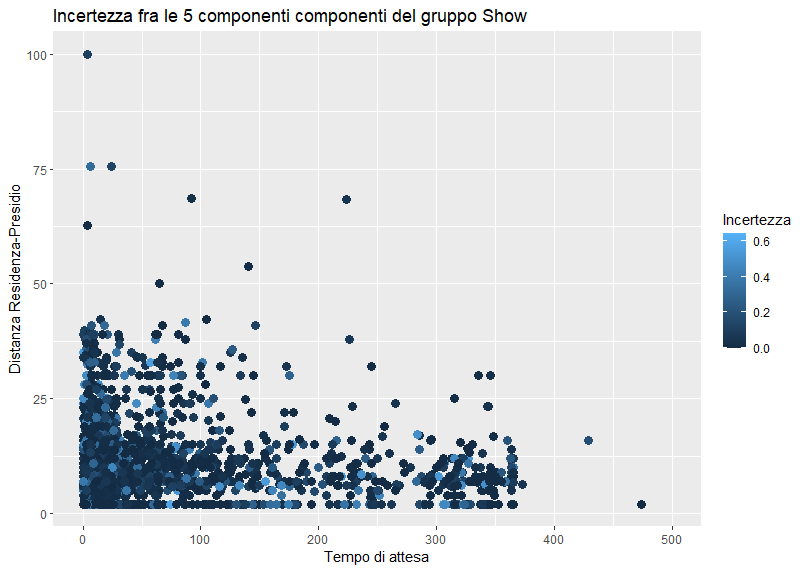


Figura 44. Scatterplot dell’incertezza all’interno della classe degli show. Variabili prese in considerazione: tempo di attesa e distanza fra residenza e presidio, dati bilanciati.

L'1.15% delle osservazioni della classe show ha un'incertezza maggiore di 0.5; pertanto la classe degli show è determinata con minore incertezza.

Per ultimo si cerca di stabilire il miglior modello e il miglior numero di componenti all’interno di ogni gruppo tramite il criterio della V-Fold Cross Validation.

In questo caso si è scelto di imporre un numero massimo G di componenti pari a 10 mentre e di dividere casualmente le unità statistiche in 37 blocchi.

Si stabilisce il parametro B dato dal numero delle osservazioni diviso il numero di blocchi V.

Poi si costruisce e si applica una funzione che utilizza il metodo della cross validation, ovvero stima il modello dopo aver tolto dai dati ogni volta un numero di records pari a B, in questo caso toglie ogni volta 218 record e poi stima il modello MDA sul dataset con n-B unità statistiche.

Tramite il criterio della v-fold cross validation si evince che il miglior numero delle componenti all'interno di ogni gruppo è 3

**CLUSTERING CON VARIABILI MISTE**

Per effettuare questo tipo di clustering (rif. par 2.6) e la successiva classificazione, ora è possibile utilizzare tutte le tipologie di variabili, non solo quelle quantitative.

A questo punto si scelgono le variabili più importanti e significative da inserire all’interno della matrice di distanza costruita tramite la metrica di Gower.

Poi si applica il metodo del clustering gerarchico con legame completo fornendo in input la matrice delle distanze calcolate al punto precedente.

Viene fissato il numero dei gruppi k pari a 2, per poter effettuare confronti e classificazione con la realtà.

In questo caso calcolando l’errore di classificazione si evince come il metodo misclassifichi il 25% delle osservazioni.

In particolare, l’errore più importante è legato alla classificazione di unità come show quando in realtà sono no show producendo una Specificity del 6.2% e una Sensitivity del 93.8%.

Provando a ripetere le medesime analisi sul dataset bilanciato si ottiene un errore di classificazione pari a 47.88% delle istanze mostrando come il clustering gerarchico non produca ottimi risultati.

Si prova ad utilizzare gli altri linkage. Con quello singolo, l'errore di classificazione è poco più alto ma il vero problema è che questo metodo di clustering considera 8074 osservazioni su 8076 come appartenenti al gruppo 1 non riuscendo quindi a distinguere fra le due classi quando invece le vere etichette sono equidistribuite.

Anche con il legame medio e il metodo di ward si ottengono dei risultati simili con un accuracy che varia attorno al 50%.

Provando a cambiare metodo di clustering e passando al k-means i risultati rimangono sempre non ottimi con un’accuracy che peggiora leggermente attestandosi attorno al 46%. Come mostrato dalla tabella 45. Infatti, ci sono 2812 u.s. che vengono classificate show pur essendo no show e 1558 u.s. previste come no show quando invece sono show.

|  |  |  |
| --- | --- | --- |
|  | **Predicted** | |
| **Class** | No show | Show |
| No show | 2480 | 2812 |
| Show | 1558 | 1226 |

Tabella 45. Confusion matrix con k-means, variabili miste e dati bilanciati

**Mixture of experts model (MEM):**

Per la stima di questo modello è stato necessario raggruppare le osservazioni in base alla data della prestazione, ottenendo 320 records ai quali sono state tolte 12 osservazioni anomale relative alle date con percentuale di no show superiore al 25%.

Per ogni data pertanto sono state calcolate il tempo di attesa medio, la distanza media fra casa e ospedale, la tariffa media della prestazione, e l’età media oltre al numero totale di show e di no show.

Per quanto riguarda invece l’implementazione dell’algoritmo si è scelto di inizializzarlo ponendo un numero di gruppi pari a 2 e un expert network che modellizza un GLM binomiale poiché siamo nel caso di variabile risposta dicotomica con numero totale noto per ogni data.

Il metodo restituisce una partizione del dataset in due gruppi formati da 87 e 221 (size) osservazioni avendo raggiunto la convergenza dopo 75 iterazioni. Il numero totale di parametri stimati è 11 mentre il valore assunto dall’AIC e dal BIC è rispettivamente 1563.337 e 1604.368. La prima componente/gruppo quindi ha un peso (prior) del 35.7% mentre quello della seconda componente è 64.3%. Ci sono 304 (Post>0) osservazioni del primo gruppo che hanno una probabilità positiva di appartenenza al gruppo stesso ma soltanto il 28.6% (ratio) vengono effettivamente classificate al gruppo 1. La “ratio” ha a che fare con la sicurezza/incertezza con cui alloco le unità statistiche ai gruppi. Dato che il primo gruppo ha una ratio bassa, l’incertezza della classificazione è elevata. Invece, il numero di osservazioni con una probabilità positiva di appartenenza al gruppo 2 è pari a 301 e di queste il 73.4% viene effettivamente assegnato al gruppo 2. Pertanto, al contrario di quanto accadeva nel primo gruppo, nel secondo c’è una buona sicurezza nell’assegnazione delle unità statistiche. Tutte queste considerazioni sono ben riassunte nella tabella 46.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prior** | **Size** | **Post>0** | **Ratio** |
| Comp.1 | 0.357 | 87 | 304 | 0.286 |
| Comp.2 | 0.643 | 221 | 301 | 0.734 |

Tabella 46. Output del summary del modello stimato

Per quanto concerne la significatività delle covariate, questo aspetto è suddiviso per ogni componente.

Per il gruppo 1 le variabili significative sono il tempo di attesa, la distanza in km e la tariffa della prestazione, elencate in ordine di importanza e peso come raffigurato nella tabella 47.

Per il gruppo 2 le variabili significative sono il tempo di attesa e la distanza in km, sempre elencate in ordine di rilevanza, mentre la tariffa della prestazione non è più significativa come variabile, come si evince dalla tabella 48.

Entrambe le variabili hanno un’influenza fortemente negativa sul numero di “show” sia per la componente 1 che per la componente 2 come dimostra il segno negativo davanti alla stima del coefficiente di regressione.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Componente 1** | **Estimate** | **Std. Error** | **z value** | **Pr(>|z|)** |
| (Intercept) | 4.2290375 | 1.1056833 | 3.8248 | 0.0001309 \*\*\* |
| Media\_distanza\_km | -0.1483669 | 0.0525516 | -2.8233 | 0.0047538 \*\* |
| Media\_età | -0.0046254 | 0.0159194 | - 0.2905 | 0.7713965 |
| media\_tempo\_attesa | -0.0146770 | 0.0025808 | - 5.6871 | 1.292e-08 \*\*\* |
| Media\_tariffa\_prestaz | 0.009264 | 0.0040168 | 2.3064 | 0.0210893 \* |

Tabella 47. Output della significatività dei parametri per la componente 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Componente 2** | **Estimate** | **Std. Error** | **z value** | **Pr(>|z|)** |
| (Intercept) | 3.7159373 | 1.2266577 | 3.0293 | 0.0024511 \*\* |
| Media\_distanza\_km | -0.1341967 | 0.0422971 | -3.1727 | 0.0015102 \*\* |
| Media\_età | 0.0266766 | 0.0172953 | 1.5424 | 0.1229723 |
| media\_tempo\_attesa | -0.0114943 | 0.0034247 | -3.3563 | 0.0007899 \*\*\* |
| Media\_tariffa\_prestaz | -0.0023705 | 0.0023871 | -0.9931 | 0.3206834 |

Tabella 48. Output della significatività dei parametri per la componente 2

Analizzando l’incertezza si nota come le date con meno sicurezza di classificazione sono le seguenti: 2019-08-09, 2019-09-11, 2019-09-13, 2019-09-28, 2019-11-09, 2022-01-13, 2022-04-05.

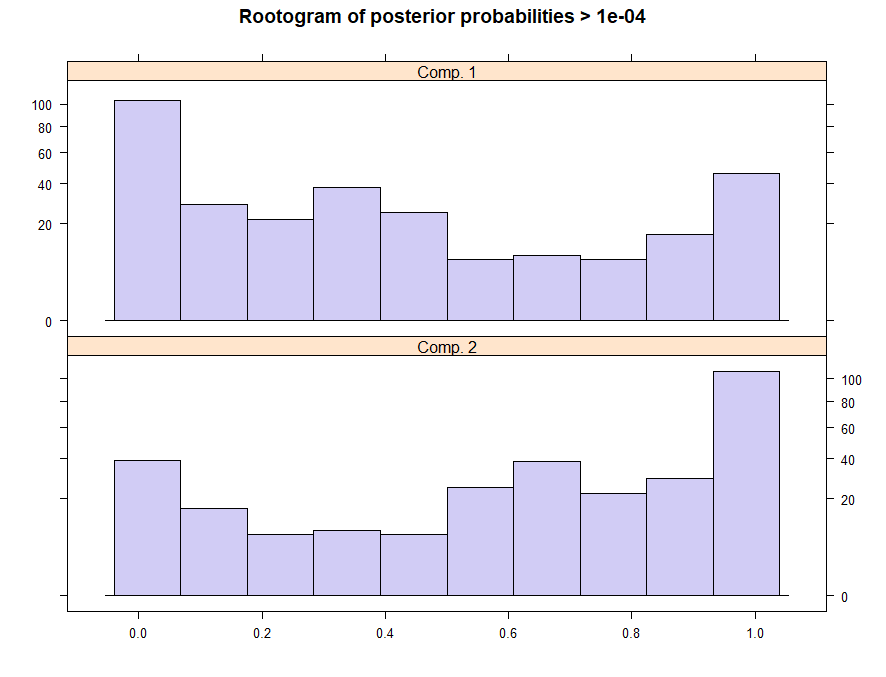


Figura 49. Rootogram delle probabilità a posteriori

Dalla figura 49, si trae che ci sono molte osservazioni con probabilità a posteriori alta ma anche molte con probabilità intermedia di appartenenza alla componente j-esima e quest’ultima situazione denota un’elevata incertezza per un discreto numero di unità statistiche.

