SICURI

Modello di Clustering - Sintesi della Ricerca

EXCELLENCE expert.ai SAPIENZA UNIVERSITÀ DI ROMA

Sviluppo del modello di Clustering

Definizioni

Un modello di clustering è una tecnica di apprendimento non supervisionato utilizzata per raggruppare osservazioni in sottoinsiemi, chiamati cluster, in modo che:

- le osservazioni all'interno dello stesso cluster siano simili tra loro
- diverse da quelle in altri cluster.

L'obiettivo principale è scoprire strutture nascoste o raggruppamenti naturali nei dati senza alcuna conoscenza predefinita delle etichette delle classi.

Obiettivo

1. realizzare una analisi mirante alla identificazione di pattern di rischio nei termini di gruppi di entità (la unità di analisi del dataset) definite da specifiche configurazioni di variabili, ognuno dei quali sia sufficientemente consistente da un punto di vista statistico ;
2. fornire criteri di interpretabilità dei risultati, va nella direzione della XAI (*Explanable Artificial Intelligence*): i risultati del modello di clustering non tipicamente output di una black box, sono direttamente interpretabili e fruibili

Dataset

Per dimostrare le potenzialità dell'uso di tali modelli, abbiamo definito un dataset virtuale, basato sull'estrazione di dati da parte del crawler semantico di ExpertAI applicato ad alcuni social media: lo scopo è quello di catturare dinamiche dei comportamenti degli utenti che possano essere messe relazione con contesti di rischio e di attività potenzialmente criminogene.

A partire da un configurazione basata su link e parole chiave, il modulo WebCrawler-NLU ricerca contenuti corrispondenti sui quali effettuare la Natural Language Understanding (NLU).
L'output dell'analisi è una serie di response nel formato json in cui, oltre all'analisi strettamente semantica e sintattica e NER (Named Entity Recognition), saranno riportate le entità di interesse per il dataset di machine learning.

I contenuti sono articolati secondo le seguenti tassonomie:

1. CRIME
2. INTELLIGENCE
3. TOPICS
4. EMOTIONS

I Dati

Il dataset virtuale rappresenta una estrazione dati da forum di utenti di quattro siti specifici, che possono contenere elementi “attenzionabili”:

1. Telegram
2. Reddit
3. Discord
4. Dw_Hydra

I fattori comuni che rendono queste piattaforme “attenzionabili” in un’indagine sono:

- Anonimato o pseudonimia (difficile collegare identità reali).
- Canali chiusi o cifrati che limitano il monitoraggio.
- Presenza di comunità tematiche che possono aggregare soggetti interessati ad attività illegali o estremiste.
- Facilità di diffusione di contenuti (file, link, manuali).

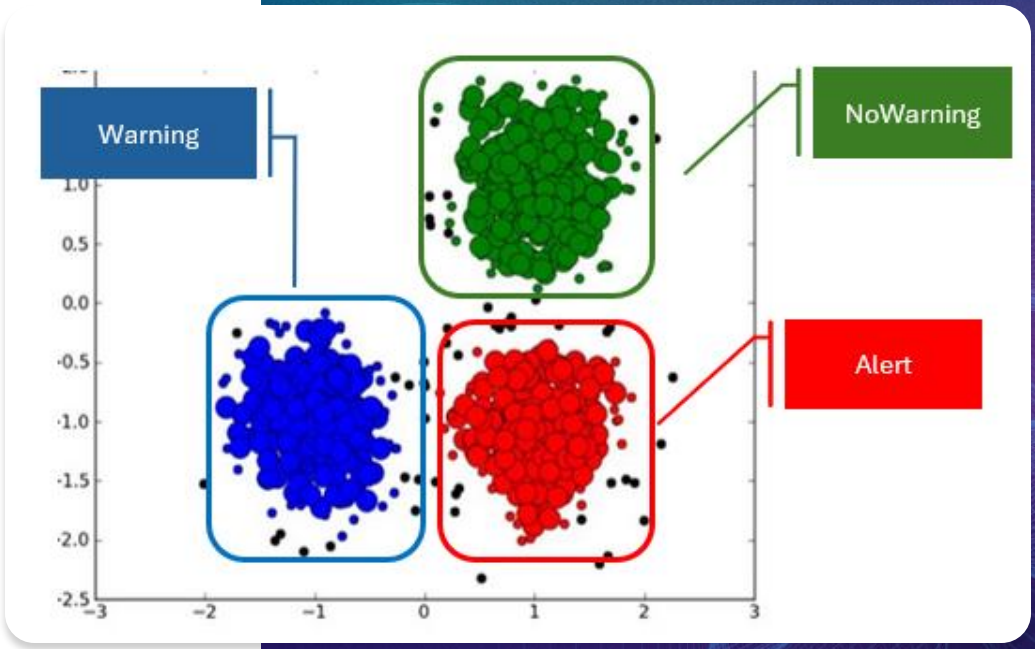
Le variabili

Per ognuno dei siti abbiamo selezionato le 41 seguenti variabili, raggruppate per tassonomia,

