SPOTIFY ANALYSIS

Fondamenti di Analisi Dati 2023/2024

Matteo Imbrosciano 1000014829

Dataset da analizzare

IL DATASET È STATO PRESO DAL SEGUENTE INDIRIZZO : SPOTIFY-DATASET-2023

SONO PRESENTI I SEGUENTI DATI:

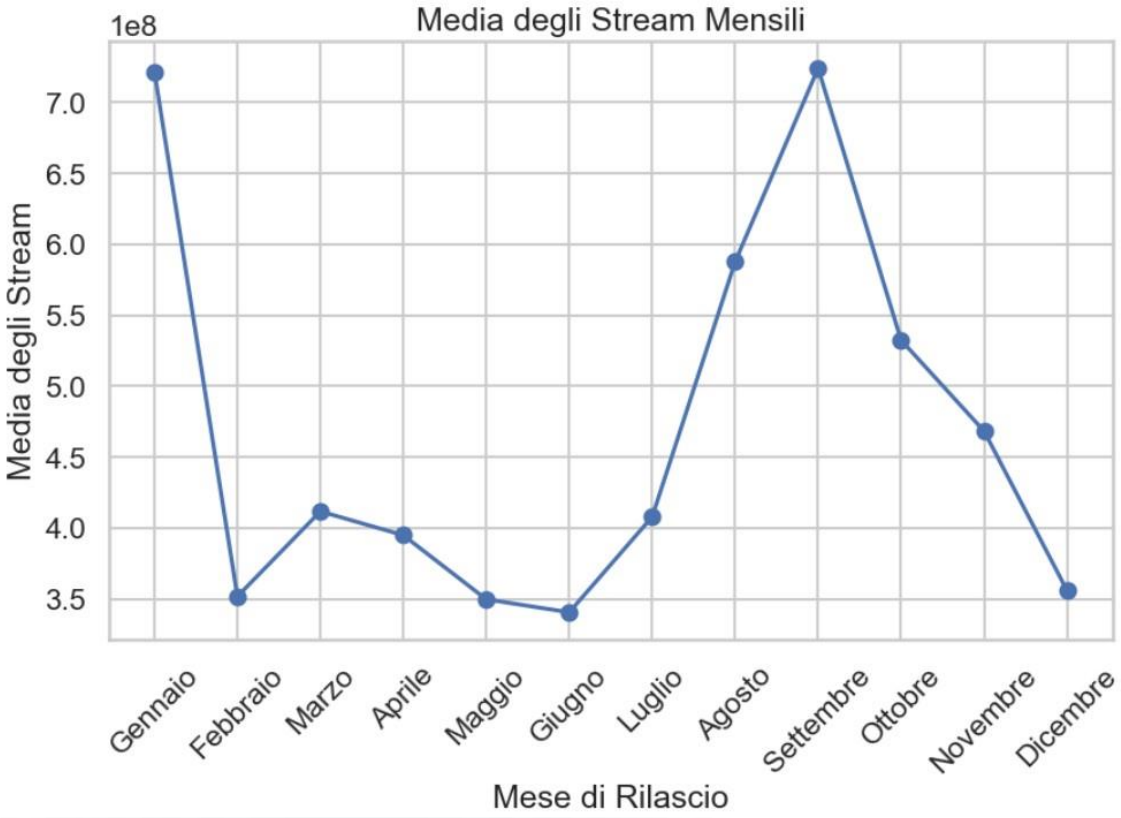
TRACK_NAME, ARTIST(S)_NAME, ARTIST_COUNT, RELEASED_YEAR,
RELEASED_MONTH, RELEASED_DAY, IN_SPOTIFY_PLAYLISTS,
IN_SPOTIFY_CHARTS, STREAMS, IN_APPLE_PLAYLISTS, IN_APPLE_CHARTS,
IN_DEEZER_PLAYLISTS, IN_DEEZER_CHARTS, IN_SHAZAM_CHART, BPM,
KEY, MODE, DANCEABILITY_%, VALENCE_%, ENERGY_%,
ACOUSTICNESS_%, INSTRUMENTALNESS_%, LIVENESS_%, SPEECHINESS_%,
RELEASED_DATE, RELEASED_WEEKDAY, STREAMS_CATEGORY, RELEASE_YEAR,
IN_SPOTIFY_PLAYLIST, IN_APPLE_PLAYLIST, IN_DEEZER_PLAYLIST

