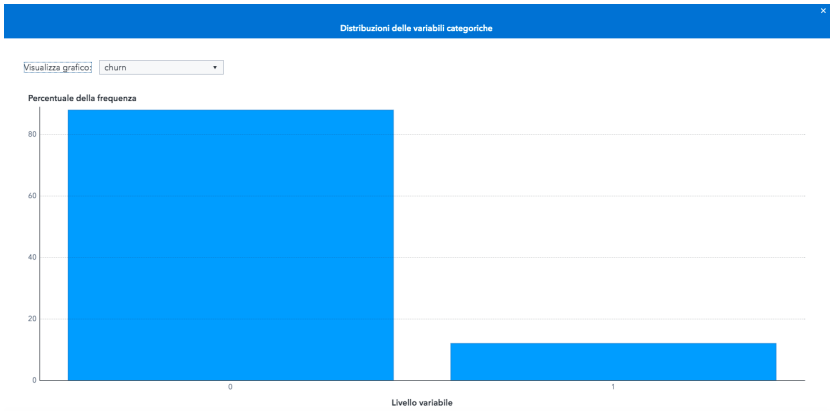
1 Progetto Finale- Gabriele D'Andrea 1744433

1.1 Analisi Preliminare

Selezionare il dataset COMMSDATA
Opzioni dell'assistente: lasciare i valori di default Partizionare 70-30-0 (quindi senza test)
Fissare CHURN come target
Call Center Category 1 come testo
Call Center Category 2 Call Center Issue level 1 Call Center Issue level 2 rifiutata come testo rifiutata
Quindi in totale le variabili di testo saranno
- Call Center Category 1
- Call Center Issue level 1
- Survey Verbatim

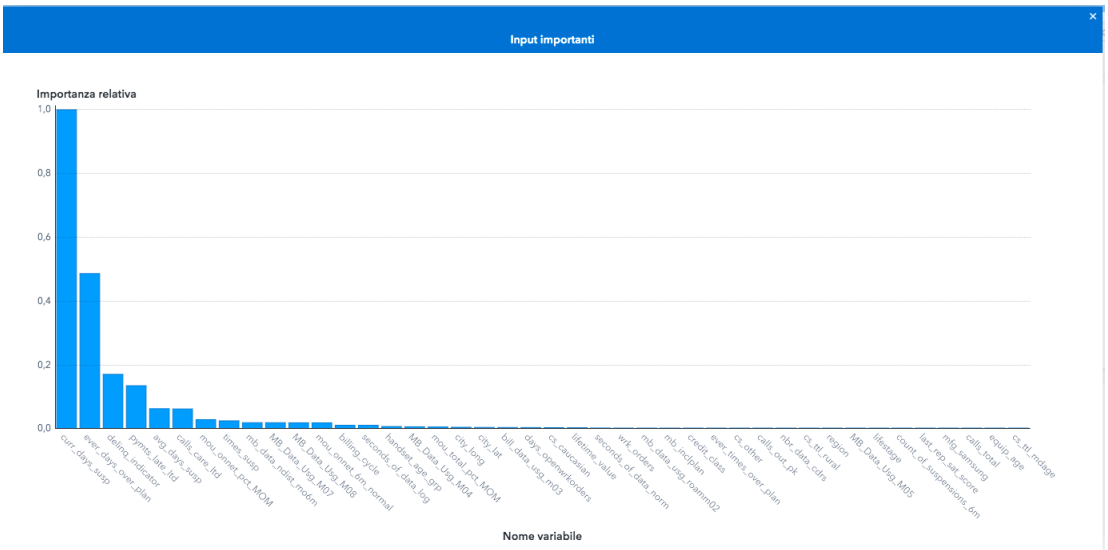
Utilizzare un nodo esplora dati (mettere num variabili =150)