Tassonomia	Variabile
Sentiment	Positivity, Negativity
Crime	Traffico Armi, Droga, Terrorismo Criminalità Informatica Reati, Frode Truffa. Delitti Ordine pubblico
Intelligence	Agitazioni, Conflitti Guerre, Infrastrutture Lavoro, Politica Religioni, Fedi Sociale
Topics	Amministrazione pubblica Armi, Chimica Criminalità, Ebraismo Ecologia, Energia Finanza, Islam Istituzioni, Parlamento Polizia, Religione Reti, Sette Software, Università
Emotions	Odio, Rabbia Paura, Esasperazione

Mappa di rischio

Il modello di Clustering ha lo scopo di generare una “mappa di rischio”, in cui sarà possibile identificare pattern, configurazioni specifiche di variabili del dataset, ognuno corrispondente ad una fascia di rischio: dall’assenza totale al rischio massimo, con un gradiente derivante dai risultati del modello.
La mappa di rischio prodotta dal modello di clustering può essere rappresentata dalla seguente figura.



La mappa è il prodotto della segmentazione dello spazio delle variabili al fine di definire gruppi delle entità analizzate, omogenei e, soprattutto distinti dagli altri gruppo: ognuno di questi gruppi condivide un “pattern” di comportamento, definito da una configurazione specifica del set delle variabili utilizzate.

Le tre aree di rischio sono:

1. “NoWarning”,
2. “Warning”
3. “Alert”.

Sviluppo del modello di Clustering

Framework AutoML

Abbiamo progettato un framework sperimentale per lo sviluppo di modelli di clustering basato sul paradigma ormai consolidato della AutoML.
Abbiamo previsto l'uso dei seguenti modelli:

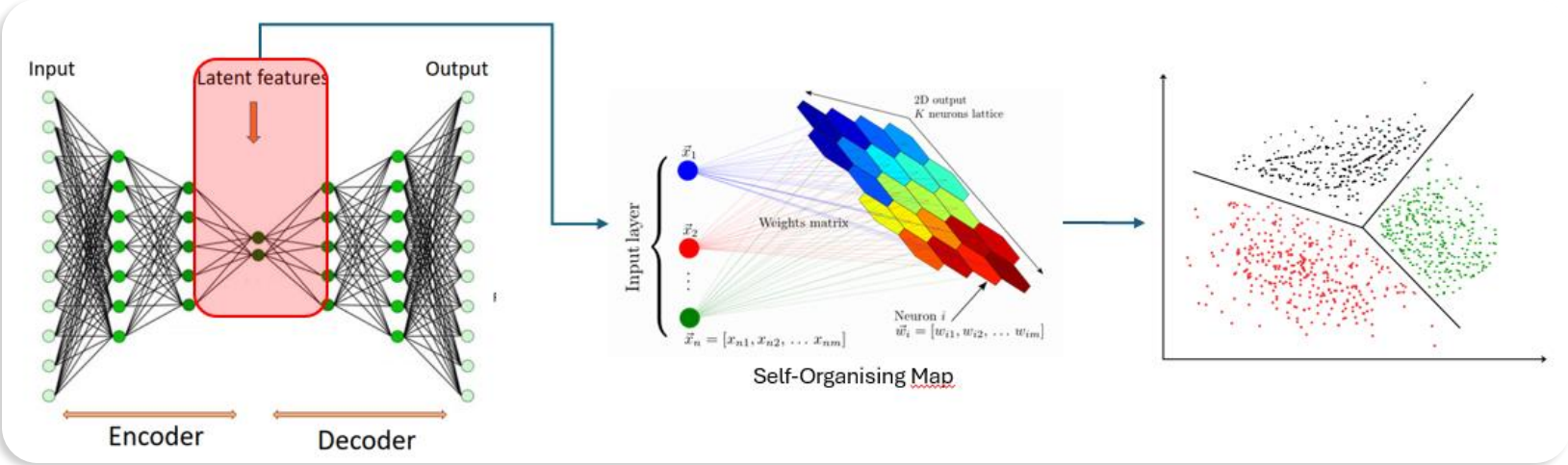
1. Bisecting k-means;
2. Gaussian Mixture
3. K-means
4. Affinity propagation
5. Mean shift
6. Spectral clustering
7. DBSCAN
8. HDBSCAN
9. Optics
10. Self-Organising Maps (Neural Network)
11. Deep clustering