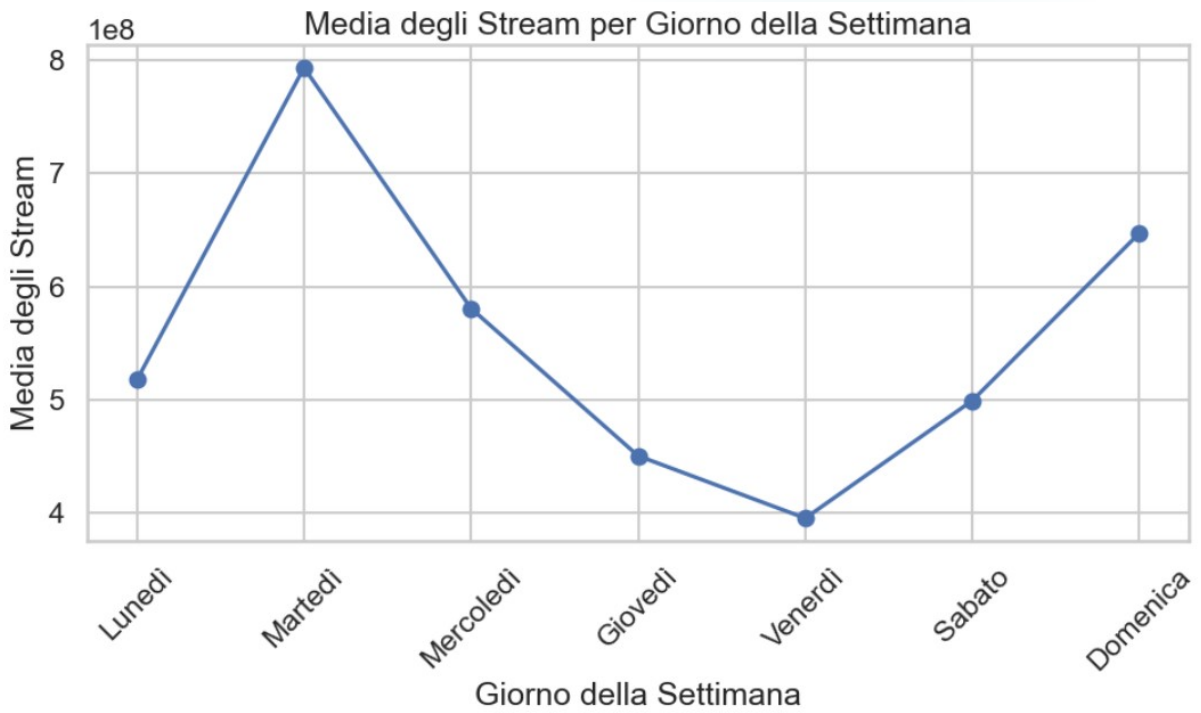
RAPPRESENTANO INFORMAZIONE SULLE VARIE CANZONI DEL 2023.

Obiettivi analisi

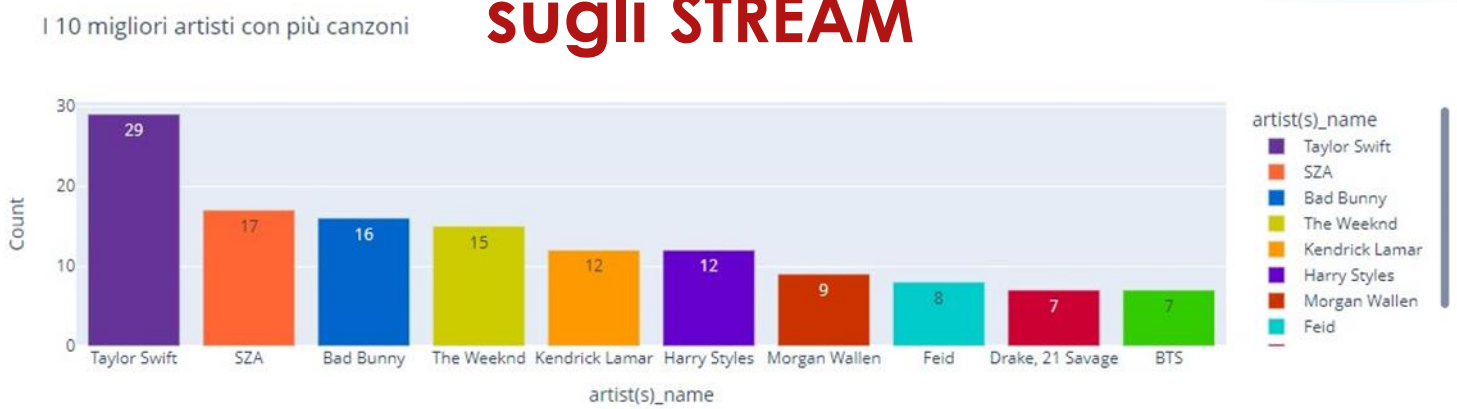
- Effettuare un'analisi esplorativa su:
  - Trend Temporali;
  - Stream;
  - Valori che può assumere 'key';
- Analisi delle caratteristiche stilistiche della musica: correlazioni tra queste variabili e la frequenza di apparizione nelle classifiche;
- Date determinate caratteristiche posso prevedere gli stream?
- Qual'è l'artista con la più altra probabilità di avere maggiori stream?
- Quali sono le canzoni che potrebbero trovarsi nella top 10?
  - Due soluzioni:
    1. Regressore Logistico;
    2. Naive Bayes;
  - Confronto accuracy delle due soluzioni trovate;