Usando il nodo esplorazione dati, si ottiene il seguente istogramma



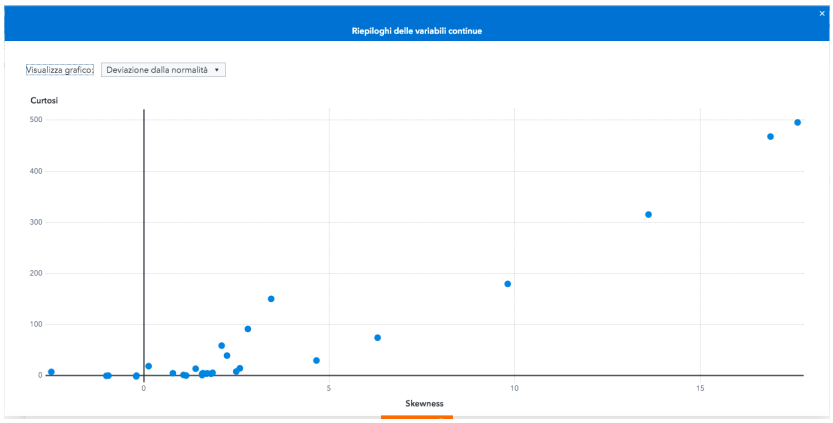
che riporta la frequenza con la quale la variabile churn, che rappresenta l'abbandono o meno del cliente, assume valore 0 o 1 e vale, rispettivamente, 88 e 12 per cento.

Nel grafico successivo è riportata l'importanza relativa di ogni singola variabile nella previsione della variabile churn



Le variabili più rilevanti sono curr-days-susp e ever-days-over-plan con un' importanza relativa, rispettivamente, di 1 e 0.5. Quindi l'abbandono è fortemente influenzato dal numero di giorni di sospensione e dal numero di giorni oltre il piano previsto.

Per quanto riguarda le variabili che presentano una asimmetria maggiore di 5, il seguente grafico

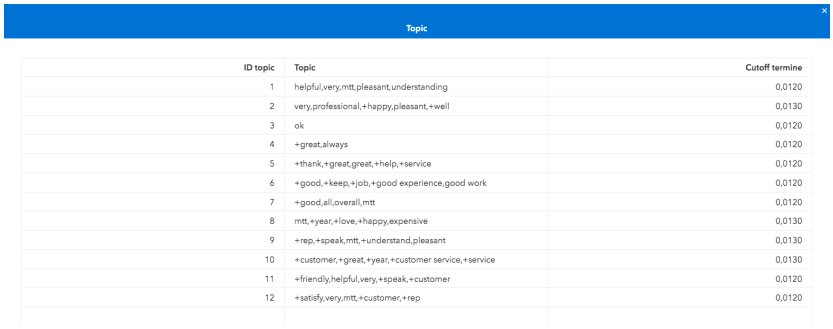


ci dice che le variabili in questione sono: days-openwrkorders, MB-Data-Usg-M04, MB-Data-Usg-M05, MB-Data-Usg-M06, MB-Data-Usg-M08 e MB-Data-Usg-M07.

In base ai risultati - Trasformare (log) le variabili con skewness > 5
Bill-data-usg-m09, MB-Data-Usg-M04, MB-Data-Usg-M06,
MB-Data-Usg-M07, MB-Data-Usg-M08
- Mettere tutti i valori negativi a 0 (solo per le variabili di input!)
- Eliminare le seguenti variabili con valori mancanti:
mb-data-ndist-mo6m, mou-onnet-pct-MOM, seconds-of-data-log

Imputare tutte le altre variabili con valori mancanti

Aggiungere un nodo di testo (max 12 topics)
Il seguente grafico mostra i topic selezionati



Topic

ID topic	Topic	Cutoff termine
1	helpful,very,mtt,pleasant,understanding	0,0120
2	very,professional,+happy,pleasant,+well	0,0130
3	ok	0,0120
4	+great,always	0,0120
5	+thank,+great,great,+help,+service	0,0120
6	+good,+keep,+job,+good experience,good work	0,0120
7	+good,all,overall,mtt	0,0120
8	mtt,+year,+love,+happy,expensive	0,0130
9	+rep,+speak,mtt,+understand,pleasant	0,0130
10	+customer,+great,+year,+customer service,+service	0,0130
11	+friendly,helpful,very,+speak,+customer	0,0120
12	+satisfy,very,mtt,+customer,+rep	0,0120

1.2 Regressione Logistica

La Regressione Logistica appartiene alla classe dei modelli lineari generalizzati, la cui famiglia esponenziale è la bernulliana che è caratterizzata dalla terna

$$\{Y^N, f(y|\pi), \pi \in \Pi\}$$

dove

- $Y^N = \{0,1\}^N$ è lo spazio campionario ;
- la legge di probabilità è la legge di una bernulliana

$$f(y_i, \pi_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

e per l'ipotesi di indipendenza

$$f(y|\pi) = \prod_{i=1}^{N} f(y_i, \pi_i);$$

- $\Pi = [0,1]$ spazio dei parametri.