Oltre ai modelli ampiamente noti in letteratura, abbiamo voluto verificare le potenzialità del modello definito *Deep Clustering*

Deep Clustering

Descrizione funzionale

Il modello Deep Clustering sperimentale è costituto da una pipeline illustrata in figura



Gli stage della pipeline:

1. **AutoEncoder**: I modelli di questo tipo sono ormai ampiamente utilizzati in letteratura per gli scopi più disparati, Nel nostro caso, il compito di questo modello è l'estrazione delle feature latenti, una rappresentazione più significativa e compatta (un minor numero di variabili) del dato in ingresso da parte del sub-modulo dell'Encoder.
2. Queste nuove feature costituiscono il nuovo input al modello di clustering **Self Organising Map**

Autoencoder

Un autoencoder è una rete neurale artificiale progettata per apprendere una rappresentazione compatta (codifica) dei dati in input e poi ricostruire l'input originale a partire da quella rappresentazione. L'obiettivo è comprimere i dati e poi "decomprimerli", cercando di ridurre al minimo la perdita di informazione, minimizzando il *reconstruction error*.
Un autoencoder è composto da tre parti principali:

1. **Encoder (codificatore**)
   Trasforma l'input originale in una rappresentazione più piccola e astratta, detta codice o embedding. È una funzione $f(x)$ che riduce la dimensionalità.
2. **Latent Space (spazio latente**)
   Lo "spazio compresso" in cui i dati vengono rappresentati. È una forma di compressione non lineare, dove i pattern nascosti vengono catturati.
3. **Decoder (decodificatore**)
   Ricostruisce l'input originale a partire dal codice latente. È una funzione $g(z)$ che cerca di riportare i dati alla forma iniziale.

Un autoencoder non ha etichette (è un modello non supervisionato).
Il suo compito è minimizzare l'errore di ricostruzione: $L(x,x\hat{})=\|x-x\hat{}\|^2$

La pipeline

Nel nostro caso usiamo l'autoencoder come riduttore di dimensionalità ed estrattore di feature latenti, per poi applicare un modello di clustering (es. SOM, k-means, DBSCAN, ecc.) nello spazio latente.

In pratica:

1. Autoencoder come estrattore di feature latenti dalla bottleneck (o latent space layer). Per ottenere una rappresentazione compatta e non lineare, per ridurre il rumore e mettere in evidenza le caratteristiche salienti.
2. Clustering nello spazio latente

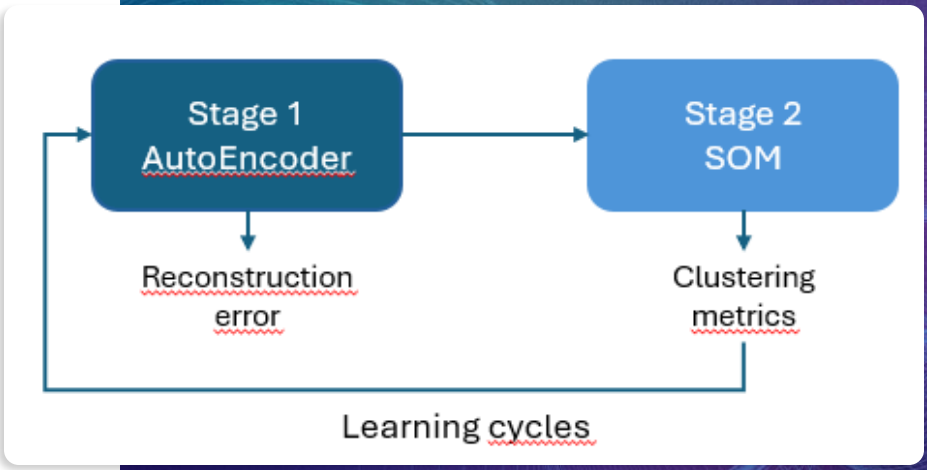
Il clustering sullo spazio originale può essere difficile (dati ad alta dimensionalità, rumore). Lo spazio latente è più "pulito" e meno ridondante, per ottenere cluster più netti e separabili.

3. Self-Organising Map (SOM)

La SOM proietta i dati in uno spazio 2D preservando le relazioni topologiche. Se alimentata con feature latenti di qualità può produrre mappe molto più interpretabili.

