Supervised Clustering con variabile target: *un'applicazione in Teleassistenza*

Università Campus Bio-Medico di Roma

Collaboratori

Alessia Rossi
Fabio Di Gregorio
Ignazio Emanuele Piccichè
Martina Bertazzoni

Challenge Campus Biomedico

1 Problema

- La **Piattaforma Nazionale di Telemedicina (PNT)** coordina i processi di telemedicina in Italia, con l'obiettivo di armonizzarli a livello nazionale. Tra i principali obiettivi della telemedicina ci sono la gestione semplificata delle malattie croniche, la riduzione delle ospedalizzazioni e l'uso di strumenti innovativi per migliorare la qualità dei servizi sanitari.

- La **Teleassistenza**, un servizio incluso nella piattaforma, permette visite mediche a distanza tra pazienti e professionisti sanitari. La piattaforma registra ogni intervento per garantire una documentazione completa e accurata.

- L’obiettivo della challenge si concentra sul **profilare i pazienti** in base al loro contributo all'incremento dell'uso della teleassistenza, utilizzando tecniche di clustering per identificare gruppi con comportamenti simili.

2 Architettura del Progetto

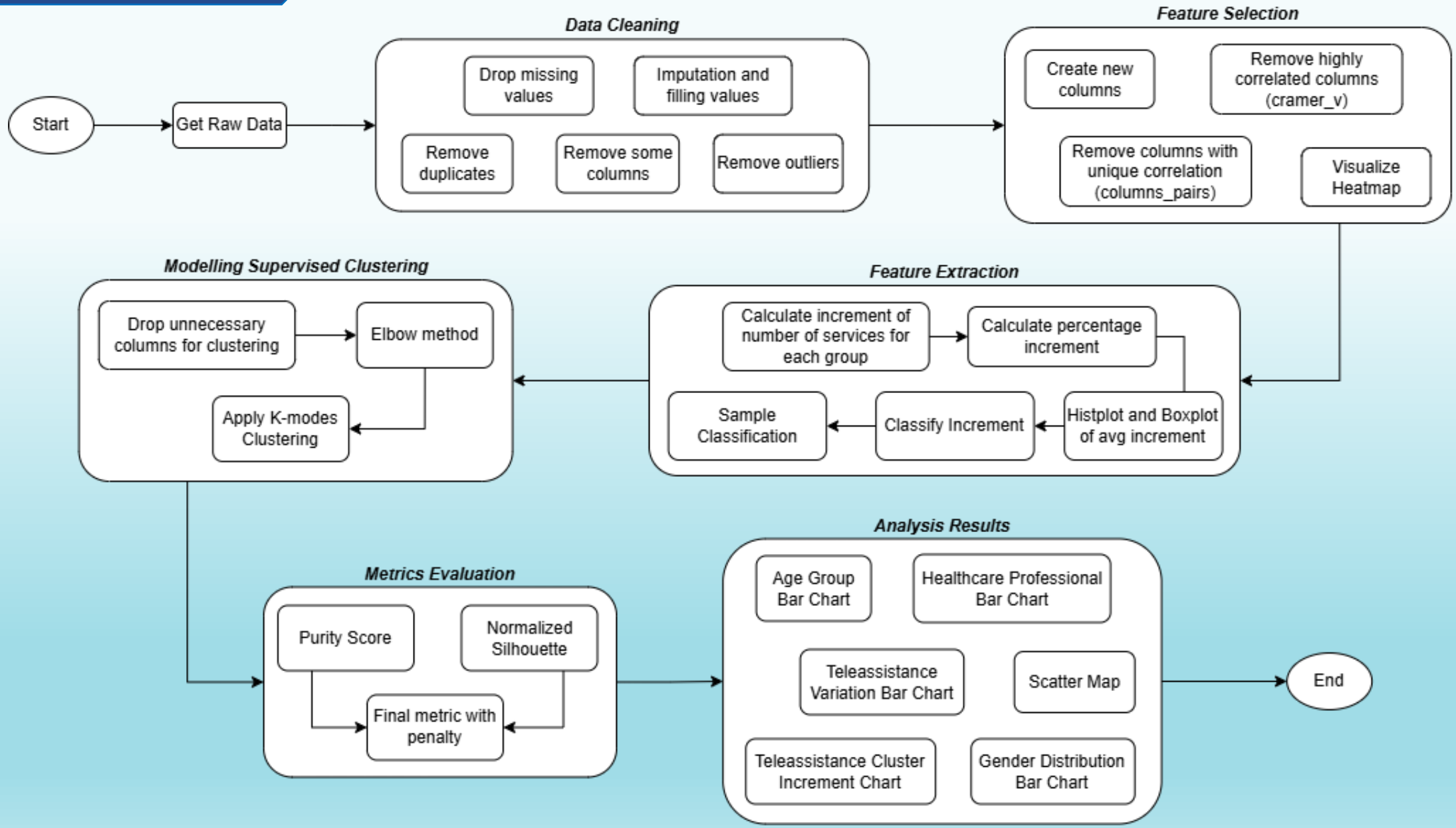
- **SUDDIVISIONE IN CARTELLE:** per una gestione efficiente dei dati e del codice.

data/	dati grezzi e processati
graphs/	grafici e plot
logs&metrics/	log e metriche di performance
models/	modelli finali salvati per utilizzi futuri
notebooks/:	notebook Jupyter contenenti lo sviluppo, l'analisi esplorativa dei dati (EDA), e gli esperimenti
src/:	script per la preparazione dei dati, modelling, evaluation e visualization

- **FILE DI CONFIGURAZIONE: config.yaml** e **requirements.txt** gestiscono le impostazioni e le dipendenze necessarie al progetto, mentre **main.py** esegue l'intero processo.

```
├── data/
│   ├── processed/
│   ├── raw/
│   └── README.md
├── graphs/
│   ├── analysis/
├── logs&metrics/
├── models/
├── myLib/
├── notebooks/
│   ├── development/
│   ├── EDA/
│   └── experiments/
├── src/
│   ├── data_prep/
│   ├── __init__.py
├── .gitattributes
├── .gitignore
├── config.yaml
├── main.py
├── README.md
└── requirements.txt
```