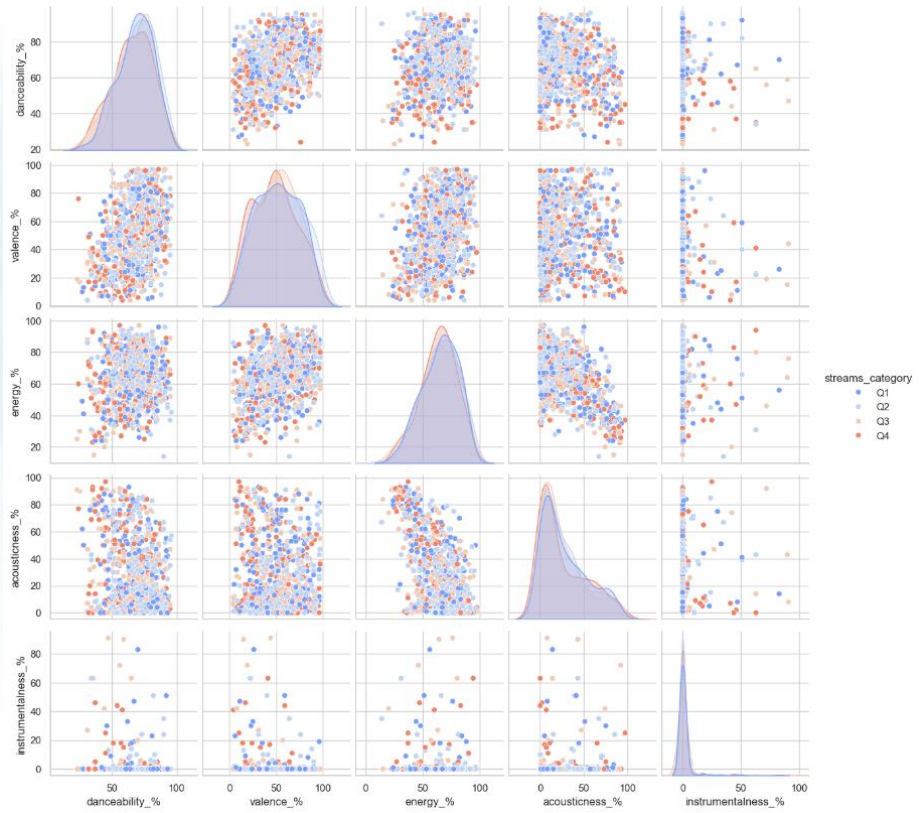
EFFETTUARE UN'ANALISI ESPLORATIVA SU TREND TEMPORALI

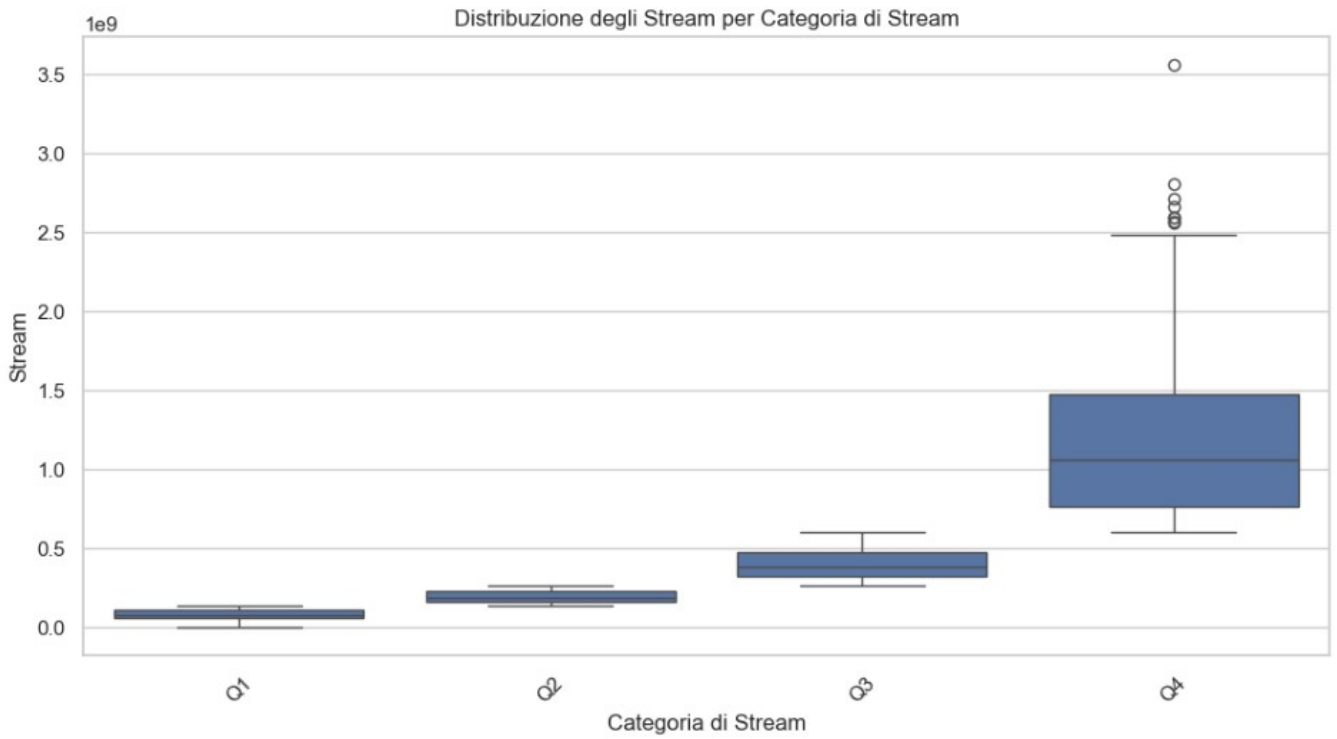




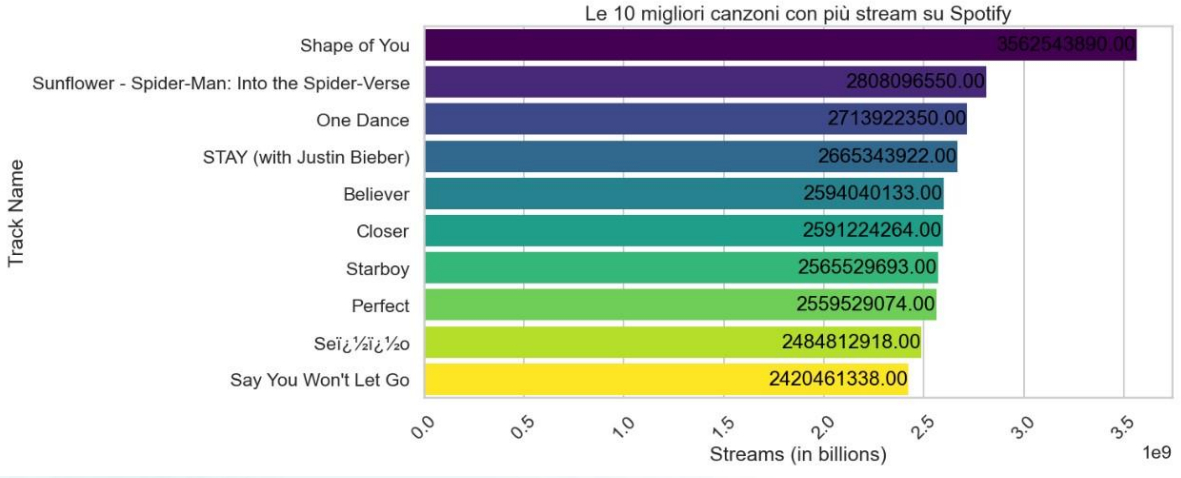
