Terrorismo Globale: ricerca di un'impronta

Riccardo Cervero 794126[1], Marco Apa 848154[2], Pasquale Paolicelli 848804[3]

Sommario
Il terrorismo globale è ormai, in ogni sua forma, dipendente dall'applicazione di *Information and Communication Technologies* sempre più potenti, utilizzate per comunicare attraverso la rete degli affiliati, convertire nuovi adepti alla propria ideologia e organizzare gli attacchi. Allo stesso modo, le forze statali e internazionali impegnate nella difesa investono nello sviluppo di tecniche e modelli predittivi avanzati, in grado di ridurre l'imprevedibilità degli attentati e impostare strategie efficaci. Il seguente lavoro si propone di trovare una metodologia di *Machine Learning* capace di individuare il *modus operandi* di un gruppo terroristico. Attraverso varie categorie di modelli di classificazione supervisionata, si cercherà, dunque, di estrapolare un set di caratteristiche ricorrenti per ogni associazione, con un dato livello di *F-measure* e Accuratezza. Dopo un'importante fase di pulizia, preparazione dei dati e selezione degli attributi esplicativi, l'obiettivo ha presentato due questioni fondamentali: l'adozione di specifici approcci per la risoluzione del problema di classificazione supervisionata multi-classe e la gestione del forte sbilanciamento delle osservazioni verso alcuni valori della variabile d'interesse, ovvero la cosiddetta problematica di *class imbalance*. Inoltre, nell'ultima sezione, sarà impiegato il metodo di *Boosting* per stimare in maniera migliore la misura di Accuratezza.

Keywords
Terrorism — Multiclass problem – Imbalanced class

[1] *Universita' degli Studi di Milano Bicocca, CdLM Data Science*
[2] *Universita' degli Studi di Milano Bicocca, CdLM Data Science*
[3] *Universita' degli Studi di Milano Bicocca, CdLM CLAMSES*

Indice

Introduzione

Negli ultimi anni, il terrorismo globale ha accresciuto il proprio impatto sociale e culturale all'interno delle comunità colpite, grazie alla crescente frequenza di attentati compiuti, all'incremento del numero di agenti coinvolti, e un sempre maggiore uso dei nuovi mass media. Le *Intelligence* impegnate nell'attività di Contro-terrorismo necessitano di tecnologie migliori per combattere le minacce, incrementando le analisi sulla mole di dati relativi ai diversi gruppi terroristici, al fine di prevedere, con maggiore affidabilità, le caratteristiche dei futuri attentati. La seguente trattazione si propone di suggerire un modello di analisi predittiva mirato ad identificare l'esistenza di una "firma" riferita ad ogni associazione terroristica. I dati utilizzati provengono dal *Global Terrorism Database* [1] (GTD). Esso consiste in una raccolta di osservazioni a livello globale, relative ad attentati terroristici verificatisi nel periodo fra il 1970 e il 2017. Il database è amministrato e aggiornato dal *National Consortium for the Study of Terrorism and Respondes to Terrorism*.

Innanzitutto, è opportuno cominciare l'analisi dalla decrizione del fenomeno oggetto di studio. A tal proposito, la documentazione riferita al dataset GTD definisce l'attentato terroristico come *'uso minacciato o effettivo della forza e della violenza illegale da parte di un attore non statale, per raggiungere un obiettivo politico, economico, religioso o so-*

[1] https://www.start.umd.edu/gtd/downloads/Codebook.pdf

ciale attraverso la paura, la coercizione o l'intimidazione'. Il database raccoglie originariamente più di 180000 osservazioni, descritte da 153 attributi di vario genere - binari, nominali, numerici -, raggruppati in 9 aree tematiche.

1. Preprocessing

1.1 Feature Selection

La selezione degli attributi significativi per la domanda d'analisi non ha implicato l'utilizzo di algoritmi specifici, bensì è stata condotta seguendo alcuni criteri logici:

1. Eccessiva presenza di *missing value* all'interno di un campo, condizione che influenza negativamente la classificazione.

2. Assenza sistematica dei valori di un attributo: è il caso, ad esempio, delle variabili *claim* e *individual*, i cui dati sono stati registrati soltanto a partire dall'anno 1997. È pertanto necessario rimuovere questi campi per non falsare il risultato dell'analisi.

