Progetto Data Science Lab: Antichurn

Julia Lan Bui Xuan[1], Cosimo Simone Farallo[1], Michele Salvaterra[1], Luca Sammarini[1], Eugenio Tarolli Bramè[1]

Sommario
L'obiettivo di questo progetto mira a predire se un cliente abbandonerà o meno il servizio, partendo da un dataset antichurn. Il problema è stato studiato nel contesto di classificazione binaria e affrontato attraverso tecniche di feature selection, formal concept analysis, cross validation, undersampling e oversampling, al fine di risolvere problematiche relative alla dimensionalità dei dati e delle classi sbilanciate. Modelli come il Random Forest, la Logistic Regression, il Multi-Layer Perceptron e il Gaussian Naive Bayes sono stati implementati; le performance sono state valutate in termini di Accuracy, Precision, Recall, F-measure e AUC. Infine, è stato applicato anche il metodo dei quantili come ulteriore valutazione del modello più performante.

Parole chiave
Modelli di classificazione — FCA — Cross Validation — Class Imbalance — Feature Selection

[1] *CDLM in Data Science, Università degli Studi di Milano-Bicocca*

Indice

1. Introduzione

Ogni giorno moltissimi clienti effettuano acquisti, sottoscrivono programmi fedeltà, scoprono nuove realtà ed abbandonano prodotti o servizi dei quali hanno usufruito fino a quel momento. Una delle migliori strategie per fidelizzare i propri clienti è rappresentata dalla capacità di creare un Customer Service in grado di valorizzare e personalizzare la Customer Experience. Secondo un recente studio di SAS e Loyalty360, il 68% del business proviene da clienti esistenti. Questo significa che, oltre a ricercarne di nuovi, la vera sfida per un'azienda consiste soprattutto nel riuscire a non perdere i clienti già acquisiti (Customer Retention). Un indicatore numerico di questo fenomeno è il churn rate, il quale rappresenta il tasso con cui i clienti cessano il rapporto con un'azienda. Si tratta di un indicatore critico per misurare la salute di un business e aiuta a ottenere il benchmark della Customer Satisfaction, ovvero il grado di soddisfazione dei clienti. Attraverso i dati a disposizione, si vogliono estrarre le informazioni utili al fine di predire il comportamento di un cliente. Ciò può rivelarsi utile all'azienda in questione, per adottare strategie mirate al fine di non perdere clienti. L'obiettivo principale di questo progetto, dunque, consiste nel trovare un modello di classificazione, capace di predire se un cliente abbandonerà o meno un particolare servizio.

2. Struttura del dataset

Il dataset preso in considerazione rappresenta un "cluster" di clienti, con varie informazioni e una variabile target, che indica se il cliente ha abbandonato il servizio oppure no. Nello specifico le variabili prese in considerazione sono le seguenti:

1. ***external_id***, identificativo alfanumerico dell'utente;
2. ***how_many_ok_urls***, numero di siti che il motore semantico è riuscito ad analizzare espresso tramite valori interi;
3. ***how_many_ko_urls***, numero di siti che il motore semantico non è riuscito ad analizzare espresso tramite valori interi;

4-10. ***os_*****, sistema operativo utilizzato, in forma dummy;

11-20. ***browser_*****, browser utilizzato, in forma dummy;

21-24. ***feriale_*****, frazione di siti visitati in un determinato momento della giornata di un giorno lavorativo (sabato escluso), normalizzata a 1;

25-28. ***weekend_*****, frazione di siti visitati in un determinato momento della giornata di un giorno di weekend, normalizzata a 1;

29-37. ***L***_*****, frazione di pagine web contenenti un testo della lunghezza indicata, in numero di parole, normalizzata a 1;

38-63. ***categories_*****, frazione di siti web visitati appartenenti alla categoria indicata, espressa in percentuale;

64-87. ***admants_*****, categorie semantiche ad hoc, i valori risultano tutti nulli;

88-91. ***CINEMA***, pacchetto di canali Sky scelto dall'utente, in forma dummy;

92-96. ***FLG_*****, abbonamento Sky scelto dell'utente, in forma dummy;

97-100. ***STB_*****, decoder utilizzato dall'utente, in forma dummy;

101. ***DATA_RIF***, data di rilevazione dell'osservazione;
102. ***Pdisc***, variabile target, riguarda gli utenti che hanno disdetto l'abbonamento, in forma dicotomica.