Effettuare un'analisi esplorativa sugli STREAM

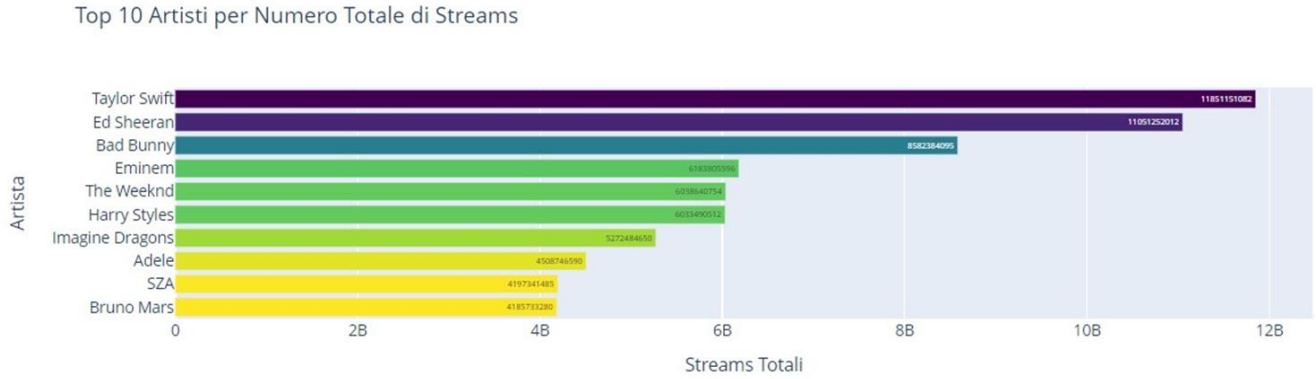






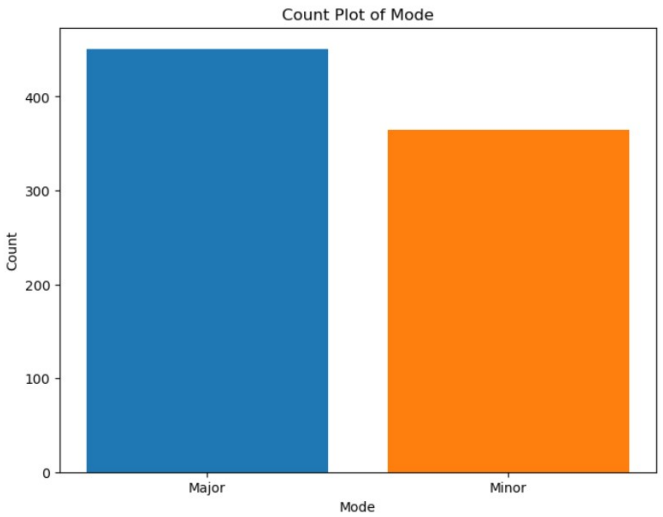
Effettuare un'analisi esplorativa sugli STREAM