La regressione logistica non stima direttamente la risposta, ma stima le probabilità delle risposte condizionatamente ai dati osservati. Nella regressione logistica si assume che

$$log(\frac{\pi}{1-\pi}) = \vartheta^T x$$

da cui

$$\pi = \sigma(\vartheta^T x) = \frac{e^{\vartheta^T x}}{1+e^{\vartheta^T x}}$$

quindi le stime saranno date da

$$\hat{y}_i = \sigma(\vartheta^T x_i)$$

. La regola decisionale che assegna la classe ad una data unità è

$$d(x) = \begin{cases} 1 & \text{se } \sigma(\vartheta^T x) > s \\ 0 & \text{altrimenti} \end{cases}$$

dove s è una soglia tra [0,1]. L'obiettivo diventa quindi stimare i parametri incogniti non osservabili.
La funzione di perdita che si considera è la log-verosimiglianza cambiata di segno, detta anche cross-entropy, data da

$$J(\vartheta) = -\sum_{i=1}^{N} [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$$

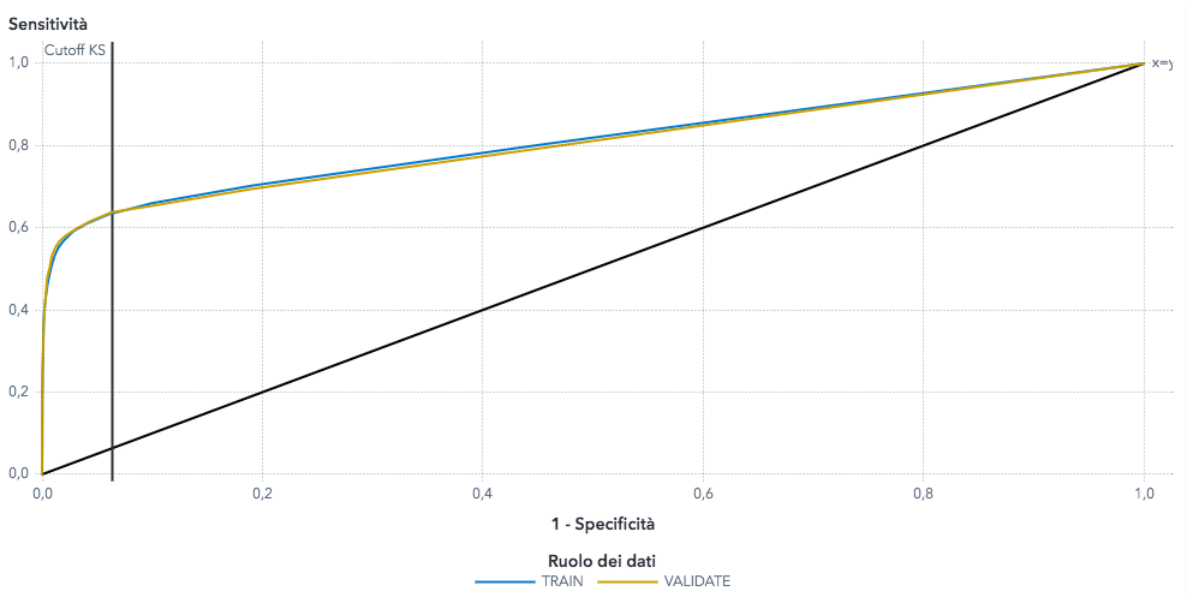
e il metodo di ottimizzazione è la discesa del gradiente o varianti

$$\vartheta_{k+1} = \vartheta_k - \alpha_k \nabla J(\vartheta_k)$$

con α_k parametro non negativo detto learning rate.