Il dataset è composto da 330,586 righe e 102 colonne. Nell'elenco di attributi sopra riportato è stato deciso di considerarne solo una per categoria affinché fosse più immediato e meno ridondante l'interpretazione del dataset.

3. Metodologie e analisi

3.1 Riduzione delle dimensioni del dataset

Tramite una prima analisi descrittiva del dataset è stato possibile osservare come questo fosse privo di dati mancanti e quindi non è stato necessario applicare alcuna tecnica di risoluzione al problema. Ciò ci ha permesso di considerare il dataset completo di tutti i suoi elementi.

Si è rivelato però necessario rimuovere alcune colonne poiché presentavano solo valori nulli; le colonne infatti non erano una fonte di informazione aggiuntiva.

Le colonne eliminate sono le seguenti: *os_bsd*, *os_osx*, *browser_android*, *browser_chromium*, *browser_edge*, *browser_ie*, *categories_emotions* e tutte le categorie *admants_**.

Il nuovo dataset contiene dunque 71 colonne.

Inoltre, al fine di non perdere informazioni, è stato necessario unire le righe con lo stesso *external_id*, in quanto riferite allo stesso utente. È stato notato che gli stessi utenti presentino sempre il medesimo valore della variabile target, sottolineando coerenza per quanto riguarda la scelta di abbandonare il servizio o meno, anche in tempi di rilevazione differenti. Ciò, però, non vale per le variabili esplicative ad essi associate. Pertanto, per determinate variabili numeriche, è stato scelto di unire le informazioni attraverso calcoli come media o somma e per le variabili dicotomiche si è mantenuta la classe positiva (1) laddove apparisse almeno una volta. Dopo l'unione il dataset contiene 326,414 righe.

3.2 Formal Concept Analysis

Prima di iniziare ad implementare ed analizzare i classificatori è stata applicata la tecnica della Formal Concept Analysis (FCA). Si tratta di un metodo per derivare una gerarchia di concetti da una collezione di oggetti e dalle loro proprietà. Ogni concetto della gerarchia rappresenta gli oggetti con il medesimo insieme di proprietà e ogni sotto-concetto della gerarchia rappresenta un sottoinsieme degli oggetti nei concetti che lo precedono.

3.3 Cross validation

Le performance dei modelli di classificazione dipendono dalle tecniche usate per dividere il dataset in training set e in test set. È stata applicata la tecnica di k-fold cross validation con k pari a 3.

Questa tecnica presenta una distorsione minore rispetto alle procedure di Standard Holdout e di Iterated Holdout, perché è in grado di ridurre l'impatto dei valori anomali. Inoltre, garantisce che ogni istanza del dataset sia inclusa nel training set lo stesso numero di volte e nel test set esattamente una volta. Il dataset è suddiviso in k-fold (k sottoinsiemi collettivamente esaustivi e mutualmente esclusivi), contenenti lo stesso numero di record. Le iterazioni vengono eseguite utilizzando un fold diverso come test set ad ogni iterazione. Infine, le misure di performance si ottengono calcolando la media di tutte le misure calcolate durante le k iterazioni.

3.4 Modelli di classificazione

In questo progetto sono state implementate diverse tecniche di classificazione allo scopo di trovare la più appropriata al problema in questione. Sono stati applicati quattro modelli appartenenti a quattro diverse categorie:

- **Modelli Euristici**: seguendo un approccio ad albero, questi algoritmi consentono di costruire soluzioni interpretabili. In particolare è stato utilizzato il Random Forest;
- **Modelli base di regressione**: Logistic Regression, usata per risolvere problemi di classificazione binaria. Inoltre, è applicabile ad attributi continui e, con una certa accuratezza, anche ad attributi nominali;
- **Modelli di separazione**: Multi-Layer Perceptron, un tipo di rete neurale artificiale organizzata in più strati all'interno della quale l'informazione fluisce solo dallo strato di input a quello di output (rete feedforward);
- **Modelli probabilistici**: Gaussian Naive Bayes, algoritmo basato sul teorema di Bayes e ipotesi di partenza molto semplificate. In particolar modo, si considerano indipendenti tra loro le varie caratteristiche (features) del modello.