Come mostrato dal grafico 50, il modello MEM distingue le osservazioni in due gruppi e per ognuno di essi è stato calcolato un modello di regressione. Ad esempio, per il gruppo 1 la retta di regressione risulta essere caratterizzata da una bassa pendenza significando che il tempo medio di attesa influisce ma in maniera lieve sulla percentuale di no show; al contrario, per il gruppo 2, un aumento del tempo medio di attesa produce una forte variazione della percentuale di no show e ciò è ben rappresentato dalla retta azzurrina contraddistinta da un’elevata pendenza.

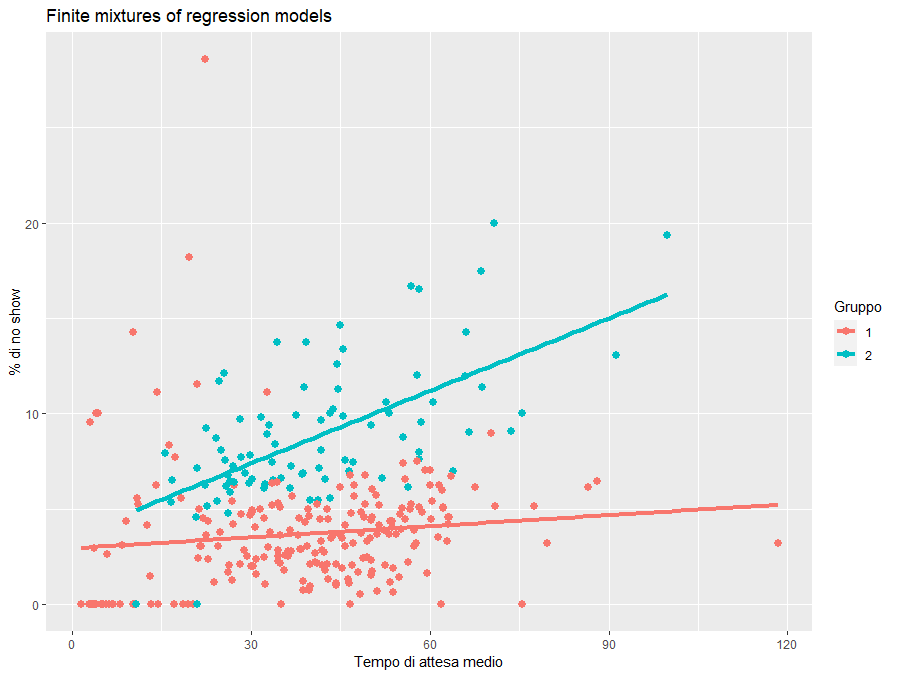


Figura 50. Scatterplot della regressione tramite MEM considerando come covariata il tempo di attesa medio

Poiché l’algoritmo EM si basa sull’estrazione casuale di valori iniziali è conveniente ripetere la stima del modello più volte e tra queste scegliere il modello migliore in termini di BIC.

Si sceglie di iterare l’algoritmo per 35 volte, ritenuto un numero sufficiente e non così grande in modo tale da ottenere risultati in tempi brevi. Il BIC nelle 35 iterazioni rimane pressoché costante tranne per l'iterazione 31 dove aumenta fino a 1676.042.

Per il resto delle iterazioni invece le variazioni riguardano soltanto la terza cifra decimale del BIC e questo è un fattore positivo perché significa che non c'è grossa variabilità nella stima del modello.

Come ultima analisi si passa alla stima del modello MEM scegliendo un numero di gruppi che varia da 1 a 4 e verificando quale numero di gruppi conduce al miglior modello in termini di BIC o ICL.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Iter** | **Converged** | **K** | **K0** | **logLik** | **AIC** | **BIC** | **ICL** |
| 2 | TRUE | 1 | 1 | -862.8894 | 1735.779 | 1754.429 | 1754.429 |
| 32 | TRUE | 2 | 2 | -770.6679 | 1563.336 | 1604.367 | 1714.478 |
| 52 | TRUE | 3 | 3 | -761.3282 | 1556.656 | 1620.068 | 1816.434 |
| 108 | TRUE | 4 | 4 | -755.2558 | 1556.512 | 1642.304 | 1997.090 |

Tabella 51. Summary della funzione stepFlexmix facendo variare il numero dei gruppi da 1 a 4

Come riportato dai valori della tabella 51, l'ICL e il BIC assumono valori più bassi e quindi migliori con 2 gruppi. Questo risultato è confortante sapendo che anche nella realtà le osservazioni sono distinte in due gruppi. Inoltre, tutti i modelli raggiungono convergenza nella stima di massima verosimiglianza e il numero di iterazioni necessarie aumenta notevolmente all’aumentare del numero di componenti.

Si evidenza inoltre come l’AIC, a differenza del BIC e dell’ICL, prediliga un numero di gruppi maggiore poiché il miglior risultato si ottiene con K=4 e subito dopo con K=3. Come sottolineato anche nei capitoli precedenti questo è un difetto della metrica AIC poiché non penalizza per la complessità del modello.

1. **CONCLUSIONE:**

Il fenomeno dei no show è un problema rilevante e consistente all’interno degli ospedali dell’ASST BRIANZA.

Infatti, su 30617 pazienti, ben 1875 non si sono presentati all’appuntamento, dato che corrisponde al 6.12% del totale delle prenotazioni e costituisce una perdita di 148mila euro in termini di mancato rimborso del ticket, relativa alla sola diagnostica per immagini e nei mesi presi in considerazione.

La media dei soldi persi per ogni no show è 79.17. Tenendo conto che sono state prese in considerazione soltanto 30617 su 130000 e mantenendo una percentuale di no show costante al 6.12%, si stima che la perdita totale si aggiri fra 560mila e 640 mila euro soltanto per i 13 mesi presi in considerazione.

Come anticipato nel capitolo di introduzione, il fenomeno risulta essere complesso da spiegare e da prevedere. Infatti le varie tecniche statistiche di classificazione portano a risultati buoni ma non perfetti come invece può accadere con dataset più scolastici dove si riescono a individuare relazioni molto forti tra variabili.

|  |  |  |
| --- | --- | --- |
| Metodo | Dataset originale | Dataset bilanciato |
| Clustering gerarchico | Balanced Accuracy: 49.9% | Accuracy= 50.6% |
| K means clustering | Balanced Accuracy: 54.9% | Accuracy= 51.1% |
| Regressione logistica | Balanced Accuracy: 78% | Accuracy= 92.75% |
| Model based clustering | Balanced Accuracy: 59.2% | Accuracy= 62.44% |
| Clustering con variabili miste | Balanced Accuracy: 50% | Accuracy= 52.12% |

Tabella 45. Tabella riassuntiva della classificazione per tutte le tecniche adottate e le relative performance

Come mostrato dalla tabella 45 il miglior metodo di classificazione, sia con i dati originali che con i dati bilanciati, è la regressione logistica che infatti per sua caratteristica ben si addice alla previsione di fenomeni dicotomici con variabili quantitative ma anche qualitative.

Le altre tecniche di classificazione producono performance peggiori perché ammettono soltanto variabili numeriche, limitando così il contributo delle numerose variabili categoriche che risultano significative.

Inoltre, all’interno di queste tecniche che utilizzano soltanto variabili numeriche, il model based clustering si distingue comunque per risultati migliori rispetto alle tecniche di clustering tradizionali che si basano sulla distanza fra le osservazioni.

Pertanto, sulla base delle motivazioni appena citate, i risultati ottenuti erano in parte attesi e previsti anche alla luce del fatto che tecniche come la regressione logistica e il model based clustering sfruttano metodologie più avanzate che solitamente conducono a metriche migliori soprattutto analizzando dati complessi.

Per quanto riguarda invece le cause che provocano questo fenomeno, come preannunciato anche nel paragrafo introduttivo e come mostrato nel capitolo 3.2, esse sono molteplici e di varia natura; tra queste, le più importanti ed esplicative sono il tempo di attesa, il possesso o meno dell’esenzione dal ticket, la cittadinanza italiana o estera, la distanza da percorrere per raggiungere l’ospedale, l’orario dell’appuntamento e il presidio dove vengono erogate poiché Seregno e Vimercate presentano una percentuale nettamente inferiore rispetto agli altri due ospedali.

Inoltre, come mostrato dal paragrafo 3.3 le variabili che influenzano il fenomeno in oggetto non hanno un andamento monotono, ovvero sempre crescente/decrescente. Per alcuni intervalli la percentuale di no show aumenta, per altri diminuisce; pertanto, l’influenza delle varie cause non è assoluta ma è relativa in base a ciascun range che si prende in esame.

Script R:

#Importazione del dataset

dati\_filtrati\_agg<- Dati\_no\_Show\_Totali\_filtrato

str(dati\_filtrati\_agg)

summary(dati\_filtrati\_agg)

dati\_filtrati\_agg<- dati\_filtrati\_agg[, -c(1,2,5,6,7,8,16,19,23,27,28,32,35,36)]

#Analisi esplorativa

table(dati\_filtrati\_agg$Trimestre)/nrow(dati\_filtrati\_agg)

table(dati\_filtrati\_agg$MESE)

table(dati\_filtrati\_agg$Ora)

table(dati\_filtrati\_agg$Sez\_giornata)

table(dati\_filtrati\_agg$Sez\_giornata)/nrow(dati\_filtrati\_agg)

table(dati\_filtrati\_agg$`Ricetta Digitale`)

table(dati\_filtrati\_agg$`Ricetta Digitale`)/nrow(dati\_filtrati\_agg)

summary(dati\_filtrati\_agg$PrezzoPrestazione)

table(dati\_filtrati\_agg$Esente)

table(dati\_filtrati\_agg$Esente)/nrow(dati\_filtrati\_agg)

table(dati\_filtrati\_agg$Sesso)

table(dati\_filtrati\_agg$Sesso)/nrow(dati\_filtrati\_agg)

table(dati\_filtrati\_agg$Presidio); table(dati\_filtrati\_agg$Presidio)/nrow(dati\_filtrati\_agg)

table(dati\_filtrati\_agg$Anno), table(dati\_filtrati\_agg$Anno)/nrow(dati\_filtrati\_agg)

table(dati\_filtrati\_agg$Descrizione)

table(dati\_filtrati\_agg$Descrizione)/nrow(dati\_filtrati\_agg)

table(dati\_filtrati\_agg$Tipo\_di\_esenzione);

table(dati\_filtrati\_agg$Straniero; table(dati\_filtrati\_agg$Straniero)/nrow(dati\_filtrati\_agg)

summary(dati\_filtrati\_agg$data\_diff); summary(dati\_filtrati\_agg$KM\_Presidio\_Resid); summary(dati\_filtrati\_agg$Età)

par(mfrow=c(2,2))

boxplot(dati\_filtrati\_agg$Età)

boxplot(dati\_filtrati\_agg$KM\_Presidio\_Resid); boxplot(dati\_filtrati\_agg$data\_diff); boxplot(dati\_filtrati\_agg$PrezzoPrestazione)

par(mfrow=c(2,2))

hist(dati\_filtrati\_agg$Età, main="Istogramma Età", xlab="Età", col="Red")

hist(dati\_filtrati\_agg$KM\_Presidio\_Resid, main="Istogramma Distanza Tragitto Stradale", xlab="KM Tra il domicilio e il presidio osp.", col="Blue")

hist(dati\_filtrati\_agg$data\_diff, main="Istogramma tempo di attesa", xlab="Tempo di attesa", col="Green")

hist(dati\_filtrati\_agg$PrezzoPrestazione, main="Istogramma Tariffa", xlab="Tariffa prestazione", col="yellow")

par(mfrow=c(1,1))

hist(log(dati\_filtrati\_agg$KM\_Presidio\_Resid))

hist(log(dati\_filtrati\_agg$data\_diff))