3 Pipeline


Start
Get Raw Data
Data Cleaning
Drop missing values
Imputation and filling values
Remove duplicates
Remove some columns
Remove outliers
Feature Selection
Create new columns
Remove highly correlated columns (cramer_v)
Remove columns with unique correlation (columns_pairs)
Visualize Heatmap
Feature Extraction
Calculate increment of number of services for each group
Calculate percentage increment
Sample Classification
Classify Increment
Histplot and Boxplot of avg increment
Modelling Supervised Clustering
Drop unnecessary columns for clustering
Elbow method
Apply K-modes Clustering
Metrics Evaluation
Purity Score
Normalized Silhouette
Final metric with penalty
Analysis Results
Age Group Bar Chart
Healthcare Professional Bar Chart
Teleassistance Variation Bar Chart
Scatter Map
Teleassistance Cluster Increment Chart
Gender Distribution Bar Chart
End

4 Preprocessing – Data Cleaning

OBIETTIVO: Preparazione dei dati grezzi, trasformandoli in un formato pulito e coerente per garantire analisi precise e affidabili

1. **Rimozione delle Colonne con Valori Mancanti Eccessivi:**
   Rimuoviamo colonne con oltre il 60% di valori mancanti, eccetto data_disdetta
2. **Imputazione dei Valori Mancanti:**
   Comune_residenza: Completiamo i dati mancanti utilizzando i codici ISTAT dei comuni
3. **Gestione delle righe incomplete:**
   Rimuoviamo righe con valori mancanti in colonne cruciali e quelle con data_disdetta non nulli
4. **Trattamento degli Outlier:**
   Filtriamo gli outlier con il metodo del boxplot per evitare distorsioni nei dati
5. **Smussatura dei Dati Rumorosi:**
   Applichiamo una media mobile per ridurre il rumore nei dati
6. **Rimozione dei Duplicati:**
   Eliminiamo righe duplicate per garantire l'unicità dei record

4 Preprocessing – Feature Selection

Rimozione e Pulizia Dati:

- Pulisce colonne con codici non coerenti come *codice_struttura_erogazione* e rimozione della colonna *data_disdetta*
- Rimuove righe duplicate o non informative e gestisce gli outlier nelle colonne come durata_erogazione (collegato al punto successivo)

Correlazione e Rimozione Colonne:

- Analizza la correlazione tra variabili categoriali usando la matrice di correlazione (**Cramér's V**).
- Rimuove colonne altamente correlate per ridurre la complessità e multicollinearità, migliorando le prestazioni del modello.

Creazione di Nuove Caratteristiche:

Crea colonne come

- *durata_erogazione*: differenza tra *ora_inizio_erogazione* e *ora_fine_erogazione*
- *fascia_eta*: categorizzazione in gruppi demografici
- colonne temporali: anno e quadrimestre