L'algoritmo

L’esecuzione della pipeline non è semplicemente sequenziale, in cui alla conclusione dello stage AutoEncoder, con l’estrazione delle feature dal bottleneck, è eseguito lo stage di clustering.
Lo scopo della pipeline è un modello di clusterizzazione valutabile nei termini di una metrica, come la **silhouette**.
Nel caso dell’AutoEncoder, la metrica è **l’errore di ricostruzione**, che indica praticamente la capacità delle feature latenti di “rappresentare” il dato in ingresso. In questa chiave, una adeguata capacità delle feature dovrebbe teoricamente comportare un adeguato modello di clustering.
Ma in realtà, questa corrispondenza deve essere verificata: non possiamo avere la certezza che al diminuire dell’errore dell’AutoEncoder corrisponda un valore alto della metrica di clustering e quindi una corretta separazione tra le classi. Si tratta del classico problema della curva di apprendimento, che non sempre ha un andamento monotono.
La pipeline è eseguita in cicli di stage, come illustrato in figura.



I cicli della fase di learning

I cicli di learning sono costituiti da

- un numero di epoche di training dell'AutoEncoder e
- un numero di epoche del modello di clustering SOM.

Relativamente a quest'ultimo, abbiamo apportato modifiche al codice sorgente delle librerie utilizzate (GEMA e TF2 SOM) per implementare la logica dell'early stopping: poiché non è certo che al progredire delle epoche la metrica utilizzata assuma valori sempre più alti (andamento monotono), manteniamo in memoria il modello migliore durante la fase di learning basato sul valore della metrica scelta e salveremo il modello. La metrica del clustering è il criterio guida di tutta la pipeline.

Ad ogni ciclo l'AutoEncoder non viene re-inizializzato, ma opera in modalità di incremental learning. I cicli di learning avranno il ruolo quindi di verificare la relazione tra l'errore di ricostruzione dell'AutoEncoder e il valore della metrica di valutazione del clustering.

I vantaggi di questo modello sono evidenti:

1. Vantaggio del protocollo: ricerca del migliore modello di clustering possibile;
2. Efficienza: il modello AutoEncoder sfrutta il processamento sulle GPU.

Interpretabilità

Un modello è interpretabile se possiamo capire:

- Come prende le decisioni (quali feature influenzano il risultato).
- Perché ha dato un certo output in un caso specifico.

Le caratteristiche di un modello che soddisfa questo requisito possono essere così riassunte.

- Fiducia e trasparenza.

  Se il modello è una “black box”, gli utenti e i decisori fanno fatica a fidarsi. La interpretabilità favorisce una maggiore accettazione e adozione.

- Debug e miglioramento del modello.

  Se si comprendono quali feature guidano le decisioni, è possibile individuare errori, bias o dati inconsistenti.

- Compliance normativa.

  In molti settori (bancario, assicurativo, sanitario, PA) le normative richiedono spiegazioni delle decisioni automatizzate.

- Etica e fairness.

  Un modello non interpretabile può introdurre bias discriminatori (es. razza, genere, età). L’interpretabilità aiuta a verificare che le decisioni siano eque.

- Conoscenza del dominio.

  Un modello interpretabile può fornire insight utili agli esperti di dominio.

La metrica di valutazione del modello di clustering

Silhouette

Nella nostra sperimentazione abbiamo utilizzato la *Silhouette:* una metrica che valuta la qualità di un modello di clustering, misurando quanto bene ogni punto dati si adatta al proprio cluster rispetto ai cluster vicini.

Varia tra -1 e +1, dove:

- un valore vicino a +1 indica un'ottima appartenenza al proprio cluster e una buona separazione dagli altri,
- 0 suggerisce che il punto è vicino al confine tra due cluster,
- un valore negativo indica che il punto potrebbe essere stato assegnato al cluster sbagliato.

Questo score è utile per determinare il numero ottimale di cluster.

Template di esecuzione

L'input è standardizzato.
Il template di esecuzione riportato nella figura.

Come anticipato, abbiamo implementato la logica **dell'early stopping**: sia l'auto-encoder che la Self Organising Map hanno il parametro early_stop_patience.
Se l'errore di ricostruzione dell'auto-encoder non si riduce, o la silhouette dele modello di clustering non aumenta per un numero determinato di epoche, il training si interrompe.
Ad ogni miglioramento delle metriche vengono salvati i pesi e i codebook (centroidi) del modello. Alla fine del training sarà disponibile il best model.


```
"deep_clustering": {
        "use": true,
        "template": {
            "use": true,
            "autoencoder": {
                    "model": {
                            "hidden_layers": 4,
                            "hidden_units": 128,
                            "bottleneck_units": 24,
                            "activation": "sigmoid",
                                "activity_regularizer": {
                                "reg": "none",
                                "penalty": 0.1
                            },
                            "kernel_regularizer": {
                                "reg": "none",
                                "penalty": 0.1
                            },
                            "compile": {
                                "optimizers": "adam",
                                "learning_rate": 0.001
                            }
                    },
                    "fit": {
                        "cycles_count": 5,
                        "epochs": 100,
                        "early_stop_patience": 50,
                        "batch_size": 128,
                        "shuffle": true,
                        "validation_data": true,
                        "validation_steps": "None"
                        "validation_freq": 1
                        "verbose": 1
                        }
                    },
            "clustering": {
                "model": "som_gema",
                "size":[6],
                "period":50,
                "initial_lr":0.01,
                "initial_neighbourhood":4,
                "distance":"euclidean",
                "use_decay":false,
                "presentation":"random",
                "weights":"sample",
                "evaluation_interval":1,
                "reinforcement_epochs":0,
                "reinforcement_extension":1,
                "reinforcement_rate_decay":1,
                "early_stop_patience": 30
            }
        }
    }
```