3. Forte *class imbalance* nello specifico caso di attributi binari: per questo tipo di variabili – come, fra le altre, *crit1, crit2, extended*, non è possibile ovviare al problema di sbilanciamento delle classi, mediante, ad esempio, aggregazione. Pertanto, si è deciso di escluderle.

4. Parte dei dati è affetta da varie tipologie di ridondanza informativa:

    - Alcuni campi - come *targsubtype1 e weapsubtype1* - sono stati esclusi dall'analisi poiché nel database è già presente la loro versione aggregata, più interpretabile e utile all'addestramento del modello di classificazione;
    - Il GTD contiene versioni del tutto identiche della stessa variabile, come accade per *targtype1, targtype2 e targtype3*;
    - Un particolare attributo *corp1*, accoglie valori identici a *targtype1* soltanto in alcune osservazioni, perciò il suo utilizzo nel processo di classificazione potrebbe penalizzare l'analisi;
    - Attributi come *latitude* e *longitude* sono state scartate poiché presentano un elevatissimo numero di classi, rischiando di aumentare eccessivamente il costo computazionale e peggiorare l'interpretabilità del modello;
    - La nostra analisi si propone di offrire un approccio in grado di capire se sia possibile identificare una firma dei gruppi terroristici in base alle caratteristiche del loro modus operandi, e non in base alla provenienza geografica o all'anno di riferimento. È il caso, ad esempio, di *iyear, region, country e natly1*, la quale registra la nazionalità delle vittime. Queste variabili devono essere necessariamente estromesse durante l'addestramento del modello, poiché distorsive per la ragione appena riportata. In più, questi valori sono spesso in stretta dipendenza con il gruppo terroristico a cui si riferiscono, producendo un elevato rischio di *overfitting*;
    - Gli altri attributi esclusi sono irrilevanti per la domanda d'analisi.

Come risultato del procedimento di *Feature Selection* appena descritto, le caratteristiche degli attentati scelte per la risoluzione del problema di classificazione sono:

- *gname*, la variabile target nominale che registra il nome del gruppo terroristico;
- *imonth*, il mese – espresso in numero – in cui è avvenuto l'attacco;
- *multiple*, che presenta valore 1 se l'atto fa parte di una serie di attacchi concatenati, 0 altrimenti;
- *success*, pari a 1 se l'attacco ha avuto successo, cioè se si sono verificate delle conseguenze tangibili in base al tipo di attacco, valutando in maniera globale l'azione nel caso di un attentato multiplo;
- *suicide*, "1" se l'attentato ha implicato il suicidio degli attentatori, 0 altrimenti;
- *targtype1*, che riassume, in 21 classi, la categoria generale relativa alla vittima o al target;
- *weaptype1* indica i 12 tipi di arma utilizzata dagli attentatori;
- *nkill* riporta il numero di vittime confermate per il dato attacco;
- *nwound* il numero di feriti confermati per il dato attacco;
- *INT-ANY*, che presenta valore 1 se l'attacco è internazionale dal punto di vista di tre dimensioni: logistica, ideologica o l'unione di entrambe; 0 altrimenti;
- *attacktype1* rappresenta la metodologia di attacco in base a 8 categorie: *Assassination, Hijacking,Kidnapping, Barricade Incident, Bombing Explosion, Armed Assault, Unarmed Assault, Facility/Infrastructure Attack.*

1.2 Data Cleaning

Dopo aver convertito gli attributi al tipo migliore per condurre l'analisi in *Knime*, e quindi per elaborare in modo corretto i record, si è proceduto alla gestione dei valori mancanti. Le variabili nel dataset originario, come indicato nel *Codebook*, presentavano 8 diverse codifiche per i *missing value*[2]. Inoltre,

[2] "?", "Unknown", "-9", "-99", "9", "13", "0", "20"

la fase di preparazione dei dati ha implicato il trattamento della variabile *nwound*: il numero di feriti presentava un valore anomalo pari a 8.5 in due osservazioni. Poichè le informazioni a nostra disposizione non erano sufficienti per sostituire correttamente i valori, si è deciso di rimuoverli.