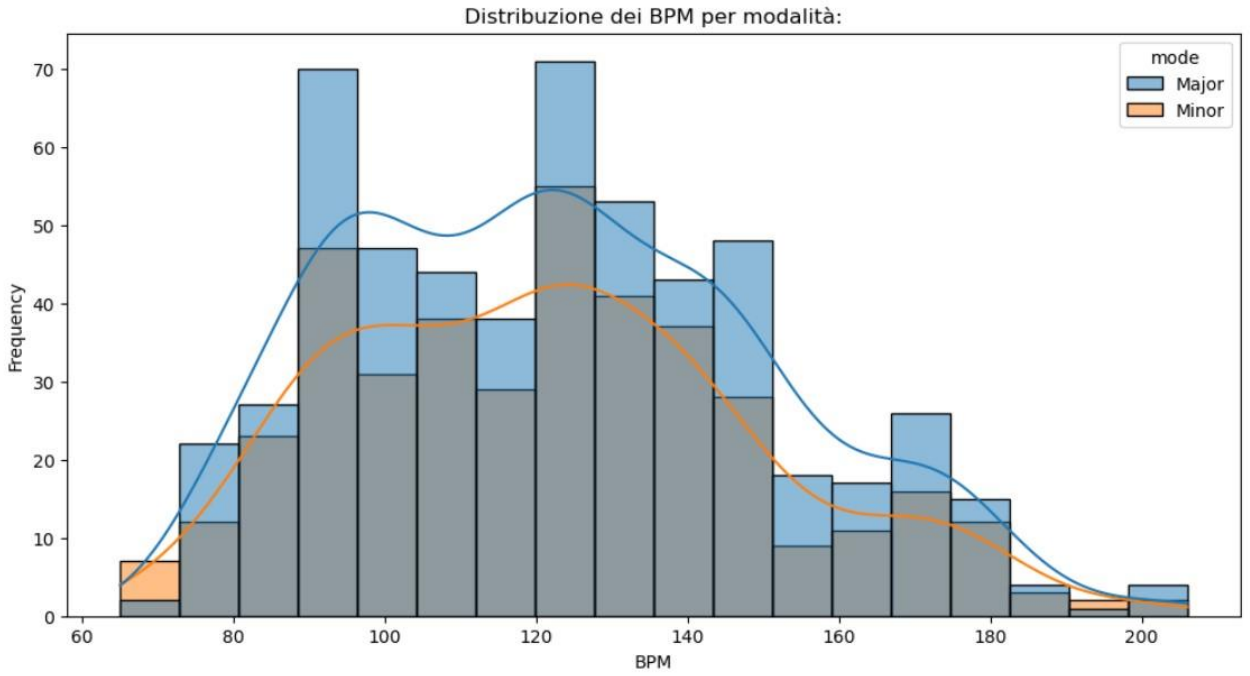

```
Calcolo delle statistiche descrittive per la colonna streams

Mean: 468985764.4080882
Median: 263836779.5
Variance: 2.7366159469724688e+17
Standard Deviation: 523126748.2142802
25th Percentile: 134284821.0
50th Percentile (Median): 263836779.5
75th Percentile: 601198591.25
```

Effettuare un’analisi esplorativa sui valori che può assumere ‘key’;



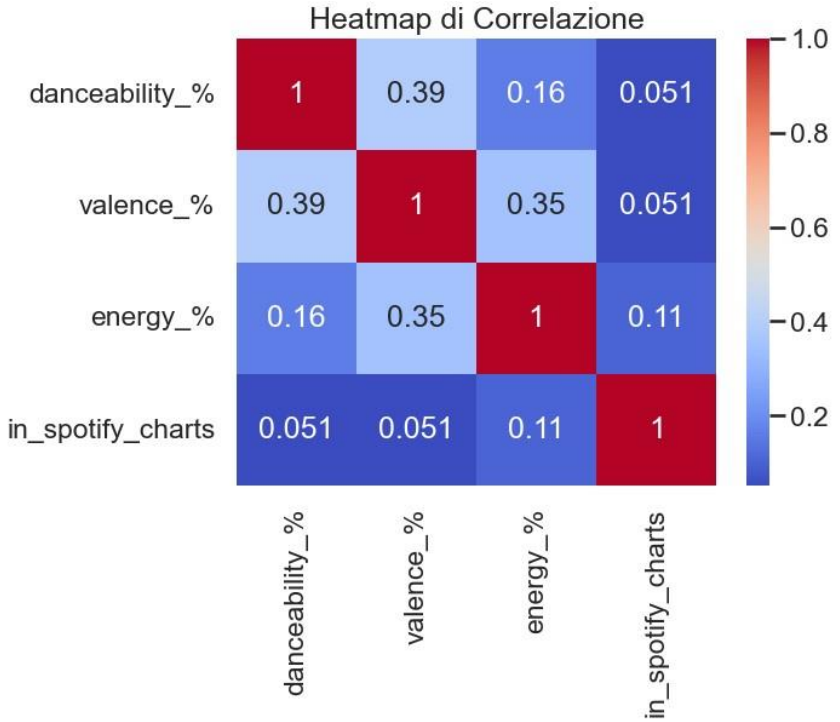
Distribuzione dei BPM per modalità:

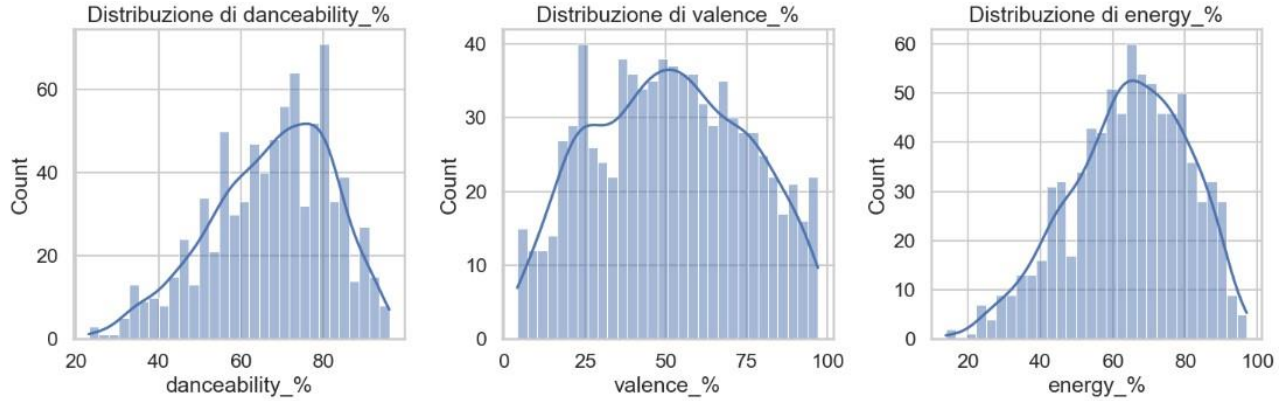
Il risultato che possiamo dedurre dal grafico è che esiste una distinzione nella distribuzione dei BPM tra le canzoni in modalità maggiore e quelle in modalità minore, con una tendenza a tempi leggermente più lenti per le canzoni in modalità minore. Questo può riflettere le tendenze stilistiche e emotive nella musica, dove le modalità minori sono spesso associate a sentimenti più riflessivi o malinconici, che possono essere espressi attraverso tempi più lenti.