plot(factor(dati\_filtrati\_agg$`NO SHOW`),data\_diff,xlab="Y",ylab="Tempo di Attesa", ylim=c(0,120))

plot(factor(dati\_filtrati\_agg$`NO SHOW`),Età,xlab="Y",ylab="Età")

plot(factor(dati\_filtrati\_agg$`NO SHOW`),KM\_Presidio\_Resid,xlab="Y",ylab="Distanza Residenza\_Presidio", ylim=c(0,50))

plot(factor(dati\_filtrati\_agg$`NO SHOW`),PrezzoPrestazione,xlab="Y",ylab="Tariffa\_prestazione", ylim=c(0,200))

length(boxplot.stats(dati\_filtrati\_agg$Età)$out)/nrow(dati\_filtrati\_agg)\*100 #0.62%

length(boxplot.stats(dati\_filtrati\_agg$PrezzoPrestazione)$out)/nrow(dati\_filtrati\_agg)\*100 #13.28%

length(boxplot.stats(dati\_filtrati\_agg$KM\_Presidio\_Resid)$out)/nrow(dati\_filtrati\_agg)\*100 #8.031%

length(boxplot.stats(dati\_filtrati\_agg$data\_diff)$out)/nrow(dati\_filtrati\_agg)\*100 #14.92 %

dati\_filtrati\_agg$Trimestre<- as.factor(dati\_filtrati\_agg$Trimestre)

dati\_filtrati\_agg$Ora<- as.numeric(dati\_filtrati\_agg$Ora)

dati\_filtrati\_agg$Sez\_giornata<- as.factor(dati\_filtrati\_agg$Sez\_giornata)

dati\_filtrati\_agg$Data\_app...10<-as.Date(dati\_filtrati\_agg$Data\_app...10)

dati\_filtrati\_agg$Data\_pren...9<-as.Date(dati\_filtrati\_agg$Data\_pren...9)

dati\_filtrati\_agg$`Ricetta Digitale`<- as.factor(dati\_filtrati\_agg$`Ricetta Digitale`)

dati\_filtrati\_agg$PrezzoPrestazione<- as.numeric(dati\_filtrati\_agg$PrezzoPrestazione)

dati\_filtrati\_agg$Esente<- as.factor(dati\_filtrati\_agg$Esente)

dati\_filtrati\_agg$Sesso<- as.factor(dati\_filtrati\_agg$Sesso)

dati\_filtrati\_agg$Presidio<- as.factor(dati\_filtrati\_agg$Presidio)

dati\_filtrati\_agg$`NO SHOW`<- as.factor(dati\_filtrati\_agg$`NO SHOW`)

dati\_filtrati\_agg$MESE<- as.factor(dati\_filtrati\_agg$MESE)

dati\_filtrati\_agg$Anno<- as.factor(dati\_filtrati\_agg$Anno)

dati\_filtrati\_agg$Descrizione<- as.factor(dati\_filtrati\_agg$Descrizione)

dati\_filtrati\_agg$Tipo\_di\_esenzione<- as.factor(dati\_filtrati\_agg$Tipo\_di\_esenzione)

dati\_filtrati\_agg$Fascia\_età<- as.factor(dati\_filtrati\_agg$Fascia\_età)

dati\_filtrati\_agg$Range\_di\_attesa<- as.factor(dati\_filtrati\_agg$Range\_di\_attesa)

dati\_filtrati\_agg$Fascia\_Prezzo<- as.factor(dati\_filtrati\_agg$Fascia\_Prezzo)

dati\_filtrati\_agg$Fascia\_dist\_km<- as.factor(dati\_filtrati\_agg$Fascia\_dist\_km)

dati\_filtrati\_agg$Straniero<- as.factor(dati\_filtrati\_agg$Straniero)

table(dati\_filtrati\_agg$`NO SHOW`)

table(dati\_filtrati\_agg$`NO SHOW`)/nrow(dati\_filtrati\_agg)

anyNA(dati\_filtrati\_agg)

p\_filt<-dati\_filtrati\_agg%>%

group\_by(dati\_filtrati\_agg$Data\_app...10)%>%

count(dati\_filtrati\_agg$`NO SHOW`)

colnames(p\_filt)<- c("data\_app", "Type", "Numero")

p\_filt<- p\_filt%>%

spread(key = Type, value = Numero)

colnames(p\_filt)<- c("data\_app", "NO\_SHOW", "SHOW")

which(is.na(p\_filt$NO\_SHOW))

p\_filt$NO\_SHOW[6]<- 0; p\_filt$NO\_SHOW[13]<- 0; p\_filt$NO\_SHOW[21]<- 0; p\_filt$NO\_SHOW[26]<- 0; ;p\_filt$NO\_SHOW[32]<- 0; p\_filt$NO\_SHOW[39]<- 0 ;p\_filt$NO\_SHOW[45]<- 0; p\_filt$NO\_SHOW[51]<- 0; p\_filt$NO\_SHOW[57]<- 0; p\_filt$NO\_SHOW[63]<- 0; p\_filt$NO\_SHOW[69]<- 0; p\_filt$NO\_SHOW[70]<- 0

p\_filt$NO\_SHOW[82]<- 0; p\_filt$NO\_SHOW[93]<- 0; p\_filt$NO\_SHOW[99]<- 0; p\_filt$NO\_SHOW[105]<- 0

p\_filt$NO\_SHOW[115]<- 0; p\_filt$NO\_SHOW[121]<- 0; p\_filt$NO\_SHOW[125]<- 0; p\_filt$NO\_SHOW[127]<- 0; p\_filt$NO\_SHOW[133]<- 0; p\_filt$NO\_SHOW[139]<- 0; p\_filt$NO\_SHOW[145]<- 0; p\_filt$NO\_SHOW[152]<- 0; p\_filt$NO\_SHOW[158]<- 0; p\_filt$NO\_SHOW[175]<- 0; p\_filt$NO\_SHOW[177]<- 0; p\_filt$NO\_SHOW[223]<- 0; p\_filt$NO\_SHOW[269]<- 0; p\_filt$NO\_SHOW[287]<- 0; p\_filt$NO\_SHOW[299]<- 0; p\_filt$NO\_SHOW[305]<- 0

p\_filt$NO\_SHOW+p\_filt$SHOW

p\_filt$somma\_app<-p\_filt$NO\_SHOW+p\_filt$SHOW

p\_filt$perc\_no\_show<- (p\_filt$NO\_SHOW)/p\_filt$somma\_app\*100

g3\_filt<-dati\_filtrati\_agg%>%

group\_by(dati\_filtrati\_agg$Data\_app...10)%>%

summarize(media\_km=mean(KM\_Presidio\_Resid),

media\_eta=mean(Età),

media\_prezzo\_prestaz=mean(PrezzoPrestazione),

media\_data\_diff=mean(data\_diff),

media\_ora=mean(Ora))

colnames(g3\_filt)<-c("data\_app", "Media\_distanza\_km", "Media\_età", "Media\_tariffa\_prestaz", "media\_tempo\_attesa", "media\_ora")

dati\_num\_filt\_agg<-full\_join(g3\_filt, p\_filt, by="data\_app")

outliers<-which(dati\_num\_filt\_agg$perc\_no\_show>25)

dati\_num\_filt\_agg<- dati\_num\_filt\_agg[-outliers,]

########## VISUALIZZAZIONE GRAFICA

ggplot(dati\_num\_filt\_agg, mapping=aes(x=dati\_num\_filt\_agg$media\_tempo\_attesa, y=dati\_num\_filt\_agg$perc\_no\_show))+

geom\_point(alpha= .6, size=3)+

geom\_smooth(method="loess", se=TRUE,size=1.5)+

labs(title="Relazione fra % di no show e Tempo di Attesa",

subtitle="",

x = "Tempo di attesa",y = "% di no show")

ggplot(dati\_num\_filt\_agg, mapping=aes(x=dati\_num\_filt\_agg$Media\_distanza\_km, y=dati\_num\_filt\_agg$perc\_no\_show))+

geom\_point(alpha= .6, size=3)+

geom\_smooth(method="loess",

se=TRUE,

size=1.5)+

labs(title="Relazione fra % di no show e Distanza fra residenza e presidio",

subtitle="",

x = "Distanza fra residenza e presidio",y = "% di no show")

ggplot(dati\_num\_filt\_agg, mapping=aes(x=dati\_num\_filt\_agg$Media\_età, y=dati\_num\_filt\_agg$perc\_no\_show))+

geom\_point(alpha= .6, size=3)+

geom\_smooth(method="loess",

se=TRUE,

size=1.5)+

labs(title="Relazione fra % di no show ed età",

subtitle="",

x = "Età",y = "% di no show")

ggplot(dati\_num\_filt\_agg, mapping=aes(x=dati\_num\_filt\_agg$Media\_tariffa\_prestaz, y=dati\_num\_filt\_agg$perc\_no\_show))+

geom\_point(alpha= .6, size=3)+

geom\_smooth(method="loess",

se=TRUE,

size=1.5)

correlazione<- round(cor(dati\_num\_filt\_agg[,c(2,3,4,5)]),2)

############# REGRESSIONE LOGISTICA

attach(dati\_filtrati\_agg)

model\_logit\_filt\_agg <- glm(dati\_filtrati\_agg$`NO SHOW` ~ `Ricetta Digitale`+Presidio+PrezzoPrestazione+Esente+Sesso+Età+KM\_Presidio\_Resid+data\_diff+Ora+MESE+Sez\_giornata+Anno+Tipo\_di\_esenzione+Descrizione+Fascia\_età+ Straniero, data = dati\_filtrati\_agg, family = binomial)

summary(model\_logit\_filt\_agg)

dev\_filt\_agg<- model\_logit\_filt\_agg$deviance

pchisq(dev\_filt\_agg, model\_logit\_filt\_agg$df.residual); par(mfrow=c(2,2)); plot(model\_logit\_filt\_agg)

outlierTest(model\_logit\_filt\_agg); influenceIndexPlot(model\_logit\_filt\_agg\_2) ; influencePlot(model\_logit\_filt\_agg)

#provo con altri link

model\_logit\_filt\_agg\_probit <- glm(dati\_filtrati\_agg$`NO SHOW` ~ `Ricetta Digitale`+Presidio+PrezzoPrestazione+Esente+Sesso+Età+KM\_Presidio\_Resid+data\_diff+Ora+MESE+Sez\_giornata+Anno+Tipo\_di\_esenzione+Descrizione+Fascia\_età+Straniero , data = dati\_filtrati\_agg, family = binomial(link = "probit"))

model\_logit\_filt\_agg\_cauchy <- glm(dati\_filtrati\_agg$`NO SHOW` ~ `Ricetta Digitale`+Presidio+PrezzoPrestazione+Esente+Sesso+Età+KM\_Presidio\_Resid+data\_diff+Ora+MESE+Sez\_giornata+Anno+Tipo\_di\_esenzione+Descrizione+Fascia\_età+Straniero , data = dati\_filtrati\_agg, family = binomial(link = "cauchit"))

model\_logit\_filt\_agg\_cloglog <- glm(dati\_filtrati\_agg$`NO SHOW` ~ `Ricetta Digitale`+Presidio+PrezzoPrestazione+Esente+Sesso+Età+KM\_Presidio\_Resid+data\_diff+Ora+MESE+Sez\_giornata+Anno+Tipo\_di\_esenzione+Descrizione+Fascia\_età+Straniero , data = dati\_filtrati\_agg, family = binomial(link = "cloglog"))

model\_logit\_filt\_agg\_modif <- glm(dati\_filtrati\_agg$`NO SHOW` ~ `Ricetta Digitale`+Presidio+PrezzoPrestazione+Esente+Sesso+Età+log\_KM\_Presidio\_Resid+log\_data\_diff+Ora+MESE+Sez\_giornata+Anno+Tipo\_di\_esenzione+Descrizione+Fascia\_età+ Straniero, data = dati\_filtrati\_agg, family = binomial)

summary(model\_logit\_filt\_agg\_probit)$aic ; summary(model\_logit\_filt\_agg)$aic; summary(model\_logit\_filt\_agg\_cauchy)$aic ; summary(model\_logit\_filt\_agg\_cloglog)$aic

model\_logit\_filt\_agg\_probit$deviance ; model\_logit\_filt\_agg$deviance; model\_logit\_filt\_agg\_cauchy$deviance; model\_logit\_filt\_agg\_cloglog$deviance step.model <- stepAIC(model\_logit\_filt\_agg\_cloglog, direction = "backward", trace = FALSE) summary(step.model)

attach(dati\_filtrati\_agg); dati\_filtrati\_agg$`NO SHOW`<- ifelse(dati\_filtrati\_agg$`NO SHOW`=="SHOW", 1,0)

dati\_filtrati\_agg$`NO SHOW`<- as.factor(dati\_filtrati\_agg$`NO SHOW`)

model<- glm(dati\_filtrati\_agg$`NO SHOW`~ KM\_Presidio\_Resid , data=dati\_filtrati\_agg, family=binomial) summary(model)

plot(KM\_Presidio\_Resid,dati\_filtrati\_agg$`NO SHOW`,col = "red",pch=19,main="curva stimata")

grid<-seq(0,100,by=.01); eta<-coef(model)[1]+coef(model)[2]\*grid; mu<-exp(eta)/(1+exp(eta))

lines(grid,mu,type="l",col=3, lwd=3)