1.3 Riduzione delle classi di *gname*

La variabile target presenta un numero di classi evidentemente troppo elevato – 2971, molti dei principali gruppi terroristici esistiti dal 1970 fino ad oggi –, rendendo quasi impossibile o del tutto inefficace la conduzione di una classificazione supervisionata utile alla nostra analisi. In più, *gname* è affetta da un grave sbilanciamento delle classi, evidente dal grafico sottostante (Figura 1).

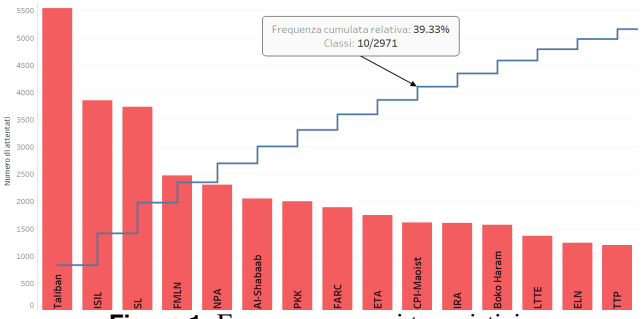

Figura 1. Frequenza gruppi terroristici

Pertanto, è stato opportuno focalizzare l'analisi sulle prime dieci classi più frequenti, le quali coprono il 39.33% delle osservazioni totali del dataset originario.

1.4 Aggregazione

L'attributo *weaptype1* presenta un grave problema di *class imbalance*: le classi *Firearms e Explosives*[3] coprono il 92.64% delle 27066 osservazioni totali rimaste dopo il compimento delle fasi precedenti di *preprocessing*. La restante porzione di record è divisa fra ulteriori 10 classi di armi, le quali, avendo una frequenza molto bassa, rischiano di essere sottorappresentate nel *Training Set* e *Test Set*. È opportuno garantire, perciò, che il *Test Set* ed il *Training Set* comprendano lo stesso numero di possibili valori, affinché il procedimento di *Hold-out* possa essere svolto in maniera esatta e lo specifico nodo *Inducer* di *Knime* possa effettuare le previsioni senza rilevare alcuna incongruenza fra i due subset. Pertanto, si è deciso di aggregare in una nuova classe denominata *"Other"* le altre 10.

Infine, è opportuno specificare che non sia stato eseguito alcun campionamento poiché la dimensionalità del data set originale è già ridotta significativamente durante la fase di *preproccesing*.

[3] Valore 5 e 6 nel dataset

2. Metodi di valutazione delle prestazioni

La valutazione delle prestazioni dei vari classificatori mediante stima della misura di Accuratezza è stata ottimizzata con l'approccio *K-folds Cross Validation*, poiché permette di ottenere una stima più robusta e affidabile rispetto al tradizionale metodo *Hold-out*. È chiaro, infatti, come il valore di *Accuracy* basato su quest'ultimo approccio possa dipendere quasi essenzialmente dalla scelta - il più delle volte, casuale – del *Test Set*. Ciò comporta un rischio molto alto di sopravvalutare o sottovalutare le reali misure. Posto k uguale a 10, tramite un processo iterativo, la tecnica di *Cross Validation* estrae una misura di Accuratezza più robusta, poiché successivamente stimata come mediana dei 10 risultati calcolati sui K *subset*. Tuttavia, il calcolo dell'Accuratezza si basa erroneamente sull'ipotesi che tutte le classi siano equamente distribuite nello spazio dei dati, cioè si ripetano con la medesima frequenza nel dataset. Pertanto, nel caso multi-classe, quando la distribuzione della variabile d'interesse è fortemente sbilanciata verso determinati valori, i modelli di classificazione prescelti rischiano di etichettare tutti i record con le classi più frequenti [4]. Nonostante la riduzione delle classi descritta al punto 1.3, l'attributo *gname* continua a presentare una forte concentrazione: i primi tre gruppi terroristici per frequenza costituiscono quasi il 50% del subset delle osservazioni considerate[5]. Pertanto, assieme all'Accuratezza – comunque calcolata, con lo scopo di offrire un'analisi completa dei risultati -, è necessario valutare la performance dei classificatori attraverso la stima delle metriche *F-measure,Recall e Precision*. Infatti, la capacità del modello di prevedere correttamente ogni classe non viene valutata sul totale dei *records*, come effettuato dall'Accuratezza. Al contrario, esse analizzano la prestazione del classificatore in base alla dimensione di ogni valore della variabile target[6], quindi considerando la frequenza di ogni gruppo terroristico. Nel dettaglio, il risultato globale delle tre metriche è stato calcolato come mediana dei valori associati ad ogni classe, e non come media, al fine di evitare la distorsione provocata dalla grande variabilità dei risultati per gruppo terroristico. Questo aspetto verrà approfondito nel dettaglio all'interno della quarta sezione.