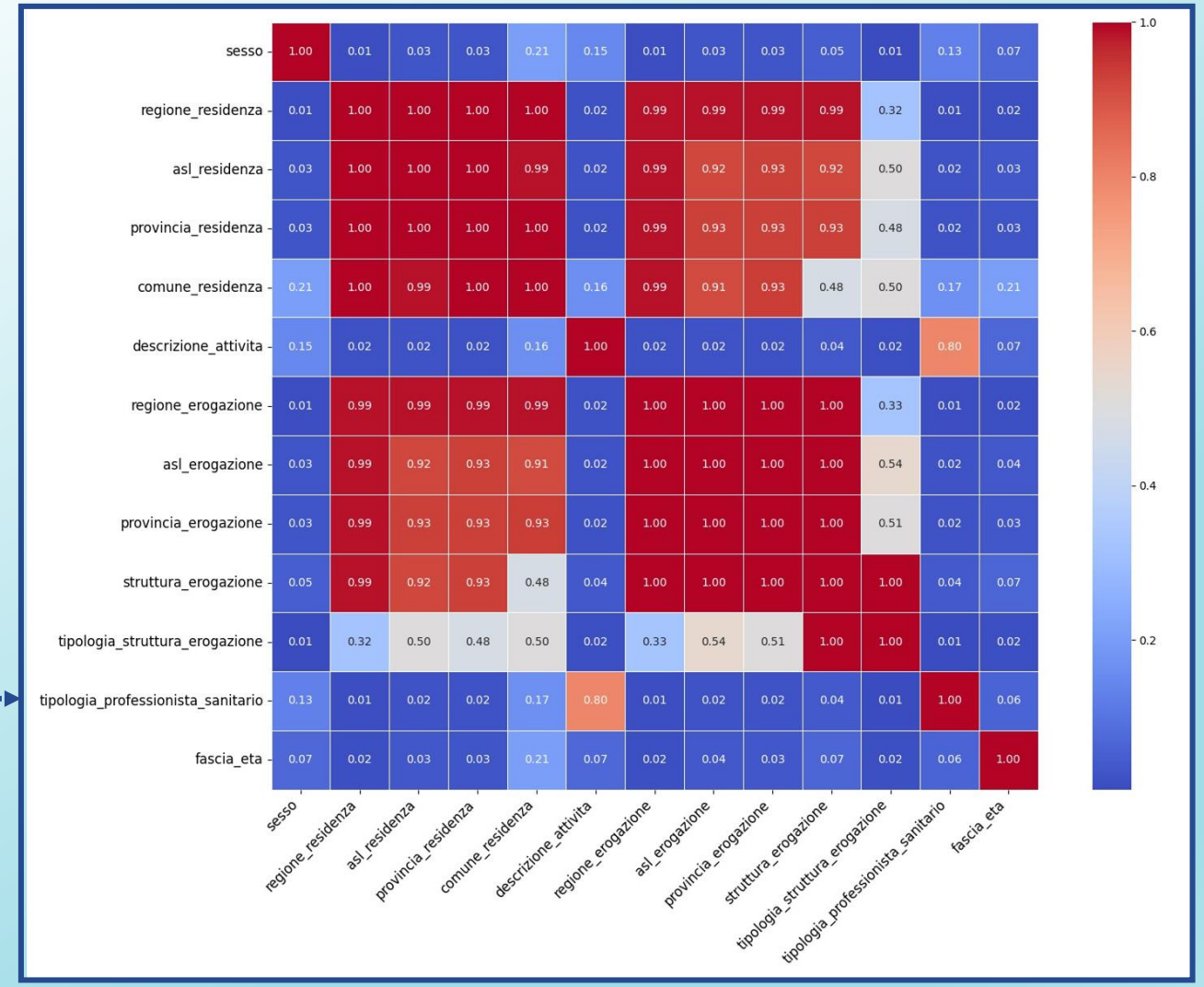
4 Preprocessing – Feature Selection

Esportazione del Dataset:

Salva il dataset finale con le caratteristiche selezionate in un file parquet per utilizzi futuri

Visualizzazione della Correlazione:

Crea e salva una heatmap che mostra la matrice di correlazione



4 Preprocessing – Feature Extraction

Raggruppamento dei Dati: 1

Il codice raggruppa il dataset in base a colonne specifiche [anno, quadrimestre, fascia_eta, regione_residenza], contando il numero di servizi per ogni gruppo/anno.

Calcolo dell'Incremento: 2

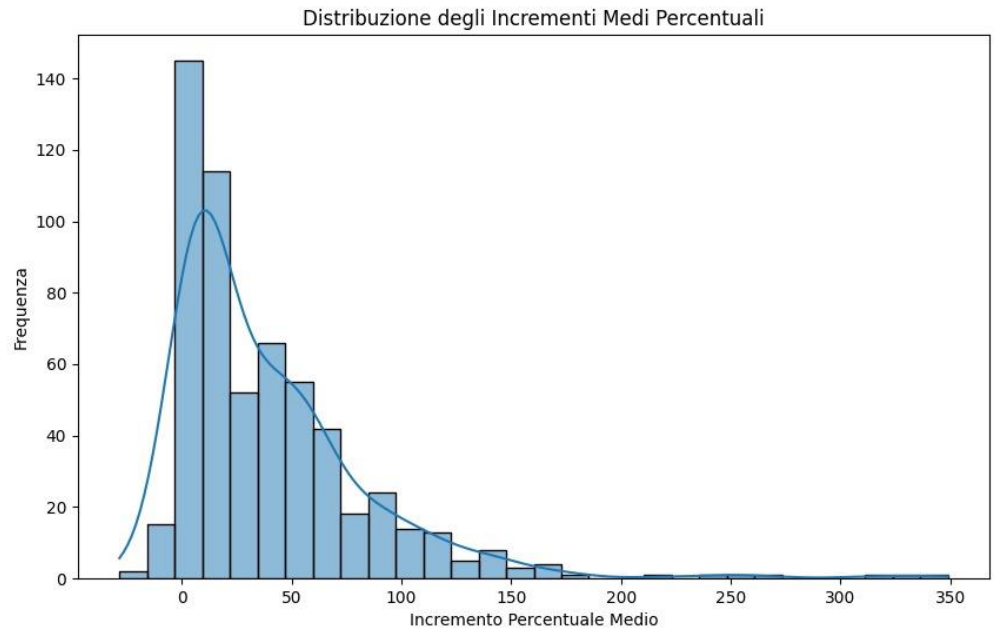
Calcoliamo l'incremento percentuale per ciascun gruppo tra anni consecutivi, e successivamente si calcola la media degli incrementi percentuali per tutti i gruppi.

Classificazione dell'Incremento: 3

Gli incrementi percentuali medi vengono classificati in categorie

	Categoria Incremento	Range di Incremento
1	decrement	< 0%
2	low_increment	0 - 15%
3	medium_increment	15 - 40%
4	high_increment	> 40%

Visualizzazione: 4



Integrazione nel dataset: 5

La classificazione degli incrementi viene aggiunta al dataset originale.

Esportazione del risultato 6

4 Preprocessing – Feature Extraction

Raggruppamento dei Dati: 1

Il codice raggruppa il dataset in base a colonne specifiche [anno, quadrimestre, fascia_eta, regione_residenza], contando il numero di servizi per ogni gruppo/anno.

Calcolo dell'Incremento: 2

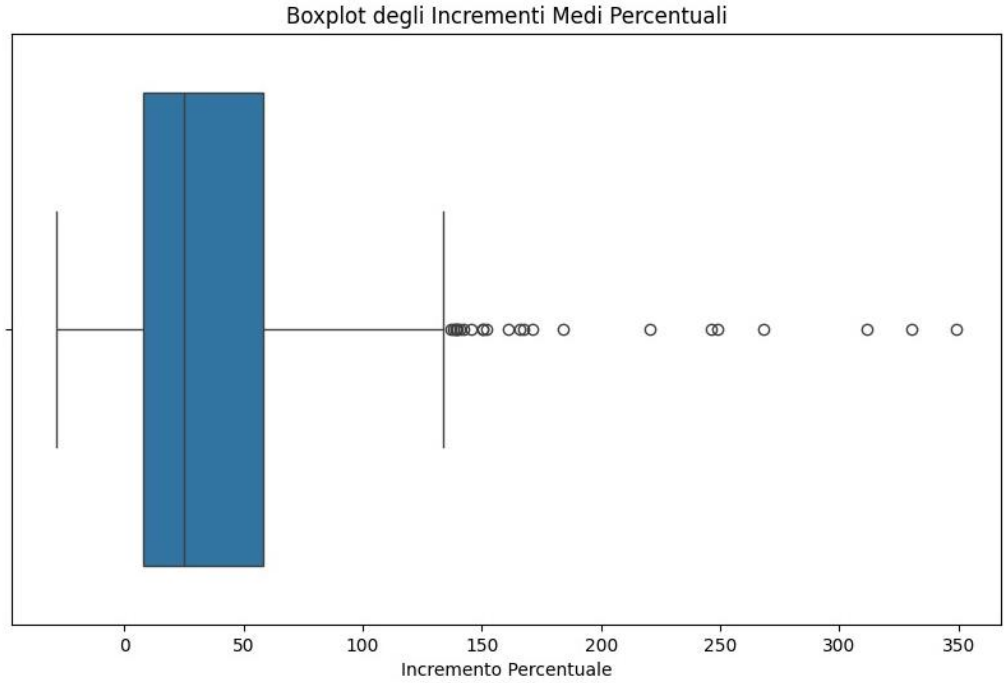
Calcoliamo l'incremento percentuale per ciascun gruppo tra anni consecutivi, e successivamente si calcola la media degli incrementi percentuali per tutti i gruppi.