#######REGRESSIONE LOGISTICA PER LA CLASSIFICAZIONE

train\_size\_filt\_agg <- floor(0.80\*nrow(dati\_filtrati\_agg))

set.seed(123); train\_index\_filt\_agg <- sample(seq\_len(nrow(dati\_filtrati\_agg)), size = train\_size\_filt\_agg)

training\_filt\_agg <- dati\_filtrati\_agg[train\_index\_filt\_agg, ]; test\_filt\_agg <- dati\_filtrati\_agg[-train\_index\_filt\_agg, ]

subtrain\_size\_filt\_agg <- floor(0.65\*nrow(training\_filt\_agg)); subtrain\_index\_filt\_agg <- sample(seq\_len(nrow(training\_filt\_agg)), size = subtrain\_size\_filt\_agg)

sub\_training\_filt\_agg <- training\_filt\_agg[subtrain\_index\_filt\_agg, ]; sub\_training\_original\_agg <- sub\_training\_filt\_agg

validation\_filt\_agg <- training\_filt\_agg[-subtrain\_index\_filt\_agg, ]; validation\_original\_agg <- validation\_filt\_agg

nrow(sub\_training\_original\_agg)/nrow(dati\_filtrati\_agg) #52%; nrow(validation\_original\_agg)/nrow(dati\_filtrati\_agg) #28%

nrow(test\_filt\_agg)/nrow(dati\_filtrati\_agg) #20%

round(prop.table(table(sub\_training\_filt\_agg$`NO SHOW`)), 2)

round(prop.table(table(validation\_filt\_agg$`NO SHOW`)), 2)

round(prop.table(table(test\_filt\_agg$`NO SHOW`)), 2)

sub\_training\_num\_filt<- sub\_training\_filt\_agg[,c(5,13,19,21)]

medianAll\_filt <- matrix(0, nrow = 1, ncol = (dim(sub\_training\_num\_filt)[2]))

rownames(medianAll\_filt) <- "Median"

colnames(medianAll\_filt) <- colnames(sub\_training\_num\_filt)

averageAll\_filt <- matrix(0, nrow = 1, ncol = (dim(sub\_training\_num\_filt)[2]))

rownames(averageAll\_filt) <- "Mean"

colnames(averageAll\_filt) <- colnames(sub\_training\_num\_filt)

sub\_training\_num\_filt<- as.data.frame(sub\_training\_num\_filt)

for (i in 1:(dim(sub\_training\_num\_filt)[2])){

averageAll\_filt[,i] <- median(sub\_training\_num\_filt[, i], na.rm = F)

medianAll\_filt[,i] <- mean(sub\_training\_num\_filt[, i], na.rm = F)

}

model\_logit\_filt\_agg\_sub <- glm(sub\_training\_filt\_agg$`NO SHOW` ~ `Ricetta Digitale`+Presidio+PrezzoPrestazione+Esente+Sesso+Età+KM\_Presidio\_Resid+data\_diff+Ora+MESE+Sez\_giornata+Anno+Tipo\_di\_esenzione+Descrizione+Fascia\_età+Straniero , data = sub\_training\_filt\_agg, family = binomial(link = "cloglog"))

pred\_logit\_agg <- predict(model\_logit\_filt\_agg\_sub, validation\_filt\_agg[, -c(3,4,6,8,14,22)], type = "response")

validation\_filt\_agg<- as.data.frame(validation\_filt\_agg)

validation\_filt\_agg$`NO SHOW`<- ifelse(validation\_filt\_agg$`NO SHOW`=="SHOW",1,0)

validation\_filt\_agg$`NO SHOW`<- as.factor(validation\_filt\_agg$`NO SHOW`)

pred\_logit\_class\_agg <- ifelse (pred\_logit\_agg > 0.5, 1, 0)

confusionMatrix(factor(validation\_filt\_agg[, 23]), factor(as.vector(pred\_logit\_class\_agg)))

pred\_logit\_class\_agg <- ifelse (pred\_logit\_agg > 0.55, 1, 0)

confusionMatrix(factor(validation\_filt\_agg[, 23]), factor(as.vector(pred\_logit\_class\_agg)))

pred\_logit\_class\_agg <- ifelse (pred\_logit\_agg > 0.45, 1, 0)

confusionMatrix(factor(validation\_filt\_agg[, 23]), factor(as.vector(pred\_logit\_class\_agg)))

set.seed(123)

table(sub\_training\_filt\_agg$`NO SHOW`)/nrow(sub\_training\_filt\_agg)\*100

obs\_noshow<- which(sub\_training\_filt\_agg$`NO SHOW`=="NO SHOW")

data\_no\_show<- sub\_training\_filt\_agg[obs\_noshow,]

table(sub\_training\_filt\_agg\_bal$`NO SHOW`)/nrow(sub\_training\_filt\_agg\_bal)\*100

obs\_show<- which(sub\_training\_filt\_agg$`NO SHOW`=="SHOW")

samp<- sample(1:length(obs\_show), size = 2\*nrow(data\_no\_show), replace = F)

data\_show<- sub\_training\_filt\_agg[obs\_show[samp],]

sub\_training\_filt\_agg\_bal<- rbind(data\_show, data\_no\_show, data\_no\_show)

table(sub\_training\_filt\_agg\_bal$`NO SHOW`)/nrow(sub\_training\_filt\_agg\_bal)\*100

model\_logit\_filt\_agg\_bal <- glm(sub\_training\_filt\_agg\_bal$`NO SHOW` ~ `Ricetta Digitale`+Presidio+PrezzoPrestazione+Esente+Sesso+Età+KM\_Presidio\_Resid+data\_diff+Ora+MESE+Sez\_giornata+Anno+Tipo\_di\_esenzione+Descrizione+Fascia\_età+Straniero , data = sub\_training\_filt\_agg\_bal, family = binomial(link = "cloglog"))

samp\_val<- sample(1:nrow(validation\_filt\_agg), size = 2000, replace = F )

pred\_logit\_agg <- predict(model\_logit\_filt\_agg\_bal, validation\_filt\_agg[samp\_val, -c(3,4,6,8,14,22)], type = "response")

pred\_logit\_class\_agg\_bal <- ifelse (pred\_logit\_agg > 0.5, 1, 0) confusionMatrix(factor(validation\_filt\_agg[samp\_val, 23]), factor(as.vector(pred\_logit\_class\_agg\_bal)))

accuracy<-c(0.624, 0.68, 0.6355, 0.632, 0.6735,0.6915,0.6435,0.6475,0.6405,0.6495)

mean(accuracy)

pred\_logit\_class\_agg\_bal <- ifelse (pred\_logit\_agg > 0.55, 1, 0)

confusionMatrix(factor(validation\_filt\_agg[samp\_val, 23]), factor(as.vector(pred\_logit\_class\_agg\_bal)))

pred\_logit\_class\_agg\_bal <- ifelse (pred\_logit\_agg > 0.15, 1, 0)

confusionMatrix(factor(validation\_filt\_agg[samp\_val, 23]), factor(as.vector(pred\_logit\_class\_agg\_bal)))

test\_filt\_agg<- as.data.frame(test\_filt\_agg)

samp\_test<- sample(1:nrow(test\_filt\_agg), size = 2000, replace = F )

test\_filt\_agg$`NO SHOW`<- ifelse(test\_filt\_agg$`NO SHOW`=="SHOW",1,0)

pred\_logit\_agg <- predict(model\_logit\_filt\_agg\_bal, test\_filt\_agg[samp\_test, -c(3,4,6,8,14,22)], type = "response")

pred\_logit\_class\_agg\_bal <- ifelse (pred\_logit\_agg > 0.15, 1, 0)

confusionMatrix(factor(test\_filt\_agg[samp\_test, 23]), factor(as.vector(pred\_logit\_class\_agg\_bal)))

step.model\_bal<- stepAIC(model\_logit\_filt\_agg\_bal, direction = "backward", trace = FALSE)

summary(step.model\_bal)

summary(model\_logit\_filt\_agg\_bal)

pred\_logit\_agg <- predict(step.model\_bal, validation\_filt\_agg[samp\_val, -c(3,4,6,10,14,15,19,22)], type = "response")

pred\_logit\_class\_agg <- ifelse (pred\_logit\_agg > 0.5, 1, 0)

confusionMatrix(factor(validation\_filt\_agg[samp\_val, 23]), factor(as.vector(pred\_logit\_class\_agg)))

pred\_logit\_class\_agg <- ifelse (pred\_logit\_agg > 0.6, 1, 0)

confusionMatrix(factor(validation\_filt\_agg[samp\_val, 23]), factor(as.vector(pred\_logit\_class\_agg)))

pred\_logit\_class\_agg <- ifelse (pred\_logit\_agg > 0.15, 1, 0)

confusionMatrix(factor(validation\_filt\_agg[samp\_val, 23]), factor(as.vector(pred\_logit\_class\_agg)))

# Single link clustering

tmp <- hclust (dist(dati\_filtrati\_agg[, c(5,13,19,21)]),method="single")

noshow.single <- rect.hclust (tmp , k=2)

(n.dati <- length (dati\_filtrati\_agg [,1]))

noshow.single; dati.single.class <- rep (NA , n.dati)

for (g in 1:2) dati.single.class[noshow.single [[g]]] <- g

classError(dati.single.class , dati\_filtrati\_agg$`NO SHOW`) table(dati.single.class)

adjustedRandIndex (dati.single.class , dati\_filtrati\_agg$`NO SHOW`)

# Average link clustering

tmp <- hclust (dist(dati\_filtrati\_agg[, c(5,13,19,21)]),method="average"); dati.average <- rect.hclust (tmp , k=2); n.dati <- length (dati\_filtrati\_agg [,1])

dati.average.class <- rep (NA , n.dati)

for (g in 1:2) dati.average.class [dati.average [[g]]] <- g

classError (dati.average.class , dati\_filtrati\_agg$`NO SHOW`)

adjustedRandIndex (dati.average.class , dati\_filtrati\_agg$`NO SHOW`)