Analisi del modello (1)

- **La silhouette del modello è 0.9977705**, decisamente alta, ed è una indicazione iniziale della qualità del modello dal punto di vista della separazione dei gruppo della mappa.
- La fase successiva è quindi l'analisi dettagliata del modello, cioè la valutazione della consistenza statistica dei cluster individuati, e cioè dei record in essi contenuti, al fine di definire una mappa del rischio attraverso una "taggatura" dei cluster: la mappa finale sarà costituita da gruppi di cluster identificati da "tag" semantici basati sulle proprietà statistiche dei record, proprietà che dovranno essere statisticamente distintive di ogni gruppo.

L'analisi è quindi articolata nelle seguenti fasi:

- Analisi dei record per ogni cluster, attraverso una analisi invariata delle variabili.
- Analisi comparata tra i cluster per individuare similitudini e, soprattutto differenze nelle distribuzioni per la individuazione di tag e pattern.

Analisi del modello (2)

Un file prodotto dall'analisi contiene le seguenti informazioni per ogni cluster:

- Identificativo del cluster;
- numero di record contenuti;
- percentuale di record sul totale:
- per ogni variabile numerica una statistica di base sui valori dei re:
    - Minimo
    - Q1
    - Mediana
    - Q3
    - Massimo
    - Media
    - Deviazione standard
    - Intervallo di confidenza

Sono questi dati che ci consentiranno di valutare la separabilità dei cluster, e cioè la identificazione di pattern specifici per ogni "classe"

- NoWarning,
- Warning,
- Alert

Analisi del modello (3)

Riportiamo un esempio tratto dal file json prodotto dall'analisi, (cluster_profiles.json)
In primo luogo, esponiamo i dati dei cluster relativi ai numeri dei record, tenendo presente che la mappa della Self Organising Map selezionata è una 6 x 6 (36 cluster)

Cluster	Rilevazioni	Percentuale sul totale
C30	405	21.33
C33	95	5.0
C35	1399	73.67

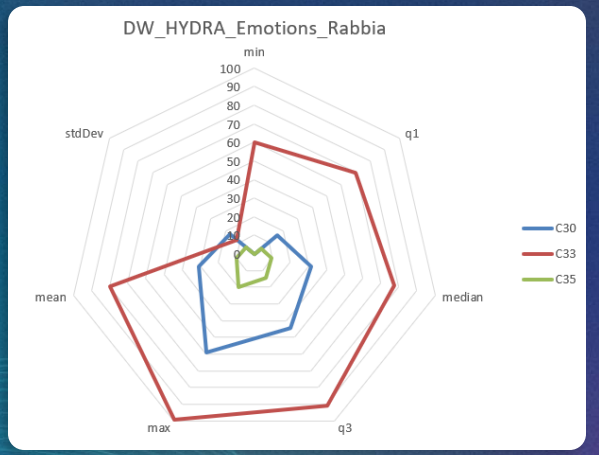
È interessante la concentrazione delle 2000 rilevazioni su soli 3 cluster. **Occorre verificare se ognuno di questi è "classificabile" con le classi specificate.**

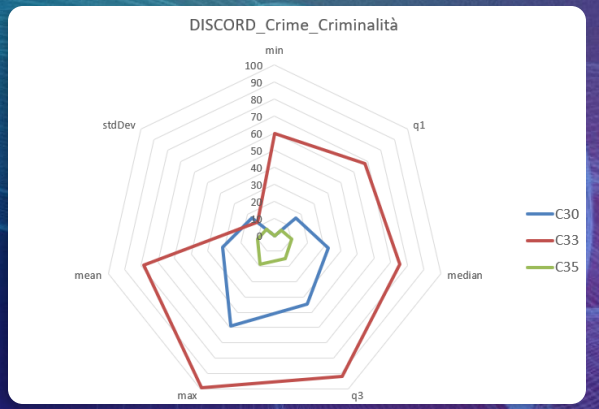
Analisi del modello (4)