Il calcolo delle metriche è stato eseguito seguendo, nella maggior parte dei casi, l'approccio *1-against-1*. Questo criterio suddivide il problema multi-classe in molteplici classificazioni binarie. Quando non specificato, il metodo sottinteso è *1-against-all*.

[4] Come un modello ZeroR Rule

[5] Taliban : 5547, Islamic State of Iraq and the Levant (ISIL) : 3859, Shining Path (SL) : 3734, il 48.23% sul totale di 27247 osservazioni

[6] La Precision basa la valutazione sul totale delle previsioni del classificatore su una data classe, siano esse corrette (True Positive) o non corrette (False Positive). Invece, la Recall considera la mera frequenza della classe.

3. Modelli di classificazione e interpretazione dei risultati

Per svolgere il compito di classificazione supervisionata, si è scelto di adoperare sei categorie differenti di modelli predittivi:

3.1 Euristici

Tra le metodologie euristiche di classificazione, si è scelto di utilizzare la famiglia degli alberi decisionali, ovvero quei modelli in cui il percorso da un nodo radice ad uno foglia rappresenta la previsione per la class *attribute*. In particolare, sono state selezionate due tipologie sviluppate da *Weka*[1]: J48 e *Random Forest*. Il primo sfrutta l'algoritmo *C4.5*, il quale consente di rimuovere i rami che non si sono rivelati utili e rimpiazzarli con dei nodi foglia[7]. Per tutelarsi dal rischio di *overfitting*, è mantenuta attiva l'impostazione *'useMLDcorrection'*, che opera una correzione sugli attributi numerici[8]. Il parametro *'minNumObj'* [9] è lasciato di default, al fine di migliorare la specificità del modello, in modo tale da avere un basso numero di istanze per nodo e, quindi, diminuire la generalità dell'albero di decisione [2]. Invece, il *Random Forest* si basa sull'idea che la combinazione dei risultati dei singoli *decision trees* sia in grado di aumentare la casualità del classificatore, migliorando la performance e incrementando la stabilità della previsione[3]. È stato, infatti, rilevato un aumento delle prestazioni rispetto al J48, relativamente alle misure di *Accuracy, Precisione Recall*, e di conseguenza anche dell'*F-measure*. Per quanto riguarda il nodo *Random-Forest* in *Knime*, sono stati mantenuti i parametri preimpostati per evitare un appesantimento del costo computazionale.

3.2 Support Vector Machine

Le macchine a vettori di supporto tentano di ripartire lo spazio dei dati in modo da massimizzare l'ampiezza del margine rigido che separa le partizioni. Quando il dataset non è linearmente separabile, l'algoritmo trasferisce in dati in un nuovo spazio, sfruttando una funzione *Kernel*. Un tipo di macchina a vettori di supporto è la *Sequential Minimal Optimization*, sulla quale è stata impostata una funzione *Kernel* polinomiale[10]. Poiché la variabile target è composta da dieci classi, è stato opportuno adottare la specifica procedura di *pairwise coupling* - accoppiamento "a due a due"-, la quale permette di estendere la classificazione da un problema binario a un problema multi-classe[4]. La metodologia corrispondente è ottenuta impostando un approccio *1-against-1* [11] nella fase di addestramento del modello. È stato poi preso in considerazione l'algoritmo *Spegasos*, comparando i risultati ottenuti dall'applicazione di due differenti *Loss Function*: una *logistic regression loss function e Hinge loss function*[12]. La *'log loss'* produce un peggioramento per quanto concerne la stima della *Precision*, ma comunque risultati migliori nella *F-measure*, grazie ad un più elevato valore di *Recall*. Infatti, questa funzione, applicata agli output della macchina a vettori di supporto, produce stime di probabilità più corrette per le analisi multiclasse. In ogni caso, l'algoritmo *SMO* con funzione *kernel* polinomiale fornisce una miglior prestazione rispetto alle due tipologie di *Spegasos*. Nonostante ciò, le macchine di vettori a supporto producono risultati scarsi sul GTD, come visibile nella tabella 1.