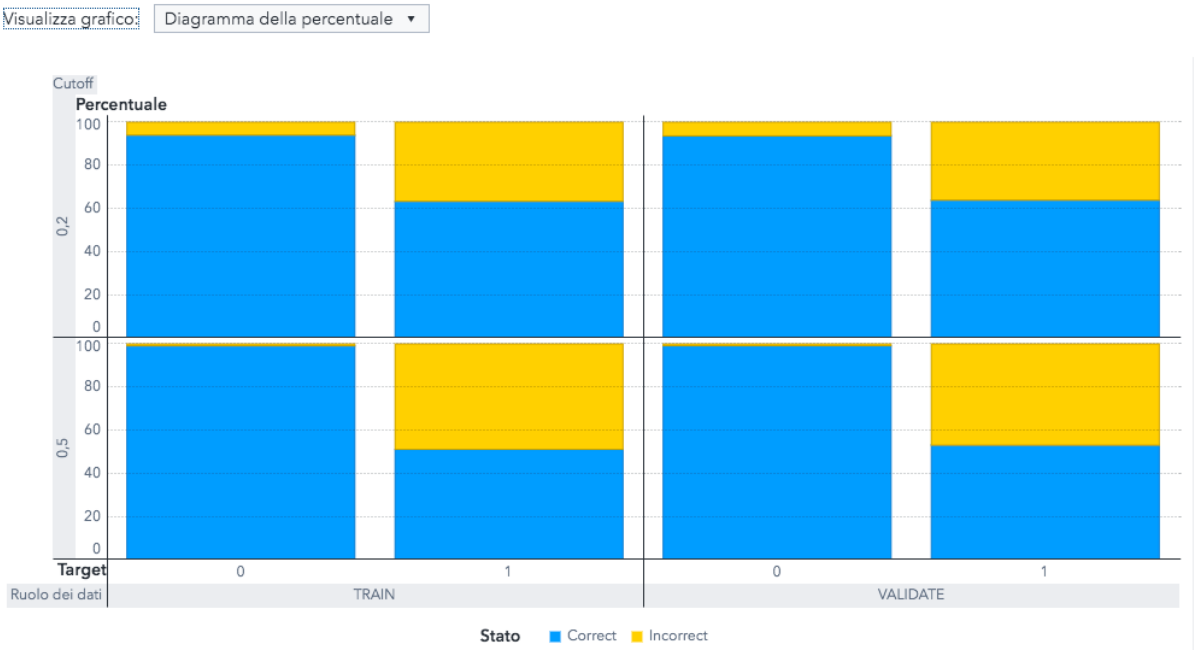
Dalla tabella delle statistiche di bontà del modello abbiamo che le più rilevanti sono l'errore di validate che assume valore 0.0673 e la statistica di KS con un valore pressoché identico sia per il training sia per il validate ossia 0.57. Il più interessante è l'errore di validate poiché ci dà informazioni sulla capacità predittiva del modello su dati che non conosce e che non ha mai visto prima.

Per quanto riguarda il grafico della curva ROC



il cutoff migliore, ossia la soglia per cui è massima la distanza tra la curva del modello logistico e quella del modello casuale rappresentata dalla bisettrice, è 0.2 con un 1-Specificità pari a 0.063 e una sensitività del 0.638. La sensitività è la percentuale di positivi correttamente classificati, mentre la specificità è la percentuale di negativi correttamente classificati, quindi , in corrispondenza di 0.2, abbiamo che il classificatore classifica correttamente circa il 64 per cento dei positivi e classifica erroneamente circa il 6 per cento dei negativi.

Per quanto riguarda la matrice di confusione, la situazione è la seguente



per il cutoff ottimale abbiamo che la percentuale dei falsi positivi nel training e nel validate è rispettivamente 37% e 36%, mentre il numero di falsi negativi è del circa 6% per entrambi i dataset.

1.3 Gradient Boosting

Il Gradient Boosting è un tipo di classificatore ensemble, ossia combina più alberi di decisione identici e indipendenti al fine di ottenere un classificatore più performante. A differenza del bagging e random forest, dove gli alberi sono costruiti in parallelo, il gradient boosting costruisce dli alberi in sequenza in modo da ridurre sia la varianza sia la distorsione. Si chiama così poiché affronta i problema del boosting come un problema di ottimizzazione e lo risolve applicando il gradient descent. Sempre a differenza del bagging e del boosting, l'ideale è che glia alberi siano poco profondi a causa del numero di iterazioni. Lo schema algoritmico è il seguente

- Siano $\hat{f}(x) = 0$ e $r_i = y_i$.
- for b=1,. . . ,B repeat
  - Stima l'albero $\hat{f}^b(x)$ con d divisioni sui dati di training;
  - for i=1,. . . ,N
    - r_i<- $r_i - \lambda \hat{f}^b(x_i)$.
- $\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$.

Anzitutto, il grafico di statistiche di bontà del modello generato con la sintonizzazione automatica

Per prima cosa eseguendo il modello con i parametri di default si ha l'indice di errore di classificazione pari a 0.0579, mentre l'indice KS risulta essere pari a 0.595 ci dice che l'errore di validate è pari a circa 0.058, mentre la statistica di KS per il validate è pari a circa 0.59.

