MMHAi

Music & Mental Health Analysis – Uno sguardo intelligente alle abitudini di ascolto musicale in correlazione alla salute mentale

Questo progetto è stato sviluppato da William Tinella, studente d'Informatica presso l'Università degli Studi "Aldo Moro" di Bari. Matricola 774904.

Link: GitHub

INDICE

INDICE 2

INTRODUZIONE 3

ANALISI DEI DATI E PREPROCESSING 6

- Pre-Elaborazione Dati 9
- Conclusioni e Grafici 13

APPRENDIMENTO NON SUPERVISIONATO 18

- Fondamenti Teorici 19
- Clustering e risultati 20
  - Primo Appoccio 21
  - Secondo Approccio 24

APPRENDIMENTO SUPERIVISIONATO 28

- Introduzione 28
- Algoritmi e Modelli Usati 29
- Ulteriori Concetti Chiave 33
- Primo Approccio (Fallimentare) 33
- Secondo Approccio 37
- Conclusioni 40
- Belief Network (bonus) 41

INTRODUZIONE

Da circa 50 anni si studia in maniera approfondita la possibile correlazione tra l'ascolto di musica e le sue abitudini e la salute mentale dell'individuo. Ad inizio anni 80' si è sviluppata la musicoterapia, una forma di terapia nel campo della crescita e del supporto psicofisico dell'individuo.

Mentre negli ultimi anni diversi studi hanno cercato di evidenziare un possibile nesso tra le abitudini musicali e la qualità della salute mentale. Come, ad esempio, la relazione che potrebbe esistere tra la quantità di musica ascoltata quotidianamente, i generi, le frequenze di ascolto e anche altri fattori musicali con l'insorgere di patologie come depressione, ansia, insonnia e OCD (disturbo ossessivo compulsivo).

> *"secondo la teoria dell'omeostasi, la musica viene utilizzata per regolare l'umore e i livelli di stress nei momenti difficili, prevenendo così patologie come la depressione (Cummins, 2010)."*

L'idea di base di questo progetto è quindi sviluppare un sistema intelligente per verificare, tramite dati empirici e correlazioni matematiche, che questa relazione esista o meno. Per lo sviluppo di questo progetto è stato preso come campione una ricerca statistica svolta da Catherine Rasgaitis, dottoranda nel settore biomedico pressa l'Università di Washington. I suoi dati sono pubblicati in un dataset presso Kaggle.

Obiettivi Principali del Progetto

1. Analizzare il dataset per esplorare le relazioni tra le abitudini di ascolto musicale e gli indicatori di salute mentale (Ansia, Depressione, Insonnia, OCD).

2. Definire profili di salute mentale oggettivi tramite tecniche di clustering non supervisionato, raggruppando gli utenti in base ai loro livelli di benessere psicologico.
3. Addestrare e valutare comparativamente diversi modelli di Machine Learning (tra cui *RandomForest, Gradient Boosting, XGBoost, LightGBM*) per la classificazione di un individuo all'interno di uno dei profili identificati.
4. Sviluppare e testare un modello di regressione capace di predire un punteggio aggregato di salute mentale con un margine di tolleranza predefinito.
5. Selezionare i modelli più performanti per entrambi i compiti (classificazione e regressione) basandosi su metriche di valutazione robuste come l'accuratezza e la cross-validation.

Metodologia Adottata

La metodologia adottata si può definire tramite le seguenti fasi:

1. Data Collection and Preprocessing: Partendo dal dataset originale, è stata eseguita una fase approfondita di pulizia e preparazione dei dati. Questa fase ha incluso la gestione di valori anomali, la conversione di variabili categoriche (es. genere musicale preferito) in formato numerico tramite one-hot encoding e la normalizzazione delle feature numeriche per renderle confrontabili.
2. Feature Engineering: Sono state create nuove feature derivate da quelle esistenti per integrare conoscenza di fondo e migliorare il potenziale predittivo dei modelli:
    2.1. **Mental_Health_Score**: Un indice numerico aggregato, creato sommando i valori dei quattro indicatori di salute mentale, per semplificare e rendere più trattabile il problema della regressione.
    2.2. **Diversity_Score**: Una metrica originale per quantificare la varietà dei generi musicali ascoltati da un utente, basata sulla frequenza di ascolto dei generi non preferiti.

3. Unsupervised Analysis and Target Definition (Analisi non supervisionata e Definizione del Target): Prima della classificazione, è stato utilizzato l'algoritmo di clustering K-Means sulle quattro feature di salute mentale. Questo ha permesso di identificare due cluster naturali e distinti all'interno dei dati, che sono stati poi usati come le classi target per il modello di classificazione supervisionata.
4. Model Selection and Training: Sono stati selezionati, addestrati e ottimizzati diversi modelli di apprendimento supervisionato per i due compiti:

    4.1. Classificazione: Confronto tra RandomForestClassifier, GradientBoostingClassifier, XGBClassifier e LGBMClassifier.

    4.2. Regressione: Sviluppo di modelli basati su RandomForestRegressor e altri.

5. L'ottimizzazione degli iperparametri è stata eseguita tramite tecniche di ricerca come RandomizedSearchCV e GridSearchCV, con una valutazione robusta garantita dalla cross-validation (5-fold).
6. Model Evaluation and Comparison: Le performance dei modelli addestrati sono state valutate tramite metriche appropriate per ciascun compito e confrontate per selezionare l'approccio migliore:
    6.1. Per la classificazione, è stata usata l'accuracy.
    6.2. Per la regressione, è stata implementata una metrica di accuratezza con tolleranza, più adatta a valutare la bontà di una predizione numerica in un contesto reale.
    6.3. L'analisi ha rivelato che un modello più semplice, senza feature engineering complesse, forniva le performance migliori e più stabili, indicando che le feature originali contenevano già il segnale predittivo massimo.

Analisi dei Dati

Descrizione e Struttura del Dataset

Come già detto in calce, è stato usato il dataset creato e fornito da Catherine Rasgaitis disponibile su Kaggle.

Il dataset, che da ora in poi verrà chiamato in maniera abbreviata “mxmh”, è stato realizzato e pubblicato nel 2022. I dati sono auto inseriti tramite Google Form.

MXMH è composto da 727 record, contenenti 33 colonne così divise.

- **Numero di Attributi (Colonne):** 36, che possono essere raggruppati in categorie logiche.

Struttura Dettagliata delle Feature

Le colonne del dataset possono essere suddivise nei seguenti gruppi tematici:

1. Dati Demografici e Comportamentali Questo gruppo descrive le caratteristiche anagrafiche e le abitudini generali dei partecipanti.

- *Age*: L'età del partecipante (variabile numerica).
- *Hours_per_day*: Il numero medio di ore giornaliere dedicate all'ascolto di musica (numerica).
- *While_working:* Indica se il partecipante ascolta musica mentre lavora o studia, tramite sì o no.
- *Instrumentalist*: Indica se il partecipante suona uno strumento musicale, tramite sì o no.
- *Composer*: Indica se il partecipante compone musica, tramite sì o no.
- *Exploratory*: Indica se al partecipante piace esplorare attivamente nuova musica, tramite sì o no.
- *Foreign_languages*: Indica se il partecipante ascolta musica in lingue straniere, tramite sì o no.

Tutti i valori “si/no” sono stati convertiti in variabili booleane.

2. Indicatori di Salute Mentale (Auto-valutazione) Queste feature rappresentano il cuore dei dati sulla salute mentale, con punteggi auto-riferiti su una scala da 0 a 10.

- *Anxiety*: Livello di ansia percepito.
- *Depression*: Livello di depressione percepito.
- *Insomnia*: Livello di insonnia percepito.
- *OCD* (Disturbo Ossessivo-Compulsivo): Livello di sintomi OCD percepiti.

3. Abitudini Musicali (Frequenza di Ascolto) Questo gruppo di 16 feature descrive la frequenza con cui ogni partecipante ascolta un determinato genere musicale.

- *Frequency_[**Genere**]*: Una serie di colonne (es. Frequency_Rock, Frequency_Pop, Frequency_Classical, etc.) che indicano la frequenza di ascolto per ciascun genere, con valori come "Never", "Rarely", "Sometimes", "Very frequently". *Nota: nel preprocessing, queste sono state convertite in valori numerici ordinali.*
- Fav_genre: Il genere musicale preferito, testuale (poi fatto one hot encoding).
- *Primary_service_streaming*: la piattaforma di streaming musicale preferita. Testuale

4. Caratteristiche Audio Oggettive

- *BPM*: I battiti per minuto (BPM) medi della musica preferita dal partecipante. Numerico.

5. Percezione Soggettiva della Musica

- *Music_effects*: Descrive l'effetto che la musica ha sull'umore del partecipante (es. "Improve", "Worsen", "No effect").

Tramutato in *Music_effects_scaled,* versione numerica della feature precedente.

6. Feature Ingegnerizzate (Aggiunte durante il Preprocessing). Queste sono le feature create per arricchire il dataset e facilitare la modellazione.

- *Mental_Health_Score*: Un punteggio aggregato (da 0 a 40) calcolato come somma dei quattro indicatori di salute mentale. Utilizzato come target per il modello di regressione.
- *Diversity_Score:* Una metrica calcolata per misurare la varietà di generi ascoltati da un utente. Questo punteggio, il più interessante, è una media dei punteggi di frequence[genere] escluso il punteggio della frequenza d'ascolto del genere preferito dell'individuo (testato anche escludendo i primi 3 generi musicali preferiti). Vuole, quindi, misurare quanto una persona tende ad ascoltare generi musicali diversi da quelli preferiti.
- Cluster: La classe di appartenenza assegnata a ogni utente dall'algoritmo K-Means, basata sui suoi indicatori di salute mentale. Utilizzata come target per il modello di classificazione.

Sono stati esclusi dati come "permissions" e "timestamp" che sono completamente inutili ai fini del progetto.

Feature	Valori
Age	numerico
Hours per day	numerico
While Working	Testuale: si o no
Instrumentalist	Testuale: si o no
Composer	Testuale: si o no
Exploratory	Testuale: si o no
Foreign Language	Testuale: si o no
Frequency[Genre*] Genre ha 16 possibili valori diversi	Testuale: "never", "rarely", "sometimes", "very frequentaly"
Fav Genre	Testuale
BPM	Numerico
Music Effects	Testuale: si o no

Mental Health Score*	Numerico
Diversity Score*	Numerico
Primary Service Streaming	Testuale
Permission**	Testuale
Timestamp**	Testuale

*aggiunti dopo feature engineering
**rimossi successivamente

Pre-Elaborazione dei Dati

Analizzando il dataset e avendo in mente lo scopo del progetto è stata condotta scrupolosa fase di pre processing sui dati.
Per prima cosa sono stati rimossi dal dateset i campi riguardanti il timestamp (momento di sottoscizione del questionario) e del permissions(permesso per la prolificazione dei dati).

Dopo di che sono stati standardizzati tutte le variabili testuali.

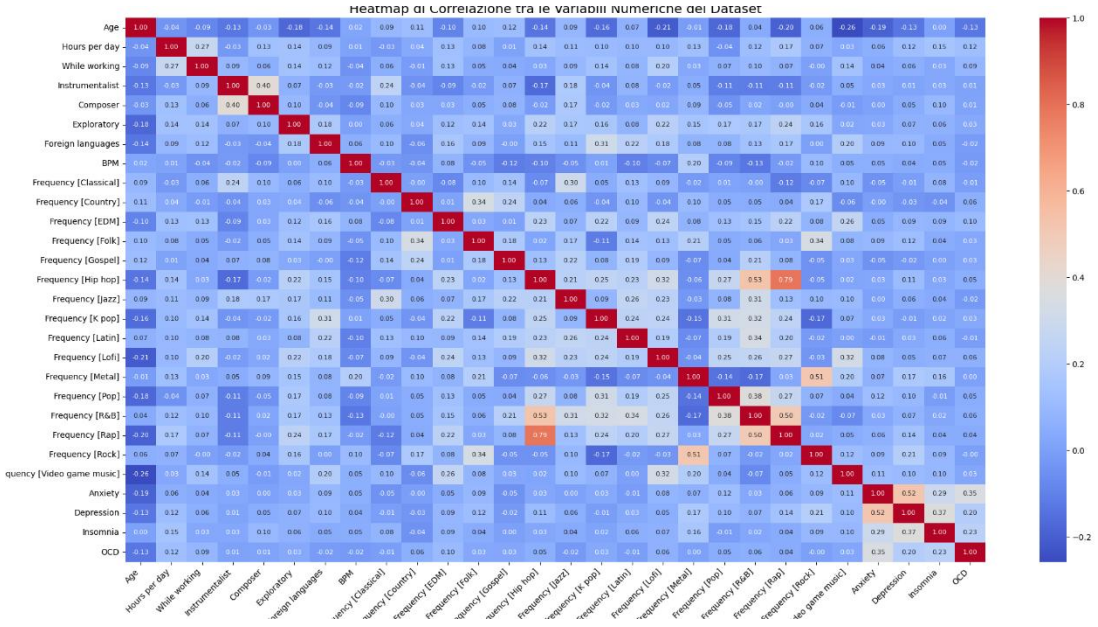
Tutte le variabili contenenti valori “si” o “no” sono diventate variabili booleane.