3.3 Modelli probabilistici

Questa categoria comprende i classificatori bayesiani. In particolare, un algoritmo bayesiano noto è il *Naive Bayes*, il quale assume l'indipendenza condizionale tra gli attributi, quando la classe corrispondente è nota. Il metodo migliore per la risoluzione del problema multi-classe consiste nell'approccio *1-against-1*. È stato notato che l'implementazione di una procedura di *pairwise coupling* produce, in questo caso, un leggero decremento della misura di F-measure. Un secondo modello sviluppato per affrontare il problema di indipendenza condizionale degli attributi è *l'Averaged one dependence estimators (A1DE)*. La particolarità di questo classificatore risiede nella capacità di condurre una classificazione più accurata, calcolando la media sugli stimatori ottenuti da un insieme di modelli *naive-bayesiani*, basati, questa volta, su un'ipotesi di indipendenza più debole, e quindi meno restrittiva, rispetto al classico *Naive Bayes*. Infatti, grazie a questa caratteristica, l'algoritmo *A1DE* offre la seconda performance migliore fra tutti i modelli utilizzati, decisamente superiore al *Naive Bayes*. Inoltre, prestazioni migliori sono state ottenute a seguito dell'adozione di un approccio *1-against-1* con tecnica di *pairwise coupling*.

3.4 Bayesian Nets

Una rete bayesiana - o causale - è un modello in cui la radice rappresenta la variabile di classe, i nodi le variabili casuali, e i collegamenti le condizioni di dipendenza statistica tra le variabili/nodi collegati.

Tramite misura *"BAYES"*, sono state analizzate e giudicate due reti la cui struttura è ottenuta tramite l'utilizzo di due algoritmi diversi: *K2R* e *TAN (Tree-Augmented Naive Bayes)*. Per entrambi è stata Sfruttata la metodologia *1-against-1*.

La rete K2[5] è ottenuta tramite un metodo euristico, in cui inizialmente si assume che tutti i nodi non abbiano collegamenti. Successivamente, in modo incrementale, questi ultimi vengono aggiunti singolarmente, finché l'aumento di probabilità della struttura non è nullo, salvo con l'aggiunta di collegamenti multipli. Nel caso presentato, il concetto appena descritto è stato invertito tramite il parametro *initAsNaiveBayes*, partendo da una struttura *Naive Bayes* in cui tutti i nodi sono collegati tra loro, e applicando una ricerca *greedy* per

[7] L'impostazione corrispondente, in Knime, è: unpruned = False.

[8] La correzione si basa sul principio Minimum Description Length (MDL), ed opera durante il partizionamento degli attributi numerici

[9] Numero minimo di istanze per nodo foglia

[10] 'PolyKernel'in Knime

[11] Parametro method impostato su "1-against-1", in combinazione con l'opzione usePairwisecoupling (True)

[12] In Knime, 'log loss' e 'Hinge loss'

la rimozione degli archi. Questa versione è chiamata K2R *(Reverse)*.

La seconda rete analizzata - TAN[6]- ottiene risultati migliori rispetto alla precedente. L'algoritmo TAN fa uso di una struttura ad albero per estendere la rete bayesiana. Questa si rivela utile per ottenere un'approssimazione delle interazioni tra gli attributi, considerando la correlazione tra questi in accordo ad una specifica istanza della variabile di classe.