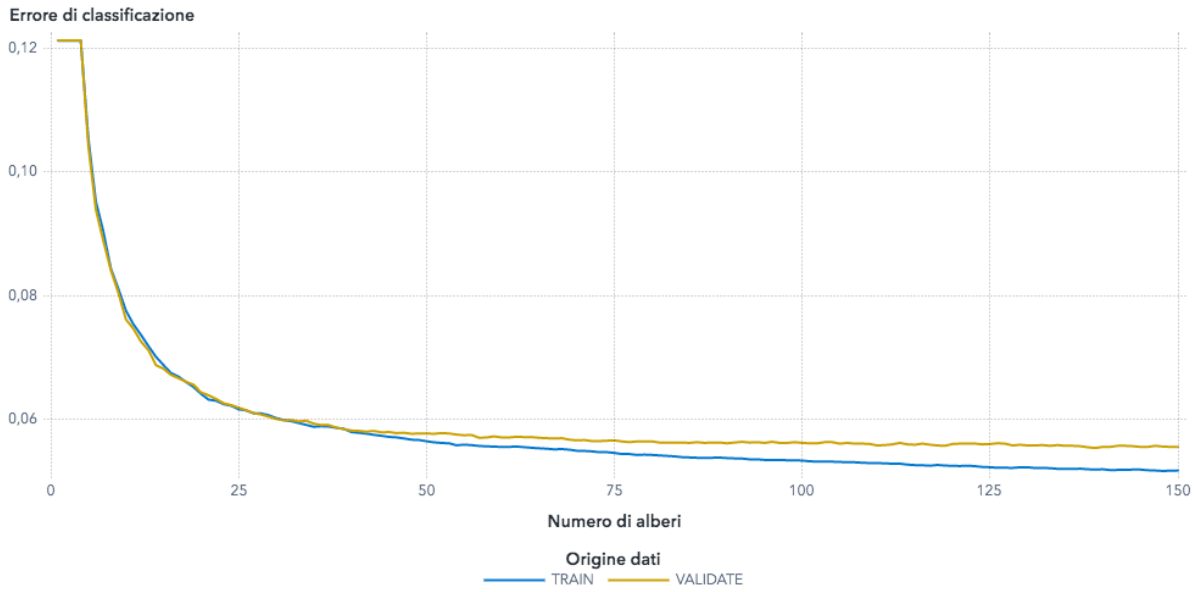
Applicando la sintonizzazione automatica, otteniamo la seguente situazione ottimale

Configurazione migliore della sintonizzazione automatica

Parametro	Valore
Valutazione	43
Num. di variabili per prova	23
Tasso di apprendimento	0,12
Tasso di campionamento	1
Lasso	6,66666667
Ridge	7,22222222
Num. di raggruppamenti	56
Livelli massimi albero	6
Dimensione foglia	100
Kolmogorov-Smirnov	0,6073146967

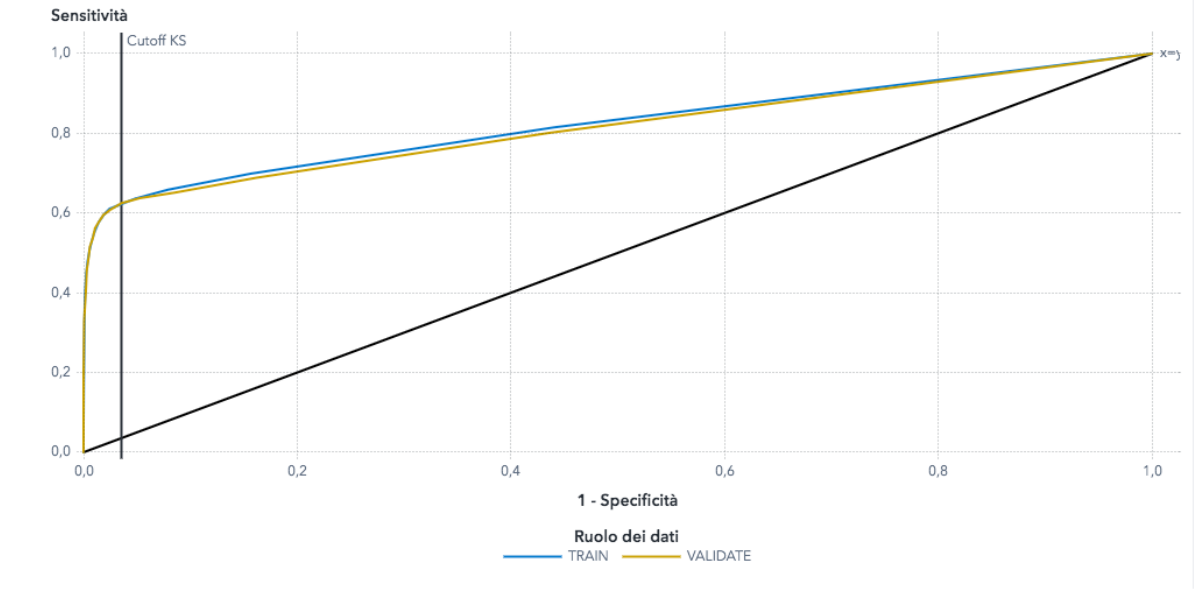
che riporta i parametri di tuning del modello, come ad esempio il tasso di apprendimento che controlla la velocità del modello e il valore delle costanti di regolarizzazione.