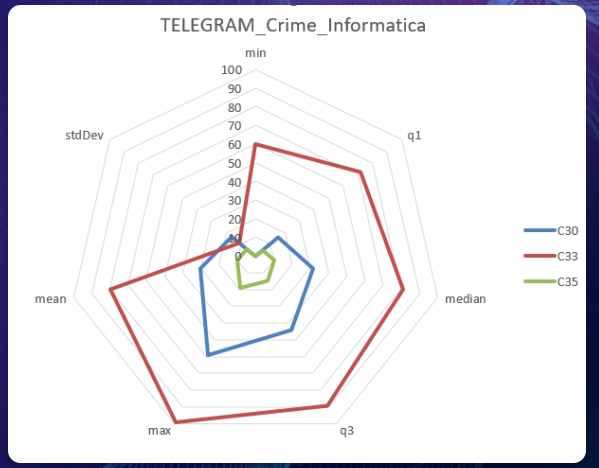
Come esempio di analisi, selezioniamo la variabile **DISCORD_Crime_Criminalità**, che indica il livello di “attivazione” di questo tipo di contenuto rilevato dal motore WEB-NLU. In base alla statistica univariata riportiamo il grafico seguente di tipo “radar”.

Questo tipo di grafico evidenzia immediatamente il pattern di variazione all’interno dei cluster.
Analogamente per le altre variabili TELEGRAM_Crime_Informatica e DW_HYDRA_Emotions_Rabbia.

Da questi esempi di analisi, possiamo dedure che il cluster C33 ha la connotazione “Alert” mentre C35 potrebbe avere una connotazione di NoWarning.



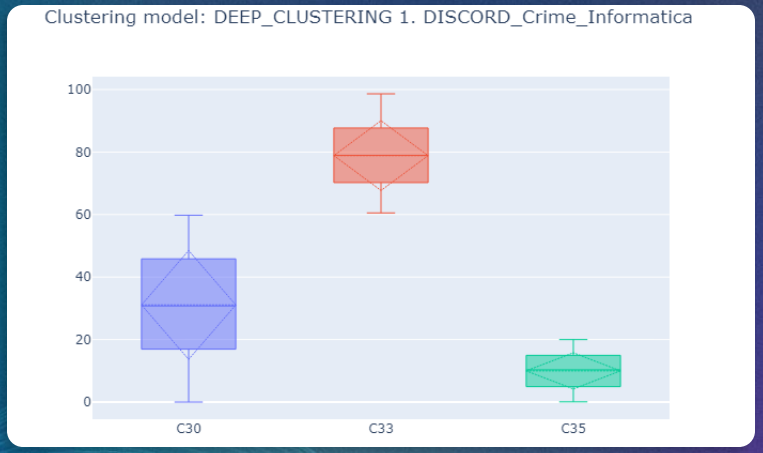


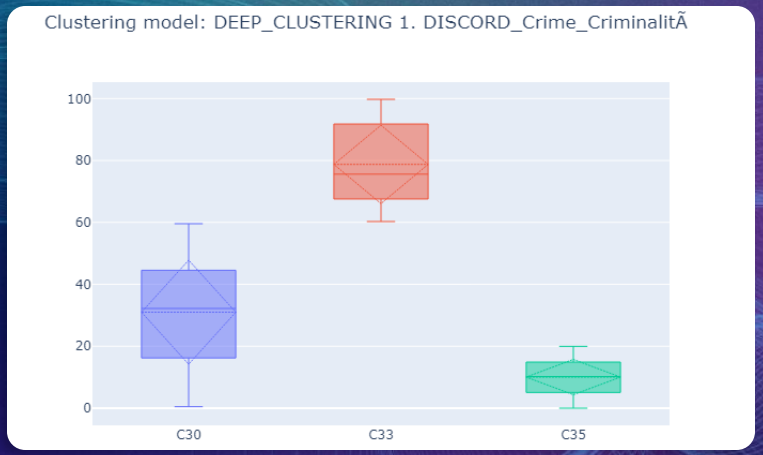


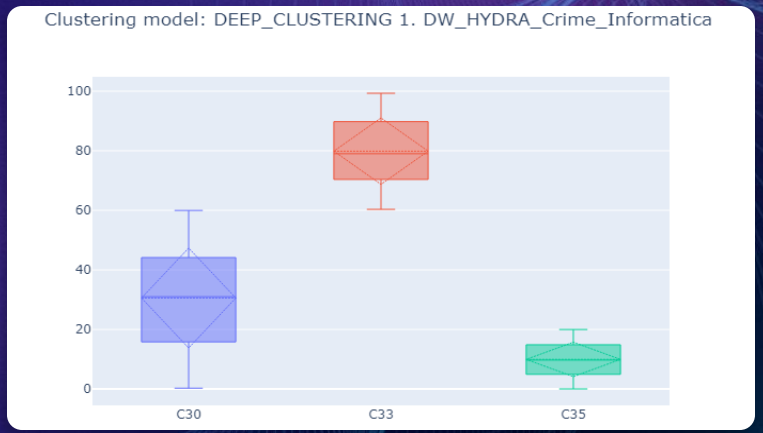
Analisi del modello (5)

