Generali Data Challenge

Riccardo Cadei
MSc Data Science, EPFL, Switzerland

Abstract

In questo report presento la mia soluzione al problema di Churn Classification di polizze assicurative proposto da Assicurazioni Generali S.p.A. Le particolarità del problema sono la forte perturbazione dei dati e lo sbilanciamento delle labels. A seguito di una dettagliata fase di analisi dei dati e pre-processing, formulo $Alghora$: un algormitmo di classificazione costruito ad hoc per questa challenge e sperabilmente generalizzabile a nuovi problemi con simili patologie. $Alghora$ utlizza una struttura gerarchica, combinando tra loro 7 moderni algoritmi di classificazione allenati individualmente facendo ensambling sul training set. Insieme i modelli generano 7 nuove features sintetiche, attraverso le quali risolvo un nuovo problema di classificazione più semplice ottenendo il secondo miglior $F1 - score$ (pari a 0,379221) sul Public Test Set di Generali Data Challenge.

I. Introduzione

L'obiettivo di Generali Data Challenge è predire la churn di un cliente da una polizza assicurativa, note 294 features anonime (continue, discrete o categoriche). Non conoscendo il significato di ciascuna feature il problema perde di interpretabilità; ciò nonostante è ancora possibile studiare il dataset attraverso diverse analisi statistiche che non possiamo però ricondurre alla letteratura sul dominio di applicazione (polizze assicurative). Propongo di risolvere il problema in maniera supervisionata formulando $Alghora$: una mia proposta di classificatore ad hoc per questo problema basato sulla combinazione di 7 diversi classificatori allenati individualmente su diversi training bilanciati.

Nella Sezione 2 presento la mia analisi ed elaborazione del dataset, seguita dalla formulazione di $Alghora$. Nella Sezione 3 discuto i risultati e la Sezione 4 è dedicata alla conclusione e alle proposte per sviluppi futuri.

II. Modelli e Metodi

Ho studiato e risolto questo problema di classificazione in due fasi: analisi e rielaborazione dei dati (data exploration, data visualization, preprocessing, features selection, features expansion) e design del modello ($Alghora$).

A. Analisi e rielaborazione dei dati

Il training set consiste in 10'000 utenti ciascuno descritto da 294 features e una label binaria (1 'Churn' e 0 'No Churn'). Il test consiste in altrettanti 10'000 utenti di cui, note le stesse 294 features, dobbiamo prevedere la label. Le particolarità di questo dataset sono la sua disomogeneità (variabili continue, discrete e categoriche), la sua incompletezza (numerosi valori mancanti e possibili perturbazioni) e lo sbilanciamento delle labels (solo 12% di 'Churn' sul Train). Attraverso il pre-processing e feature engineering cerco di risolvere i primi due problemi mentre il terzo è affrontato direttamente nella formulazione dell'algoritmo di classificazione.

1) Definizione dei Missing values: Il dataset è ricco di dati mancanti. Insieme a questi definisco come dati mancanti le stringhe accidentalmente presenti in features numeriche, i valori di features categoriche presenti solo nel test set e il carattere '#' (non altrimenti specificato).

2) Features categoriche: Non tutte le features sono numeriche. In particolare `feature 36, 37, 38 e 39` sono categoriche. Per queste determino dunque tutte le possibili quadruple (24) e per ciascuna di queste definisco una nuova feature binaria. Osservo inoltre che molte altre features discrete potrebbero essere in realtà categoriche (ma non è facile compiere ulteriori ipotesi senza una conoscenza del dominio di applicazione).

3) Imputare i missing values: Comparo tra loro 3 diverse tecniche per imputare i missing values:

- mediana (statistica robusta a eventuali outliers)
- k-NN [1]
- Multivariate Imputation attraverso Chained Equation (MICE) [2]

Massimizzando l'$F1 - score$ attraverso Grid Search con Cross Validation trovo sorprendentemente che l'imputazione con la mediana è la più performante strategia (nonché la più efficiente computazionalmente).

Definisco inoltre una nuova feature pari al numero di missing values di ciascun example.

4) Outliers: Dai plot delle distribuzioni di ciscuna feature osservo che sono presenti diversi outliers (probabilmente causati da errori di registrazione o perturbazioni volontarie del dataset). Decido dunque di 'tagliare' i valori degli elementi di ciascuna feature a un minimo (massimo) α-percentile (1-α). Ancora determino l'α ottimale (=0.2) attraverso Grid Search con Cross Validation massimizzando l' $F1 - score$.