# Complete link clustering

tmp <- hclust (dist(dati\_filtrati\_agg[, c(5,13,19,21)]),method="complete")

dati.complete <- rect.hclust (tmp , k=2)

n.dati <- length (dati\_filtrati\_agg [,1])

dati.complete.class <- rep (NA , n.dati)

for (g in 1:2) dati.complete.class [dati.complete [[g]]] <- g

classError (dati.complete.class , dati\_filtrati\_agg$`NO SHOW`); table(dati.complete.class)

adjustedRandIndex (dati.complete.class , dati\_filtrati\_agg$`NO SHOW`)

dati\_filtrati\_agg$`NO SHOW`<- ifelse(dati\_filtrati\_agg$`NO SHOW`=="SHOW",2,1)

confusionMatrix(factor(dati.complete.class) , factor(dati\_filtrati\_agg$`NO SHOW`))

dati.kmeans <- kmeans (dati\_filtrati\_agg[, c(5,13,19,21)] , centers=2, nstart =20)

classError (dati.kmeans$cluster , dati\_filtrati\_agg$`NO SHOW`)

table(dati.kmeans$cluster)

confusionMatrix(factor(dati.kmeans$cluster), factor(dati\_filtrati\_agg$`NO SHOW`))

adjustedRandIndex (dati.kmeans$cluster , dati\_filtrati\_agg$`NO SHOW`)

################ PROVIAMO AD EFFETTUARE TUTTE LE ANALISI PRECEDENTI MA SUL DATASET BILANCIATO

table(dati\_filtrati\_agg$`NO SHOW`)/nrow(dati\_filtrati\_agg)\*100

obs\_noshow<- which(dati\_filtrati\_agg$`NO SHOW`=="NO SHOW")

data\_no\_show<- dati\_filtrati\_agg[obs\_noshow,]; obs\_show<- which(dati\_filtrati\_agg$`NO SHOW`=="SHOW")

samp<- sample(1:length(obs\_show), size = 2\*nrow(data\_no\_show), replace = F)

data\_show<- dati\_filtrati\_agg[obs\_show[samp],]

dati\_filtrati\_agg\_bal<- rbind(data\_show, data\_no\_show, data\_no\_show)

table(dati\_filtrati\_agg\_bal$`NO SHOW`)/nrow(dati\_filtrati\_agg\_bal)\*100

# Single link clustering

tmp <- hclust (dist(dati\_filtrati\_agg\_bal[, c(5,13,19,21)]),method="single")

noshow.single <- rect.hclust (tmp , k=2)

(n.dati <- length (dati\_filtrati\_agg\_bal [,1]))

dati.single.class <- rep (NA , n.dati)

for (g in 1:2) dati.single.class[noshow.single [[g]]] <- g

classError(dati.single.class , dati\_filtrati\_agg\_bal$`NO SHOW`)

table(dati.single.class)

table(dati\_filtrati\_agg\_bal$`NO SHOW`)

adjustedRandIndex (dati.single.class , dati\_filtrati\_agg\_bal$`NO SHOW`)

# Average link clustering

tmp <- hclust (dist(dati\_filtrati\_agg\_bal[, c(5,13,19,21)]),method="average")

dati.average <- rect.hclust (tmp , k=2)

n.dati <- length (dati\_filtrati\_agg\_bal [,1])

dati.average.class <- rep (NA , n.dati)

for (g in 1:2) dati.average.class [dati.average [[g]]] <- g

classError (dati.average.class , dati\_filtrati\_agg\_bal$`NO SHOW`)

adjustedRandIndex (dati.average.class , dati\_filtrati\_agg\_bal$`NO SHOW`)

tmp <- hclust (dist(dati\_filtrati\_agg\_bal[, c(5,13,19,21)]),method="complete")

dati.complete <- rect.hclust (tmp , k=2); n.dati <- length (dati\_filtrati\_agg\_bal [,1])

dati.complete.class <- rep (NA , n.dati)

for (g in 1:2) dati.complete.class [dati.complete [[g]]] <- g

classError (dati.complete.class , dati\_filtrati\_agg\_bal$`NO SHOW`)

dati\_filtrati\_agg\_bal$`NO SHOW`<- ifelse(dati\_filtrati\_agg\_bal$`NO SHOW`=="SHOW",2,1)

confusionMatrix(factor(dati.complete.class), factor(dati\_filtrati\_agg\_bal$`NO SHOW`))

adjustedRandIndex (dati.complete.class , dati\_filtrati\_agg\_bal$`NO SHOW`)

dati.kmeans <- kmeans (dati\_filtrati\_agg\_bal[, c(5,13,19,21)] , centers=2, nstart =50)

classError (dati.kmeans$cluster , dati\_filtrati\_agg\_bal$`NO SHOW`)

adjustedRandIndex (dati.kmeans$cluster , dati\_filtrati\_agg\_bal$`NO SHOW`)

confusionMatrix(factor(dati.kmeans$cluster), factor(dati\_filtrati\_agg\_bal$`NO SHOW`))

sub\_training\_num\_filt\_mclust<-as.data.frame(sub\_training\_filt\_agg[,c(5,13,19,21)])

##### CLUSTERING MODEL BASED

library(mclust)

cluster1<- Mclust(sub\_training\_filt\_agg[,c(5,13,19,21)], G=2)

summary(cluster1); cluster1$parameters$pro;cluster1$parameters$mean; str(cluster1)

cluster1\_icl<- mclustICL(sub\_training\_filt\_agg[,c(5,13,19,21)], G=2); cluster1\_icl; str(cluster1\_icl)

cluster\_density\_1<- densityMclust(sub\_training\_filt\_agg[,c(5,13,19,21)])

plot(cluster\_density\_1, what = "BIC")

plot(cluster\_density, what = "BIC", ylim= c(-28000, -24000))

cluster1\_icl\_tutti<- mclustICL(sub\_training\_filt\_agg[,c(5,13,19,21)], G=1:5)

plot(cluster1\_icl\_tutti)

summary(cluster1\_icl\_tutti)

uncerPlot(z=cluster1$z,truth=cluster1$classification)

length(which(cluster1$uncertainty>0.45))

coordProj(data=sub\_training\_num\_filt\_mclust, dimens=c(1,4), what="uncertainty",

parameters=cluster1$parameters , z=cluster1$z, xlim = c(0,300))

coordProj(data=sub\_training\_num\_filt\_mclust, dimens=c(1,4), what="uncertainty",

parameters=cluster1$parameters , z=cluster1$z, xlim = c(0,50), ylim = c(0,40))

coordProj (data=sub\_training\_num\_filt\_mclust, dimens=c(1,2), what="uncertainty",

parameters=cluster1$parameters , z=cluster1$z, xlim = c(0,400), ylim = c(0,240))

cluster1$classification[1]

table(cluster1$classification)

coordProj (data=sub\_training\_num\_filt\_mclust, dimens=c(4,3), what="uncertainty",

parameters=cluster1$parameters , z=cluster1$z, xlim = c(0,70), ylim = c(0,140))

coordProj (data=sub\_training\_num, dimens=c(4,3), what="uncertainty",

parameters=cluster1$parameters , z=cluster1$z, xlim = c(0,600), ylim = c(0,100))

KL\_S<-function(mu1,mu2,sigma1,sigma2)

t(mu1-mu2)%\*%(solve(sigma1)+solve(sigma2))%\*%(mu1-mu2)/2+

sum(diag(sigma1%\*%solve(sigma2)+solve(sigma1)%\*%sigma2))/2-length(mu1)

mu1<- cluster1$parameters$mean[,1]; mu2<- cluster1$parameters$mean[,2]

(sigma1<-cluster1$parameters$variance$sigma[, ,1]); (sigma2<-cluster1$parameters$variance$sigma[, ,2])

drop(KL\_S(mu1,mu2,sigma1,sigma2))

par(mfrow=c(1,2))

coordProj (sub\_training\_num\_filt\_mclust, dimens=c(1,4), what="classification",

classification=cluster1$classification, col=c("red2", "dodgerblue2"), symbols=c(6,16),xlim = c(0,500),

sub="(a) Model-Based Clustering")

coordProj(sub\_training\_num\_filt\_mclust, dimens=c(1,4), what="uncertainty",

classification=sub\_training\_filt\_agg$`NO SHOW`,

col=c("red2","dodgerblue2"), xlim = c(0,500), z=cluster1$z,

sub="(b) True classification")

classError(cluster1$classification, sub\_training\_filt\_agg$`NO SHOW`)

cluster1\_completo<- Mclust(dati\_filtrati\_agg[,c(5,13,19,21)], G=2)

classError(cluster1\_completo$classification, dati\_filtrati\_agg$`NO SHOW`); adjustedRandIndex(cluster1\_completo$classification, dati\_filtrati\_agg$`NO SHOW`)

misclassif<- classError(cluster1$classification, sub\_training\_filt\_agg$`NO SHOW`)$misclassified

points(sub\_training\_num\_filt\_mclust[misclassif,c(1,4)],pch=19)

set.seed(123)

mixmod<- mixmodLearn(dati\_filtrati\_agg[,c(5,13,19,21)], dati\_filtrati\_agg$`NO SHOW`, models= mixmodGaussianModel(family = "all")) ;summary(mixmod)

mixmod@bestResult@parameters@mean; table(mixmod@bestResult@partition)

mixmod@bestResult@parameters@proportions;

mixmod@bestResult@model #Gaussian\_pk\_L\_C; mixmod@bestResult@criterionValue #0.06594376

mixmod@bestResult@model #Gaussian\_pk\_L\_C; mixmod@bestResult@criterionValue #0.06594376 mixmod@bestResult@model #Gaussian\_pk\_L\_C; mixmod@bestResult@criterionValue #0.06594376

mixmod\_bic<- mixmodLearn(dati\_filtrati\_agg[,c(5,13,19,21)], dati\_filtrati\_agg$`NO SHOW`, models= mixmodGaussianModel(family = "all"), criterion="BIC") ; summary(mixmod\_bic)

mixmod\_bic@bestResult@parameters@proportions

mixmod\_bic@bestResult@parameters@mean

res = mixmodLearn(dati\_filtrati\_agg[,c(5,13,19,21)], dati\_filtrati\_agg$`NO SHOW`,

models=mixmodGaussianModel(family="all"), criterion=c('CV','BIC'))

BIC = CV = rep(NA ,length(res@models@listModels) )

for (i in 1: length(res@models@listModels)){

ind = which(res@results [[i]] @model == res@models@listModels)

CV[ind] = res@results [[i]] @criterionValue [1]

BIC[ind] = res@results [[i]] @criterionValue [2]

}

par(mfrow=c(2,1))

plot(BIC ,type='b',xlab='',xaxt='n',col =2); axis(1,at=1: length(

res@results),labels=substr(res@models@listModels ,10 ,30),cex.axis =0.7

,las =2)

abline(v=which.min(BIC), col=1, lty =2)

plot(CV ,type='b',xlab='',xaxt='n',col =3); axis(1,at=1: length(

res@results),labels=substr(res@models@listModels,10 ,30),cex.axis =0.7

,las =2)

abline(v=which.min(CV), col=1, lty =2)

par(mfrow=c(1,1))