Come si vede dal seguente grafico



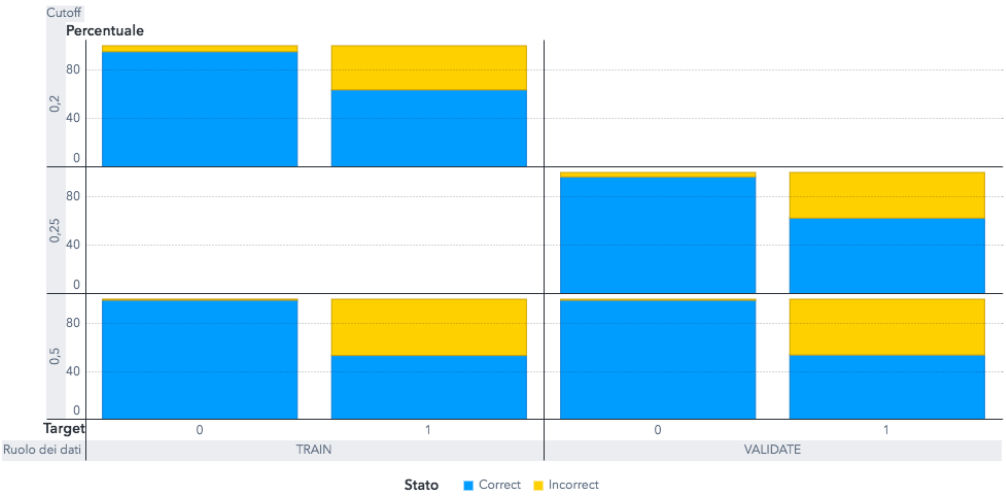
il modello, al crescere del numero di alberi, presenta un leggero overfitting poiché notiamo che mentre l'errore del training si abbassa sempre di più, quello del validate si mantiene stabile e questa differenza potrebbe diventare sempre più marcata. Per evitare l'overfitting l'idea è di modificare manualmente i parametri del modello; per ottenere una situazione più stabile portiamo il numero di alberi a 400, scegliamo come tasso di apprendimento il valore 0.08, modifichiamo la dimensione minima della foglia a 200, lasciamo invariati i coefficienti di regolarizzazione e si pone a 40 il numero di iterazioni consecutive per la stagnazione. A seguito di queste modifiche otteniamo un errore di classificazione per il training pari a 0.0635 e per il validate pari a 0.0637, quindi maggiore rispetto alla sintonizzazione automatica, ma otteniamo un andamento della curva dell'errore del validate molto più vicina a quella del training.

La curva ROC che otteniamo a seguito di queste modifiche è la seguente



il cutoff ottimale è 0.25 con una sensitività del 0.6251 e una 1-specificità del 0.0358 per il validate.

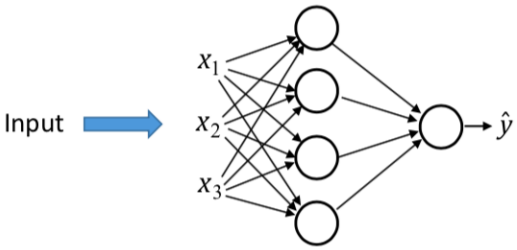
Per quanto riguarda la matrice di confusione, la situazione è la seguente



per il cutoff ottimale del validate abbiamo che la percentuale dei falsi positivi è 37,5%, mentre il numero di falsi negativi è del 3,5%.