5) Feature Engineering: Esistono diverse strategie per trasformare e/o ridurre lo spazio delle features continue. Tuttavia il dataset, come elaborato finora, è composto sia da features continue che discrete e binarie; inoltre la mia ipotesi è che diverse features siano irrilevanti. Per questo comparo e combino tra loro due diverse strategie di features selection:

- Features selection 1:
  - elimino tutte le features che hanno più di β (=0.6) missing values
  - elimino tutte le features a valore costante (ben 55)
  - elimino tutte le features con una correlazione maggiore di γ (=0.97)
  - elimino le features con una importanza comulativa minore dell'1% (attraverso Gradient Boosting Machine),
- Features selection 2:

– elimino tutti i regressori che per una regressione logistica sul training set hanno un $p - value$ sul t-test maggiore di un threshold fissato (=0.25).

Propongo inoltre due strategie di data expansion per enfatizzare la dipendenza non lineare tra features e label:

- Features expansion 1:
  - Espando polinomialmente al secondo grado lo spazio delle features.
- Features expansion 2:
  - Per ogni feature non negativa definisco una nuova feature pari a $log(1 + x)$.

6) Standardizzazione: Standardizzo infine ogni feature (sia in training che in test set) sottraendo la media e dividendo per la deviazione standard calcolate rispetto al training set.

Combinando tra loro diversi i parametri propongo dunque 5 diverse rielaborazioni del dataset iniziale tra le quali privilegio (attraverso Grid Search) il pre-processing con tutte le tecniche e i parametri sopra riportati, senza features selection 2 e senza entrambe le features expansion.

B. $Alghora$

Una volta rielaborato il dataset voglio costruire un classificatore ad hoc per fare fronte al carattere fortemente unbalanced delle labels. Propongo dunque $Alghora$: la composizione di 7 celebri classificatori, dove a sua volta ciascun classificatore è allenato individualmente su diversi training set bilanciati attraverso la tecnica di ensamble [3]. Essendo il rapporto tra le due classi ('Churn'/'No Churn') circa 1 a 7, divido infatti il training set in 7 nuovi training set, ciascuno composto da un settimo degli elementi della classe maggiore (undersampling), e tutti gli elementi della classe maggiore. In Figura 1 una rappresentazione grafica di questa tecnica.

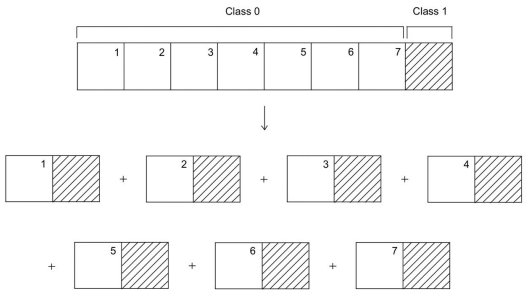

Figure 1: *Rappresentazione grafica della tecnica di ensambling: per ogni classificatore di $Alghora$ divido il training set sbilanciato in 7 nuovi training set bilanciati e lo alleno separatamente su ciascuno di questi.*

I classificatori proposti sono:

- Logistic Regression (LR)
- Support Vector Machine (SVM)
- Random Forest (RF)
- Adaptive Boosting (ADA) [4]
- Gradient Boosting Classifier (GB)
- eXtreme Gradient Boosting Classifier (XGB) [5]
- CatBoost Classifier (CAT) [6]

Regolo i parametri di ciascun classificatore individualmente attraverso Grid Search (massimizzando l'$F1 - score$ in Cross Validation), tuttavia la trattazione completa delle ipotesi e dei risultati di questa ricerca esula dal contenuto di questo report.

Alleno ciascuno classificatore 7 volte, una per ogni nuovo training set, e per ciascuna di esse memorizzo la previsione sia sul training che sul test set. Ottengo dunque per ogni example del training e del test, il numero di previsioni positive ('Churn') ottenute attraverso ciascun modello allenato e valutato per 7 volte su training diversi. Abusando della nomenclatura della struttura gerarchica di una neural network, queste 7 nuove features ($\in \{0, ..., 7\}$) sono i 7 neuroni dell'hidden layer di $Alghora$. Attraverso questo primo step ho infatti ridotto lo spazio delle features da $\tilde{3}00$ a 7, ciascuna (sperabilmente) rappresentante un particolare pattern enfatizzato da un particolare modello. Combino dunque tra loro i 7 neuroni dell'hidden layer attraverso un classificatore finale, quale Logistic Regression o Gradient Boosting Classifier, pesando questa volta nella funzione di costo l'importanza di ciascuna classe. In Figura 2 riporto una rappresentazione grafica di $Alghora$.