3.5 Class Imbalance Problem

Il problema della Class Imbalance si verifica quando una classe ha una frequenza estremamente elevata rispetto a un'altra. Nel caso specifico del problema in questione, la classe positiva è rappresentata solamente dal 3% dei valori. Si è quindi nella situazione di un grande sbilanciamento tra le due classi.

La presenza di questo problema può avere un effetto significativo sulle prestazioni del modello di classificazione: se il dataset è fortemente sbilanciato, il modello tende a comportarsi come la regola ZeroR. Ciò significa che l'algoritmo prevede il valore della classe con il maggior numero di osservazioni nel training set. Per risolvere questo problema è stato utilizzato un approccio basato sul campionamento. È importante sottolineare che è proprio la classe minoritaria a rappresentare la classe di interesse, ovvero quella dei churn. Le diverse soluzioni che eseguono un lavoro di pre-processing sui dati, possono essere classificate in due categorie:

- **Oversampling**, ovvero la tecnica di sovra-campionamento, mira a bilanciare la distribuzione di classe attraverso la replicazione casuale di osservazioni di classe minoritaria o la sintetizzazione di nuove osservazioni;
- **Undersampling**, ovvero la tecnica di sotto-campionamento, mira a bilanciare la distribuzione di classe attraverso l'eliminazione casuale di osservazioni appartenenti alla classe maggioritaria.

È importante sottolineare che conviene non eccedere con l'oversampling, in quanto si rischia di far allontanare molto i dati dalla realtà osservata. Attraverso il sotto-campionamento invece, si ha il rischio di scartare dati potenzialmente utili che potrebbero essere importanti per il processo di apprendimento.

Entrambi i metodi sono stati usati per risolvere il problema delle classi sbilanciate. Per evitare di basarsi su approcci di selezione randomici, è stata applicata la Synthetic Minority Oversampling Technique (SMOTE), in base alla quale si generano nuovi esempi che combinano le funzionalità del caso di destinazione con le funzionalità dei relativi vicini. Questo approccio, quindi, aumenta le caratteristiche disponibili per ciascuna classe e rende i campioni più generali. Per il sotto-campionamento invece, è stata scelta la tecnica basata sui collegamenti Tomek; le osservazioni della classe maggioritaria che hanno i valori più bassi della distanza euclidea con la classe minoritaria vengono rimossi, in quanto la loro distinzione è ambigua.

3.6 Feature selection

È possibile ottenere una misura di prestazione migliore del modello di classificazione eliminando dell'informazione non rilevante nell'input. Esistono situazioni in cui usare tutti gli attributi presenti nel dataset è poco efficiente; uno dei metodi più comuni e utilizzati per ridurre la dimensionalità consiste nel selezionare un sottoinsieme degli attributi, ovvero fare Feature Selection. Nel caso specifico di questo progetto sono state scelte 11 colonne; ciò è stato possibile grazie alla libreria di *Sklearn*, che tramite la funzione *SelectKBest*, in base ai valori di score relativi alla F_1 measure sceglie gli attributi più rilevanti studiando la dipendenza lineare tra gli stessi. Si è osservato che le 11 features selezionate presentano uno score pari al 99% dello score totale.

È stato deciso di utilizzare questo metodo e non la tecnica della PCA (Principal Component Analysis) in quanto, avendo molte variabili categoriche, quest'ultima risulta meno ottimale.

3.7 Misure di performance

Per confrontare le prestazioni dei modelli implementati sono state utilizzate diverse misure. In particolare sono state calcolate Accuracy, Precision, Recall, F_1 measure e AUC:

- **Accuracy**: misura la capacità del modello di fornire classificazioni affidabili su nuovi record; normalmente è affidabile quando le classi sono equamente bilanciate rispetto alla variabile target. In questo caso specifico, viene analizzato un set di dati sbilanciato; risulta quindi una misura non ottimale;
- **Precision**: definita come il numero di veri positivi diviso il numero totale di elementi etichettati come appartenenti alla classe positiva. Essa misura quanto è accurata la predizione del classificatore rispetto a una specifica classe;
- **Recall**: è definita come il numero di veri positivi diviso il numero totale di elementi che effettivamente appartengono alla classe positiva. Essa misura la capacità del modello di identificare una specifica classe;
- **F_1 measure**: è definita come la media armonica tra Recall e Precision, mantenendo così un bilancio fra queste due misure. Per questo motivo la F-measure è più appropriata nel caso di dataset con classi sbilanciate, il cui valore alto segnala la capacità del modello di classificare correttamente la maggior parte dei dati;
- **ROC curve**: è un diagramma grafico creato tracciando il tasso di veri positivi rispetto al tasso di falsi positivi a varie impostazioni di soglia. Rappresenta la performance di un classificatore senza considerare la distribuzione della classe e quindi viene utilizzato per confrontare diversi modelli tenendo conto dell'area sotto la curva (AUC).

Essendo necessario rilevare i clienti che hanno intenzione di abbandonare il servizio, risulta più costoso classificare un churner come non churner, piuttosto che viceversa. In questo caso, risulta più importante avere un alto valore di Recall che un alto valore di Precision. Per non abbandonare però completamente il punteggio di Precision rilevato, si è deciso di utilizzare la F_1 measure, che in un dataset con classi sbilanciate è la misura ottimale.

3.8 Metodo dei quantili

Il dataset risulta fortemente sbilanciato e non contiene informazioni su un cambiamento delle abitudini del consumatore, che lo porterebbero ad abbandonare o disdire un determinato abbonamento, quanto piuttosto informazioni sugli interessi del consumatore, che tecnicamente lo porterebbero a "essere un churner ontologicamente". È chiaro che la correlazione tra le variabili esplicative e la variabile target risulta molto debole; è quindi intrinsecamente complesso trovare un modello che performi in modo ottimale con queste premesse, cioè che sia in grado di classificare correttamente ogni osservazione. Risulterebbe quindi già sufficiente se il modello riuscisse a predire l'andamento generale della probabilità di abbandono degli utenti. Per studiare questo si è deciso di utilizzare il metodo di seguito descritto.

Prendendo in considerazione il modello che performa meglio in termini di F_1 measure, si calcola la probabilità per ogni osservazione del test set di essere classificato come churn. Dopo aver ordinato il dataset in ordine crescente di probabilità predetta dal modello, lo si suddivide in 5 parti uguali e viene calcolata la percentuale di churners presenti all'interno di ogni gruppo. Se questa percentuale ricade tra i valori di probabilità degli elementi agli estremi del gruppo, si può ritenere che il classificatore è in grado di prevedere l'andamento della probabilità di churn presente nei dati.

4. Risultati

L'obiettivo principale di questo studio è quello di predire la volontà degli utenti di abbandonare o rimanere legati ad un determinato abbonamento tramite l'utilizzo di variabili esplicative, quali ad esempio relative al tipo di browser utilizzato, al sistema operativo impiegato e alle pubblicità osservate.

Per ottenere informazioni riguardo a quanto detto si è ricorsi all'utilizzo di quattro modelli di classificazione, ovvero: Random Forest, Logistic Regression, Multi-Layer-Perceptron e Gaussian Naive Bayes.

4.1 FCA

La dimensionalità dei dati e gli strumenti limitati non hanno reso possibile effettuare l'analisi sull'intero dataset, a causa dell'elevato costo computazionale. È stato dunque estratto un campione costituito da 1/10,000 del dataset iniziale (ottenuto sia tramite un campionamento stratificato sia uno casuale) al fine di ottenere informazioni riguardanti la gerarchia fra le variabili, anche se ciò ha comportato una perdita di informazioni. Il risultato ottenuto è rappresentato dalle figure 1 e 2.

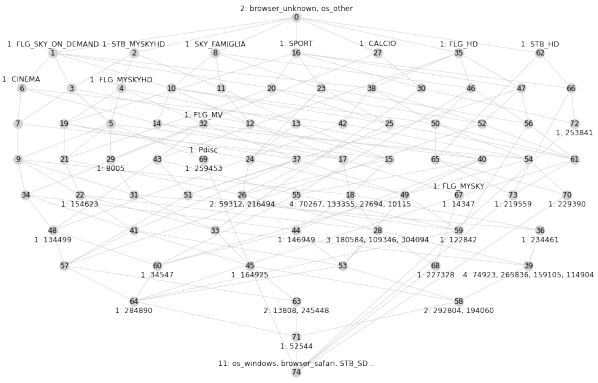

Figura 1. FCA con campionamento stratificato.