3.5 Artifical neural network

Le reti neurali artificiali fanno parte dei modelli di separazione, come le *Support Vector Machine*, e sono caratterizzate da diversi strati di neuroni di input e di output. Una sottocategoria importante della tipologia di reti neurali artificali definite *FeedForward* all'interno delle quali le connessioni fra i nodi non formano cicli - è rappresentata dal *Multi-Layer Perceptron* (MLP). In particolare, esso si basa su un algoritmo di tipo *back-propagation*, il quale, durante il processo di *training*, minimizza la differenza tra la rete di output e il valore desiderato dello stesso, modificando i pesi delle connessioni per diminuire il valore del gradiente[7]. Questo processo iterativo fa sì che il classificatore abbia un notevole peso computazionale, specie se abbinato all'impostazione di un numero elevato di epoche[13]. Così come il numero di epoche, anche gli altri parametri sono stati lasciati di default. Infine, grazie alla sua struttura, il *MLP* offre una naturale estensione dal problema binario al multi-classe, producendo discreti risultati.

3.6 Classificatori ibridi

La miglior prestazione sul *GTD* è stata ottenuta mediante l'algoritmo ibrido *NBtree*, derivato dall'unione del concetto di albero decisionale e della metodologia bayesiana per l'assegnazione dell'etichetta della variabile d'interesse all'osservazione. In particolare, questo procedimento, sfruttando la nozione di probabilità condizionata, attribuisce la classe più probabile ad ogni *record*, computandola all'interno di nodi foglia. La combinazione degli approcci euristico e probabilistico mantiene le proprietà e i pregi dei due modelli. L'interpretabilità della segmentazione fatta dai nodi dell'albero decisionale e quella grafica dei classificatori Naive-bayesiani consentono di ottenere scalabilità sul dataset analizzato[8]. La velocità dell'albero decisionale e la robustezza rispetto gli attributi irrilevanti dei *Naive Bayes*, uniti all'approccio *1-against-1*, hanno permesso di raggiungere alti risultati riguardo l'*Accuracy* e i migliori risultati di classificazione per quanto concerne *Precision, Recall,* e, di conseguenza, *F-measure*.

In conclusione, la seguente tabella (Tabella 1) mostra i quattro risultati stimati per ogni modello di classificazione:

Modello	Accuracy	F-measure	Recall	Precision
NBtree	0.528	0.473	0.495	0.453
A1DE	0.519	0.435	0.458	0.415
Bayesian Net (TAN)	0.513	0.417	0.449	0.39
Random Forest	0.5	0.416	0.429	0.403
MultiLayer Perceptron	0.484	0.4	0.396	0.404
J48	0.466	0.375	0.369	0.381
Bayesian Net (K2)	0.455	0.343	0.367	0.322
SMO PolyKernel	0.466	0.33	0.286	0.39
Naive Bayes	0.35	0.293	0.321	0.27
Spegasos (log loss)	0.379	0.217	0.211	0.279
Spegasos (Hinge loss)	0.368	0.18	0.124	0.325

Tabella 1. Risultati analisi per modello

4. Analisi dei risultati per ogni classe della variabile target

I valori di *F-measure, Precision e Recall* calcolati in precedenza rappresentano il risultato mediano per ogni classe di *gname* in corrispondenza di ogni classificatore. La mediana fornisce una mera sintesi della performance generale, senza descrivere la grande variabilità che caratterizza i singoli risultati per gruppo terroristico. Pertanto, è opportuno analizzare questa caratteristica per estrarre ulteriori informazioni sul fenomeno analizzato. In particolare, in questa sezione verranno esaminate le misure ottenute dall'algoritmo capace di raggiungere la miglior prestazione: l'*NBTree*. Le metriche relative alle dieci classi studiate, come visibile dal grafico *BoxPlot* sottostante, si distribuiscono su un *range* molto sparso, e non vengono influenzate dalla frequenza della classe su cui sono state calcolate, come ci si aspetterebbe. Questo connotato della variabile target evidenzia come l'abilità di uno stesso modello di effettuare previsioni corrette sul *GTD* vari notevolmente a seconda dell'etichetta considerata, a prescindere dal numero di osservazioni ad essa associate. Infatti, perfino le prestazioni di un algoritmo complesso come l'*NBTree* oscillano da un *F-measure* estremamente elevata – ottenuta sulla classe *"Islamic State of Iraq and the Levant"* - a una decisamente bassa – nel caso di *"Revolutionary Armed Forces of Colombia"* -, nonostante rientrino entrambe nel subset di gruppi terroristici più spesso incontrati nell'intero dataset (Figura 2).