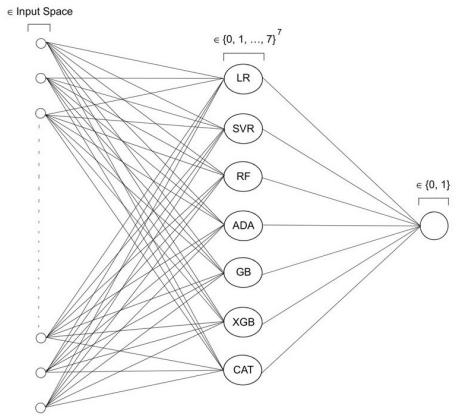

Figure 2: *Rappresentazione grafica della classificazione di un example attraverso $Alghora$.*

Accorgimento finale: : Risolvo in un primo momento il problema utilizzando ensamble con un singolo classificatore e classifico facendo voting con un threshold fissato. Salvo le 4 migliori soluzioni ottenute (con modelli diversi) in cross validation (e coerentemente anche sul public test set) e ne calcolo l'intersezione (operatore logico 'and') che nomino 'sintetic'. L'ipotesi è che 'sintetic' contenga meno 1 ('Churn') di quanti attesi dalla legge dei grandi numeri, ma robusti, poiché confermati da modelli singoli diversi tra loro. Combino infine la 'sintetic' con la previsione ottenuta attraverso $Alghora$.

III. Risultati

Essendo il dataset fortemente unbalanced è importante scegliere un'opportuna metrica per la valutazione delle performance: l'$Accuracy$ ad esempio non è la migliore proposta in quanto può essere facilmente hackerata focalizzandosi su i True Negative. Decido dunque di valutare le performance del mio modello sulla base dell'$F1 - score$ (stessa metrica utilizzata per la classifica

tecnica di questa challenge) definito come:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Attraverso Grid Search e Cross Validation determino tutti i parametri del pre-processing e di ogni singolo modello massimizzando proprio l'$F1 - score$ sui validation set. Una volta regolati i parametri alleno $Alghora$ sull'intero training set, e faccio previsione sul test set aggiungendo l'accorgimento finale. $Alghora$ ottiene il secondo migliore $F1 - score$ sul Public Test set pari a 0.379221, in accordo con i valori ottenuti in Cross Validation (non ci sono sospetti di overfitting).

IV. Conclusione e prospettive future

Generali Data Challenge è un problema di classificazione fortemente unbalanced su un dataset non documentato e fortemente perturbato. Una volta rielaborato il dataset, propongo $Alghora$: un classificatore (in forma supervisionata) costruito ad hoc per questo problema e sperabilmente generalizzabile per altri problemi con patologie simili. Le due idee principali di $Alghora$ sono fare ensambling tra training bilanciati e utilizzare un'architettura gerarchica. $Alghora$ tuttavia non è una rete neurale in senso classico, in quanto ogni neurone è collegato a tutti i neuroni del layer predente attraverso un modello allenato individualmente (no Backpropagation su tutti i parametri contemporaneamente). Le performance di $Alghora$ superano sensibilmente quelle di ciascun singolo modello utilizzato preso individualmente, tuttavia un $F1 - score < 0.5$ ci fa riflettere sulla solvibilità di questo problema. Sebbene le performance ottenute siano sensibilmente migliori che una previsione dummy o aleatoria, non possiamo dire di aver completamente risolto il problema. D'altronde non è assolutamente scontato che nelle features sia contenuta tutta l'informazione per spiegare la churn che cerco di prevedere, e allo stesso tempo non dobbiamo illuderci che un algoritmo di Machine Learning, per quanto robusto, sia in grado di indovinare una relazione che potrebbe non esistere (o più probabilmente, non essere spiegata solo da quei dati). Un possibile approccio per sviluppi futuri potrebbe essere in un primo step dividere l'intero dataset in clusters (in maniera non supervisionata), nell'ipotesi che i clienti possano essere raggruppati in categorie diverse con comportamenti diversi, e per ciascuna di queste allenare e applicare uno specifico classificatore, quale $Alghora$.

References

[1] E. Acuna and C. Rodriguez, "The treatment of missing values and its effect on classifier accuracy," in *Classification, clustering, and data mining applications*. Springer, 2004, pp. 639–647.

[2] S. v. Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in r," *Journal of statistical software*, pp. 1–68, 2010.

[3] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[4] Y. Freund, "An adaptive version of the boost by majority algorithm," *Machine learning*, vol. 43, no. 3, pp. 293–318, 2001.

[5] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, pp. 1–4, 2015.

[6] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *Advances in neural information processing systems*, 2018, pp. 6638–6648.

[7] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.