Mentre i valori testuali “never”, “rarely”, “sometimes”, “very frequently” sono state rimappate per avere valori numerici da poter usare. La logica di rimappattura è stata semplice ed è avvenuta così:

```
# Mapping delle frequenze in valori numerici
frequency_mapping = {
    'Never': 0,
    'Rarely': 1,
    'Sometimes': 2,
    'Very frequently': 3
}
```

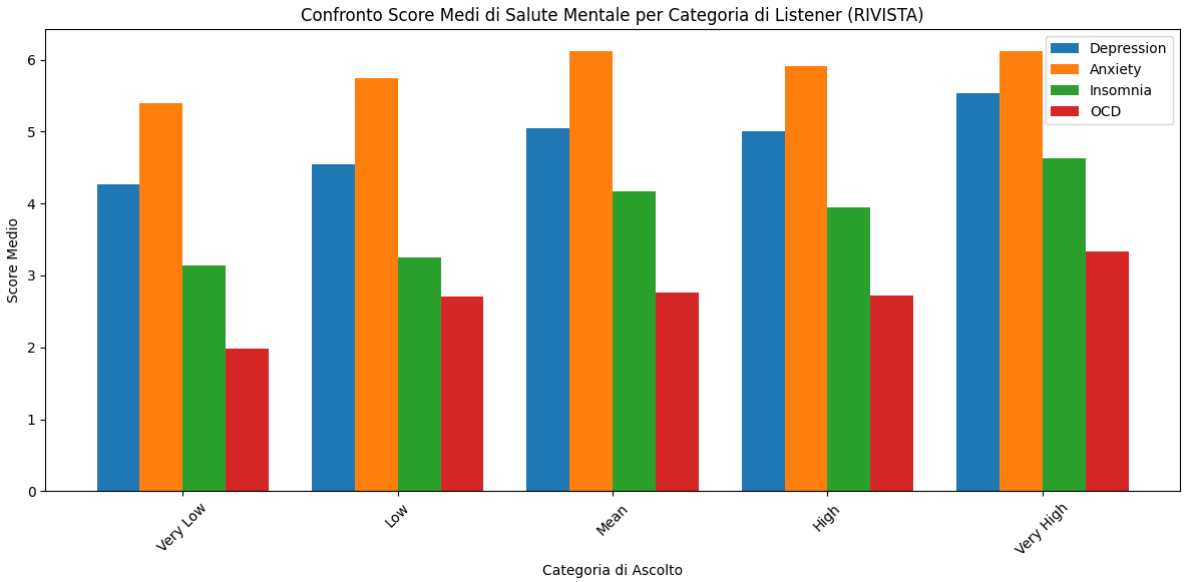
```
# Convertiamo le frequenze in valori numerici
for col in genre_cols:
    mxmh[f"{col}_numeric"] = mxmh[col].map(frequency_mapping)

# Colonne numeriche dei generi
genre_numeric_cols = [f"{col}_numeric" for col in genre_cols]
```

Successivamente sono stati eseguiti controlli e ricerche di pattern e correlazioni tra le variabili. Tramite un heatmap si evince la correlazione tra le varie feature, in maniera globale. Si nota che poche feature hanno una forte correlazione.

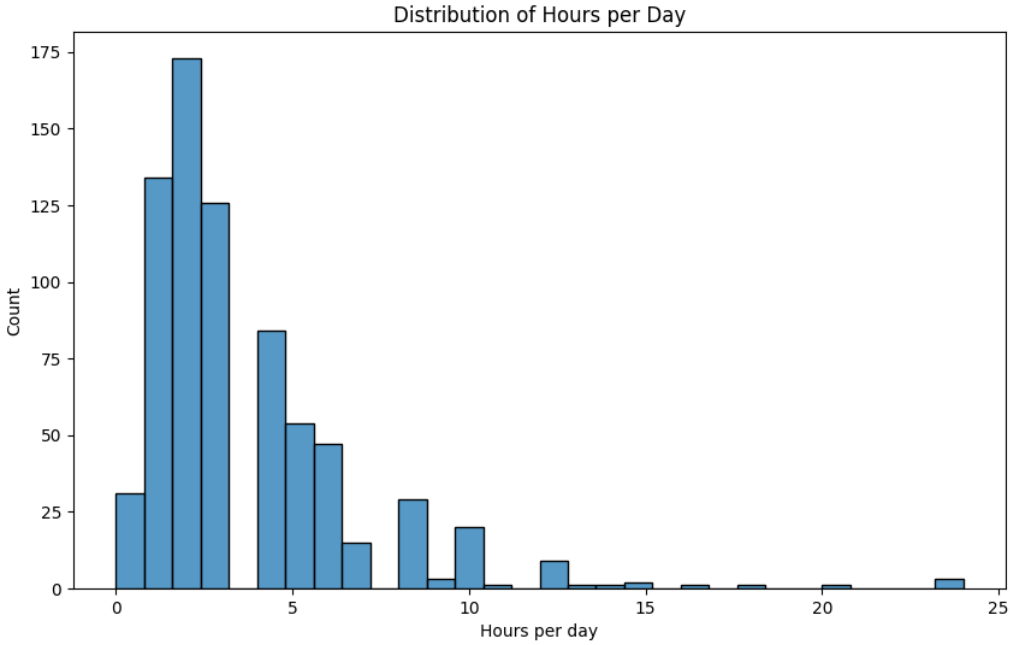


Un altro importante lavoro di analisi e manipolazione dei dati fatto è stato riguardante la feature delle ore di musica ascoltate quotidianamente. Poichè il solo dato numerico è poco interpretabile ho analizzato medie, mediana, deviazione standard e distribuzione percentile attraverso boxplot e altri grafici. Avendo calcolato le distribuzioni percentile di ascolto, ho categorizzato in cinque categorie i vari profili di ascolto.

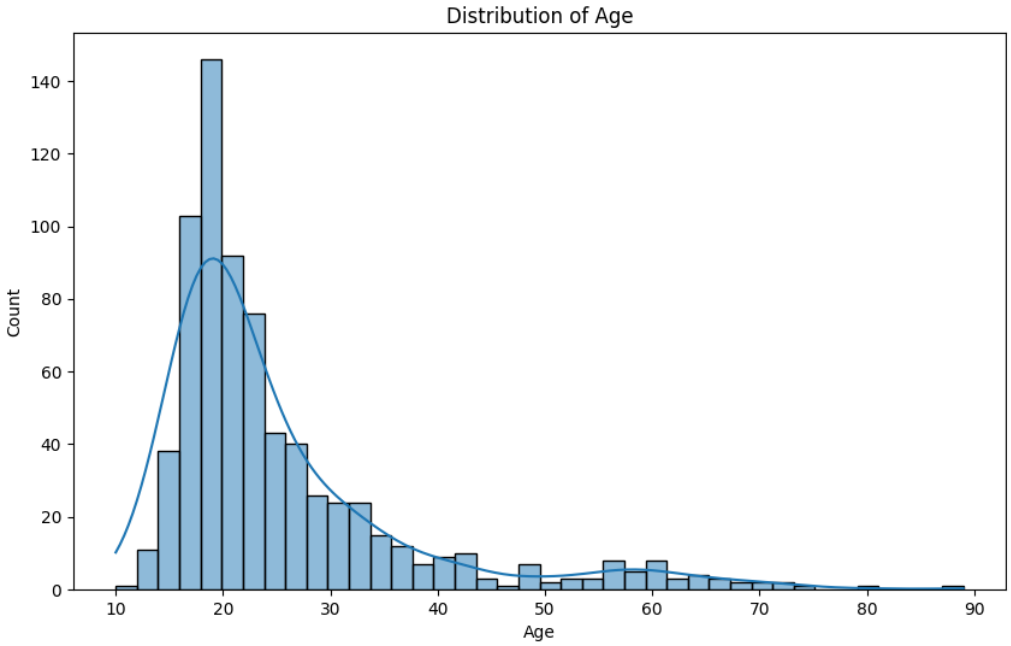
```python
def listener_categories(h):
    """
    Categorizzazione basata sui percentili del dataset:
    - Very Low: < 2 ore (sotto il 25° percentile)
    - Low: 2-3 ore (25°-50° percentile)
    - Mean: 3-5 ore (50°-75° percentile)
    - High: 5-7 ore (75°-90° percentile)
    - Very High: >= 7 ore (sopra il 90° percentile)
    """
    if h < 2:
        return "Very Low"
    elif 2 <= h < 3:
        return "Low"
    elif 3 <= h < 5:
        return "Mean"
    elif 5 <= h < 7:
        return "High"
    else:  # h >= 7
        return "Very High"
```



Il dato interessante uscito da questa categorizzazione è quello riportato di sopra. Si nota un incremento (a volte costante) dell'intensità delle quattro feature legate alla salute mentale andando sempre più in su con le ore di musica ascoltate giornalmente.

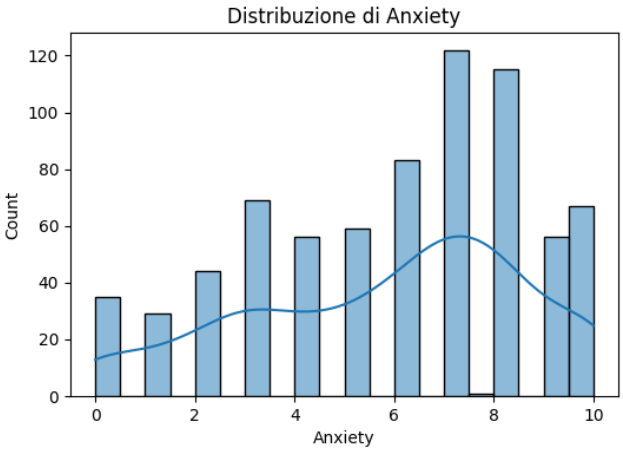


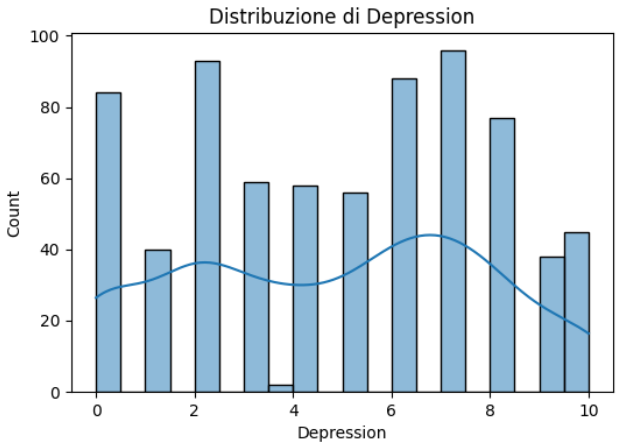
Da notare come la distribuzione di hours per day sia asimmetrico.

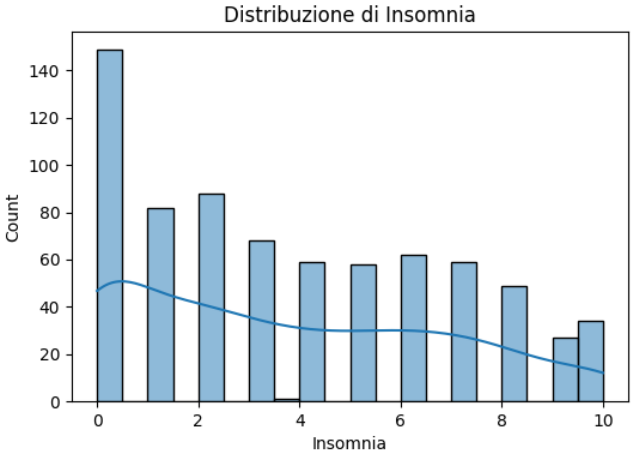


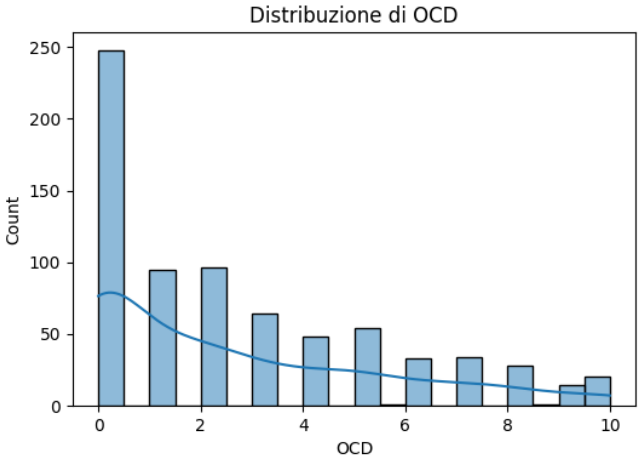
Un altro dato di interesse è stata l'età. Purtroppo, si è notato che il campione preso in esame è concentrato nella fascia d'età dai 15 ai 30 anni, con una media di 25.2 e una devianza standard di 12. Di per sé non è un problema, ma a livello scientifico la fascia d'età presa in esame potrà rilevarsi troppo uniforme.

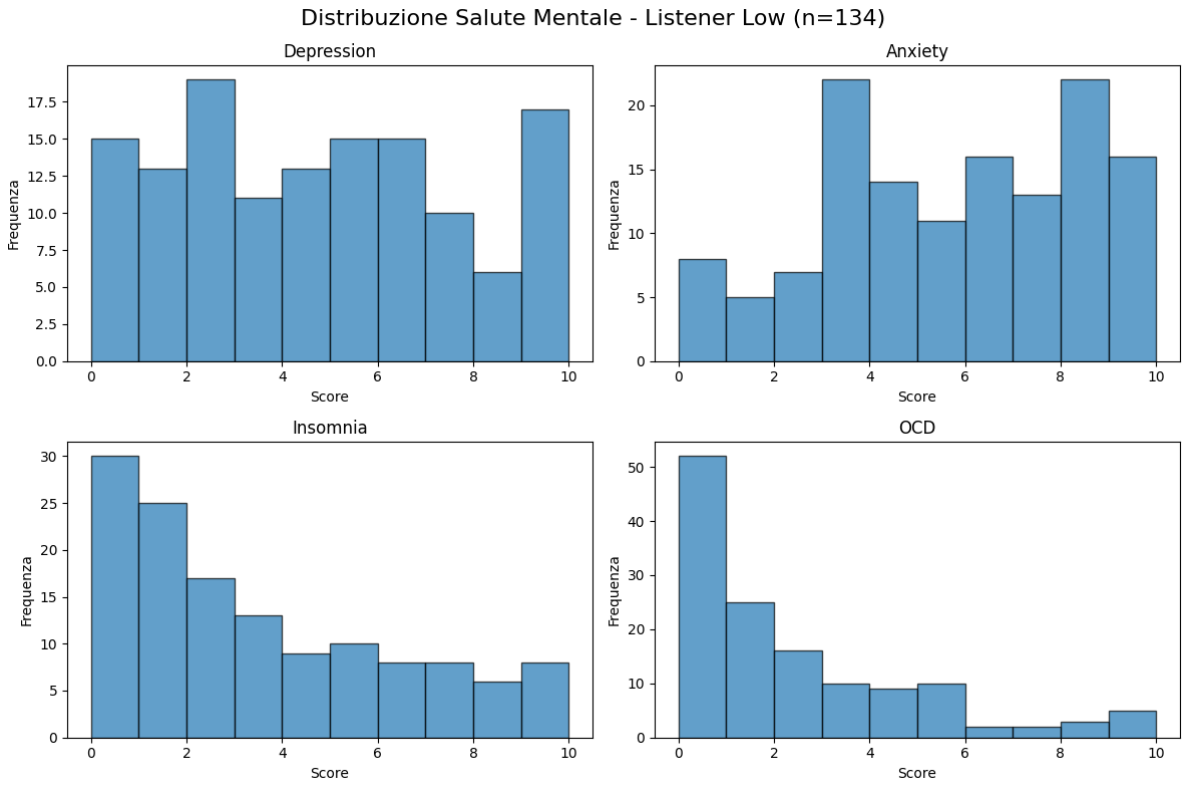
Ulteriori analisi e grafici:

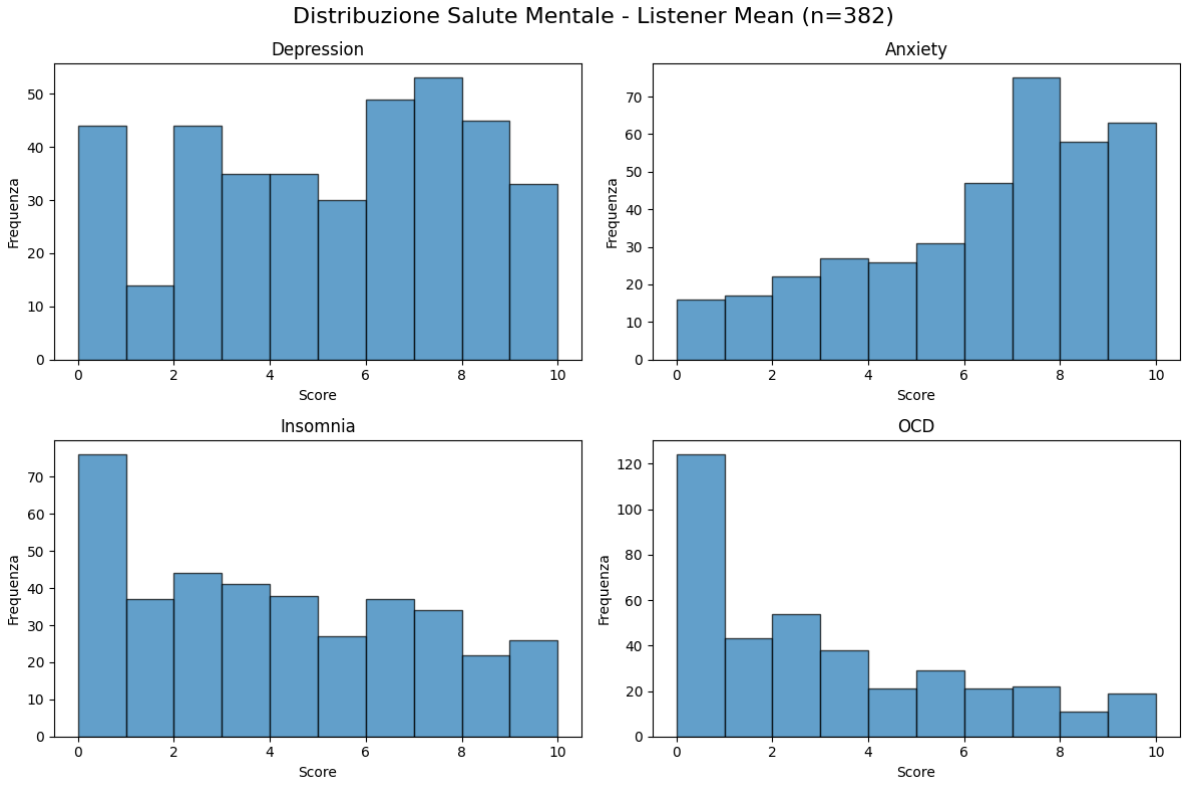


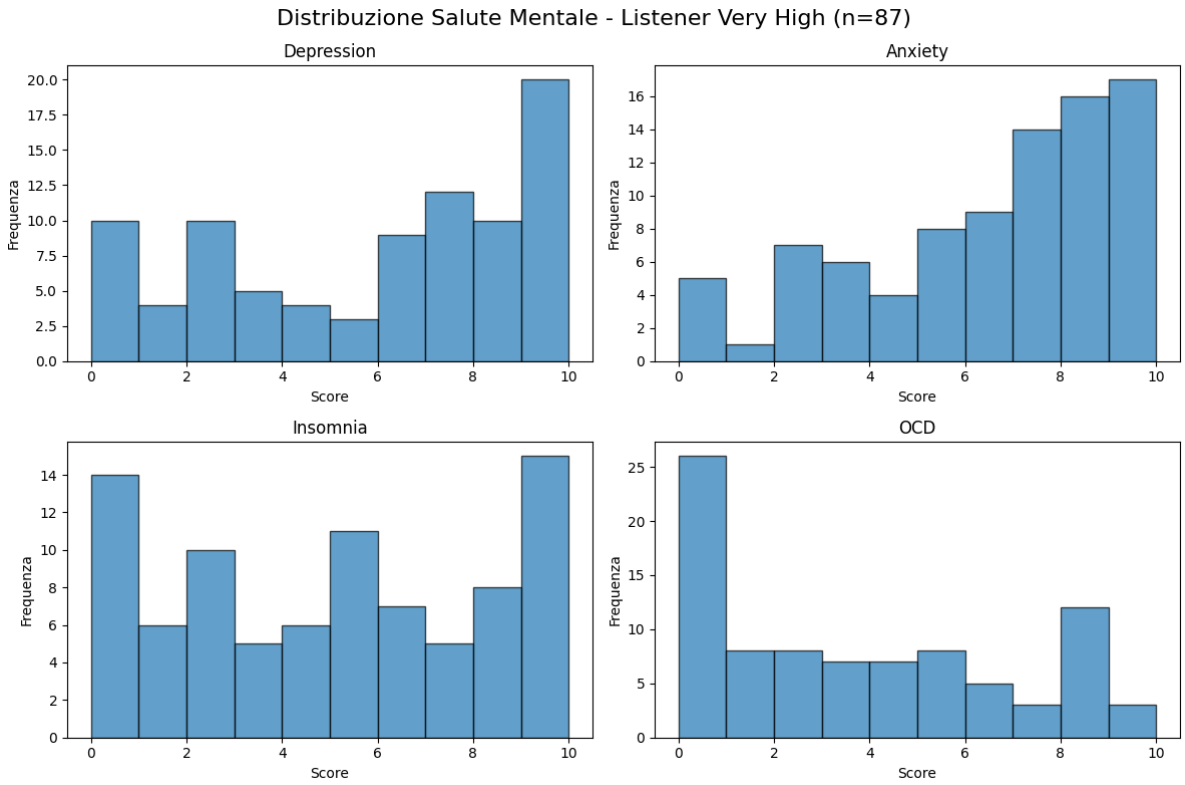


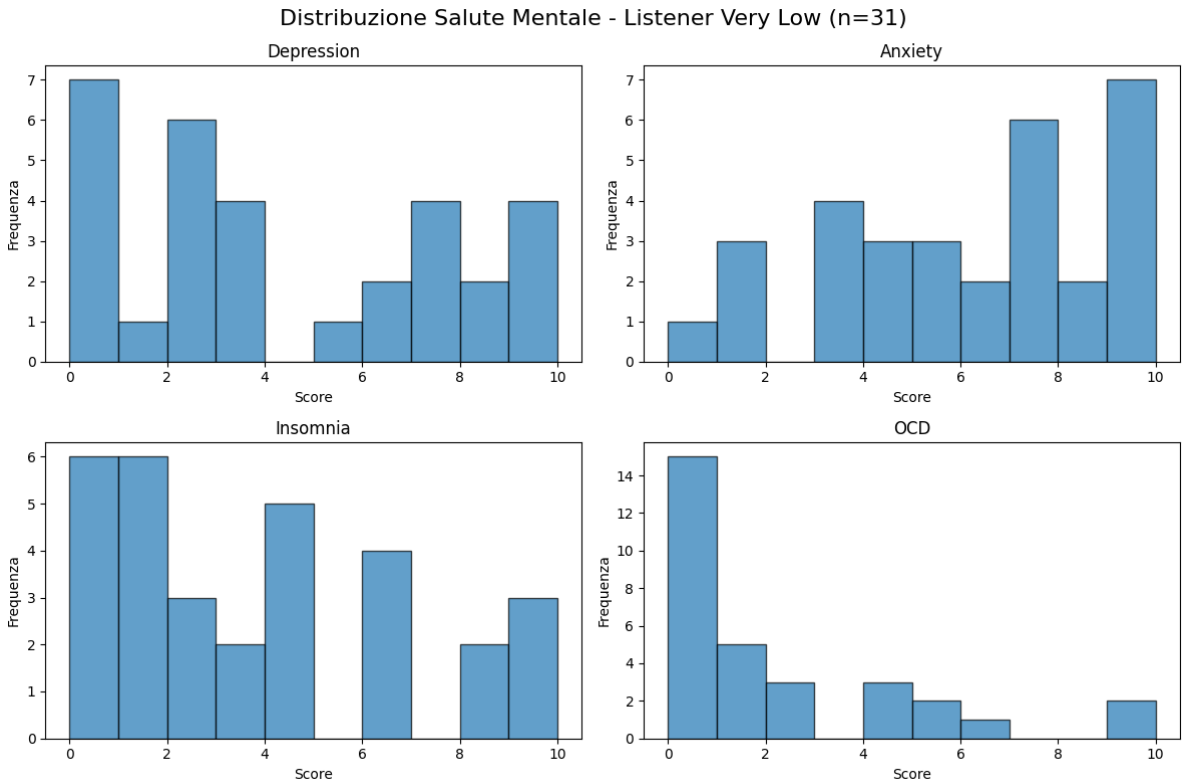


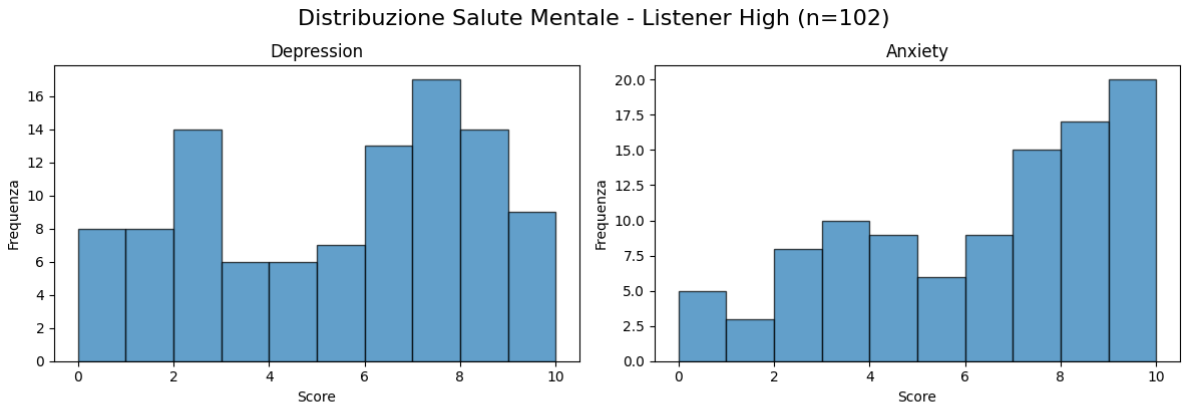


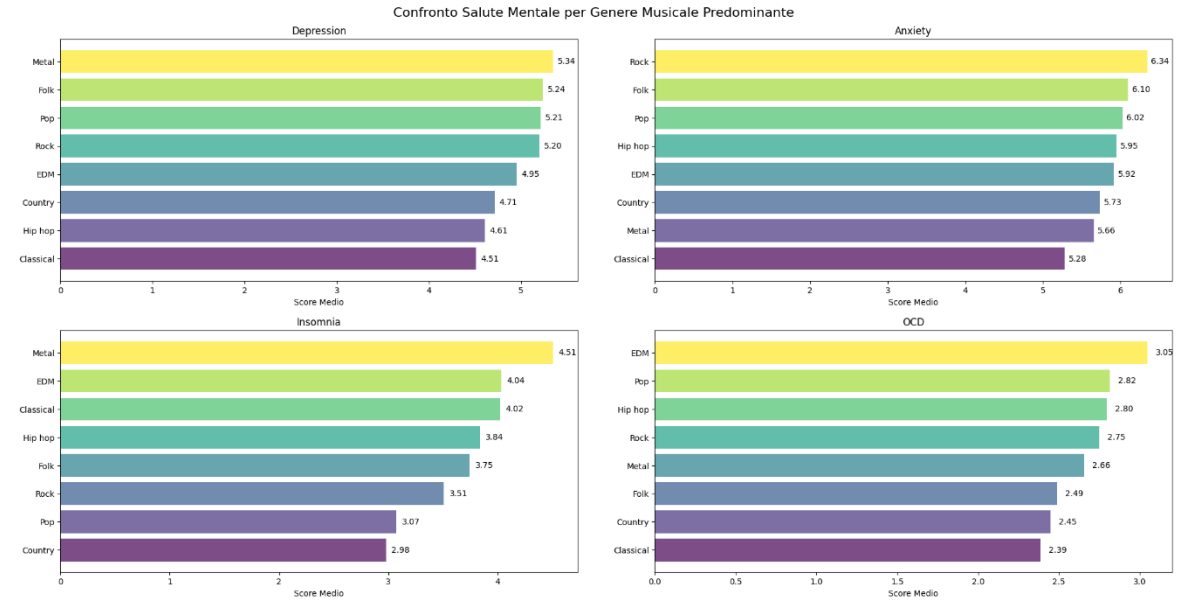

Distribuzione Salute Mentale - Listener Low (n=134)
Depression
Anxiety
Insomnia
OCD
Frequenza
Score


Distribuzione Salute Mentale - Listener Mean (n=382)
Depression
Anxiety
Insomnia
OCD
Frequenza
Score


Distribuzione Salute Mentale - Listener Very High (n=87)
Depression
Anxiety
Insomnia
OCD
Frequenza
Score


Distribuzione Salute Mentale - Listener Very Low (n=31)
Depression
Anxiety
Insomnia
OCD
Frequenza
Score





Tutti i restanti grafici sono visionabili all'interno del progetto in "\MMHAi---Music-Mental-Health-Analysis\images".

Dopo aver pulito e migliorato l'interpretabilità dei dati si è deciso di caricare tutto il lavoro nel dataset definitivo denominato "mxhm_final" (il dataset iniziale si chiama "mxhm_survey_results").

Conclusioni ed interpretazioni sulla prima analisi dei dati (EDA):

L'analisi esplorativa dei dati ha fornito una prima visione d'insieme delle complesse interazioni tra le abitudini musicali e gli indicatori di salute mentale presenti nel campione. Attraverso l'analisi delle distribuzioni e il confronto diretto tra gruppi, sono emerse diverse tendenze significative che meritano di essere approfondite.