Molto utili sono anche i grafici del confronto tra le distribuzioni dei valori delle variabili all'interno dei cluster. In questo documento ne esporremo alcuni tra i 164 prodotti dal modulo di analisi del modello.

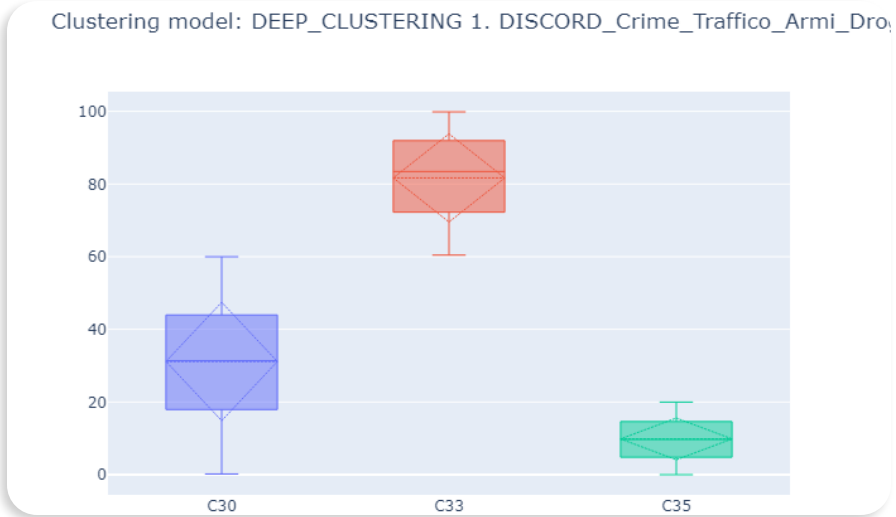
Da tutti i grafici possiamo confermare che al cluster C33 è possible assegnare il tag "Alert" mentre al cluster C35 potrebbe avere il tag di "NoWarning". Il cluster C30 non avrebbe una connotazione netta: essenzialmente NoWarning con qualche segnale di Warning.