##prevision

set.seed(124)

train\_size <- floor(0.85\*nrow(dati\_filtrati\_agg)); test\_size <- floor(0.15\*nrow(dati\_filtrati\_agg))

training\_idx <- sample(seq\_len(nrow(dati\_filtrati\_agg)), size = train\_size); test\_idx <- sample(seq\_len(nrow(dati\_filtrati\_agg)), size = test\_size)

training\_set <- dati\_filtrati\_agg[training\_idx, ]; test\_set <- dati\_filtrati\_agg[-training\_idx, ]

dati\_filtrati\_agg<- as.data.frame(dati\_filtrati\_agg)

class\_crit<- mixmodLearn(dati\_filtrati\_agg[-test\_idx,c(5,13,19,21)], dati\_filtrati\_agg$`NO SHOW`[-test\_idx],

models=mixmodGaussianModel(family="all",equal.proportions=FALSE))

previs<- mixmodPredict(data=dati\_filtrati\_agg[test\_idx,c(5,13,19,21)], classificationRule=class\_crit["bestResult"])

str(previs)

which(previs@proba > 0.30 & previs@proba < 0.70)

par(mfrow=c(1,1))

coordProj (dati\_filtrati\_agg[c(1453,2380,3791,4473,6045,6972,8383,9065 ),c(5,13,19,21)], dimens=c(1,4), what="classification",

classification=previs@partition,

col=c("red2", "dodgerblue2"), symbols=c(6,16),

sub="(b) Model-Based Clustering")

previs@partition #partizione delle u.s. ai gruppi ; table(previs@partition)

dati\_filtrati\_agg[c(1453,2380,3791,4473,6045,6972,8383,9065 ),c(5,13,19,21)]

prob\_post<- previs@proba; head(prob\_post)

err<- mean(as.integer(dati\_filtrati\_agg$`NO SHOW`[test\_idx]) != previs["partition"]); err

confusionMatrix(as.factor(dati\_filtrati\_agg$`NO SHOW`[test\_idx]), as.factor(previs["partition"]))

set.seed(124)

train\_size <- floor(0.85\*nrow(dati\_filtrati\_agg\_bal)); test\_size <- floor(0.15\*nrow(dati\_filtrati\_agg\_bal))

training\_idx <- sample(seq\_len(nrow(dati\_filtrati\_agg\_bal)), size = train\_size)

test\_idx <- sample(seq\_len(nrow(dati\_filtrati\_agg\_bal)), size = test\_size)

training\_set <- dati\_filtrati\_agg\_bal[training\_idx, ]

test\_set <- dati\_filtrati\_agg\_bal[-training\_idx, ]

dati\_filtrati\_agg\_bal<- as.data.frame(dati\_filtrati\_agg\_bal)

class\_crit<- mixmodLearn(dati\_filtrati\_agg\_bal[-test\_idx,c(5,13,19,21)], dati\_filtrati\_agg\_bal$`NO SHOW`[-test\_idx],

models=mixmodGaussianModel(family="all",equal.proportions=FALSE))

previs<- mixmodPredict(data=dati\_filtrati\_agg\_bal[test\_idx,c(5,13,19,21)], classificationRule=class\_crit["bestResult"])

str(previs)

which(previs@proba > 0.499 & previs@proba < 0.501)

par(mfrow=c(1,1))

coordProj (dati\_filtrati\_agg\_bal[c(60, 301 , 312, 328 , 332 , 587, 1271 ,1512, 1523, 1539, 1543 ,1798),c(5,13,19,21)], dimens=c(1,4), what="classification",

classification=previs@partition,

col=c("red2", "dodgerblue2"), symbols=c(6,16),

sub="(b) Model-Based Clustering")

previs@partition #partizione delle u.s. ai gruppi

dati\_filtrati\_agg\_bal[c(60, 301 , 312, 328 , 332 , 587, 1271 ,1512, 1523, 1539, 1543 ,1798),c(5,13,19,21)]

prob\_post<- previs@proba; head(prob\_post)

err<- mean(as.integer(dati\_filtrati\_agg\_bal$`NO SHOW`[test\_idx]) != previs["partition"]); err

train\_size <- floor(0.85\*nrow(dati\_filtrati\_agg\_bal)); test\_size <- floor(0.15\*nrow(dati\_filtrati\_agg\_bal))

training\_idx <- sample(seq\_len(nrow(dati\_filtrati\_agg\_bal)), size = train\_size)

test\_idx <- sample(seq\_len(nrow(dati\_filtrati\_agg\_bal)), size = test\_size); training\_set <- dati\_filtrati\_agg\_bal[training\_idx, ]

test\_set <- dati\_filtrati\_agg\_bal[-training\_idx, ]

dati\_filtrati\_agg\_bal<- as.data.frame(dati\_filtrati\_agg\_bal)

class\_crit<- mixmodLearn(dati\_filtrati\_agg\_bal[-test\_idx,c(5,13,19,21)], dati\_filtrati\_agg\_bal$`NO SHOW`[-test\_idx],

models=mixmodGaussianModel(family="all",equal.proportions=FALSE))

previs<- mixmodPredict(data=dati\_filtrati\_agg\_bal[test\_idx,c(5,13,19,21)], classificationRule=class\_crit["bestResult"])

str(previs)

err<- mean(as.integer(dati\_filtrati\_agg\_bal$`NO SHOW`[test\_idx]) != previs["partition"]); err

mean(c(0.4277457, 0.3980182, 0.3947151, 0.4260941, 0.3806771, 0.3980182, 0.4128819, 0.3955409, 0.4145334))

library(mclust)

MDA\_mod\_G2 = MclustDA(dati\_filtrati\_agg[-test\_idx,c(5,13,19,21)], dati\_filtrati\_agg$`NO SHOW`[-test\_idx],G=2)

summary(MDA\_mod\_G2); str(MDA\_mod\_G2)

MDA\_mod\_G\_var = MclustDA(dati\_filtrati\_agg[-test\_idx,c(5,13,19,21)], dati\_filtrati\_agg$`NO SHOW`[-test\_idx])

summary(MDA\_mod\_G\_var); str(MDA\_mod\_G\_var)

MDA\_mod\_G\_var2 = MclustDA(dati\_filtrati\_agg[-test\_idx,c(5,13,19,21)], dati\_filtrati\_agg$`NO SHOW`[-test\_idx], G=1:10); summary(MDA\_mod\_G\_var2); str(MDA\_mod\_G\_var2)

MDA\_mod\_G\_var$models$`NO SHOW`$parameters$mean; MDA\_mod\_G\_var$models$SHOW$parameters$mean

obs\_G1<-MDA\_mod\_G\_var$models$'1'$observations; obs\_G2<-MDA\_mod\_G\_var$models$'2'$observations

attach(dati\_filtrati\_agg)

ggplot(data = dati\_filtrati\_agg[obs\_G1,], mapping = aes(data\_diff[obs\_G1],KM\_Presidio\_Resid[obs\_G1] ))+

geom\_point(size=3, color=MDA\_mod\_G\_var$models$`NO SHOW`$classification)+

labs(title="Distinzione fra le cinque componenti del primo gruppo",

subtitle="",

x = "Tempo di attesa",y = "Distanza Residenza-Presidio", color = "Classificazione")

ggplot(data = dati\_filtrati\_agg[obs\_G2,], mapping = aes(data\_diff[obs\_G2],KM\_Presidio\_Resid[obs\_G2] ))+

geom\_point(size=3, color=MDA\_mod\_G\_var$models$SHOW$classification)+

labs(title="Distinzione fra le cinque componenti del secondo gruppo",

subtitle="",

x = "Tempo di attesa",y = "Distanza Residenza-Presidio", color = "Classificazione")

ggplot(data = dati\_filtrati\_agg[obs\_G1,], mapping = aes(data\_diff[obs\_G1],KM\_Presidio\_Resid[obs\_G1],color=round(MDA\_mod\_G\_var$models$'1'$uncertainty,4)))+

geom\_point(size=3, alpha=1)+

labs(title="Incertezza fra le 5 componenti componenti del gruppo No Show",

x = "Tempo di attesa",y = "Distanza Residenza-Presidio", color = "Incertezza")+

xlim(0,500)

length(which(round(MDA\_mod\_G\_var$models$'1'$uncertainty,3)>0.5))/length(MDA\_mod\_G\_var$models$`1`$observations)

ggplot(data = dati\_filtrati\_agg[obs\_G2,], mapping = aes(data\_diff[obs\_G2],KM\_Presidio\_Resid[obs\_G2],color=round(MDA\_mod\_G\_var$models$'2'$uncertainty,4)))+

geom\_point(size=3, alpha=1)+

labs(title="Incertezza fra le 5 componenti componenti del gruppo Show",

x = "Tempo di attesa",y = "Distanza Residenza-Presidio", color = "Incertezza")+

xlim(0,500)

length(which(round(MDA\_mod\_G\_var$models$SHOW$uncertainty,3)>0.5))/length(MDA\_mod\_G\_var$models$SHOW$observations)\*100

######### PROVO AD EFFETTUARE L'ANALISI MDA SUI DATI BILANCIATI

attach(dati\_filtrati\_agg\_bal)

library(mclust)

MDA\_mod\_G2\_bal = MclustDA(dati\_filtrati\_agg\_bal[-test\_idx,c(5,13,19,21)], dati\_filtrati\_agg\_bal$`NO SHOW`[-test\_idx],G=2)

summary(MDA\_mod\_G2\_bal); str(MDA\_mod\_G2\_bal)

MDA\_mod\_G\_var\_bal = MclustDA(dati\_filtrati\_agg\_bal[-test\_idx,c(5,13,19,21)], dati\_filtrati\_agg\_bal$`NO SHOW`[-test\_idx])

summary(MDA\_mod\_G\_var\_bal); str(MDA\_mod\_G\_var\_bal)

MDA\_mod\_G\_var2\_bal = MclustDA(dati\_filtrati\_agg\_bal[-test\_idx,c(5,13,19,21)], dati\_filtrati\_agg\_bal$`NO SHOW`[-test\_idx], G=1:10) summary(MDA\_mod\_G\_var2\_bal); str(MDA\_mod\_G\_var2\_bal)

MDA\_mod\_G\_var\_bal$models$`NO SHOW`$parameters$mean

MDA\_mod\_G\_var\_bal$models$SHOW$parameters$mean

##### VISUALIZZAZIONE GRAFICA

obs\_G1<-MDA\_mod\_G\_var\_bal$models$`NO SHOW`$observations; obs\_G2<-MDA\_mod\_G\_var\_bal$models$SHOW$observations

attach(dati\_filtrati\_agg\_bal)

ggplot(data = dati\_filtrati\_agg\_bal[obs\_G1,], mapping = aes(dati\_filtrati\_agg\_bal$data\_diff[obs\_G1],dati\_filtrati\_agg\_bal$KM\_Presidio\_Resid[obs\_G1] ))+

geom\_point(size=3, color=MDA\_mod\_G\_var\_bal$models$`NO SHOW`$classification)+

labs(title="Distinzione fra le cinque componenti del gruppo No Show",

subtitle="",

x = "Tempo di attesa",y = "Distanza Residenza-Presidio", color = "Classificazione")

ggplot(data = dati\_filtrati\_agg\_bal[obs\_G2,], mapping = aes(dati\_filtrati\_agg\_bal$data\_diff[obs\_G2],dati\_filtrati\_agg\_bal$KM\_Presidio\_Resid[obs\_G2] ))+

geom\_point(size=3, color=MDA\_mod\_G\_var\_bal$models$SHOW$classification)+

labs(title="Distinzione fra le cinque componenti del secondo gruppo",

subtitle="", x = "Tempo di attesa",y = "Distanza Residenza-Presidio", color = "Classificazione")+ xlim(0,400)

ggplot(data = dati\_filtrati\_agg\_bal[obs\_G1,], mapping = aes(dati\_filtrati\_agg\_bal$data\_diff[obs\_G1],dati\_filtrati\_agg\_bal$KM\_Presidio\_Resid[obs\_G1],color=round(MDA\_mod\_G\_var\_bal$models$`NO SHOW`$uncertainty,4)))+

geom\_point(size=3, alpha=1)+

labs(title="Incertezza fra le 5 componenti componenti del gruppo No Show",

x = "Tempo di attesa",y = "Distanza Residenza-Presidio", color = "Incertezza")+

xlim(0,500)

length(which(round(MDA\_mod\_G\_var\_bal$models$`NO SHOW`$uncertainty,3)>0.5))/length(MDA\_mod\_G\_var\_bal$models$`NO SHOW`$observations)

ggplot(data = dati\_filtrati\_agg\_bal[obs\_G2,], mapping = aes(dati\_filtrati\_agg\_bal$data\_diff[obs\_G2],dati\_filtrati\_agg\_bal$KM\_Presidio\_Resid[obs\_G2],color=round(MDA\_mod\_G\_var\_bal$models$SHOW$uncertainty,4)))+

geom\_point(size=3, alpha=1)+

labs(title="Incertezza fra le 5 componenti componenti del gruppo Show",

x = "Tempo di attesa",y = "Distanza Residenza-Presidio", color = "Incertezza")+

xlim(0,500)

length(which(round(MDA\_mod\_G\_var\_bal$models$SHOW$uncertainty,3)>0.5))/length(MDA\_mod\_G\_var\_bal$models$SHOW$observations)