[13] In Knime, rappresentate dal parametro 'trainingTime', nel nostro caso lasciato pari a 500

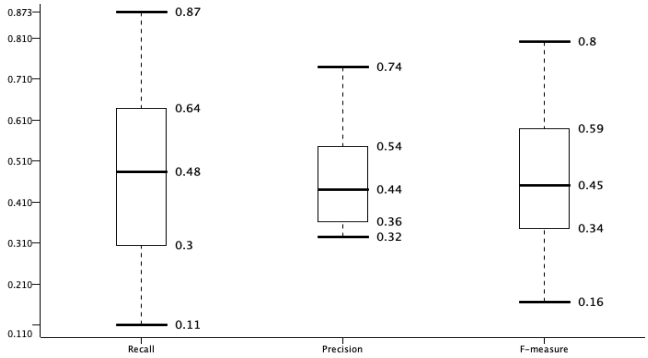

Figura 2. Distribuzione dei risultati ottenuti dal modello NBTree in corrispondenza di ogni classe di gname

La condizione si ripete anche nel caso di un classificatore con risultati peggiori: i valori dell'*F-measure* ottenuti dall'algoritmo *Naive Bayes* sono ugualmente caratterizzati da una grande escursione, come evidenziato dal boxplot seguente, in cui sono riportate le metriche calcolate in corrispondenza di ogni classe di gname (Figura 3).

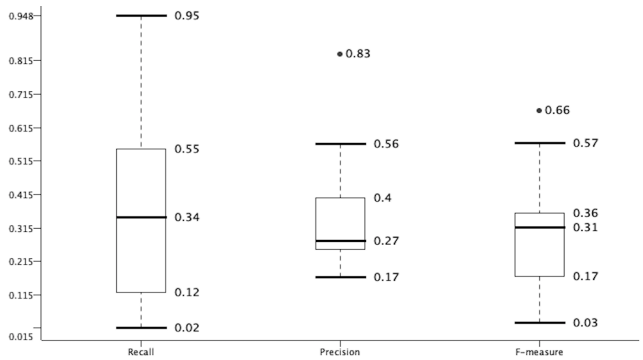

Figura 3. Distribuzione dei risultati ottenuti dal modello Naive Bayes in corrispondenza di ogni classe di gname

Una volta dimostrato che ogni modello ha un comportamento identico, è ragionevole sostenere che la possibilità di identificare una firma per ogni gruppo terroristico, sui dati selezionati, dipenda quasi essenzialmente dal gruppo terroristico stesso e non dalla complessità del modello di classificazione utilizzato. A conferma di quanto ipotizzato, vengono mostrati, in seguito, i coefficienti di variazione dell'*F-measure* ottenuta da ogni modello (Tabella 2).

Modello	CV(Recall)	CV(Precision)	CV(F-measure)
Naive Bayes	0.87	0.713	0.744
Bayesian Net (K2)	0.778	0.522	0.736
SMO PolyKernel	0.78	0.31	0.682
J48	0.933	0.337	0.67
A1DE	0.672	0.355	0.583
Bayesian Net (TAN)	0.591	0.326	0.498
NBtree	0.539	0.31	0.454
Random Forest	0.442	0.372	0.41
Spegasos (log loss)	1.107	0.598	0.819
Spegasos (Hinge loss)	1.276	0.755	0.919
MultiLayer Perceptron	0.64	0.324	0.511

Tabella 2. Coefficiente di variazione dei valori calcolati dai modelli in corrispondeza di ogni classe