Analisi delle caratteristiche di stile della musica. Correlazioni tra queste variabili e la presenza nelle classifiche

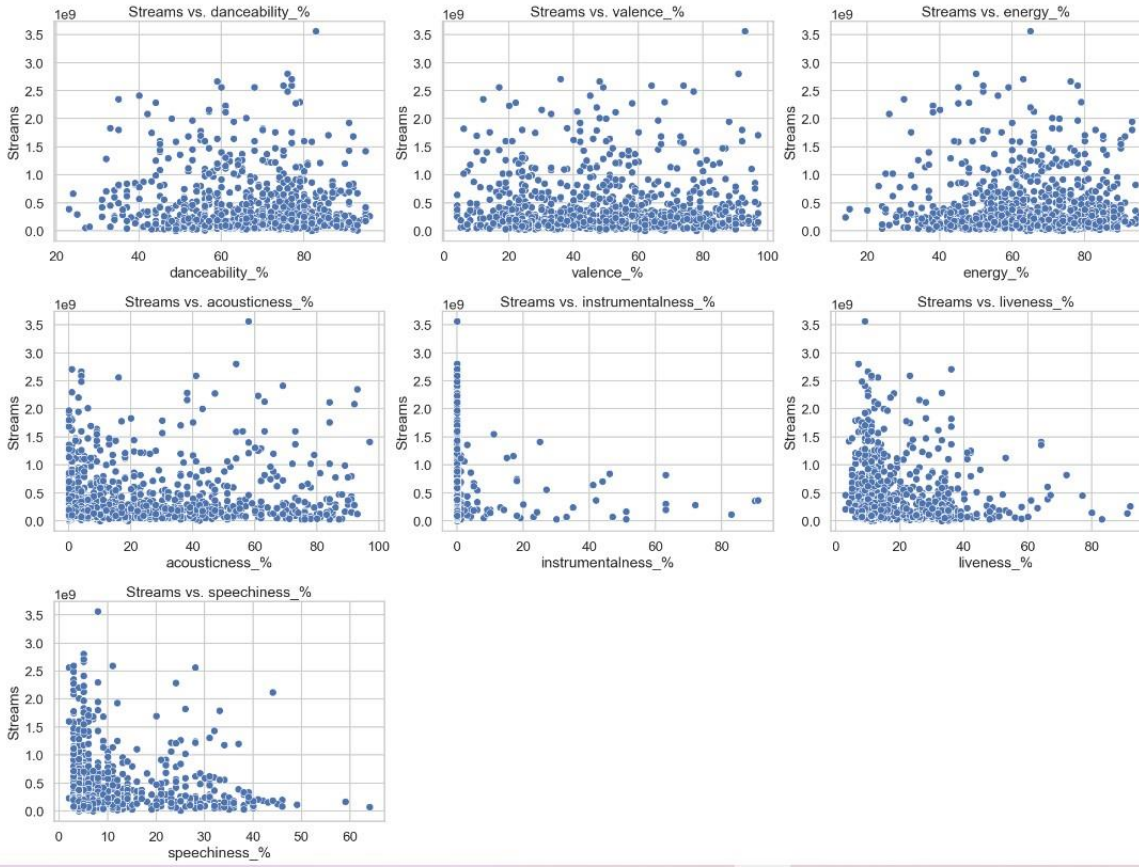
Questi grafici offrono un'analisi descrittiva delle caratteristiche delle tracce musicali e della loro relazione con la presenza nelle classifiche di Spotify.