G<-10; V=37 ; n<- nrow(dati\_filtrati\_agg\_bal); B<-round(n/V) ; err = matrix(NA ,G,V)

perm= sample(n)

for (g in 1:G){

for (v in 1:V){

test.set.labels = perm[(B\*(v-1)+1):(B\*v)]

mod = MclustDA(dati\_filtrati\_agg\_bal[-test.set.labels ,c(5,13,19,21)], dati\_filtrati\_agg\_bal$`NO SHOW`[-test.set.labels],G=g,modelNames='VVV')

err[g,v] = sum(predict(mod ,dati\_filtrati\_agg\_bal[test.set.labels,c(5,13,19,21)])$class != dati\_filtrati\_agg\_bal$`NO SHOW`[test.set.labels]) / B

}

}

Err; (round(rowMeans(err),4)); which.min(round(rowMeans(err),4))

par(mfrow=c(1,1))

plot (1:G,rowMeans(err),type='b',ylab='Classification error ',xlab='G',ylim=c(0.3,0.5))

######### ANALISI QUANTITATIVE

table(dati\_mattina$`NO SHOW`)/nrow(dati\_mattina)\*100

dati\_pomeriggio<- dati\_filtrati\_agg%>%

filter(Sez\_giornata=="Pomeriggio")

table(dati\_pomeriggio$`NO SHOW`)/nrow(dati\_pomeriggio)\*100

dati\_sera<- dati\_filtrati\_agg%>%

filter(Sez\_giornata=="Sera")

table(dati\_sera$`NO SHOW`)/nrow(dati\_sera)\*100

dat\_per\_ora<- dati\_filtrati\_agg%>%

group\_by(Ora)%>%

count(dati\_filtrati\_agg$`NO SHOW`)

colnames(dat\_per\_ora)<- c("ora\_app", "Type", "Numero")

dat\_per\_ora<- dat\_per\_ora%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_ora)<- c("ora\_app", "NO\_SHOW", "SHOW")

dat\_per\_ora<- dat\_per\_ora[-c(13,14),]

dat\_per\_ora$somma\_app<-dat\_per\_ora$NO\_SHOW+dat\_per\_ora$SHOW

dat\_per\_ora$perc\_no\_show<- (dat\_per\_ora$NO\_SHOW)/dat\_per\_ora$somma\_app\*100

dati\_filtrati\_agg\_wout\_dec<- dati\_filtrati\_agg%>%

filter(MESE!=12)

dat\_per\_ora\_wout\_dec<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(Ora)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_ora\_wout\_dec)<- c("ora\_app", "Type", "Numero")

dat\_per\_ora\_wout\_dec<- dat\_per\_ora\_wout\_dec%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_ora\_wout\_dec)<- c("ora\_app", "NO\_SHOW", "SHOW")

dat\_per\_ora\_wout\_dec<- dat\_per\_ora\_wout\_dec[-c(13,14),]

dat\_per\_ora\_wout\_dec$somma\_app<-dat\_per\_ora\_wout\_dec$NO\_SHOW+dat\_per\_ora\_wout\_dec$SHOW

dat\_per\_ora\_wout\_dec$perc\_no\_show<- (dat\_per\_ora\_wout\_dec$NO\_SHOW)/dat\_per\_ora\_wout\_dec$somma\_app\*100

dat\_per\_MESE<- dati\_filtrati\_agg%>%

group\_by(MESE)%>%

count(dati\_filtrati\_agg$`NO SHOW`)

colnames(dat\_per\_MESE)<- c("Mese\_app", "Type", "Numero")

dat\_per\_MESE<- dat\_per\_MESE%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_MESE)<- c("Mese\_app", "NO\_SHOW", "SHOW")

dat\_per\_MESE$somma\_app<-dat\_per\_MESE$NO\_SHOW+dat\_per\_MESE$SHOW

dat\_per\_MESE$perc\_no\_show<- (dat\_per\_MESE$NO\_SHOW)/dat\_per\_MESE$somma\_app\*100

dat\_per\_presidio<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(Presidio)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_presidio)<- c("Presidio\_app", "Type", "Numero")

dat\_per\_presidio<- dat\_per\_presidio%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_presidio)<- c("Presidio\_app", "NO\_SHOW", "SHOW")

dat\_per\_presidio$somma\_app<-dat\_per\_presidio$NO\_SHOW+dat\_per\_presidio$SHOW

dat\_per\_presidio$perc\_no\_show<- (dat\_per\_presidio$NO\_SHOW)/dat\_per\_presidio$somma\_app\*100

dat\_per\_ricetta<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(`Ricetta Digitale`)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_ricetta)<- c("Tipo di ricetta", "Type", "Numero")

dat\_per\_ricetta<- dat\_per\_ricetta%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_ricetta)<- c("Tipo di ricetta", "NO\_SHOW", "SHOW")

dat\_per\_ricetta$somma\_app<-dat\_per\_ricetta$NO\_SHOW+dat\_per\_ricetta$SHOW

dat\_per\_ricetta$perc\_no\_show<- (dat\_per\_ricetta$NO\_SHOW)/dat\_per\_ricetta$somma\_app\*100

dat\_per\_esente<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(Esente)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_esente)<- c("Esente", "Type", "Numero")

dat\_per\_esente<- dat\_per\_esente%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_esente)<- c("Esente", "NO\_SHOW", "SHOW")

dat\_per\_esente$somma\_app<-dat\_per\_esente$NO\_SHOW+dat\_per\_esente$SHOW

dat\_per\_esente$perc\_no\_show<- (dat\_per\_esente$NO\_SHOW)/dat\_per\_esente$somma\_app\*100

dat\_per\_straniero<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(Straniero)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_straniero)<- c("Straniero", "Type", "Numero")

dat\_per\_straniero<- dat\_per\_straniero%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_straniero)<- c("Straniero", "NO\_SHOW", "SHOW")

dat\_per\_straniero$somma\_app<-dat\_per\_straniero$NO\_SHOW+dat\_per\_straniero$SHOW

dat\_per\_straniero$perc\_no\_show<- (dat\_per\_straniero$NO\_SHOW)/dat\_per\_straniero$somma\_app\*100

dat\_per\_età<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(Fascia\_età)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_età)<- c("Fascia\_età", "Type", "Numero")

dat\_per\_età<- dat\_per\_età%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_età)<- c("Fascia\_età", "NO\_SHOW", "SHOW")

dat\_per\_età$somma\_app<-dat\_per\_età$NO\_SHOW+dat\_per\_età$SHOW

dat\_per\_età$perc\_no\_show<- (dat\_per\_età$NO\_SHOW)/dat\_per\_età$somma\_app\*100

dat\_per\_tipo\_prestazione<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(Tipo\_di\_esenzione)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_tipo\_prestazione)<- c("Tipo\_di\_Prestazione", "Type", "Numero")

dat\_per\_tipo\_prestazione<- dat\_per\_tipo\_prestazione%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_tipo\_prestazione)<- c("Tipo\_di\_Prestazione", "NO\_SHOW", "SHOW")

dat\_per\_tipo\_prestazione$somma\_app<-dat\_per\_tipo\_prestazione$NO\_SHOW+dat\_per\_tipo\_prestazione$SHOW

dat\_per\_tipo\_prestazione$perc\_no\_show<- (dat\_per\_tipo\_prestazione$NO\_SHOW)/dat\_per\_tipo\_prestazione$somma\_app\*100

dat\_per\_anno<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(Anno)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_anno)<- c("Anno", "Type", "Numero")

dat\_per\_anno<- dat\_per\_anno%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_anno)<- c("Anno", "NO\_SHOW", "SHOW")

dat\_per\_anno$somma\_app<-dat\_per\_anno$NO\_SHOW+dat\_per\_anno$SHOW

dat\_per\_anno$perc\_no\_show<- (dat\_per\_anno$NO\_SHOW)/dat\_per\_anno$somma\_app\*100

dati\_filtrati\_agg$Tipo\_di\_esenzione<- Dati\_no\_Show\_Totali\_filtrato$Tipo\_di\_esenzione

dati\_filtrati\_agg$Tipo\_di\_esenzione<- as.factor(dati\_filtrati\_agg$Tipo\_di\_esenzione)

dat\_per\_tipo\_esenzione<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(Tipo\_di\_esenzione)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_tipo\_esenzione)<- c("Tipo\_di\_esenzione", "Type", "Numero")

dat\_per\_tipo\_esenzione<- dat\_per\_tipo\_esenzione%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_tipo\_esenzione)<- c("Tipo\_di\_esenzione", "NO\_SHOW", "SHOW")

dat\_per\_tipo\_esenzione<- dat\_per\_tipo\_esenzione[-c(4,5,10,12),] #elimino quei tipi di esenzioni con pochissime <= 5 appuntamenti

dat\_per\_tipo\_esenzione$somma\_app<-dat\_per\_tipo\_esenzione$NO\_SHOW+dat\_per\_tipo\_esenzione$SHOW

dat\_per\_tipo\_esenzione$perc\_no\_show<- (dat\_per\_tipo\_esenzione$NO\_SHOW)/dat\_per\_tipo\_esenzione$somma\_app\*100

ggplot(dati\_filtrati\_agg, mapping=aes(dati\_filtrati\_agg$data\_diff,

dati\_filtrati\_agg$KM\_Presidio\_Resid,

color=dati\_filtrati\_agg$`NO SHOW`))+

geom\_point(size=2)

set.seed(123)

camp<- sample(1:nrow(dati\_filtrati\_agg), 1000, replace = F)

ggplot(dati\_filtrati\_agg[camp,], mapping=aes(dati\_filtrati\_agg[camp,]$data\_diff,

dati\_filtrati\_agg[camp,]$KM\_Presidio\_Resid,

color=dati\_filtrati\_agg[camp,]$`NO SHOW`))+

geom\_point(size=2)

camp<- sample(1:nrow(dati\_filtrati\_agg), 1000, replace = F)

ggplot(dati\_filtrati\_agg[camp,], mapping=aes(dati\_filtrati\_agg[camp,]$PrezzoPrestazione,

dati\_filtrati\_agg[camp,]$KM\_Presidio\_Resid,

color=dati\_filtrati\_agg[camp,]$`NO SHOW`))+

geom\_point(size=2)

dat\_per\_distanza\_km<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(Fascia\_dist\_km)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_distanza\_km)<- c("Fascia\_dist\_km", "Type", "Numero")

dat\_per\_distanza\_km<- dat\_per\_distanza\_km%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_distanza\_km)<- c("Fascia\_dist\_km", "NO\_SHOW", "SHOW")

dat\_per\_distanza\_km$somma\_app<-dat\_per\_distanza\_km$`NO SHOW`+dat\_per\_distanza\_km$SHOW

dat\_per\_distanza\_km$perc\_no\_show<- (dat\_per\_distanza\_km$`NO SHOW`)/dat\_per\_distanza\_km$somma\_app\*100