5. Uso dell'*Adaptive Boosting* per migliorare la stima della misura di Accuratezza

Come menzionato in precedenza, quando le etichette di classe sono caratterizzate da una forte concentrazione, l'Accuratezza si rivela una misura del tutto inaffidabile per la valutazione delle prestazioni di un classificatore. Per risolvere il problema di *class imbalance* in *gname*, si propone l'utilizzo dell'approccio *Boosting*, un metodo progettato per aumentare le performance di qualsiasi classificatore, o meglio, ridurre significativamente l'errore del *Learner* a cui viene applicato[14]. Il funzionamento consiste nell'iterare l'esecuzione del dato *WeakLearner* su varie distribuzioni estratte dal Training Set e combinare i modelli deboli in un unico classificatore composto. Il preciso algoritmo adoperato in *Knime* è l'*AdaBoostM1*, una versione del metodo *Boosting* sviluppata anche per il trattamento di variabili d'interesse nominali. I suoi principali vantaggi coincidono con la riduzione della distorsione sistematica - quindi dell'Errore, complemento all'unità dell'Accuratezza - che penalizza il *WeakLearner*, forzandolo a concentrarsi su ogni parte dello spazio dei dati, e la diminuzione della sua varianza, operando una combinazione fra le varie ipotesi deboli generate a partire dai diversi subset estratti dal *Training Set*. Tuttavia, l'utilizzo dell'Adaptive Boosting M1 è limitato dall'incapacità di gestire ipotesi deboli che presentano una misura di Errore superiore al 50%[9]. Sapendo che l'errore atteso da un modello che prevede casualmente il valore dell'attributo target – *random guessing model* - è 1 – 1/k, con k pari al numero di classi, nel caso binario è opportuno che il *WeakLearn*, per essere considerato un classificatore accettabile, dia un risultato almeno superiore a 1/2. Pertanto, se k è pari a 10, ci si aspetterebbe che il requisito sia un Errore almeno inferiore al 90%, ma nonostante ciò, l'*AdaBoostM1* continua a richiedere un risultato almeno superiore al 50%. Per questo motivo, è possibile adoperare solamente i modelli *NBTree, A1DE, Bayesian Net* con algoritmo *TAN* e *Random*

[14] Definito in questo caso algoritmo di apprendimento debole o WeakLearner a sua volta caratterizzato da un modello chiamato "ipotesi debole"

Forest. Tuttavia, l'*AdaBoostM1* è stato in grado di migliorare l'Accuratezza soltanto del Random Forest, causando addirittura un peggioramento negli altri casi. Questa diminuzione potrebbe derivare dal fatto che un classificatore già capace di offrire alte prestazioni è associato a un rischio maggiore di *overfitting*, e se ciò si verifica, l'AdaboostM1 tende a penalizzare il risultato del *WeakLearner*, offrendone uno più realistico[15] (Tabella 3).

Modello	Accuracy	Variazione %
Random Forest	0.51	+2 %
Bayesian Net (TAN)	0.499	-2.7 %
A1DE	0.488	-6 %
NBTree	0.316	-40 %

Tabella 3. Risultati applicazione AdaBoostM1

6. Conclusione e suggerimenti per future analisi

L'applicazione degli 11 modelli di classificazione supervisionata sul problema multi-classe di identificazione del nome del gruppo terroristico ha implicato la necessità di considerare la forte distorsione provocata dallo sbilanciamento del dataset. I risultati così ottenuti si sono dimostrati comunque poco utili in quanto la prestazione di ogni algoritmo, in termini di *F-measure, Recall e Precision*, è gravemente disomogenea tra tutti i gruppi. Ciò induce a sostenere che i singoli modelli analizzati non siano sufficienti a individuare con la stessa affidabilità il comportamento tipico di più associazioni terroristiche contemporaneamente. Pertanto, sarebbe opportuno non interpretare il terrorismo come fenomeno globale, bensì focalizzarsi sui singoli gruppi terroristici e costruire modelli mirati per ciascuno, al fine di estrarre informazioni utili per future analisi predittive e impostare strategie precauzionali più efficienti. Infine, l'implementazione dell'approccio *Boosting* ha aiutato a comprendere ulteriori limiti dei migliori *Classifiers*, ovvero l'alto rischio di overfitting dei dati, mostrando un solo miglioramento dell'*Accuracy* su 4.

[15] https://stackoverflow.com/questions/10591092/weka-adaboost-does-not-improve-results

Riferimenti bibliografici

[1] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[2] Peter D Grünwald. *The minimum description length principle*. MIT press, 2007.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In *Advances in neural information processing systems*, pages 507–513, 1998.

[5] Gregory F Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

[6] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.

[7] Hassan Ramchoun, Mohammed Amine, Janati Idrissi, Youssef Ghanou, and Mohamed Ettaouil. Multilayer perceptron: Architecture optimization and training. *IJIMAI*, 4(1):26–30, 2016.

[8] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *KDD*, volume 96, pages 202–207. Citeseer, 1996.

[9] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156. Citeseer, 1996.