1. Pattern di Ascolto e Coinvolgimento Musicale

L'analisi ha rivelato che non tutti gli ascoltatori sono uguali. Sono state identificate differenze notevoli basate sul livello di coinvolgimento attivo con la musica:

- **Attività Musicale e Ore di Ascolto**: I box plot hanno mostrato che sia gli **strumentisti** sia i **compositori** tendono ad avere una media di ore di ascolto giornaliere leggermente superiore rispetto a chi non ha un ruolo attivo nella creazione musicale. Questo suggerisce che un coinvolgimento pratico con la musica si associa a un maggior consumo musicale complessivo.
- **Attività Musicale e Salute Mentale**: Un risultato particolarmente interessante riguarda la salute mentale. Confrontando i gruppi, si osserva che **strumentisti e compositori riportano, in media, punteggi di depressione e ansia leggermente inferiori** rispetto ai non-musicisti. Sebbene questa differenza non implichi un nesso di causalità, suggerisce che l'impegno attivo nella musica potrebbe essere associato a un migliore benessere psicologico o a maggiori risorse per la gestione emotiva.

2. Preferenze di Genere e Benessere Psicologico

L'analisi ha esplorato anche il legame tra i generi musicali preferiti e la salute mentale, evidenziando che alcuni generi sono più frequentemente associati a determinati stati emotivi:

- **Generi e Distress Psicologico**: Dall'analisi è emerso che gli ascoltatori che prediligono generi come il **Rock** o il **Metal** tendono a riportare, in media, punteggi di depressione e ansia più alti.
- **Generi e Benessere Psicologico**: Al contrario, chi preferisce generi come la **Musica Classica** o il **Pop** mostra in media punteggi più bassi per le stesse problematiche.
- Mentre chi ha punteggi piu alti nel OCD tende a preferire, insolitamente, l'EDM

È fondamentale interpretare questo dato con cautela: non è il genere musicale a "causare" un certo stato d'animo, ma è più probabile che le persone scelgano generi musicali che rispecchiano o aiutano a elaborare il loro stato emotivo (fenomeno del *mood management*).

Ulteriori considerazioni

- Le **preferenze di genere** non sono neutre, ma sembrano correlate a diversi profili di benessere psicologico, agendo probabilmente come riflesso dello stato emotivo dell'ascoltatore.
- La **quantità di ascolto**, sebbene importante, deve essere interpretata nel contesto di *come* e *cosa* si ascolta.

Interpretare i dati porta a delle ipotesi, seppur è importante sottolineare che le correlazioni matematiche siano discrete e non eccessivamente forti. Questo potrebbe portare a non riuscire ad avere modelli intelligenti con punteggio di accuracy (o altre metriche) buone.

Apprendimento NON Supervisionato

In questa sezione, si passa dall'analisi descrittiva all'applicazione di tecniche di apprendimento non supervisionato. L'obiettivo è identificare strutture e pattern intrinseci nei dati, senza fare affidamento su etichette predefinite. In particolare, si utilizza il clustering per raggruppare gli utenti in profili omogenei, basandosi sulle loro abitudini di ascolto, sul loro coinvolgimento musicale e sul loro stato di salute mentale.

1. Fondamenti Teorici

L'**apprendimento non supervisionato** è una branca del machine learning in cui l'algoritmo apprende da dati non etichettati. Lo scopo non è predire un output noto, ma scoprire la struttura latente dei dati. Il **clustering** è la tecnica più comune in questo ambito e consiste nel partizionare un set di dati in un numero di gruppi (cluster), in modo che gli oggetti all'interno dello stesso gruppo siano più simili tra loro di quanto non lo siano con gli oggetti di altri gruppi.

Per questa analisi è stato scelto l'**algoritmo K-Means**, uno dei più diffusi per la sua efficienza e interpretabilità. Il suo funzionamento si articola nei seguenti passi:

1. **Inizializzazione**: Si sceglie il numero di cluster desiderato, *k*, e si inizializzano *k* punti casuali come "centroidi" (i centri dei cluster).
2. **Assegnazione**: Ogni punto del dataset viene assegnato al cluster il cui centroide è più vicino (secondo una metrica di distanza, solitamente euclidea).
3. **Aggiornamento**: I centroidi di ogni cluster vengono ricalcolati come la media di tutti i punti assegnati a quel cluster. I passi 2 e 3 vengono ripetuti iterativamente finché le assegnazioni dei punti ai cluster non si stabilizzano.

```
features = mxmh[["Listener_Type_Num", "Depression", "Anxiety", "Insomnia", "OCD"]]
features_clean = features.dropna()

scaler = StandardScaler()
X = scaler.fit_transform(features_clean)

# Test per trovare il numero ottimale di cluster
print("=== CLUSTERING CON NUOVA CATEGORIZZAZIONE ===")
silhouette_scores = []
for k in range(2, 8):
    km = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = km.fit_predict(X)
    score = silhouette_score(X, labels)
    silhouette_scores.append(score)
    print(f"k={k}, silhouette={score:.3f}")

best_k = silhouette_scores.index(max(silhouette_scores)) + 2
print(f"\nMigliore k: {best_k} (silhouette: {max(silhouette_scores):.3f})")

km_final = KMeans(n_clusters=best_k, random_state=42, n_init=10)
final_labels = km_final.fit_predict(X)

pca = PCA(2)
X_pca = pca.fit_transform(X)
```

Valutazione del Clustering: La scelta del numero ottimale di cluster, *k*, è un passaggio critico. Per prima cosa ho usato il piu semplice dei modi, il metodo del gomito. Anche se alla fine ho optato sull'uso della silhoutte per determinare il punteggio di ogni *k* per rendere il tutto piu autonomo. Questo indice misura la qualità del clustering valutando quanto un oggetto sia simile al proprio cluster (coesione) rispetto agli altri cluster (separazione). Il punteggio varia da -1 a +1, dove:

- Un valore vicino a **+1** indica che l'oggetto è ben inserito nel suo cluster e lontano dagli altri.
- Un valore vicino a **0** indica che l'oggetto si trova vicino al confine tra due cluster.
- Un valore vicino a **-1** indica che l'oggetto potrebbe essere stato assegnato al cluster sbagliato.

Si calcola il Silhouette Score medio per diversi valori di *k* e si sceglie il *k* che massimizza questo punteggio.

Visualizzazione: Analisi delle Componenti Principali (PCA) Poiché i nostri dati hanno più di due dimensioni (feature), per visualizzare i cluster si è utilizzata la PCA. È una tecnica di riduzione dimensionale che trasforma le feature originali in un nuovo insieme di variabili non correlate (le componenti principali), ordinate per varianza decrescente. Selezionando le prime due componenti, è possibile proiettare i dati e i cluster su un grafico 2D, preservando la massima quantità possibile della varianza originale.

2. Applicazione e Risultati

Sono state condotte due analisi di clustering distinte per esplorare i dati da due prospettive complementari. La prima idea è stata usare le feature legate direttamente alla salute mentale (ansia, depressione, insonnia, OCD) per cercare pattern e profili all'interno del dataset; il secondo approccio, invece, andava nella direzione opposta cercando pattern sull'attività musicale. L'approccio che ho ritenuto piu solido, alla fine, è stato il primo.

2.1. Clustering basato su Ore di Ascolto e Salute Mentale

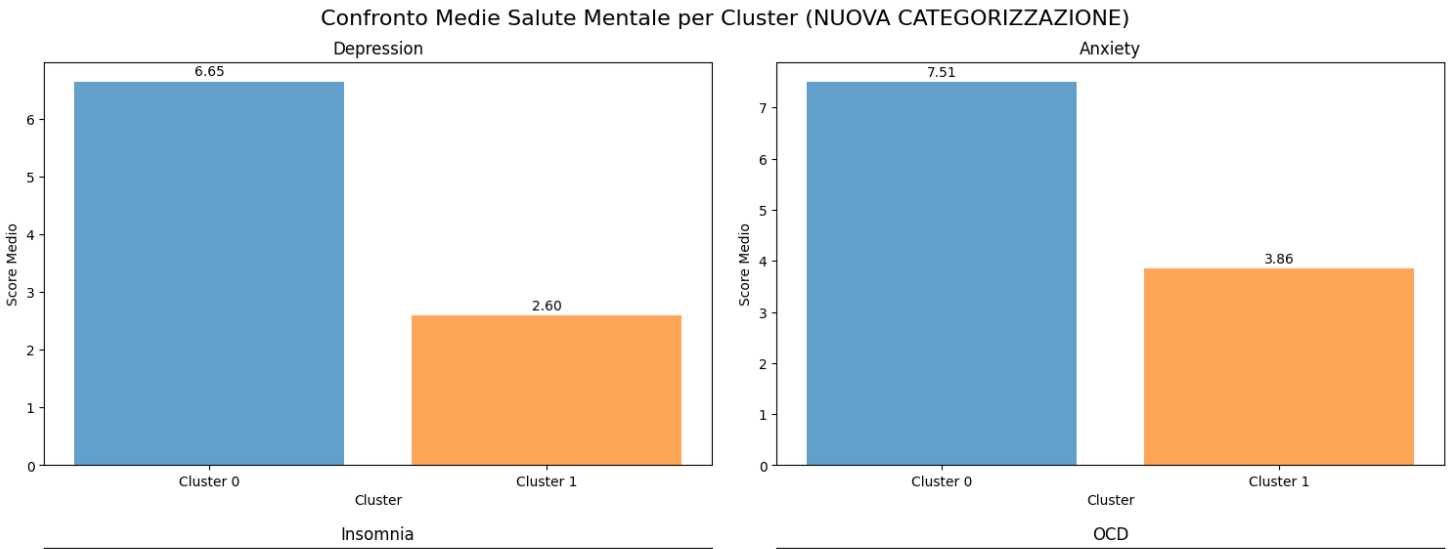
La prima e principale analisi mirava a segmentare gli utenti in base al loro **profilo di salute mentale**.

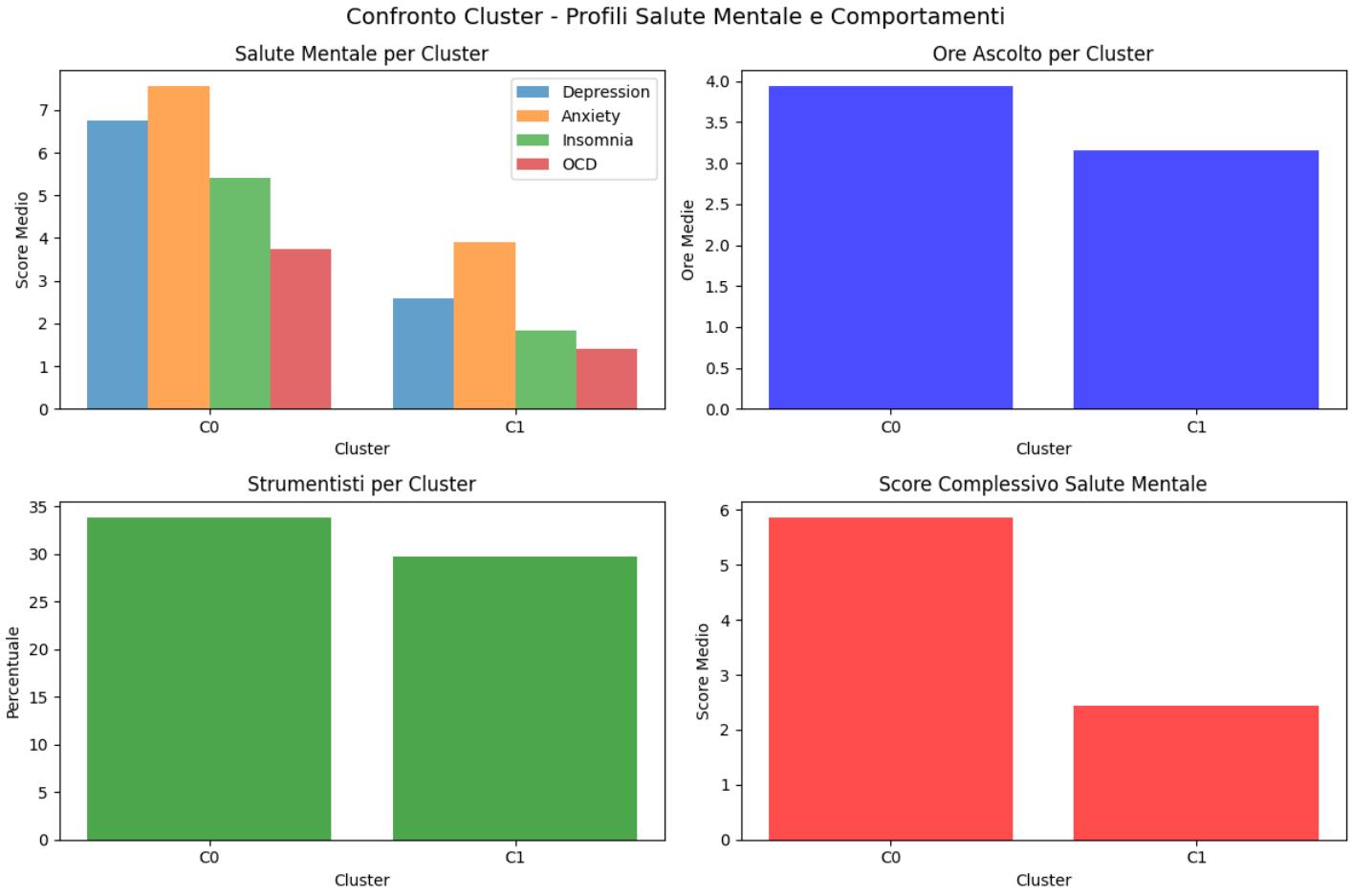
- **Feature Selezionate**: Le variabili utilizzate sono stati i punteggi di Depressione, Ansia, Insonnia e OCD. Per garantire che l'algoritmo K-Means non fosse influenzato dalla diversa scala delle variabili, i dati sono stati standardizzati con `StandardScaler`.
- **Scelta di k**: L'analisi del Silhouette Score ha indicato che la suddivisione ottimale del campione era in **k=2 cluster**. Questo suggerisce l'esistenza di due macro-profili dominanti. Da notare che il miglior punteggio, per k=2, è comunque non eccessivamente alto ma accettabile.
    - k=2, silhouette=0.256
    - k=3, silhouette=0.217
    - k=4, silhouette=0.196