dat\_per\_tempo\_di\_attesa<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(Range\_di\_attesa)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_tempo\_di\_attesa)<- c("Range\_di\_attesa", "Type", "Numero")

dat\_per\_tempo\_di\_attesa<- dat\_per\_tempo\_di\_attesa%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_tempo\_di\_attesa)<- c("Range\_di\_attesa", "NO\_SHOW", "SHOW")

dat\_per\_tempo\_di\_attesa$somma\_app<-dat\_per\_tempo\_di\_attesa$NO\_SHOW+dat\_per\_tempo\_di\_attesa$SHOW

dat\_per\_tempo\_di\_attesa$perc\_no\_show<- (dat\_per\_tempo\_di\_attesa$NO\_SHOW)/dat\_per\_tempo\_di\_attesa$somma\_app\*100

dat\_per\_tariffa\_prestaz<- dati\_filtrati\_agg\_wout\_dec%>%

group\_by(Fascia\_Prezzo)%>%

count(dati\_filtrati\_agg\_wout\_dec$`NO SHOW`)

colnames(dat\_per\_tariffa\_prestaz)<- c("Fascia\_Prezzo", "Type", "Numero")

dat\_per\_tariffa\_prestaz<- dat\_per\_tariffa\_prestaz%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_tariffa\_prestaz)<- c("Fascia\_Prezzo", "NO\_SHOW", "SHOW")

dat\_per\_tariffa\_prestaz$somma\_app<-dat\_per\_tariffa\_prestaz$NO\_SHOW+dat\_per\_tariffa\_prestaz$SHOW

dat\_per\_tariffa\_prestaz$perc\_no\_show<- (dat\_per\_tariffa\_prestaz$NO\_SHOW)/dat\_per\_tariffa\_prestaz$somma\_app\*100

dat\_per\_tariffa\_prestaz\_wout\_esente<- dati\_filtrati\_agg\_wout\_dec%>%

filter(Esente==0)

dat\_per\_tariffa\_prestaz\_wout\_esente<-dat\_per\_tariffa\_prestaz\_wout\_esente%>%

group\_by(Fascia\_Prezzo)%>%

count(dat\_per\_tariffa\_prestaz\_wout\_esente$`NO SHOW`)

colnames(dat\_per\_tariffa\_prestaz\_wout\_esente)<- c("Fascia\_Prezzo", "Type", "Numero")

dat\_per\_tariffa\_prestaz\_wout\_esente<- dat\_per\_tariffa\_prestaz\_wout\_esente%>%

spread(key = Type, value = Numero)

colnames(dat\_per\_tariffa\_prestaz\_wout\_esente)<- c("Fascia\_Prezzo", "NO\_SHOW", "SHOW")

dat\_per\_tariffa\_prestaz\_wout\_esente$somma\_app<-dat\_per\_tariffa\_prestaz\_wout\_esente$NO\_SHOW+dat\_per\_tariffa\_prestaz\_wout\_esente$SHOW

dat\_per\_tariffa\_prestaz\_wout\_esente$perc\_no\_show<- (dat\_per\_tariffa\_prestaz\_wout\_esente$NO\_SHOW)/dat\_per\_tariffa\_prestaz\_wout\_esente$somma\_app\*100

ggplot(dati\_fascia\_prezzo, mapping = aes(y=dati\_fascia\_prezzo$perc\_no\_show, x=dati\_fascia\_prezzo$Fascia\_prezzo))

######### CLUSTERING WITH MIXED DATA

samp<- sample(1:nrow(dati\_filtrati\_agg), size = 10000, replace = F)

df2<-dati\_filtrati\_agg[samp,c(1,2,5,7,9,10,11,12,13,14,16,17,19)]

head(df2)

# calculate distance

d\_dist<-daisy(df, metric = "gower")

# hierarchical clustering

hc<-hclust(d\_dist, method = "complete"); plot(hc, labels=FALSE); rect.hclust(hc, k=8, border="red")

cluster<-cutree(hc, k=8)

df<-cbind(df,as.factor(cluster)); table(cluster)

hc$height; hc$dist.method; hc$call; hc$labels

#### Proviamo con 2 gruppi (classi sbilanciate)

# hierarchical clustering

d\_dist<-daisy(df2, metric = "gower")

hc<-hclust(d\_dist, method = "complete")

plot(hc, labels=FALSE); rect.hclust(hc, k=2, border="red"); cluster<-cutree(hc, k=2)

df2<-cbind(df2,as.factor(cluster)); df2<- df2[,-c(14,15)]

labels<- dati\_filtrati\_agg[samp,23]; head(labels)

head(df2$`as.factor(cluster)`)

labels<- ifelse(labels=="SHOW",1,2); labels<- as.factor(labels); table(labels)

table(df2$`as.factor(cluster)`)

err<- mean(labels != df2$`as.factor(cluster)`)

confusionMatrix(labels, df2$`as.factor(cluster)`)

table(dati\_filtrati\_agg\_bal$`NO SHOW`) #perfetto 50 e 50

d\_dist\_bal<-daisy(dati\_filtrati\_agg\_bal[,c(1,2,5,7,9,10,11,12,13,14,16,17,19)], metric = "gower");

hc\_bal<-hclust(d\_dist\_bal, method = "complete"); plot(hc\_bal, labels=FALSE); rect.hclust(hc\_bal, k=2, border="red")

cluster<-cutree(hc\_bal, k=2)

dati\_filtrati\_agg\_bal<-cbind(dati\_filtrati\_agg\_bal,as.factor(cluster)); head(dati\_filtrati\_agg\_bal$`as.factor(cluster)`)

head(dati\_filtrati\_agg\_bal$`NO SHOW`) ; labels\_bal<- dati\_filtrati\_agg\_bal[,23]; labels\_bal<- ifelse(labels\_bal=="SHOW",2,1)

labels\_bal<- as.factor(labels\_bal)

err<- mean(labels\_bal != dati\_filtrati\_agg\_bal$`as.factor(cluster)`); which(labels\_bal != dati\_filtrati\_agg\_bal$`as.factor(cluster)`)

attach(dati\_filtrati\_agg\_bal); dati\_filtrati\_agg\_bal<- arrange(dati\_filtrati\_agg\_bal, Età)

###### PROVO CON ALTRI LINKAGE

#####SINGLE

d\_dist\_bal<-daisy(dati\_filtrati\_agg\_bal[,c(1,2,5,7,9,10,11,12,13,14,16,17,19)], metric = "gower"); hc\_bal<-hclust(d\_dist\_bal, method = "single")

plot(hc\_bal, labels=FALSE)

rect.hclust(hc\_bal, k=2, border="red"); cluster<-cutree(hc\_bal, k=2)

dati\_filtrati\_agg\_bal<-cbind(dati\_filtrati\_agg\_bal,as.factor(cluster)); dati\_filtrati\_agg\_bal<- dati\_filtrati\_agg\_bal[,-24]

head(dati\_filtrati\_agg\_bal$`as.factor(cluster)`)

err<- mean(labels\_bal != dati\_filtrati\_agg\_bal$`as.factor(cluster)`); table(dati\_filtrati\_agg\_bal$`as.factor(cluster)`)

hc\_bal<-hclust(d\_dist\_bal, method = "average")

plot(hc\_bal, labels=FALSE)

rect.hclust(hc\_bal, k=2, border="red"); cluster<-cutree(hc\_bal, k=2); dati\_filtrati\_agg\_bal<-cbind(dati\_filtrati\_agg\_bal,as.factor(cluster))

dati\_filtrati\_agg\_bal<- dati\_filtrati\_agg\_bal[,-24]

head(dati\_filtrati\_agg\_bal$`as.factor(cluster)`)

head(labels\_bal)

err<- mean(labels\_bal != dati\_filtrati\_agg\_bal$`as.factor(cluster)`)

hc\_bal<-hclust(d\_dist\_bal, method = "ward.D")

# dendrogram

plot(hc\_bal, labels=FALSE)

rect.hclust(hc\_bal, k=2, border="red")

cluster<-cutree(hc\_bal, k=2); dati\_filtrati\_agg\_bal<-cbind(dati\_filtrati\_agg\_bal,as.factor(cluster))

dati\_filtrati\_agg\_bal<- dati\_filtrati\_agg\_bal[,-24]

head(dati\_filtrati\_agg\_bal$`as.factor(cluster)`); head(labels\_bal); err<- mean(labels\_bal != dati\_filtrati\_agg\_bal$`as.factor(cluster)`)

KMEANS CLUSTERING WITH CATEGORICAL VARIABLES

dati.kmeans <- kmeans (d\_dist\_bal , centers=2, nstart =20)

err<- mean(dati.kmeans$cluster != labels\_bal); confusionMatrix(as.factor(dati.kmeans$cluster), labels\_bal)

############## MEM

attach(dati\_num\_filt); dati\_filtrati\_agg$`NO SHOW`<- as.factor(dati\_filtrati\_agg$`NO SHOW`)

fit <- flexmix(cbind(SHOW, NO\_SHOW) ~ Media\_distanza\_km+Media\_età+media\_tempo\_attesa+Media\_tariffa\_prestaz ,data=dati\_num\_filt ,k=2, model = FLXMRglm(family = "binomial"))

summary(fit); fit; parameters(fit); ICL(fit)

KLdiv(fit)

asymp.inf<-refit(fit); summary(asymp.inf); plot(asymp.inf)

posterior(fit)

incerte1<-which(posterior(fit)[,1]> 0.47 & posterior(fit)[,1] <0.53)

incerte2<-which(posterior(fit)[,2]> 0.47 & posterior(fit)[,2] <0.53)

dati\_num\_filt[incerte1,]; dati\_num\_filt[incerte2,];

head(round(posterior(fit),2)); str(fit)

fit@cluster; labs<-fit@cluster; plot(fit)

ggplot(data=dati\_num\_filt, mapping = aes(x=media\_tempo\_attesa, y=perc\_no\_show\*100,color=factor(labs)))+

geom\_point(size=2)+

geom\_smooth(method="lm", se=F, size=1.5)

ggplot(data=dati\_num\_filt, mapping = aes(x=media\_tempo\_attesa, y=perc\_no\_show\*100,color=factor(labs)))+

geom\_point(size=2.5)+

geom\_smooth(method="glm", se=F, size=1.5)+

labs(title="Finite mixtures of regression models",

x = "Tempo di attesa medio",y = "% di no show", color = "Gruppo")

bicval <- Inf

itermax <- 35

bics<-matrix(nrow=itermax,ncol=2)

for (iter in 1: itermax){

fit <- flexmix(cbind(SHOW, NO\_SHOW) ~ Media\_distanza\_km+Media\_età+media\_tempo\_attesa+Media\_tariffa\_prestaz ,data=dati\_num\_filt ,k=2, model = FLXMRglm(family = "binomial"))

bics[iter,]<-c(iter,BIC(fit))

if (bicval >BIC(fit))

{

bicval <- BIC(fit)

bestfit <- fit

}

}

Bics; summary(bestfit) ; summary(fit)

(final<-stepFlexmix(cbind(SHOW, NO\_SHOW) ~ Media\_distanza\_km+Media\_età+media\_tempo\_attesa+Media\_tariffa\_prestaz ,data=dati\_num\_filt ,k=2, model = FLXMRglm(family = "binomial"),

nrep = 10,verbose = TRUE, drop = F, unique = FALSE))

(final.v<-stepFlexmix(cbind(SHOW, NO\_SHOW) ~ Media\_distanza\_km+Media\_età+media\_tempo\_attesa+Media\_tariffa\_prestaz ,data=dati\_num\_filt ,k=1:4, model = FLXMRglm(family = "binomial"),

nrep = 10,verbose = TRUE, drop = F, unique = F))