Classificazione dell'Incremento: 3

Gli incrementi percentuali medi vengono classificati in categorie

	Categoria Incremento	Range di Incremento
1	decrement	< 0%
2	low_increment	0 - 15%
3	medium_increment	15 - 40%
4	high_increment	> 40%

Visualizzazione: 4



Integrazione nel dataset: 5

La classificazione degli incrementi viene aggiunta al dataset originale.

Esportazione del risultato 6

5 Modelling

Clustering con K-modes

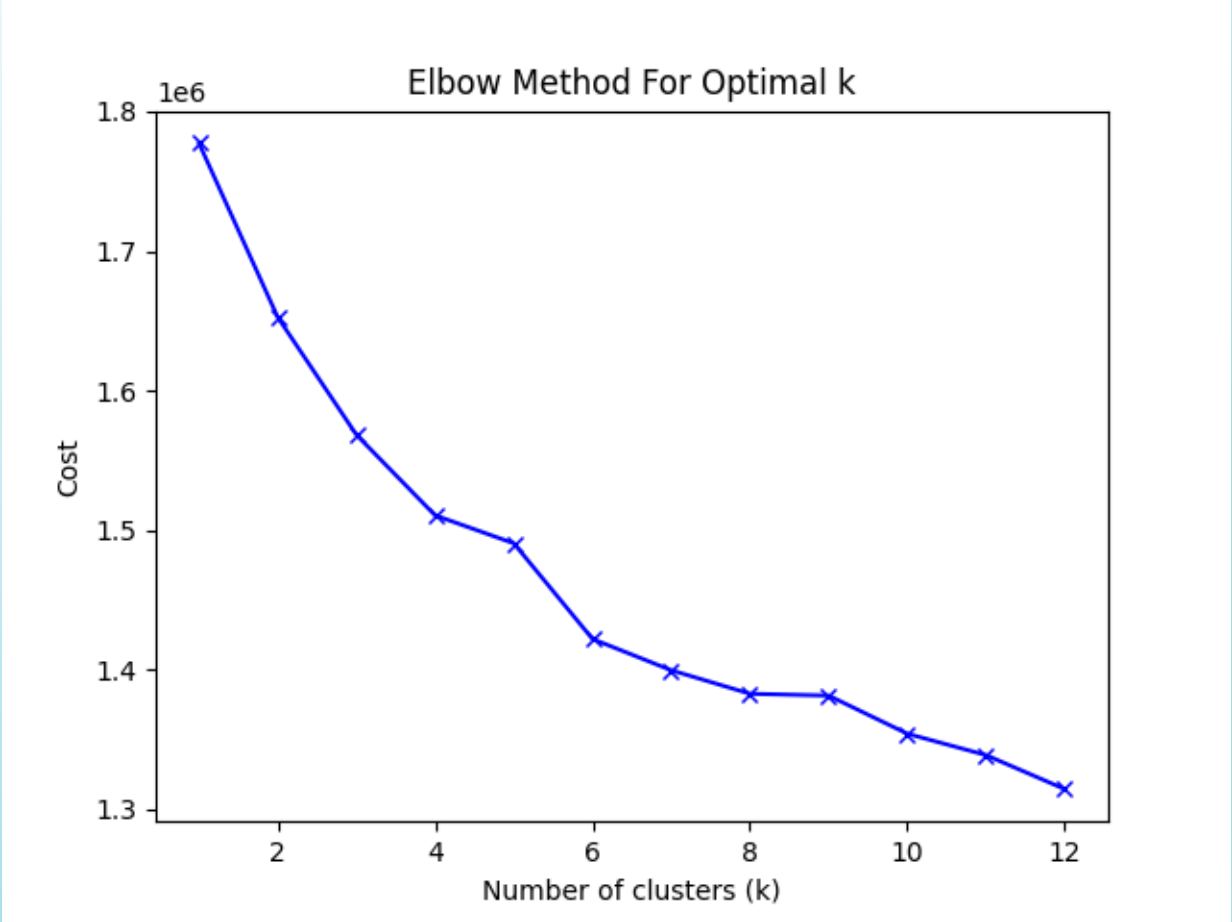
Per dati categorici

Elbow Method:

Per determinare il numero ottimale di cluster.

Esegue il clustering con un numero variabile di cluster, da 1 fino a un massimo definito (*max_clusters*), e per ogni numero di cluster, calcola il costo (la somma delle dissimilarità all'interno dei cluster).

Il risultato viene poi tracciato su un grafico, permettendo di individuare il punto in cui la riduzione del costo comincia a rallentare, suggerendo il numero ottimale di cluster.



5 Modelling

Clustering con K-modes

È una variante dell'algoritmo **K-Means**, progettato appositamente per gestire dati categorici in cui l'uso della distanza euclidea non è appropriato. Usa:

Matching Dissimilarity:	**Modalità:**
Confronta due elementi categorici misurando quante variabili tra due punti non coincidono	Il valore categorico più frequente per ciascuna caratteristica all'interno del cluster diventa il "centroide" del cluster.