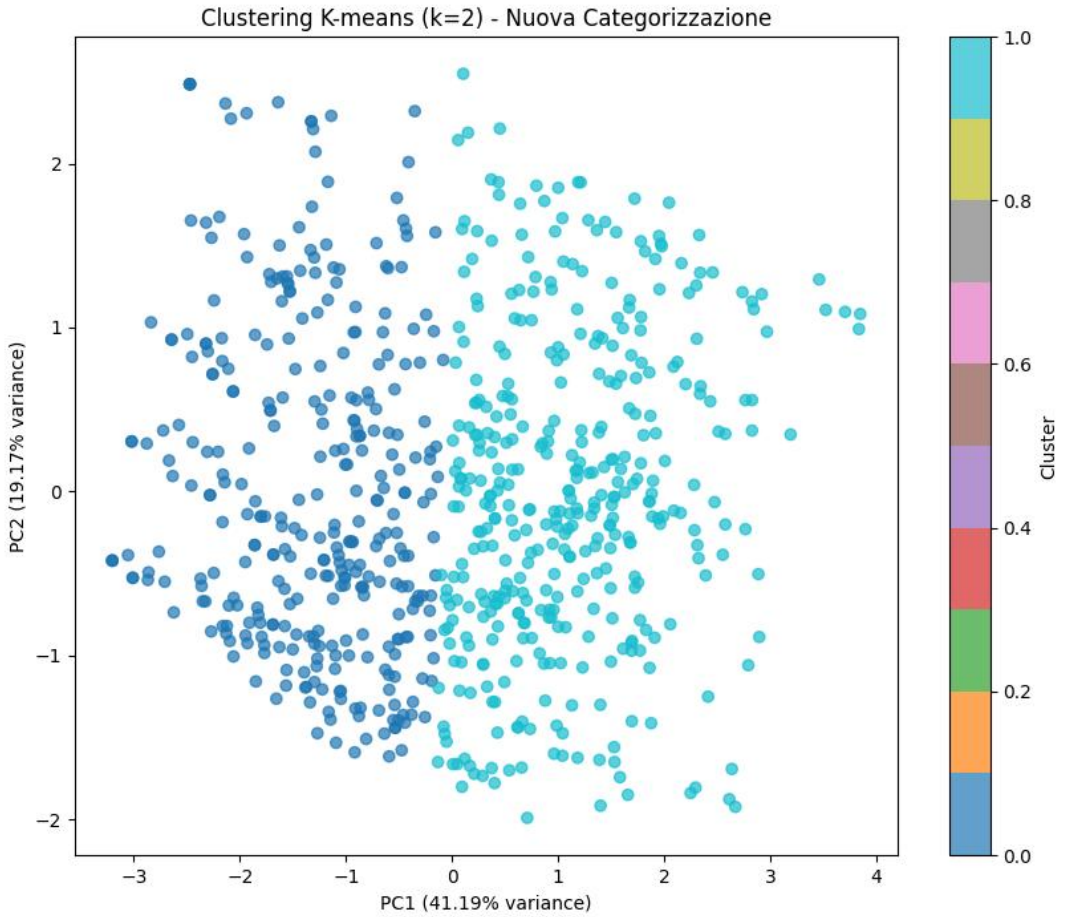
- k=5, silhouette=0.186
- k=6, silhouette=0.182
- k=7, silhouette=0.185

- **Analisi dei Profili**: dall'analisi condotta sulle quattro feature di riferimento sono emesi due profili specifici. Si è deciso, inoltre, di raggruppare i due macro-gruppi per le differenze maggiori, il che è uscito fuori che le ore di ascolto musicale fossero la più grande differenza tra i due cluster.
  - **Cluster 0 ("Ascoltatori Moderati a Basso Rischio")**: Questo gruppo, il più numeroso, è caratterizzato da **punteggi significativamente più bassi** per tutte le metriche di salute mentale. Le loro abitudini di ascolto sono moderate, con una prevalenza delle categorie "Low" e "Mean". Il che significa che hanno un ascolto giornaliero di musica che va da 1.5 a 3.0 ore (cioè i riferimenti delle categorie Low e Mean).
  - **Cluster 1 ("Grandi Ascoltatori ad Alto Rischio")**: Questo cluster è definito da **punteggi mediamente più elevati** di depressione, ansia e insonnia, associati a un **consumo musicale giornaliero decisamente superiore**. Questo profilo suggerisce un possibile utilizzo della musica come fattore legato alla salute mentale.

La significatività statistica delle differenze tra i due cluster è stata confermata tramite t-test, che hanno mostrato p-value molto bassi per tutte le variabili analizzate, rafforzando la validità di questa segmentazione. Si nota anche che l'ascolto medio nel cluster 0 è di 2,9 ore, mentre nel cluster 1 di 4,0 ore.




Confronto Cluster - Profili Salute Mentale e Comportamenti
Salute Mentale per Cluster
Depression
Anxiety
Insomnia
OCD
Score Medio
Cluster
C0
C1
Ore Ascolto per Cluster
Ore Medie
Strumentisti per Cluster
Percentuale
Score Complessivo Salute Mentale
Score Medio



2.2. Clustering basato su Attività Musicale

Il secondo approccio è partito analizzando generi di riferimento, preferenze musicale e altri fattori legati ai gusti e alle preferenze di ogni individuo.

- **Feature Selezionate**: tutte ad eccezione di quelle legate alla salute mentale
- **Scelta di k**: Anche in questo caso, il Silhouette Score ha indicato **k=2** come numero ottimale di cluster con uno score di 0.42
- **Analisi dei Profili**: L'analisi ha segmentato la popolazione in due gruppi principali:
    - PROFILO 0: "GIOVANI ASCOLTATORI INTENSIVI" (n=635 persone, 87%)
        - Caratteristiche Principali:
        - Età media: 21.3 anni (molto più giovani della media)
        - Ore di ascolto medie: 3.66 ore/giorno

- BPM medio: 123.79
- Attività Musicale: Praticamente nessuno è strumentista o compositore (0%). Sono principalmente "consumatori" di musica.
- Generi Preferiti:
- L'analisi mostra che i generi più caratteristici hanno un punteggio medio di 0, il che indica che questo cluster è molto eterogeneo e non ha una preferenza netta per un genere specifico. Sono ascoltatori onnivori.
- Profilo di Salute Mentale Associato:
- Depression: 4.96
- Anxiety: 6.02
- Insomnia: 3.70
- OCD: 2.75

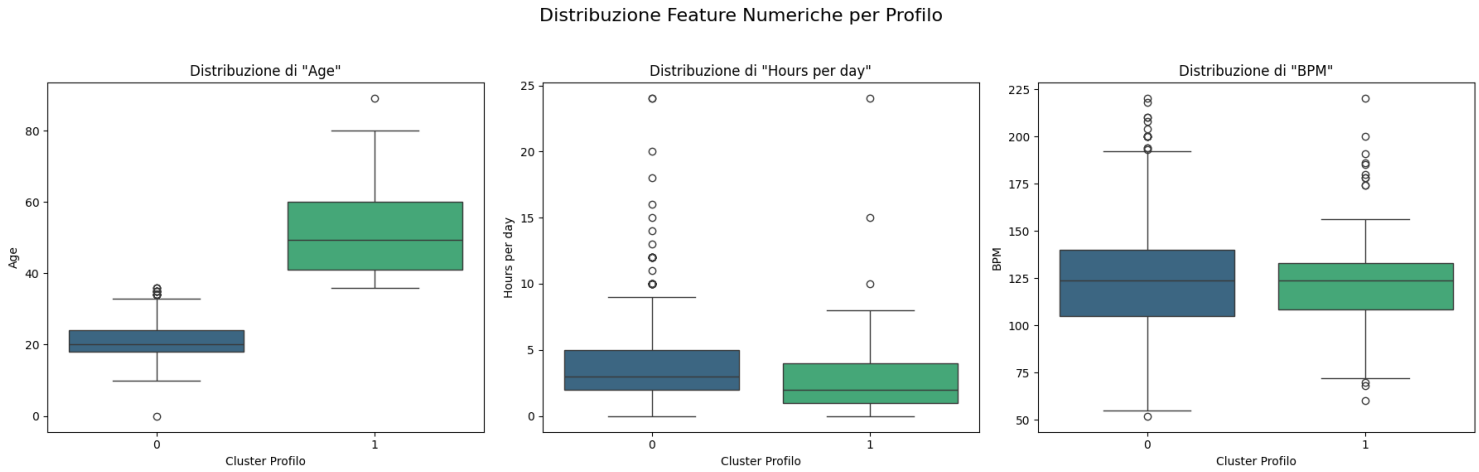
- PROFILO 1: "ASCOLTATORI ADULTI E MISURATI" (n=92 persone, 13%)
  - Caratteristiche Principali:
  - Età media: 51.3 anni (significativamente più anziani)
  - Ore di ascolto medie: 3.02 ore/giorno (ascoltano leggermente meno)
  - BPM medio: 124.58 (molto simile all'altro gruppo)
  - Attività Musicale: Anche in questo gruppo, la partecipazione attiva è quasi nulla.
  - Generi Preferiti:
  - Similmente al primo gruppo, non emerge un genere dominante, indicando gusti vari anche in questa fascia d'età.
  - Profilo di Salute Mentale Associato:
  - Depression: 3.52
  - Anxiety: 4.47
  - Insomnia: 3.90
  - OCD: 1.77

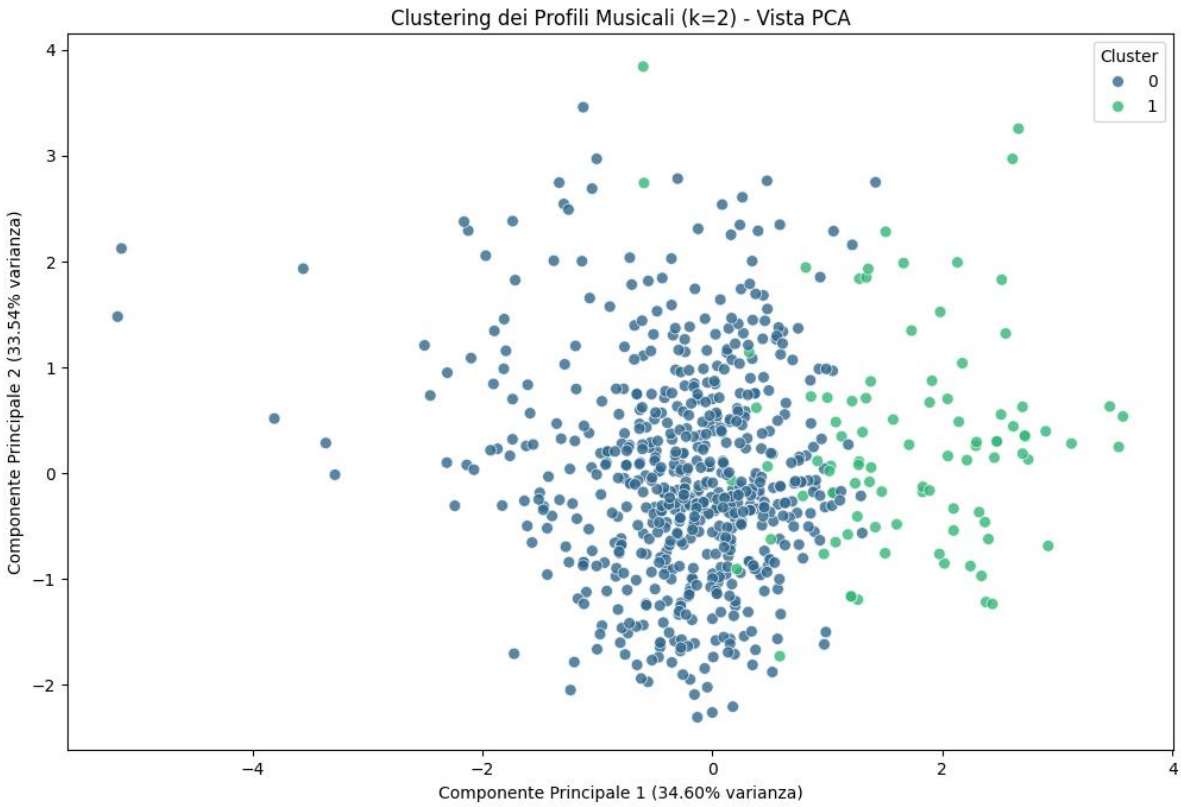
Nonostante i risultani non siano pessimi, questo approccio l'ho reputato fallace poichè la componente età (seppur valutata con un peso inferiore, di 0.1) inficia troppo i dati. Questo è un problema per la distribuzione complessiva dell'età, come accennato anche in fase di analisi e commento dei dati.

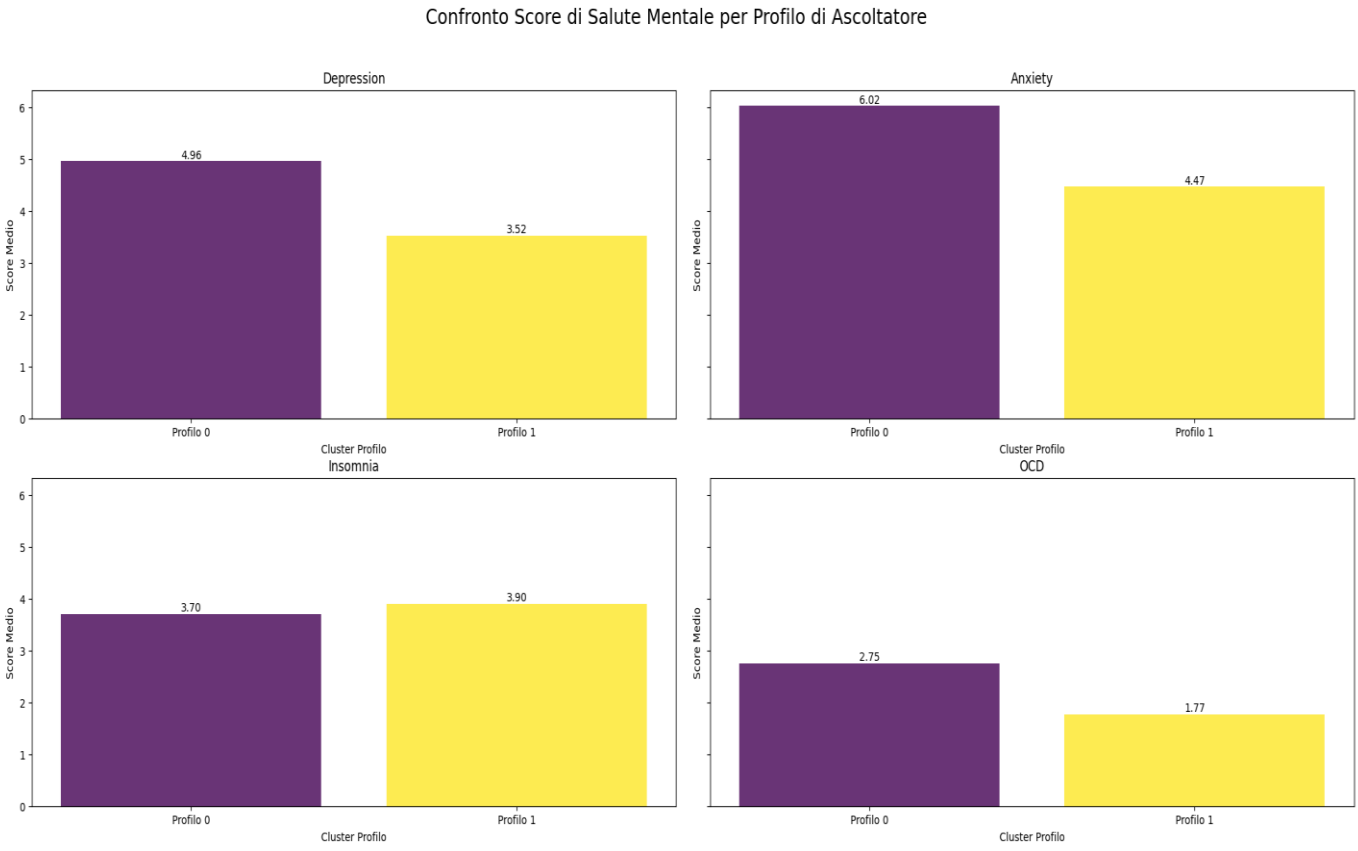
Di seguito tutti i grafici generati in riferimento al secondo approccio.

```
# Riduci l'impatto dell'età applicando un peso
age_weight = 0.1
profile_features['Age'] = profile_features['Age'] * age_weight
```

Come si puo notare, il valore età squilibra completamente l'apprendimento e per ciò è ridotto di peso.