1.4 Rete Neurale

La rete neurale è un modello statistico ispirato alle reti neurali biologiche usato per modellare relazione di tipo non lineare tra i dati. Tale modello si presenta come un grafo orientato pesato ed è caratterizzato da uno strato di input che riceve i dati di ingresso, uno o più strati nascosti che trasformano i dati in ingresso attraverso delle funzioni dette di attivazione, e uno strato di output che restituisce la stima della risposta. Un esempio di rete con un livello nascosto è rappresentata in questo modo



Ogni neurone è caratterizzato da un parametro detto soglia e da una funzione detta funzione di attivazione. Nel caso di una rete molto semplice senza livelli nascosti e con un solo neurone nello strato di output (percettrone), l'idea è che il "segnale" viene inviato lungo la rete se la somma pesata degli input supera la soglia assegnata al neurone di output, in formule

$$\hat{y} = \begin{cases} 1 & \text{se } w^T x + b > 0 \\ 0 & \text{altrimenti} \end{cases}$$

dove la funzione così definita è la funzione di Heaviside che possiamo indicare come

$$\hat{y} = H(w^T x + b)$$

Nell'ambito della classificazione binaria significa che un'unità viene assegnata alla prima classe se la quest'ultima si trova sopra l'iperpiano H=$\{x \in R | w^T x + b\}$. Dal momento che la funzione a gradino non è differenziabile ovunque, si preferisce usare funzioni continue e derivabili tra [0,1], come nel caso della logistica, o tra [-1,1], come nel caso della tangente iperbolica.
L'obiettivo è di determinare i parametri della rete, ossia i pesi e le soglie, in modo da minimizzare una funzione di perdita; le funzioni di perdita più usate sono la somma degli errori al quadrato per i problemi di regressione

$$L(W, b) = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

e la cross-entropy per i problemi di classificazione

$$J = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log \hat{y}_{ik}.$$

Il metodo di ottimizzazione principale è la discesa del gradiente in combinazione con il backpropagation, chiamato così poiché si comincia stimando i parametri del livello finale per poi tornare indietro e "propagare" all'indietro l'errore.

Dalla sintonizzazione automatica otteniamo la seguente situazione ottimale

Parametro	Valore
Valutazione	40
Layer nascosti	1
Neuroni in layer nascosto 1	88
Neuroni in layer nascosto 2	0
Regolarizzazione L1	8,77427E-6
Regolarizzazione L2	7,82284E-8
Kolmogorov-Smirnov	0,5891245915

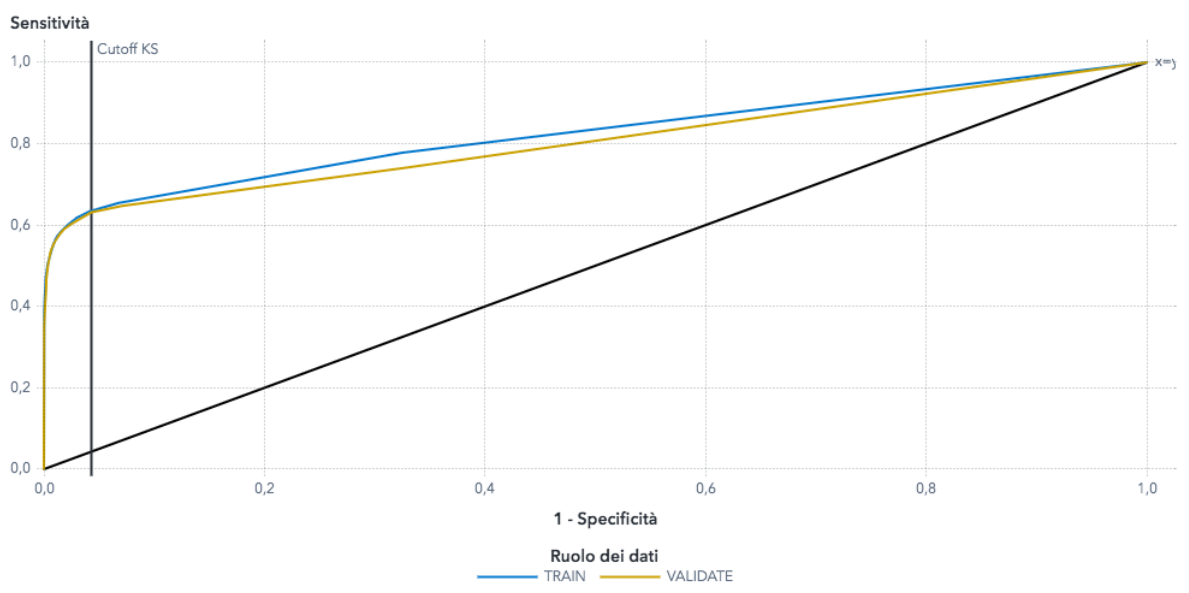
e quindi la rete ottimale è formata da un solo strato nascosto con 88 neuroni e con una leggera regolarizzazione.