Inizializzazione: Vengono scelti casualmente delle modalità (centroidi) iniziali dai dati

Assegnazione dei punti ai cluster: Ogni osservazione viene assegnata al cluster con il centroide più simile, secondo la dissimilarità categoriale

Aggiornamento dei centroidi: I centroidi vengono aggiornati per minimizzare la dissimilarità totale tra le osservazioni e i centroidi assegnati

Ripetizione: Il processo continua finché non viene raggiunta la convergenza, cioè quando i cluster non cambiano più

Inizializzazione di 15: l'algoritmo viene eseguito 15 volte con diverse condizioni iniziali per scegliere i centroidi che portano a dei cluster migliori.

Clustering con K-modes

È una variante dell'algoritmo **K-Means**, progettato appositamente per gestire dati categorici in cui l'uso della distanza euclidea non è appropriato.

Caratteristica	K-Means	K-Modes
Tipo di dati	Dati numerici	Dati categorici
Distanza utilizzata	Distanza euclidea	Distanza di matching (disaccordo)
Centroide (cluster center)	Media dei valori numerici	Modalità (valore più frequente)
Algoritmo	Minimizza la varianza all'interno del cluster	Minimizza il disaccordo all'interno del cluster
Aggiornamento del centroide	Calcolo della media dei valori numerici	Calcolo della modalità per ogni feature
Sensibile agli outlier	Sì, gli outlier possono influire sui centroidi	No, non è influenzato dagli outlier
Obiettivo	Minimizzare la somma dei quadrati delle distanze euclidee all'interno del cluster	Minimizzare il numero di dissimilitudini nei cluster

6 Risultati

METRICHE:

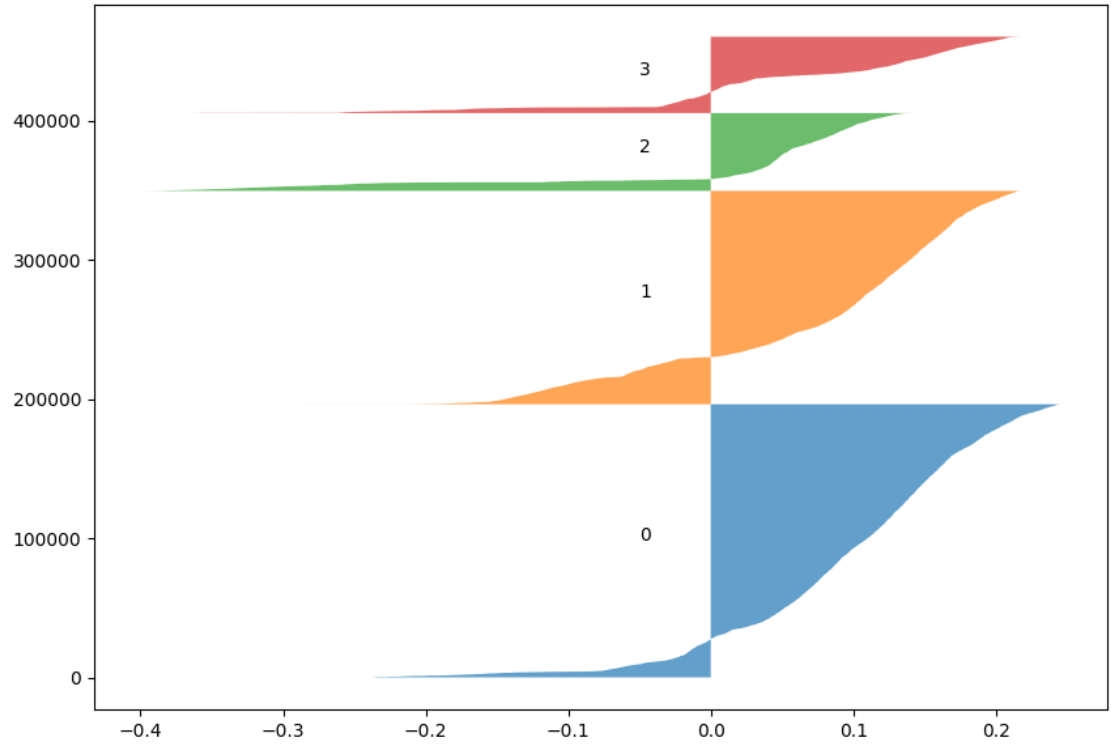
Purity score: 0.84577

Indica quanto i cluster contengono principalmente elementi di una singola classe

$$\text{Purity} = \frac{1}{N}\sum_{i=1}^{K}\max_{j}|C_i \cap L_j|$$

Più alto è il valore, più "puri" sono i cluster.

Silhouette score:0.67883



METRICA FINALE: 0.56230

Combina le due metriche con una penalità aggiuntiva proporzionale al numero di cluster per evitare che l'algoritmo porti all'overfitting

$$\text{Metrica Finale} = \left(\frac{\text{Purity} + \text{Silhouette Score Normalizzato}}{2}\right) - (0.05 \times \text{Numero di Cluster})$$

6 Risultati

Risultati Medi delle Valutazioni su n Misurazioni:

Nella tabella seguente sono riportati i valori medi delle principali metriche di valutazione ottenuti dopo n misurazioni consecutive.