Clustering dei Profili Musicali (k=2) - Vista PCA
Cluster
0
1
Componente Principale 2 (33.54% varianza)
Componente Principale 1 (34.60% varianza)
4
3
2
1
0
−1
−2
−4
−2
0
2
4


Confronto Score di Salute Mentale per Profilo di Ascoltatore
Depression
Anxiety
Insomnia
OCD
Score Medio
Cluster Profilo
Profilo 0
Profilo 1
4.96
3.52
6.02
4.47
3.70
3.90
2.75
1.77

1. **La quantità di ascolto è un forte discriminante**: La distinzione più netta emersa dai dati è tra un ascolto moderato associato a benessere e un ascolto elevato associato a disagio psicologico.
2. **Il ruolo attivo conta**: Essere un musicista attivo (strumentista/compositore) è associato a profili di salute mentale mediamente migliori rispetto all'ascolto passivo.

Dopo aver esplorato entrambi gli approcci la scelta che ho reputato migliore, in conseguenza alle ipotesi e conclusioni esposte sopra, è quella del cluster K-Means sulla base delle feature riguardanti la salute mentale. Da quel clustering si partirà come base successivamente per un modello basato sull'apprendimento supervisionato. Infatti, quel cluster fornirà le **etichette data-driven** (Cluster 0 e Cluster 1) che rappresentano la variabile target per la successiva fase di **apprendimento supervisionato**. L'obiettivo diventerà quindi costruire un modello in grado di predire, sulla base delle caratteristiche di un utente, a quale di questi profili di salute mentale è più probabile che appartenga.

Apprendimento Supervisionato

1.Introduzione all'Apprendimento Supervisionato

In questa sezione viene descritto il processo di machine learning supervisionato implementato per analizzare la relazione tra le abitudini di ascolto musicale e la salute mentale. L'obiettivo era duplice:

1. **Classificazione**: Prevedere il "cluster" di appartenenza di un utente, basato sulle sue caratteristiche.

2. **Regressione:** Prevedere un punteggio aggregato di "gravità della salute mentale", calcolato come media dei livelli di ansia, depressione, insonnia e OCD.

Il percorso è stato iterativo, partendo da un approccio di base per poi evolvere verso una pipeline complessa e ottimizzata. Questo modus operandi è stato fondamentale per comprendere meglio i problemi dell'apprendimento del modello e per avere conclusioni finali più accurate e precise.

L'apprendimento supervisionato è una branca del machine learning in cui l'obiettivo è insegnare a un modello a fare previsioni basandosi su dati di esempio. La caratteristica "supervisionato" deriva dal fatto che forniamo al modello non solo i dati di input (le "features"), ma anche le risposte corrette (le "etichette" o "target"). Il modello impara la relazione tra input e output, in modo da poter prevedere l'output per nuovi dati mai visti prima.

All'interno di questo progetto l'apprendimento supervisionato è stato perseguito attraverso due forme:

1. **Classificazione**: L'obiettivo è prevedere una categoria discreta.
2. **Regressione**: L'obiettivo è prevedere un valore numerico continuo.

2. Algoritmi e Modelli Utilizzati

2.1 Random Forest (Foresta Casuale)

È un insieme di **alberi decisionali**. Un singolo albero decisionale è un modello semplice che pone una serie di domande "sì/no" sui dati per arrivare a una decisione. Da solo, un albero è spesso impreciso e tende a "imparare a memoria" i dati di addestramento, ciò crea quindi overfitting. Con alcune tecniche, il Random Forest evita l'overfitting:

1. **Bagging (Bootstrap Aggregating):** Crea centinaia di alberi, addestrando ciascuno su un sottoinsieme casuale dei dati di training. Questo assicura che gli alberi siano diversi tra loro.
2. **Casualità delle Features**: Quando ogni albero deve decidere come dividere i dati in un nodo, non considera tutte le features disponibili, ma

solo un sottoinsieme casuale. Questo aumenta ulteriormente la diversità tra gli alberi.

La previsione finale è la **media** (per la regressione) o la **moda** (il voto di maggioranza, per la classificazione) delle previsioni di tutti gli alberi.

```
# Random Forest
rf_params = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'class_weight': ['balanced', None]
}
print("Ottimizzando Random Forest...")
rf_grid = RandomizedSearchCV(
    RandomForestClassifier(random_state=42),
    rf_params,
    n_iter=20,
    cv=5,
    scoring='accuracy',
    random_state=42,
    n_jobs=-1
)
rf_grid.fit(X_train, y_train)
models['RandomForest'] = rf_grid.best_estimator_
```

È un algoritmo robusto, difficile da mandare in overfitting (grazie alla accortezza che gestisce di suo) e gestisce bene dati complessi senza richiedere una preparazione eccessiva.

2.2 Gradient Boosting & XGBoost

A differenza della Random Forest che costruisce alberi in parallelo, il Gradient Boosting li costruisce in **sequenza**, con una strategia così strutturata:

1. Il primo albero fa una previsione.

2. Il secondo albero viene addestrato per correggere gli errori commessi dal primo.

3. Il terzo albero corregge gli errori residui del secondo, e così via.

Ogni nuovo modello si concentra sulle previsioni più difficili, migliorando progressivamente il risultato complessivo. Il "Gradient" nel nome si riferisce al fatto che usa l'algoritmo di discesa del gradiente per minimizzare gli errori ad ogni passo.

```
# Gradient Boosting
gb_params = {
    'n_estimators': [100, 200],
    'learning_rate': [0.05, 0.1, 0.2],
    'max_depth': [3, 5, 7],
    'min_samples_split': [2, 5]
}
print("Ottimizzando Gradient Boosting...")
gb_grid = RandomizedSearchCV(
    GradientBoostingClassifier(random_state=42),
    gb_params,
    n_iter=20,
    cv=5,
    scoring='accuracy',
    random_state=42,
    n_jobs=-1
)
gb_grid.fit(X_train, y_train)
models['GradientBoosting'] = gb_grid.best_estimator_
```

Il XGBoost (Extreme Gradient Boosting) è una versione **ottimizzata e potenziata** del Gradient Boosting. È noto per la sua velocità e le sue performance. Include:

1. **Regolarizzazione (L1 e L2)**: Una tecnica per prevenire l'overfitting penalizzando i modelli troppo complessi.

2. **Gestione dei valori mancanti:** Può gestirli internamente in modo intelligente.

3. **Parallelizzazione:** È molto più veloce perché ottimizza l'uso dell'hardware.

2.3 Ridge Regression

È un modello di **regressione lineare** con una modifica cruciale: la regolarizzazione L2. Un modello lineare semplice cerca di trovare la linea che minimizza la distanza tra i punti dati e la linea stessa (spesso euclidea). A volte, però, per adattarsi ai dati di training, i coefficienti del modello (i "pesi" delle features) possono diventare enormemente grandi, rendendo il modello instabile e sensibile a piccole variazioni nei dati di input (overfitting).

La regolarizzazione Ridge aggiunge una "penalità" proporzionale al quadrato dei coefficienti. Questo "costringe" il modello a mantenere i pesi delle features più piccoli e distribuiti, rendendolo più stabile e generalizzabile. Il parametro in questione, quindi, controlla l'intensità di questa penalità.

2.4 Ensemble con VotingClassifier /VotingRegressor

È la forma più diretta di ensemble. Nella letteratura scientifica spesso si usa la massima "chiediamo il parere a un comitato di esperti e prendiamo una decisione di gruppo" per spiegare in maniera semplice l'idea dietro all'Enseble.

Nel progetto, avendo prima addestrato diversi modelli, l'Ensemble li valuta insieme e fornisce voti e medie e probabilità.

Le probabilità vengono mediate e la classe con la probabilità media più alta vince. Questo è generalmente più efficace del voto "secco" (hard voting).

L'idea è che i punti di forza di un modello possano compensare le debolezze di un altro. Se i modelli sono sufficientemente diversi e bravi, la

loro decisione collettiva è quasi sempre migliore di quella del singolo modello migliore. Solitamente, è quello con prestazioni maggiori. Quando cosi non è, il problema principale è una scarsa qualità dei dati. Questa precisazione sarà importante per le conclusioni finali.

3. Ulteriori concetti chiavi usati nel progetto

Feature Engineering: ho creato nuove features (**polinomiali**, **rapporti**, **statistiche**) perché i modelli, per quanto potenti, possono imparare solo dalle informazioni che gli vengono fornite. Questo passaggio è stato cruciale nel secondo approccio del ML.

Feature Selection: ho usato tecniche come **SelectKBest** (scegli le K features migliori) e **RFE** (elimina ricorsivamente le features peggiori) per ridurre il rumore, diminuire il rischio di overfitting e velocizzare l'addestramento, concentrando il modello solo sulle informazioni più rilevanti.

Hyperparameter Tuning (RandomizedSearchCV) : ogni modello ha delle "manopole" (iperparametri, es. il numero di alberi in una foresta). **RandomizedSearchCV** non prova tutte le combinazioni possibili (che sarebbe troppo lento e infernale a livello computazionale), ma un numero fisso di combinazioni casuali per trovare una configurazione quasi ottimale in un tempo ragionevole. È un giusto compromesso tra efficienza e risultati.

4.Primo Approccio – Più semplice e più diretto (fallimentare)

Risultati di Baseline:

Il primo tentativo ha seguito un approccio standard e diretto, con l'obiettivo di stabilire una baseline di performance.

Metodologia:

1. **Selezione delle Feature:** Sono state utilizzate unicamente le colonne numeriche grezze del dataset, escludendo solo l'identificativo del cluster.

2. **Preprocessing Semplice**: È stata applicata una semplice standardizzazione (`StandardScaler`) per normalizzare i dati.

3. **Modellazione**: Sono stati addestrati modelli di classificazione standard (es. `RandomForestClassifier`, `LogisticRegression`) con i loro iperparametri di default.

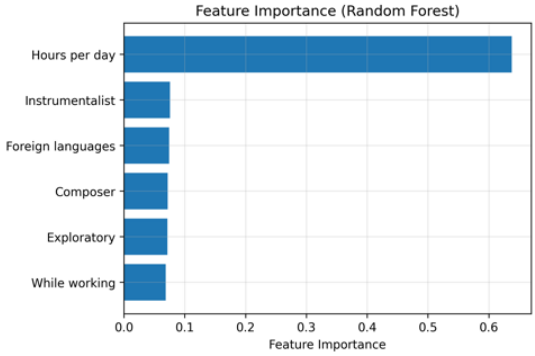
Risultati e Limiti:

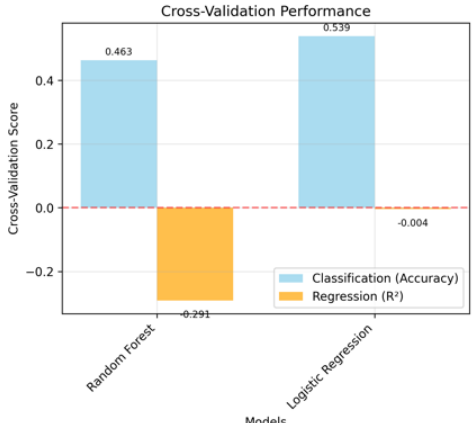
Questo approccio iniziale ha prodotto risultati deludenti, evidenziando i limiti di una metodologia troppo semplicistica per un problema così complesso:

Classificazione: L'accuratezza si attestava su valori bassi, spesso di poco superiori a una predizione casuale (60% circa). Il modello non riusciva a catturare le relazioni complesse e non lineari tra le abitudini di ascolto e il profilo psicologico dell'utente.

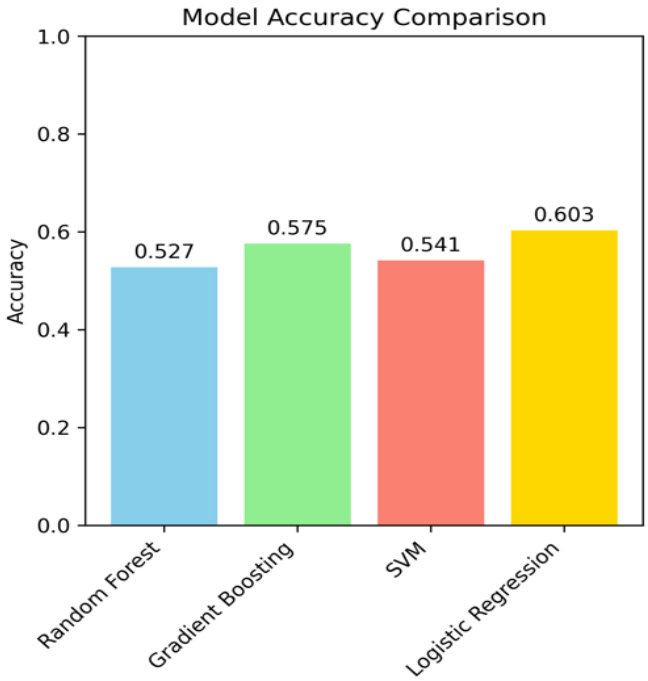
Regressione: I risultati erano ancora peggiori. Il coefficiente di determinazione (R^2) era prossimo allo zero o addirittura negativo, indicando che il modello non era in grado di spiegare la varianza nel punteggio di salute mentale. In pratica, era completamente inutile.