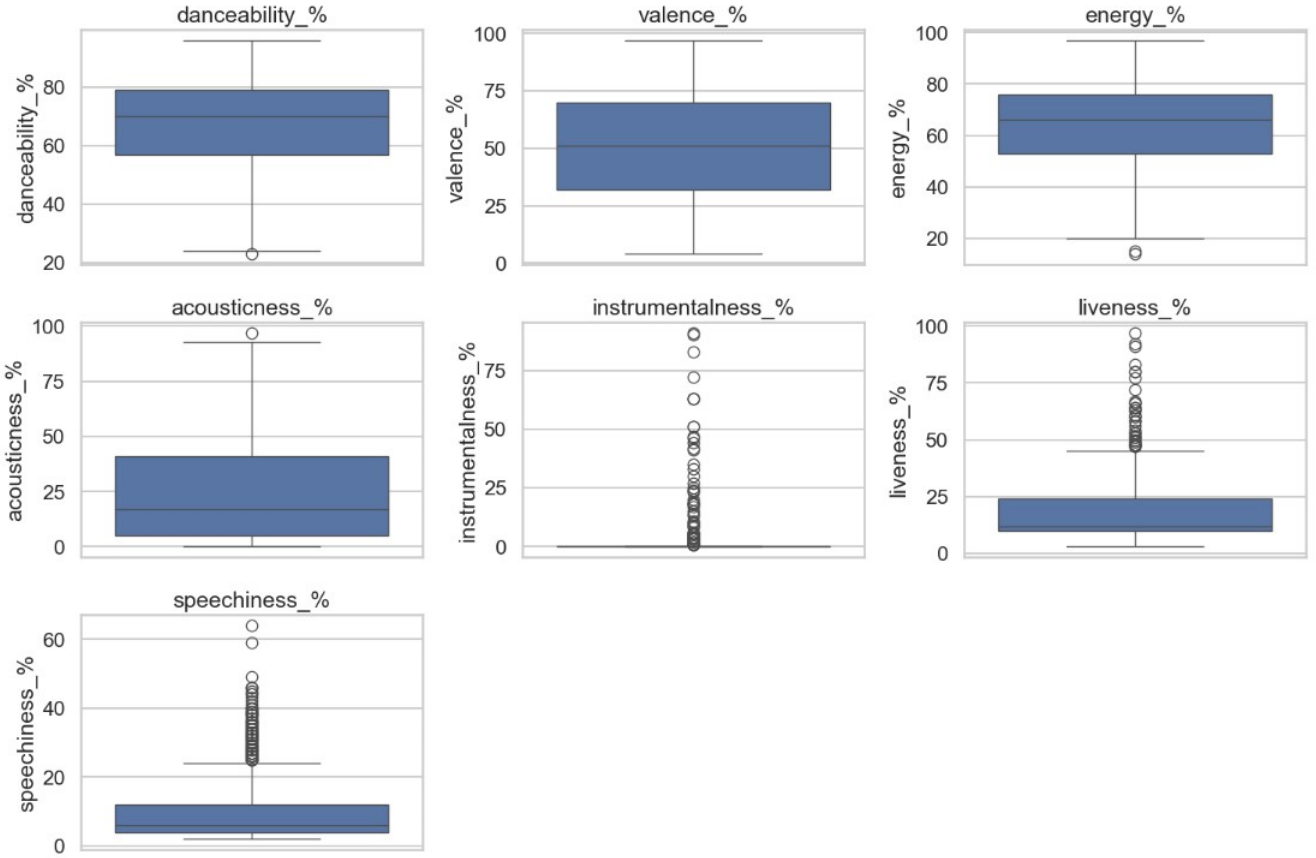




Date determinate caratteristiche posso prevedere gli stream?

Confrontiamo streams con le varie caratteristiche





Date determinate caratteristiche posso prevedere gli stream?

```
# Preparazione delle variabili indipendenti (X) e dipendente (y)
X = df_clean[['bpm','acousticness_%', 'instrumentalness_%', 'liveness_%', 'speechiness_%','danceability_%', 'valence_%', 'energy_%']]
y = df_clean['streams']
```

```
model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression
LinearRegression()
```

Ho ottenuto la soluzione attraverso l'utilizzo di un regressore lineare, cercando di prevedere gli stream dalle varie caratteristiche di una traccia.
Ho ottenuto un MSE che sembra alta però può essere relativamente basso perché gli stream si manifestano in milioni.

```
Errore quadratico medio (MSE): 2.969502977038016
```

Qual'è l'artista con la più altra probabilità di avere maggiori stream?

Calcolo la popolarità in base al numero di stream/numero di canzoni

```
# Calcola gli stream medi per traccia come proxy del successo per traccia
top_artists['popularity_score'] = top_artists['streams'] / top_artists['total_tracks']
```

Elimino tutto quello che so sull'artista per evitare di prevedere dati che il modello conosce già.
Successivamente vado ad utilizzare un regressore lineare.

```
X = top_artists.drop(['artist(s)_name', 'streams', 'popularity_score', 'total_tracks'], axis=1)
y = top_artists['popularity_score'].rank(pct=True)
```

Quali sono le canzoni che potrebbero trovarsi nella top 10?