4.2 Modelli a confronto con dataset sbilanciato

Come primo step dello studio eseguito, sono stati implementati i modelli sopra citati senza ricorrere alle metodologie della cross-validation e della feature selection, affinché fosse possibile osservare le performance dei modelli definiti come "clean". Il dataset, inoltre, presenta ancora lo sbilanciamento delle due classi. Solamente in seguito verranno applicati i metodi di oversampling e undersampling, per poter eseguire un confronto fra i due.

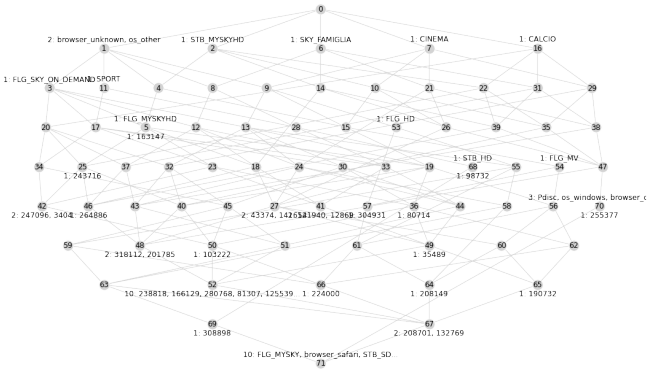

Figura 2. FCA con campionamento casuale.

I risultati di questa prima analisi sono riportati nella tabella 1.

Model	R	P	F-m	A
RF	0.00	0.19	0.01	0.97
Logistic	0.00	0.00	0.00	0.97
MLP	0.00	0.00	0.00	0.97
GNB	0.43	0.07	0.12	0.81

Tabella 1. Modelli a confronto - dataset sbilanciato. R = Recall, P = Precision, F-m = F_1 measure, A = Accuracy, AUC = Area Under Curve.

Il modello migliore è Gaussian Naive Bayes, con una F_1 measure pari a 0.12 rispetto ai valori 0.00 e 0.01 degli altri modelli. Ciò può essere giustificato dal fatto che questo modello si basa sull'assunzione di indipendenza condizionale delle classi. Infatti, lavora bene in caso di "rumore" nei dati, tendendo a non considerare gli attributi irrilevanti e mantenendo la fase di training del modello molto più semplice rispetto agli altri algoritmi.

Si nota come il forte sbilanciamento delle due classi in questione influenzi il rendimento e le performance dei modelli di classificazione. I valori della Recall, infatti, sono molto bassi, stando a significare il fallimento dei modelli a predire la classe minoritaria.

4.3 Modelli a confronto con dataset bilanciato

Successivamente all'analisi dei modelli "clean", vengono applicate le tecniche e le metodologie sopra descritte; in particolare si procede con il bilanciamento delle classi, l'esecuzione della feature selection e della cross validation. Attraverso la feature selection, le variabili

selezionate sono le seguenti:
FLG_MYSKYHD, *SPORT*, *FLG_SKY_ON_DEMAND*, *CINEMA*, *SKY_FAMIGLIA*, *STB_MYSKYHD*, *STB_HD*, *CALCIO*, *FLG_MV*, *FLG_ MYSKY*, *FLG_HD*.

Il dataset viene bilanciato utilizzando entrambe le tecniche di undersampling e oversampling. Ciò è stato fatto con lo scopo di poter osservare quale potesse essere la soluzione migliore per lo studio che si sta svolgendo.

4.3.1 Undersampling, feature selection, k-folds e ROC Curve

I risultati ottenuti in seguito alla tecnica di sottocampionamento sono riportati in tabella 2.

Model	R	P	F-m	A	AUC
RF	0.03	0.59	0.05	0.97	0.74
Logistic	0.00	0.00	0.00	0.97	0.74
MLP	0.00	0.00	0.00	0.97	0.75
GNB	0.42	0.08	0.13	0.85	0.72

Tabella 2. Modelli a confronto - dataset bilanciato. R = Recall, P = Precision, F-m = F_1 measure, A = Accuracy, AUC = Area Under Curve.