Metrica Media	Valore Medio
Purity	0.845771
Silhouette	0.678832
Metrica Finale	0.562302

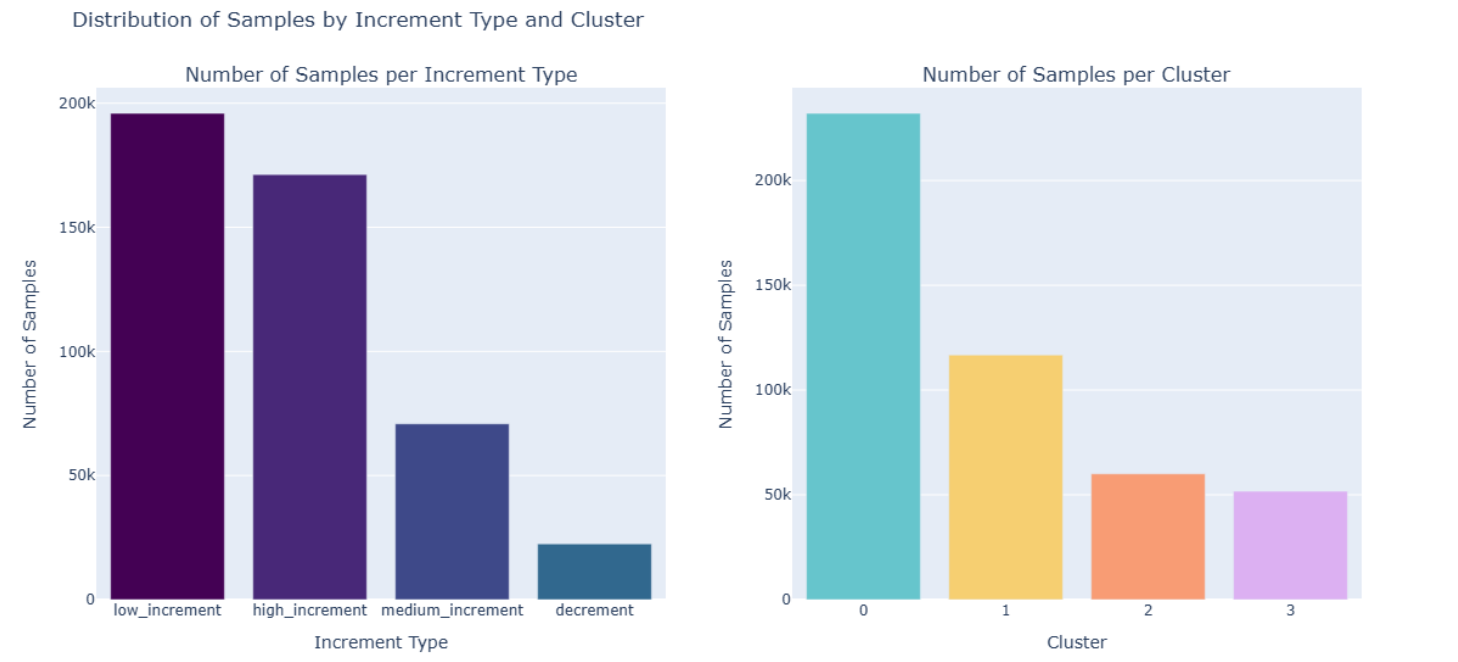
Risultati della Migliore Misurazione:

Nella tabella seguente sono riportati i risultati della misurazione che ha ottenuto i valori migliori in termini di performance complessiva. Questi risultati rappresentano il massimo raggiunto dal modello durante il processo di valutazione.

Metrica	Valore
Purity	0.87686
Silhouette	0.73871
Metrica Finale	0.60778

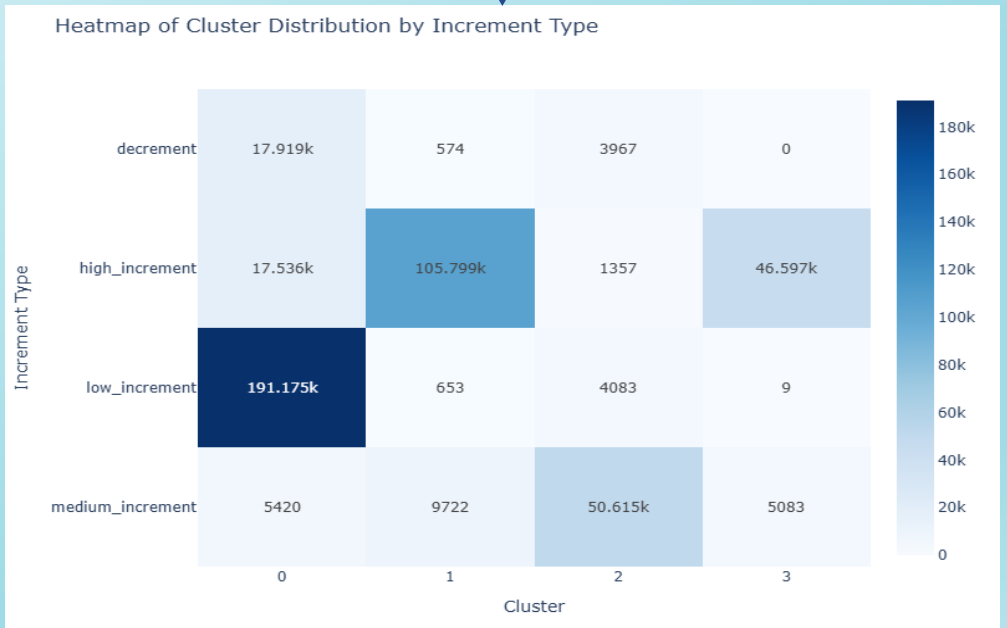
7 Grafici Interattivi

Clusterizzazione dei Campioni per Incrementi di Teleassistenza:



I campioni sono principalmente concentrati nelle categorie **low_increment** e **high_increment**, e nel **cluster 0**, con minore presenza nei **decrement** e negli altri cluster.