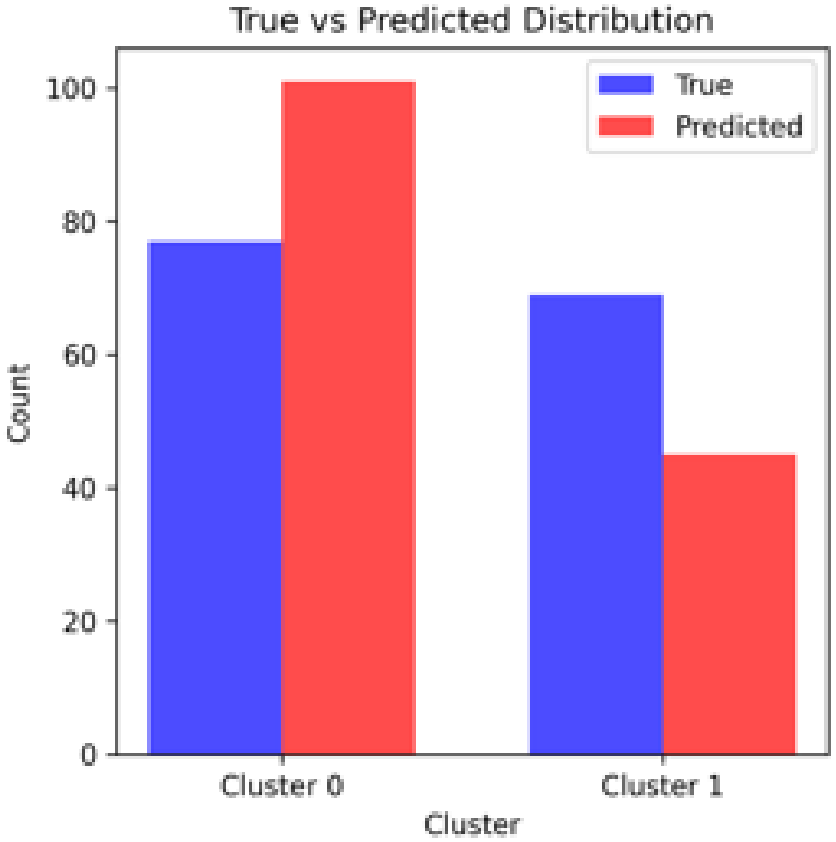
Questo approccio semi-fallimentare è stato fondamentale per comprendere che le feature grezze non erano sufficientemente informative. Le relazioni tra musica e salute mentale non sono lineari.

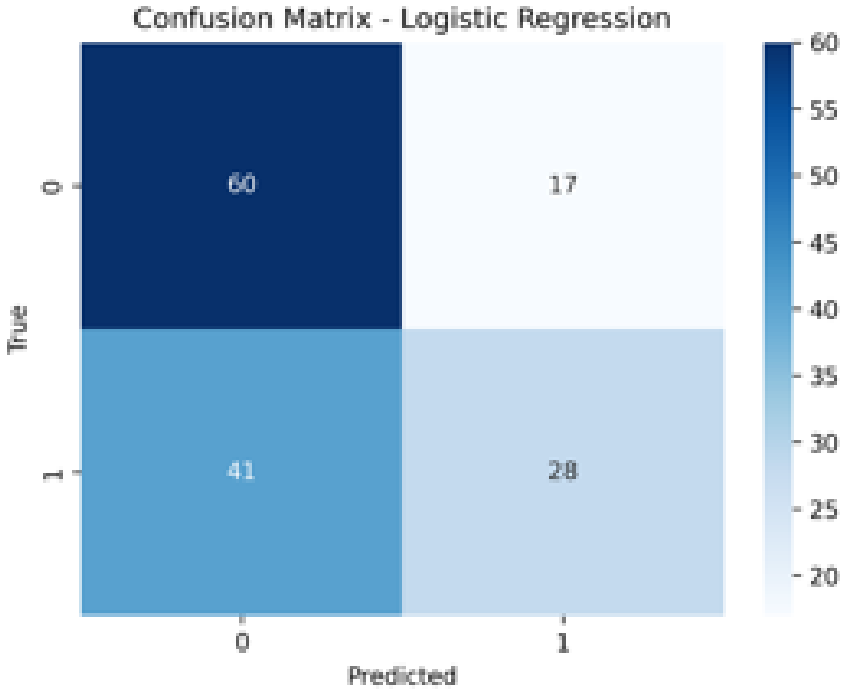




Da notare come la regressione lineare è completamente inefficiente. La feature più importante è invece hours per day.




True vs Predicted Distribution
True
Predicted
Count
100
80
60
40
20
0
Cluster 0
Cluster 1
Cluster


Confusion Matrix - Logistic Regression
60
17
41
28
True
0
1
Predicted
0
1
60
55
50
45
40
35
30
25
20

5. Secondo Approccio - Avanzato con Feature Engineering e Ottimizzazione

Questo è l'approccio finale presente nel progetto. Una considerazione notevole da fare è che la qualità del modello dipende dalla qualità dei dati di input.

Prevenzione del Data Leakage: questa è stata la scelta più critica. Per evitare che il modello "barasse", sono state rimosse dal set di feature (X) non solo la variabile target diretta ma anche tutte le variabili che potevano contenerla implicitamente. Per la regressione, ad esempio, sono state rimosse le colonne riguardanti i fattori descrittivi della salute, poiché la loro media costituiva il target. Questo garantisce che il modello impari a predire basandosi unicamente sulle abitudini musicali e demografiche, non sui sintomi stessi. Questa precisazione iniziale è dovuta per esporre anche una problematica avuta. Infatti, durante un test, una porzione di codice ciclava male e ha preso in considerazione le quattro variabili target e ha dato come score finale di classificazione e regressione qualcosa di insensato. La classificazione ha registrato un F1 score da 98% e la regressione una R2 score del 100%. Espongo didatticamente questo caso per evidenziare quanto sia importante, ai fini utili, essere molto attenti nella scelta delle variabili e feature da usare per centrare i target.

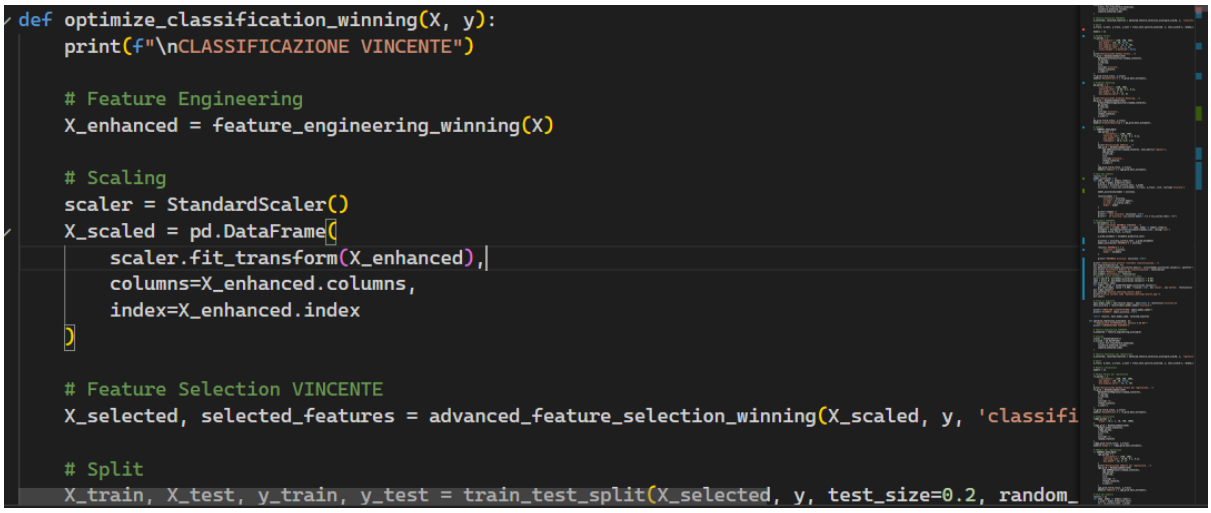
5.1 Feature Engineering: Creare Valore dai Dati

Questa fase è stata progettata per estrarre relazioni nascoste e creare nuove feature più potenti. Si sono usati tre approcci, infine combinati, per effettuare un buon feature engineering:

1. **Interazioni Polinomiali (PolynomialFeatures):** Sono state create feature di interazione tra le variabili più importanti (es., hours per day, bpm, age...). La teoria è che l'effetto di una variabile possa dipendere dal valore di un'altra. È stata usata l'opzione

`interaction_only=True` per evitare di creare potenze (es, age^2), concentrandosi solo sulle sinergie tra variabili diverse.

2. **Feature di Rapporto:** Sono state create nuove feature calcolando il rapporto tra alcune variabili numeriche. Questo aiuta a normalizzare i dati e a catturare relazioni relative che i modelli lineari faticano a vedere.
3. **Feature Statistiche Aggregate:** Per ogni utente, sono state calcolate la media, la deviazione standard, il massimo e il minimo di tutte le sue features numeriche. Questo crea un "profilo statistico" dell'utente, che si è rivelato molto informativo.



```
def optimize_classification_winning(X, y):
    print(f"\nCLASSIFICAZIONE VINCENTE")

    # Feature Engineering
    X_enhanced = feature_engineering_winning(X)

    # Scaling
    scaler = StandardScaler()
    X_scaled = pd.DataFrame(
        scaler.fit_transform(X_enhanced),
        columns=X_enhanced.columns,
        index=X_enhanced.index
    )

    # Feature Selection VINCENTE
    X_selected, selected_features = advanced_feature_selection_winning(X_scaled, y, 'classifi

    # Split
    X_train, X_test, y_train, y_test = train_test_split(X_selected, y, test_size=0.2, random_
```

5.2 Selezione delle Feature: Ridurre il Rumore e Aumentare la Performance

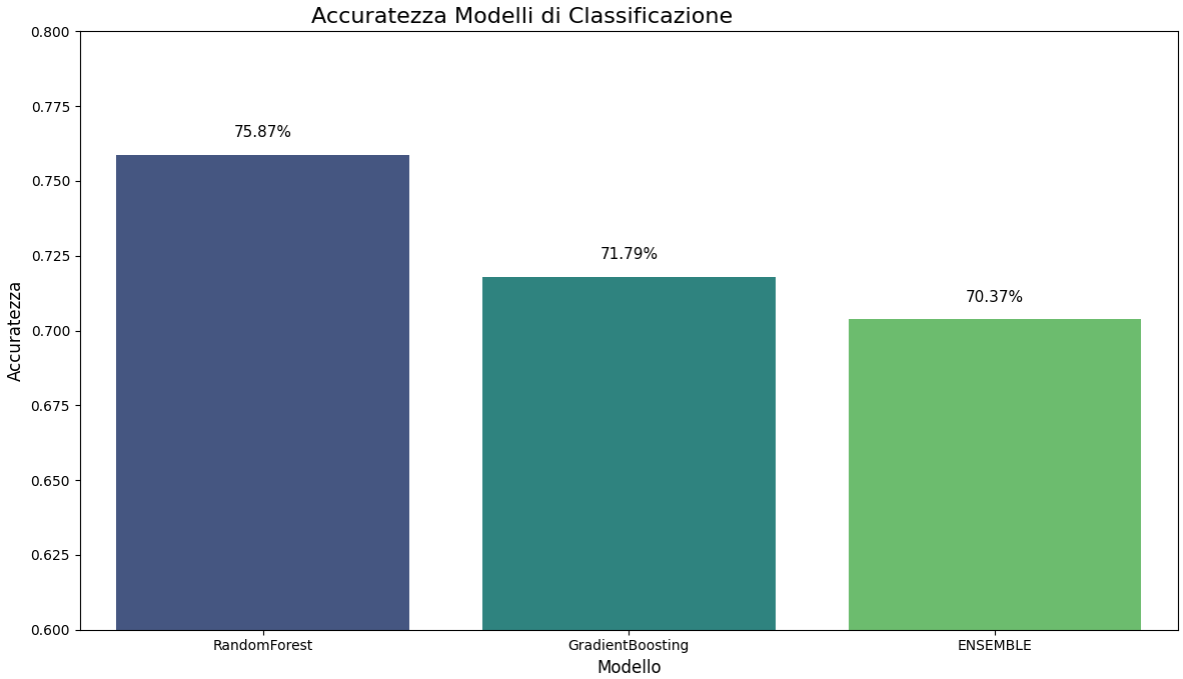
Dopo il Feature Engineering, il numero di feature era aumentato notevolmente. Si è passati da circa 25 features iniziali e semplice a più di 70 features, tra cui molte polinomiali (che non è buono avere in cosi grosso numero sbilanciato). La selezione è stata cruciale per eliminare le feature irrilevanti o ridondanti, che introducono rumore e aumentano il rischio di overfitting. È stato adottato un approccio combinato:

1. **SelectKBest:** Seleziona le feature che, individualmente, hanno la più forte relazione statistica con il target. È un metodo veloce ma ignora le interazioni.

2. **Recursive Feature Elimination (RFE)**: Addestra iterativamente un modello, rimuovendo ad ogni passo la feature meno importante. È più robusto perché considera l'interazione tra le feature.

3. **SelectFromModel**: Utilizza un modello (in questo caso, `RandomForest`) per calcolare l'importanza di ogni feature e scarta quelle al di sotto di una certa soglia.

La scelta finale è stata quella di unire le feature selezionate da tutti e tre i metodi, come già accennato prima. Questo approccio ibrido garantisce di mantenere solo le feature che sono considerate importanti da molteplici punti di vista, creando un set di dati finale piccolo ma estremamente potente.



5.3 Ottimizzazione e Addestramento dei Modelli

Prima dell'addestramento, tutte le feature sono state scalate con StandardScaler. Questo è un passo fondamentale per garantire che nessuna variabile domini le altre solo perché ha una scala numerica più grande.

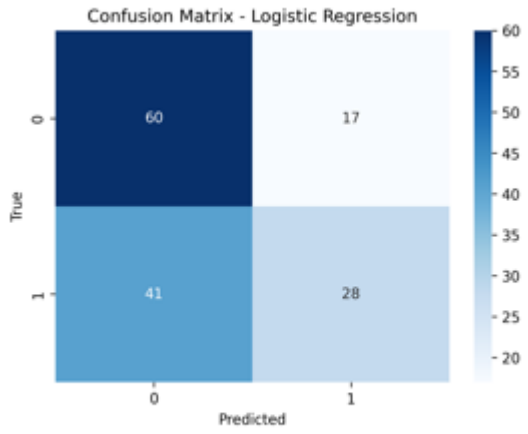
Sono stati scelti modelli ad alte prestazioni come Random Forest, Gradient Boosting e XGBoost.

Per l'iper parametrizzazione (**Hyperparameter Tuning)**, invece di provare ogni possibile combinazione di iperparametri (**GridSearchCV**), è stata usata una ricerca casuale. Questo approccio testa un numero fisso di combinazioni casuali, risultando molto più efficiente e spesso trovando soluzioni altrettanto valide, se non migliori.