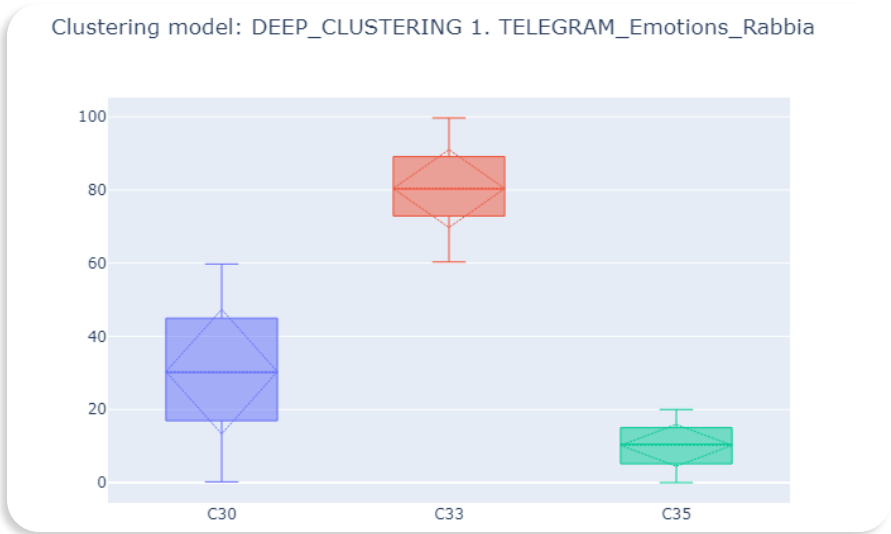


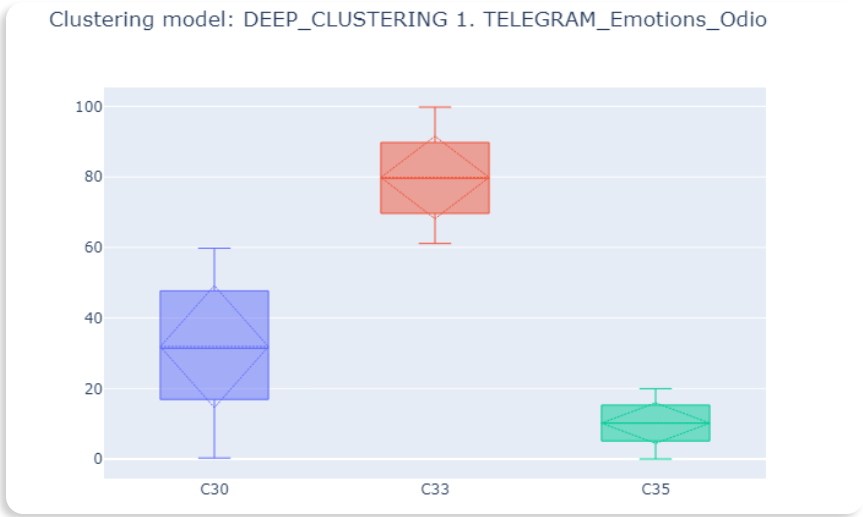


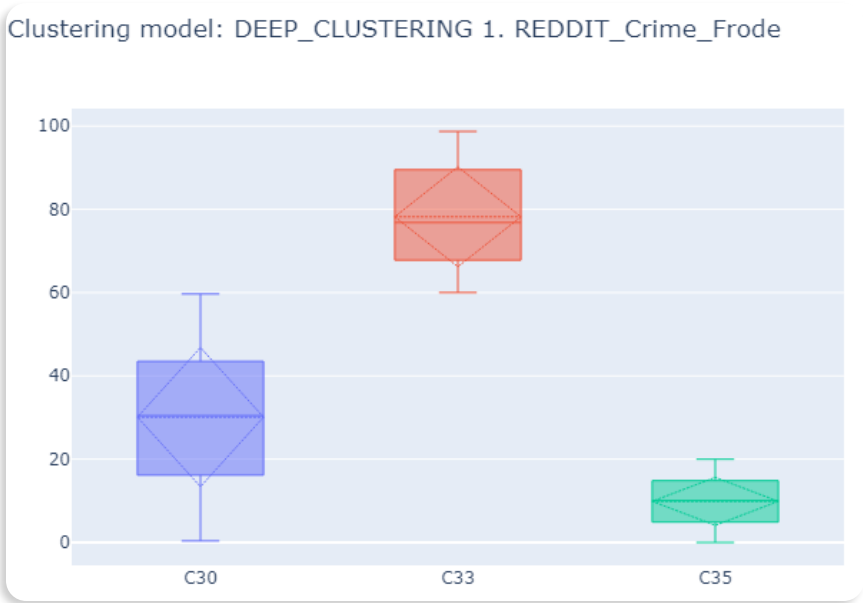


Analisi del modello (6)









Inferenza.
Il modello in ambiente di produzione

Il modello, in sede di inferenza, dovrà “classificare” un record in entrata, una rilevazione compiuta dal modulo WEB-NLU sui forum indicati.

Non si tratta naturalmente di una classificazione di tipo supervisionato, ma l’operazione di caratterizzazione dei cluster (taggatura) in sede di analisi del modello ha definito tre classi: **NoWarning, Warning e Alert**, in corrispondenza dei tre cluster estratti.

Pertanto, un record in entrata verrà assegnato ad un cluster e, conseguentemente, alla classe cui il cluster è stato associato in fase di analisi del modello.

Da questo punto di vista, l’inferenza è uno strumento di monitoring delle fonti web selezionate e consente di segnalare eventuali dinamiche da attenzionare.

Inferenza. Il container

La realizzazione del modulo di inferenza è costituita dalla creazione di un container che contiene:

- l'ambiente virtuale secondo le specifiche contenute nel file sicuri.yaml, così come esportato dall'ambiente Anaconda. Attraverso il file sicuri.yaml verranno installate tutte le librerie necessarie al framework sviluppato;
- copia del progetto del framework;
- due folder:
    - transform, contente lo script di transform dei dati (StandardScaler), il template e il modello di transform (file *.pkl);
    - model, contenente i file del modello (*artifacts*)
- script di inferenza;
- file della classe Predictor, che esegue l'inferenza.

Humanativa Group Srl

Sede Legale ed Operativa

Viale dell'Oceano Pacifico, 66 - 00144 Roma (RM)

Tel. +39.06.89161268 · Fax +39.06.89161316

Sedi Operative

Via Termini, 6 - 82010 Arpaise (BN) · Tel. +39.06.89161268

Via Confienza, 10 - 10122 Torino (TO) · Tel. +39.06.89161268

www.humanativaspa.it · info@humanativaspa.it

facebook.com/HumanativaGroupSpA

linkedin.com/company/humanativa-group-spa