Per quanto riguarda l'errore commesso dalla rete dal seguente grafico

Nome t...	Ruolo d...	Indicato...	Partizio...	Somma ...	Averag...	Divisor...	Root Av...	Errore d...	Perdita l...	KS (You...	Area so...	Coeffici...
churn	TRAIN	1	1	39.590	0,0542	39.590	0,2328	0,0621	0,2126	0,5922	0,8254	0,6508
churn	VALIDATE	0	0	16.967	0,0550	16.967	0,2346	0,0620	0,2181	0,5885	0,8061	0,6121

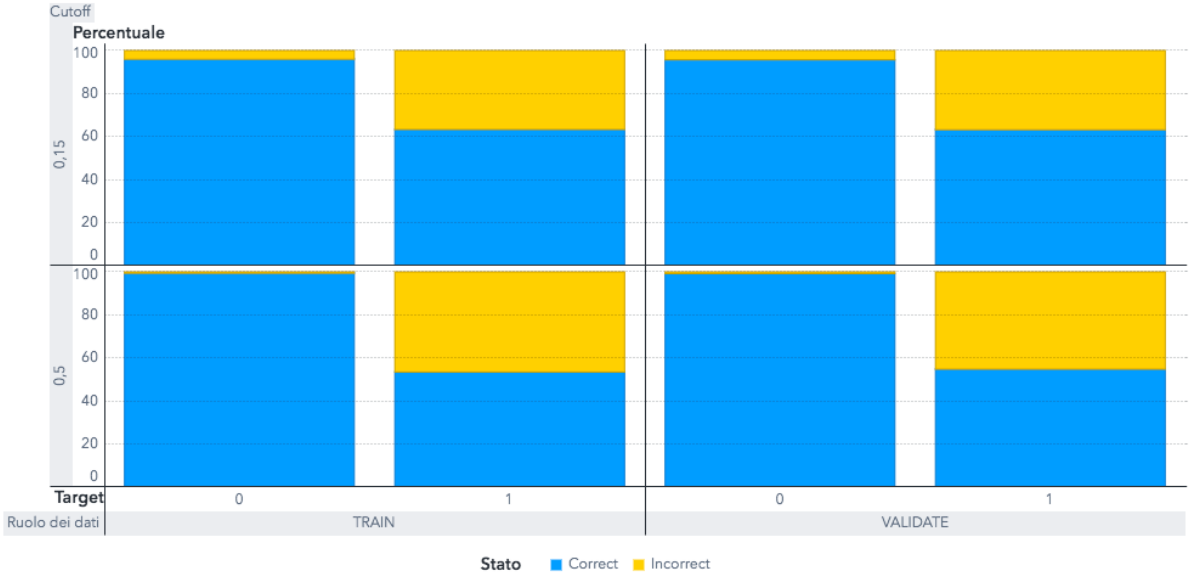
notiamo che la rete commette praticamente lo stesso errore, pari a circa 0.0620, sia per i dati di training sia per quelli di validate, quindi il modello non presenta overfitting e quindi non bisogna modificare i parametri della rete.

La curva ROC che otteniamo a seguito di queste modifiche è la seguente



il cutoff ottimale è 0.1 con una sensitività del 0.6319 e una 1-specificità del 0.0343 per il validate.

Per quanto riguarda la matrice di confusione, la situazione è la seguente



per il cutoff ottimale del validate abbiamo che la percentuale dei falsi positivi è 37%, mentre il numero di falsi negativi è del 4,3%.

1.5 Confronto tra Modelli

Alla fine, dal grafico del confronto tra modelli il vincitore è la rete neurale poiché presenta un indice KS maggiore e un errore di classificazione più basso nel validate, mentre il peggiore è la regressione logistica poiché tra i tre modelli è il più semplice e non riesce a cogliere alcune relazioni non lineari tra le risposte e i regressori.

Per completezza, alla fine viene riportato la pipeline completa