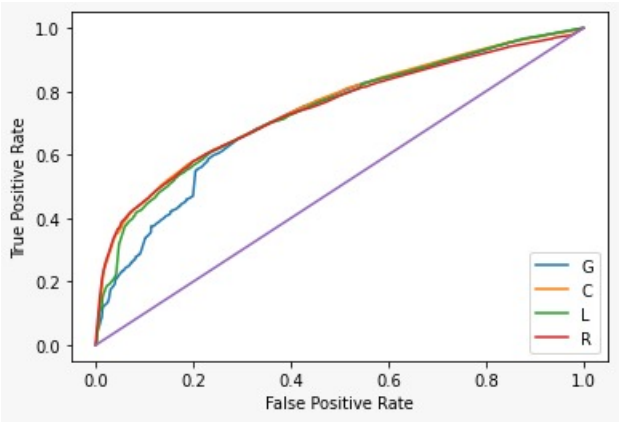

Figura 3. ROC curve undersampling. G=Gaussian Naive Baies, C=Multi-Layer Perceptron, L=Logistic Regression, R=Random Forest.

Il modello con la F_1 measure maggiore è il modello Gaussian Naive Bayes come nel caso precedente; il suo valore è pari a 0.13. Il valore della AUC, invece, è pari a 0.72. Nonostante i valori di AUC dei diversi modelli siano molto simili e accettabili, si nota dalla tabella il fatto che i valori della F_1 measure sono nulli o prossimi allo zero. Si deduce quindi che nessuno dei modelli utilizzati in questo caso sia adatto al raggiungimento dell'obiettivo che ci si è posti.

4.3.2 Oversampling, feature selection, k-folds e ROC Curve

I risultati ottenuti in seguito alla tecnica di sovracampionamento sono riportati in tabella 3.

Model	R	P	F-m	A	AUC
RF	0.55	0.09	0.15	0.80	0.74
Logistic	0.61	0.07	0.13	0.75	0.74
MLP	0.68	0.06	0.11	0.66	0.74
GNB	0.57	0.07	0.13	0.76	0.72

Tabella 3. Modelli a confronto - dataset bilanciato. R = Recall, P = Precision, F-m = F_1 measure, A = Accuracy, AUC = Area Under Curve.

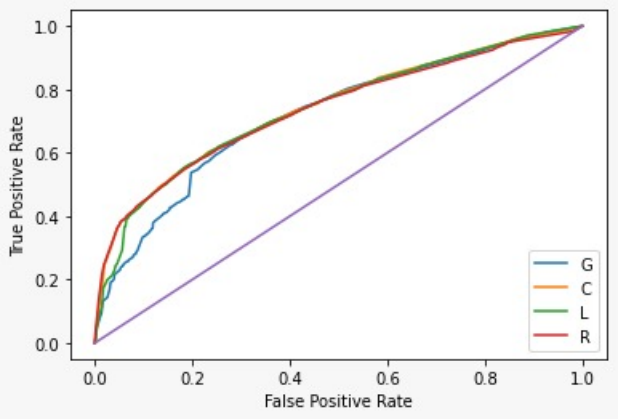

Figura 4. ROC curve oversampling. G=Gaussian Naive Baies, C=Multi-Layer Perceptron, L=Logistic Regression, R=Random Forest.

La performance migliore è data dal modello euristico, ovvero il Random Forest, con una F_1 measure pari a 0.15 e una AUC pari a 0.74. È importante sottolineare come, con l'oversampling, i valori della F_1 measure sono più elevati rispetto a quelli ottenuti attraverso l'undersampling. I valori dell'AUC, invece, rimangono dello stesso magnitudo.

4.4 Metodo dei quantili

Come si può osservare dai risultati riportati, i modelli predittivi e i metodi tipici del machine learning utilizzati, non rappresentano il metodo migliore per giungere alla soluzione del problema e quindi al raggiungimento dell'obiettivo che lo studio richiesto si pone.

Le variabili esplicative infatti non sono molto correlate alla variabile target in questione e ciò comporta e giustifica i valori molto ridotti ottenuti applicando le tecniche e le metodologie prima implementate.

Per questo motivo è estremamente importante provare a ricorrere ad una ulteriore metodologia, ovvero il metodo dei quantili.