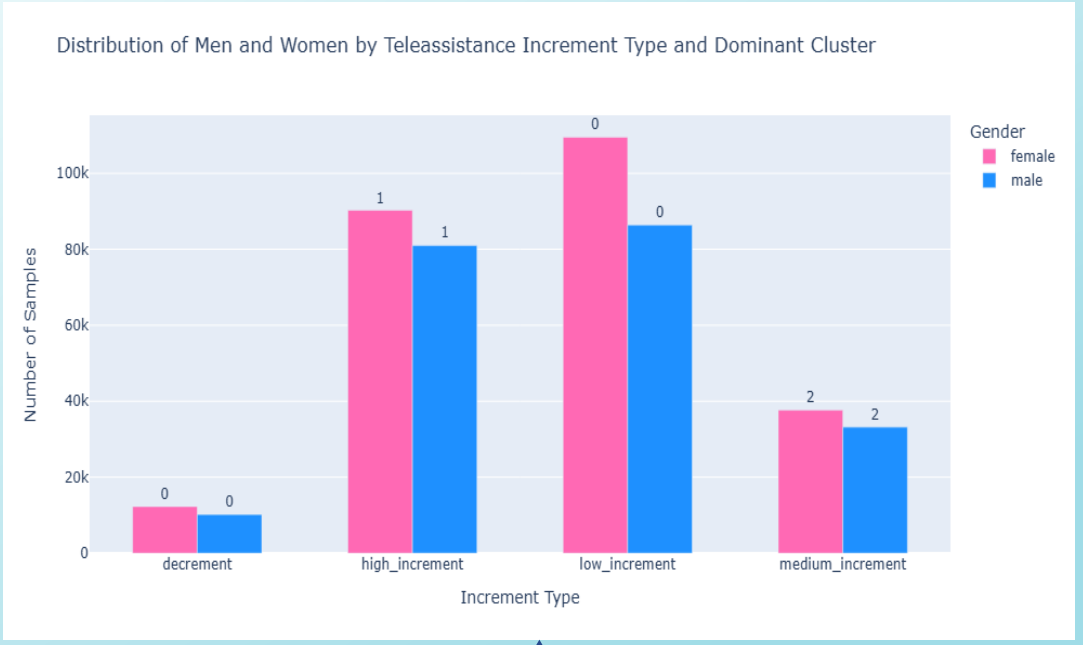
La categoria **low_increment** è fortemente concentrata nel **cluster 0**, mentre **high_increment** è maggiormente distribuita tra i **cluster 1**. I **cluster 2 e 3** hanno una distribuzione più variegata.



7 Grafici Interattivi

Impatto delle Features sulla Separazione dei Cluster

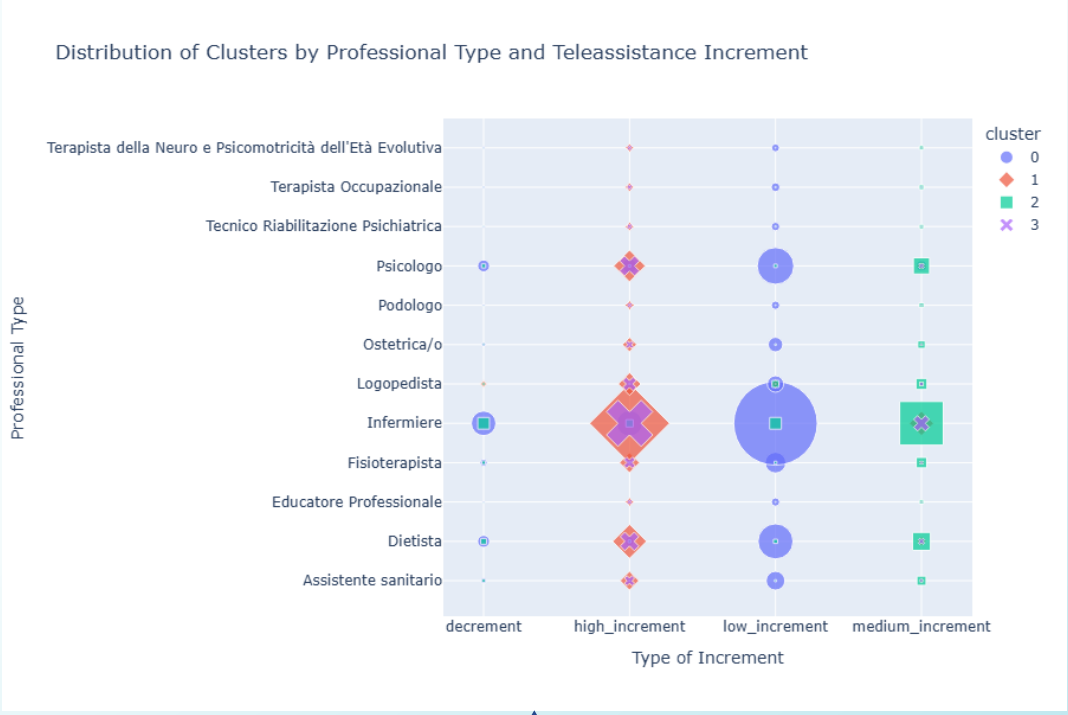
Il **low_increment** prevale nel centro-sud, mentre il **high_increment** è diversificato sul territorio.
Il **medium_increment** è meno frequente e più limitato geograficamente.