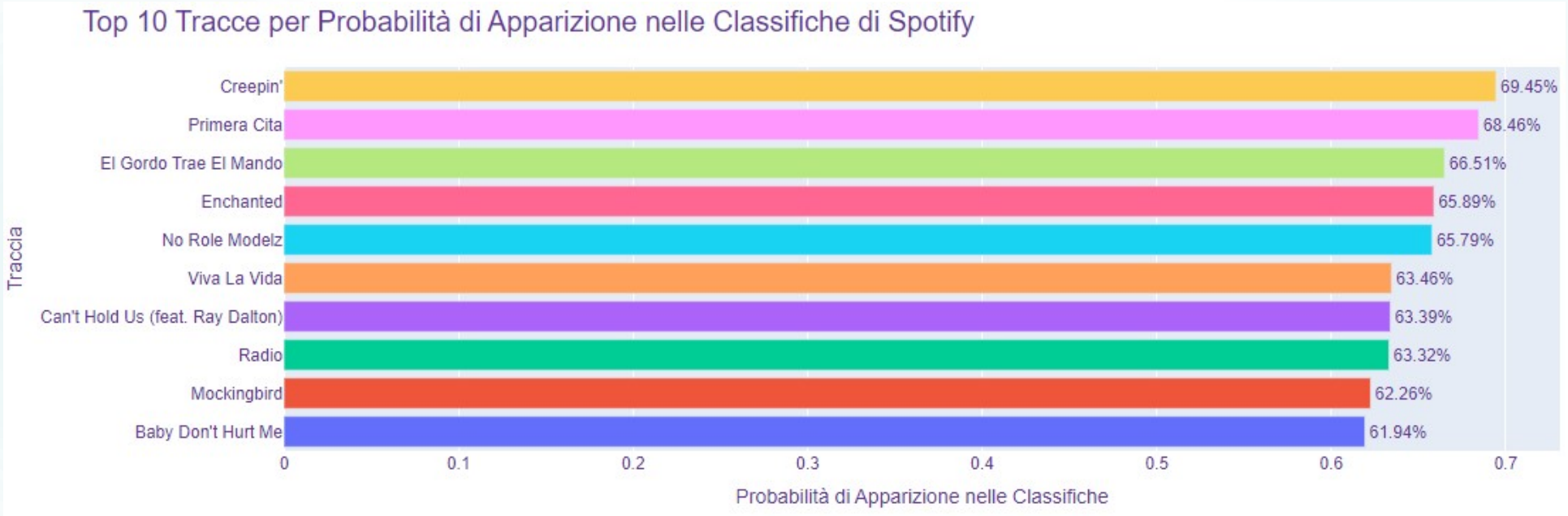
Fisso una soglia pin percentili (50°) per bilanciare bene i dati

```
X = df_clean[features]
y = (df_clean['in_spotify_playlists'] > soglia).astype(int)

y.value_counts()
```

```
in_spotify_playlists
0    408
1    408
Name: count, dtype: int64
```

```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

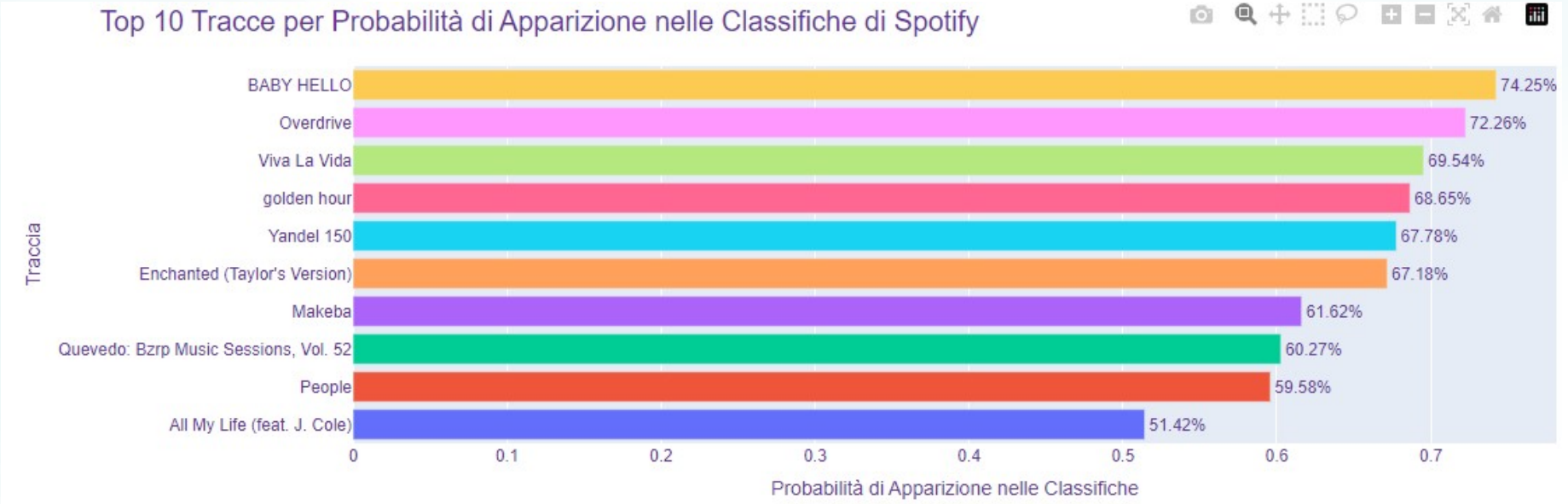
```
LogisticRegression
LogisticRegression(max_iter=1000)
```



Quali sono le canzoni che potrebbero trovarsi nella top 10?

```
soglia = np.percentile(df_clean['in_spotify_playlists'], 50)
y_binaria = (df_clean['in_spotify_playlists'] > soglia).astype(int)
```

```
model_nb = GaussianNB()
model_nb.fit(X_train, y_train)
```



Confronto tra due soluzioni trovate

REGRESSORE LOGISTICO

```
accuracy_opt = accuracy_score(y_test, y_pred_)
precision_opt = precision_score(y_test, y_pred_)
recall_opt = recall_score(y_test, y_pred_)
f1_opt = f1_score(y_test, y_pred_)

print("Ottimizzato - Accuracy:", accuracy_opt)
print("Ottimizzato - Precision:", precision_opt)
print("Ottimizzato - Recall:", recall_opt)
print("Ottimizzato - F1 Score:", f1_opt)
```

```
Ottimizzato - Accuracy: 0.8325646565655656
Ottimizzato - Precision: 0.6455696202531646
Ottimizzato - Recall: 0.5666666666666667
Ottimizzato - F1 Score: 0.6035502958579881
```

NAIVE BAYES

```
accuracy_nb = accuracy_score(y_test, y_pred_classes_nb)
precision_nb = precision_score(y_test, y_pred_classes_nb)
recall_nb = recall_score(y_test, y_pred_classes_nb)
f1_nb = f1_score(y_test, y_pred_classes_nb)

print("Naive Bayes - Accuracy:", accuracy_nb)
print("Naive Bayes - Precision:", precision_nb)
print("Naive Bayes - Recall:", recall_nb)
print("Naive Bayes - F1 Score:", f1_nb)
```

```
Naive Bayes - Accuracy: 0.6951219512195121
Naive Bayes - Precision: 0.36
Naive Bayes - Recall: 0.20930232558139536
Naive Bayes - F1 Score: 0.2647058823529412
```