Si è deciso di applicare il metodo al modello di classificazione con il valore maggiore di F_1 measure ovvero al Random Forest. I primi due quantili sono stati unificati in uno solo in quanto nel primo quantile le proabilità predette dal modello erano tutte nulle.

In figura 5 si osserva il grafico ottenuto. È possibile notare che la frazione di churner presenti in ogni quantile ricade all'interno dei valori minimo e massimo di probabilità presenti nel medesimo quantile, tranne nel primo caso. Il modello riesce quindi a predire l'andamento dei dati: a un quantile maggiore è associato un numero di churner maggiore.

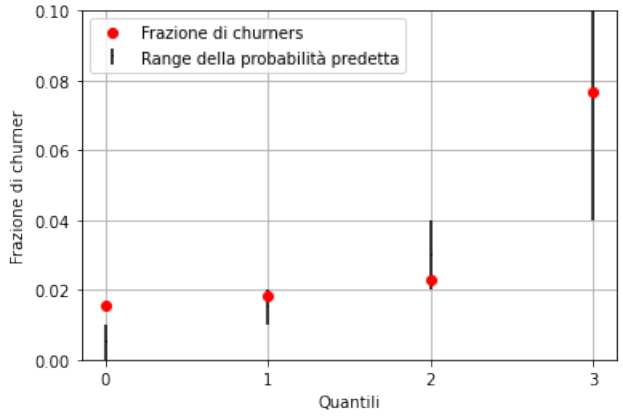

Figura 5. Metodo dei quantili con modello Random Forest. L'asse y è troncato a 0.1 per permettere una agile lettura dei primi tre quantili.

5. Conclusioni

L'obiettivo principale di questo progetto consiste nel trovare un modello di classificazione, capace di predire se un cliente abbandonerà o meno un particolare servizio. Attraverso le analisi, si ha lo scopo di aiutare aziende a mantenere, se non migliorare, un ottimo rapporto con i suoi clienti. In altre parole, aiutare a capire dove, quando e come agire affinché gli indicatori della salute del business e della Customer Satisfaction rimangano alti. Il problema è stato studiato nel contesto di classificazione binaria e affrontato attraverso tecniche di feature selection per individuare le caratteristiche più importanti, come anche attraverso tecniche di undersampling e oversampling per gestire il problema delle classi fortemente sbilanciate. I modelli implementati sono il Random Forest, la Logistic Regression, il Multi-Layer Perceptron e infine il Gaussian Naive Bayes. Il processo di sovra-campionamento ha portato a risultati migliori rispetto al sotto-campionamento; infatti insieme al modello Random Forest, ha portato a migliori performance in termini di F_1-measure e AUC, evitando di perdere utili informazioni per il processo di apprendimento. Ciò nonostante, il valore massimo raggiunto della F-measure ammonta a 0.15, abbastanza basso in quanto questa metrica risiede nell'intervallo [0,1]. Questo è spiegato dal fatto che la correlazione concettuale tra le variabili esplicative e la variabile target risulta debole, in quanto le variabili esplicative non studiano il comportamento dell'utente, una cui variazione potrebbe portare un segnale di churn, bensì i suoi interessi. Valutare quindi se gli interessi di un utente lo rendono più propenso a churnare è una impresa ardua. Non risultando possibile predire, con i dati a disposizione, precisamente la classe di appartenenza di ogni utente si è deciso di studiare se il modello predicesse almeno l'andamento della quantità dei churner. Per questo, il metodo dei quantili è stato applicato, mostrando che il modello che aveva performato meglio, il Random Forest, predicesse per ogni quantile la probabilità di churn con una buona approssimazione se confrontata alla frazione di churner nello stesso quantile.

In conclusione, dai risultati ottenuti, si evince che i dati a disposizione non sono adeguati a raggiungere una risposta precisa alla nostra domanda di ricerca, in quanto i modelli presentano valori bassi delle metriche di performance. Ciò nonostante, quest'analisi può essere usata come punto di partenza per ulteriori analisi, considerando che la maggior parte dei dati a nostra disposizione sono stati scartati perché inutilizzabili, l'integrazione con una nuova sorgente contenente dati sul comportamento degli utenti nel tempo potrebbe migliorare i risultati ottenuti.