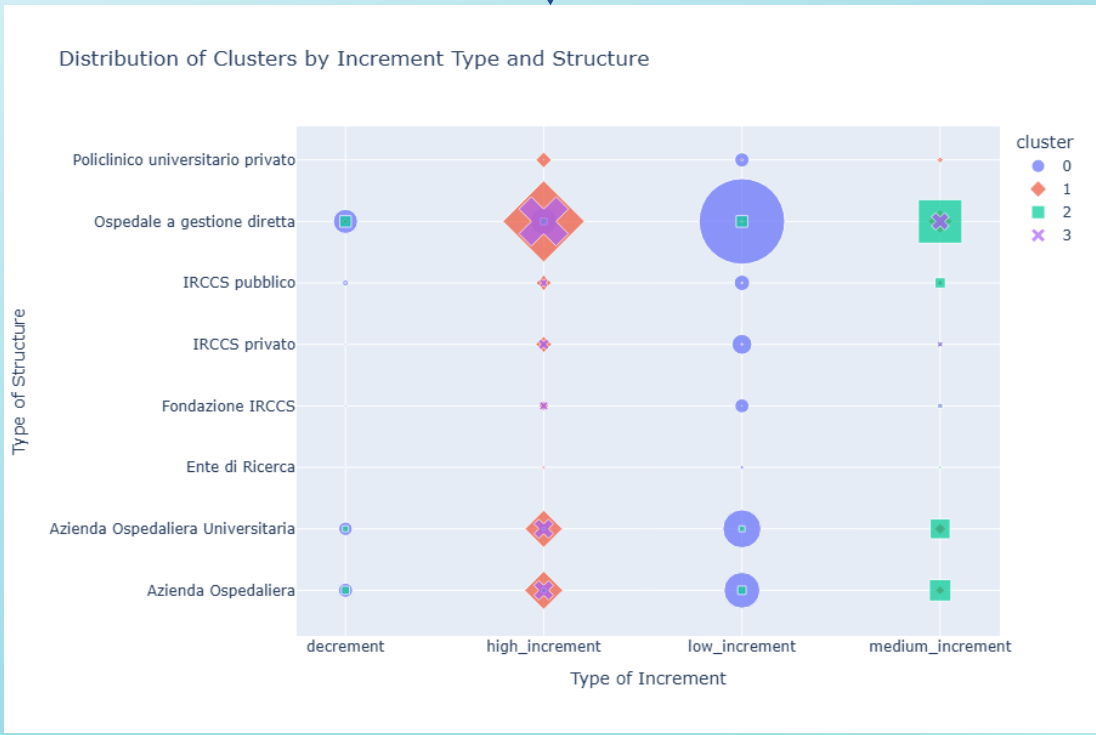
Donne e uomini presentano una distribuzione relativamente simile in quasi tutti i tipi di incremento. Le donne sono leggermente più peresenti nei gruppi di **low_increment** e **high_increment**.

7 Grafici Interattivi

Impatto delle Features sulla Separazione dei Cluster



Il **cluster 0** sembra rappresentare le strutture più stabili (spesso pubbliche) con **low_increment**, mentre il **cluster 1** raccoglie le strutture con **high_increment**, tipicamente quelle private o a gestione diretta. Evidenziando come le tipologie di strutture sanitarie si distribuiscono nei diversi cluster in base al tipo di incremento.



Il **cluster 1** è predominante nel **high_increment** professionisti quali psicologi, logopedisti, infermiere e fisioterapisti.
Il **cluster 2** è dominante nel **low_increment**, particolarmente per professioni come l'infermiere e il dietista.
Nel **medium_increment** c'è una predominanza del **cluster 3,** con i professionisti come gli infermiere, fisioterapisti e psicologi.

GRAZIE PER L'ATTENZIONE

Alessia Rossi
Fabio Di Gregorio
Ignazio Emanuele Piccichè
Martina Bertazzoni

Datasets

- `challenge_campus_biomedico_2023.parquet`

Nome Variabile	Description	Type
id_prenotazione	Unique identifier of a single Teleassistance	String
id_paziente	Patient's unique identifier code	String
data_nascita	Patient's birth date	String
sesso	Patient's sex	String
regione_residenza	Patient's residence region	String
codice_regione_residenza	Patient's residence region code	String
asl_residenza	Patient's residence ASL	String
codice_asl_residenza	Patient's residence ASL code	String
provincia_residenza	Patient's residence province	String
codice_provincia_residenza	Patient's residence province code	String
comune_residenza	Patient's residence city	String
codice_comune_residenza	Patient's residence city code	String
tipologia_servizio	Typology of offered service from telemedicine platform	String
descrizione_attivita	Description of performed activity	String
codice_descrizione_attivita	Typology of performed activity's	String
data_contatto	Patient's contact date	String

Nome Variabile	Description	Type
regione_erogazione	Service's erogation region	String
codice_regione_erogazione	Service's erogation region's code	String
asl_erogazione	Service's erogation ASL	String
codice_asl_erogazione	Service's erogation ASL code	String
provincia_erogazione	Service's erogation province	String
codice_provincia_erogazione	Service's erogation province code	String
struttura_erogazione	Service's erogation facility name	String
codice_struttura_erogazione	Service's erogation facility name's code	String
tipologia_struttura_erogazione	Service's erogation facility typology	String
codice_tipologia_struttura_erogazione	Service's erogation facility typology code	String
id_professionista_sanitario	Healtcare professional erogator's unique identifier code	String
tipologia_professionista_sanitario	Healtcare professional erogator's typology	String
codice_tipologia_professionista_sanitario	Healtcare professional erogator's typology code	String
data_erogazione	Service's erogation date	String
ora_inizio_erogazione	Service's erogation start timestamp (if already permormed)	String
ora_fine_erogazione	Service's erogation end timestamp (if already permormed)	String
data_disdetta	Service's erogation cancellation timestamp (if visit cancelled)	String

- `Codici-statistici-e-denominazioni-aggiornato-2023.xlsx`

Questo dataset contiene informazioni ISTAT, che includono codici statistici e denominazioni aggiornati dei comuni italiani. Viene utilizzato in combinazione con il primo dataset per completare i dati mancanti sul comune di residenza durante la fase di data cleaning.

6 Risultati

METRICHE:

Purity score:

Indica quanto i cluster contengono principalmente elementi di una singola classe

$$\text{Purity} = \frac{1}{N} \sum_{i=1}^{K} \max_{j} |C_i \cap L_j|$$

Più alto è il valore, più "puri" sono i cluster.

Silhouette score (normalizzato):

Quantifica quanto bene ogni punto dati si trova nel proprio cluster rispetto ai punti di altri cluste

$$\text{Silhouette Score Medio} = \frac{1}{N} \sum_{i=1}^{N} s(i)$$

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

- 1: i punti sono ben allineati con il proprio cluster e ben separati dagli altri cluster
- 0: i punti sono stati assegnati al cluster sbagliato

METRICA FINALE:

Combina le due metriche con una penalità aggiuntiva proporzionale al numero di cluster per evitare che l'algoritmo porti all'overfitting

$$\text{Metrica Finale} = \left(\frac{\text{Purity} + \text{Silhouette Score Normalizzato}}{2} \right) - (0.05 \times \text{Numero di Cluster})$$