Come passo finale, i modelli ottimizzati sono stati combinati in un "super-modello" (ensemble).

Per la regressione si sono scelte le seguenti metriche di valutazione:

- **R^2 (R-squared)**: Misura la proporzione di varianza nella variabile target che è prevedibile dalle features. Un valore di 1.0 è una previsione perfetta, 0.0 significa che il modello non è migliore di prevedere semplicemente la media del target. Valori negativi indicano un modello pessimo.
- **RMSE (Root Mean Squared Error):** Misura la deviazione media tra i valori predetti e i valori reali, nella stessa unità di misura del target. Un valore più basso è migliore.



5.4 Conclusioni ed Interpretazioni finali – Dataset non adeguato

Il passaggio da un approccio semplicistico a una pipeline avanzata ha trasformato i risultati. Mentre la baseline era inutilizzabile, l'approccio finale ha dimostrato che, con un'adeguata preparazione dei dati e tecniche di modellazione avanzate, è possibile estrarre segnali predittivi significativi.

La **Feature Engineering** si è rivelata la fase a più alto impatto, confermando che il successo nel machine learning non deriva solo dalla

scelta del modello, ma soprattutto dalla capacità di rappresentare i dati in un modo che renda le relazioni sottostanti facilmente apprendibili.

Nonostante un miglioramento delle prestazioni e degli score di accuracy, f1 e r2 si nota come il classificatore è stato comunque appena sufficiente, con un F1 Score di circa il 70-74%. Non male, ma un buon risultato soddisfacente dovrebbe aggirarsi attorno al 85-95% (di più sarebbe overfitting o data leakage). Mentre il regressore è stato un completo fallimento in ogni approccio possibile, nonostante il giusto uso e l'ottimizzazione di ogni algoritmo e strategia. La conclusione finale è che, purtroppo, nonostante gli sforzi il dataset iniziale non hanno rappresentato un KB (Knowledge Base) adeguata del mondo per poter dare un giusto input ai modelli per essere addestrati adeguatamente. Nonostante un giusto numero di record (727), il campione presente è stato troppo uniforme in molte feature. Altra considerazione da fare, anche leggendo in parte un po' di letteratura scientifica sull'argomento della salute mentale correlata all'ascolto musicale è che una possibile correlazione è possibile. Cioè, i modelli hanno rilevato una parziale correlazione ma per un motivo o per l'altro (come la scarsa qualità del dataset) non si è potuto verificare con certezza tale relazione. La tesi finale è che è ancora difficile conoscere a fondo il funzionamento della mente umana ed è difficile, specie con dati scarsi, trarre una conclusione "booleana" su tale argomento. Anche il fallimento del modello (specie del regressore) fornisce come risposta che la psiche umana è assai più complessa da analizzare. Nonostante un punteggio pessimo, quindi, il risultato da una piccola conferma in una direzione.

Belief Network (bonus)

Obiettivo

- Modellare le dipendenze probabilistiche tra fattori musicali/lifestyle e condizioni di salute mentale (Ansia, Depressione, Insonnia, OCD).

- Ottenere inferenze interpretabili (marginali, condizionali, congiunte) e visualizzazioni chiare a supporto delle conclusioni.

Una Rete Bayesiana (BN) è un grafo aciclico diretto (DAG) in cui i nodi sono variabili casuali e gli archi rappresentano dipendenze direzionali. Ogni nodo ha una tabella di probabilità condizionale (CPT) rispetto ai suoi genitori.

- Vantaggi: interpretabilità, gestione naturale dell'incertezza, inferenza probabilistica anche con dati mancanti, possibilità di analizzare scenari "what-if".
- Inferenza: calcolo di P(X|E) mediante metodi esatti (Variable Elimination) o approssimati (sampling). Qui usiamo Variable Elimination per accuratezza data la dimensione moderata della rete.

Apprendimento

Struttura: score-based (Hill Climbing) con BIC, che massimizza l'adattamento penalizzando la complessità del grafo.

Parametri: stima Bayesiana con prior BDeu per regolarizzare le CPT ed evitare probabilità zero.

Motivazioni delle scelte

- BN per questo problema: consente di esprimere relazioni direzionali e rispondere a domande del tipo "Quanto aumenta P(Depressione=Alta) se Ansia=Alta?" in modo interpretabile, a differenza di modelli puramente predittivi.
- Structure learning score-based (HC+BIC): robusto e scalabile su dataset medi, BIC bilancia fit/complessità riducendo overfitting.
- Discretizzazione: necessaria per CPT discrete e utile per interpretabilità (classi Low/Medium/High); riduce dimensionalità e rumore in variabili continue.
- Variable Elimination: inferenza esatta sufficiente per il numero di variabili selezionato.
- Prior BDeu: smoothing essenziale con categorie poco frequenti.

Implementazione (file: src/bayesian_analysis.py)

1) Preprocessing

- Selezione variabili e rinomina:

  - Età, Ore di ascolto, Genere preferito, Effetto della musica, Ansia, Depressione, Insonnia, OCD.

  - Rinominata “Music effects” in “Music_effect_on_mood”.

- Discretizzazione:

  - Age → Age_group: Teen, Young_Adult, Adult, Senior.

  - Hours per day → Listening_hours: Low, Medium, High, Very_High.

  - Condizioni mentali → {Low, Medium, High} tramite soglie 0–3, 4–7, 8–10.

- Final set (categoriche e senza NaN): Age_group, Listening_hours, Fav genre, Music_effect_on_mood, Anxiety_level, Depression_level, Insomnia_level, OCD_level.

Inferenza e visualizzazioni

- Inference engine: pgmpy.VariableElimination.
- Query automatizzate e salvataggi:
- Distribuzioni marginali per variabili di salute mentale → marginal_distributions.png + tabelle in console.
- Inferenza condizionata (es. P(Depression_level | Anxiety_level=Low/Medium/High)) → images/conditional_inference.png.
- “Matrice di correlazione bayesiana” (P(colonna=High | riga=High) tra variabili di salute mentale) → bayesian_correlation_matrix.png + tabella.
- Influenza di una variabile esterna (es. Fav genre/Age_group/Listening_hours) su P(target=High) → images/influence_analysis.png.
- Sommario modello (variabili, numero archi, metodo) → images/bayesian_model_summary.png.

- Robustezza: l’algoritmo adatta automaticamente le query alle variabili effettivamente presenti nel grafo appreso.

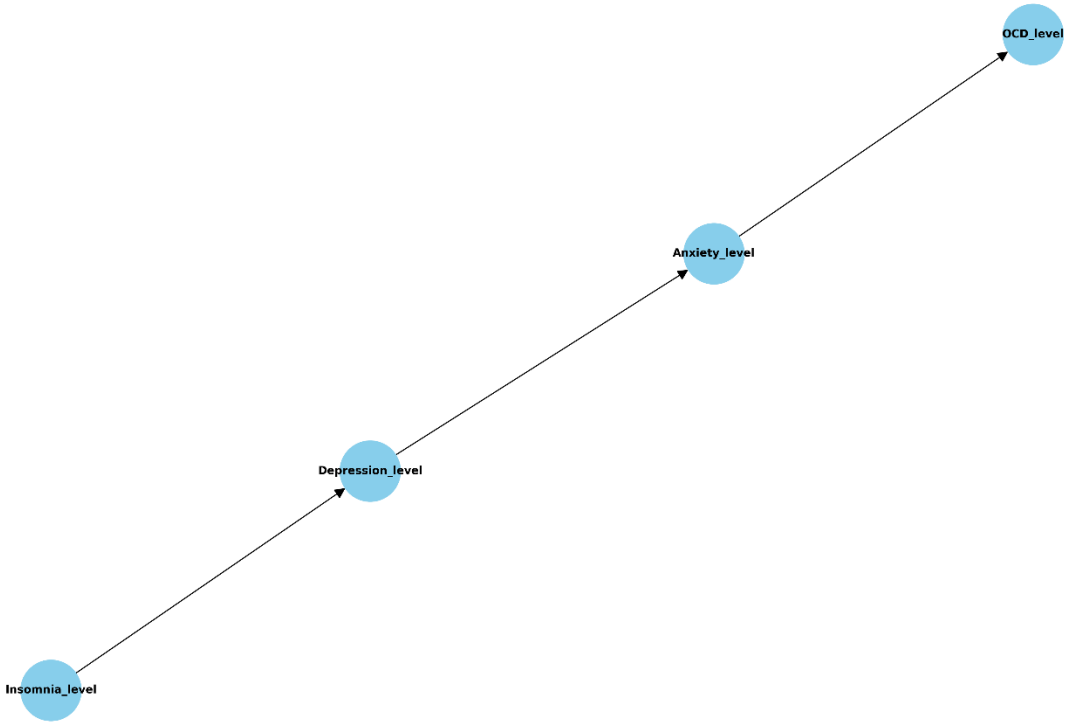
Perché queste visualizzazioni

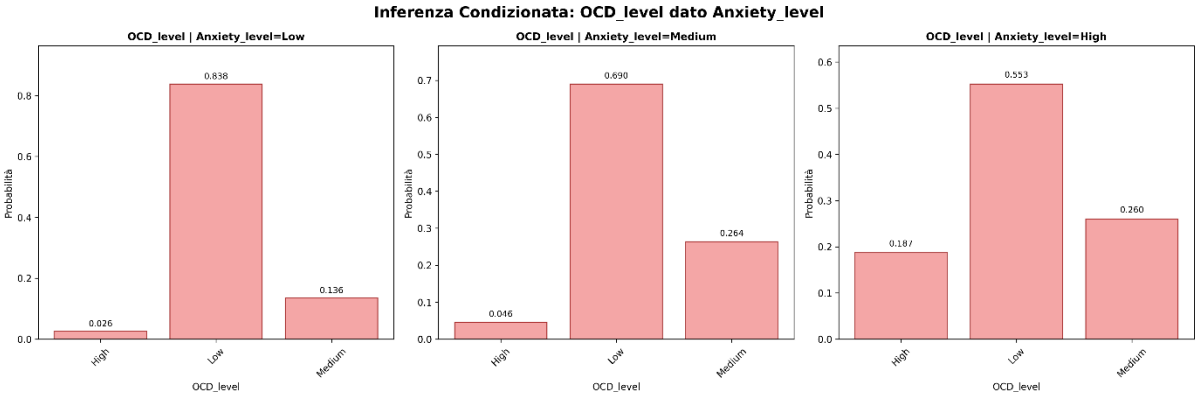
- Struttura del grafo: rende immediata la lettura delle possibili relazioni direzionali apprese dai dati.
- Marginali: indicano la distribuzione a priori delle condizioni mentali.
- Condizionali: mostrano effetti potenziali tra condizioni (es. ansia su depressione).
- Matrice P(High|High): panoramica sintetica della co-occorrenza condizionata “alta”.
- Influenza esterna: collega fattori musicali/stilistici a outcome di salute mentale in modo interpretabile.

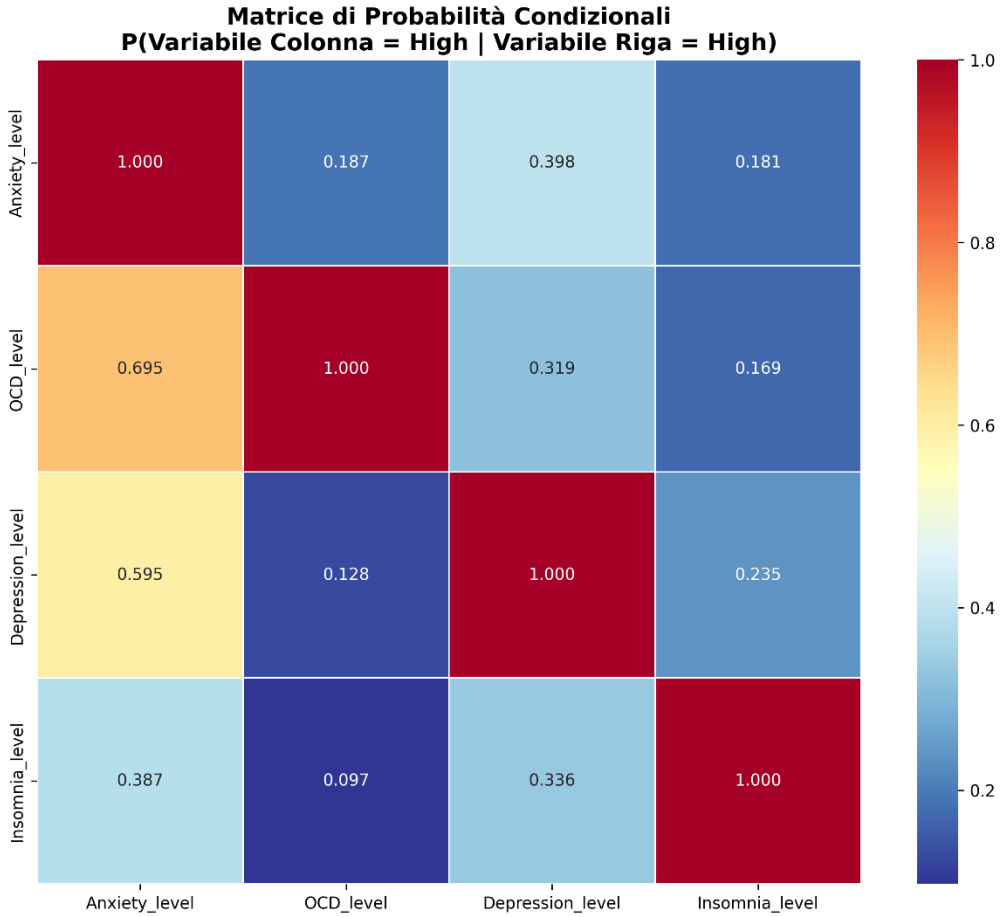
In sintesi

La BN integra conoscenza probabilistica interpretabile nel flusso analitico: apprende relazioni tra variabili chiave, consente inferenze condizionali utili alla discussione, e produce visualizzazioni immediate a supporto delle conclusioni della relazione. Non risultà un granchè sempre in relazione ai problemi del dataset iniziale

Struttura della Rete Bayesiana (appresa dai dati)







Riassunto del Modello Bayesiano

Categoria	Variabili	Descrizione
Salute Mentale	Anxiety_level, OCD_level, Depression_le	Variabili target di salute mentale
Fattori Esterni		Variabili di influenza esterna
Totale Variabili	4	Numero totale di nodi nel grafo
Archi nel Grafo	3	Connessioni causali apprese
Algoritmo	Hill Climb Search + BIC	Metodo di structure learning
Inferenza	Variable Elimination	Metodo di inferenza probabilistica