

# Hit or flop?

## **NOMINATIVO + MATRICOLA COMPONENTI GRUPPO**

Calvano Miriana - 699645

Curci Antonio - 697498

Lomonte Nunzia – 697175

**LINK REPOSITORY:** [https://github.com/NunziaL/ICon20-21\\_Cal-Cur-Lom](https://github.com/NunziaL/ICon20-21_Cal-Cur-Lom)

Questo progetto si basa su **quattro** argomenti principali: le ontologie, i modelli supervisionati, il classificatore bayesiano e l'apprendimento non supervisionato con clustering.

## Istruzioni di esecuzione

### Comandi di esecuzione

E' possibile eseguire il programma attraverso un qualsiasi IDE che supporti Python, in particolare i membri del gruppo hanno utilizzato Spyder. Per lanciare il programma è sufficiente fare il run del file 'main.py'.

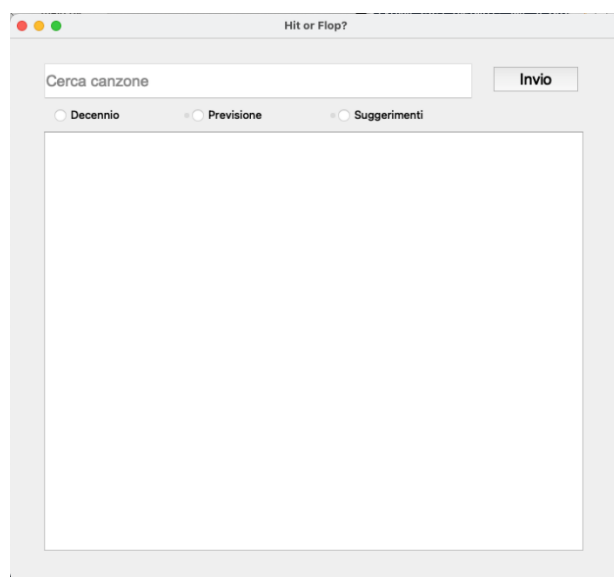
Se si vuole eseguire il file 'main.py' dal terminale, occorre scaricare la cartella del progetto da GitHub e successivamente digitare sul terminale i seguenti comandi:

1. `cd percorso_cartella/ICon20-21_Cal-Cur-Lom`
2. `python main.py`

**NB:** Le librerie utilizzate nell'implementazione sono le seguenti: pandas, PyQt5, scikit, sklearn, joblib, spotipy, Math, os, warnings, sys.

Se le librerie non sono installate sulla propria macchina, prima di eseguire il programma occorre installarle tramite il seguente comando: `pip install nome_libreria`.

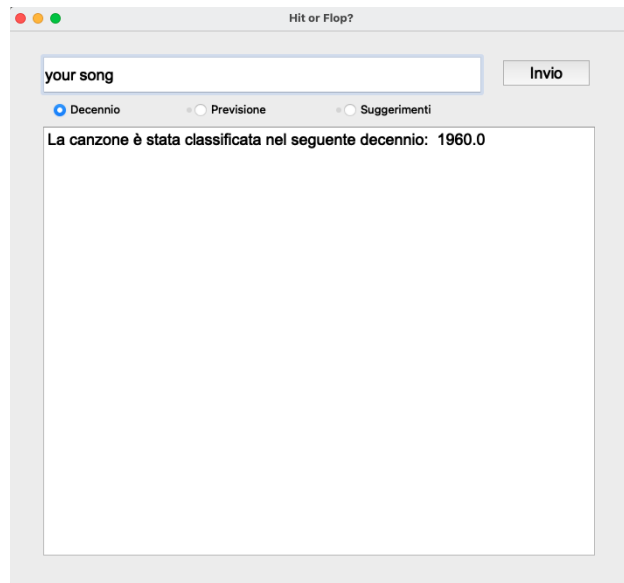
Dopo aver lanciato il programma, si aprirà la seguente schermata:



L'applicazione fornisce **3 funzionalità** principali, ognuna delle quali include degli argomenti diversi del programma:

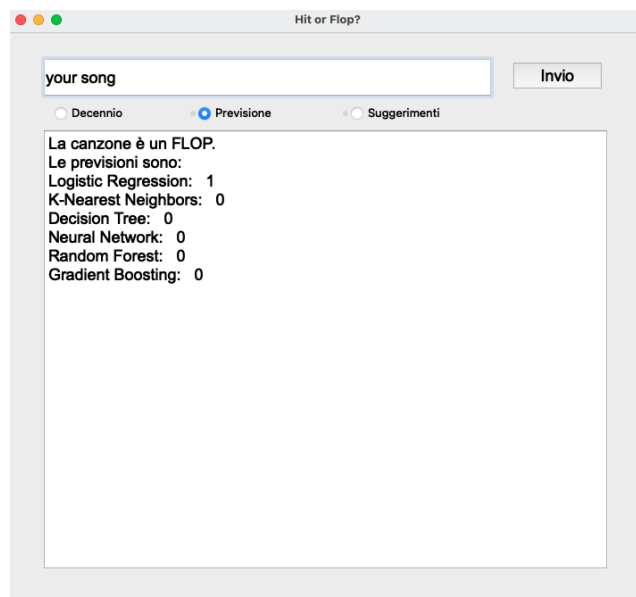
Per poter garantire un'esecuzione corretta, è necessario inserire il nome di una canzone nella barra di ricerca, selezionare un'opzione tra le tre indicate e, infine, premere "Invio":

1. Selezionare "Decennio" per **predire** il **decennio** della canzone inserita dall'utente



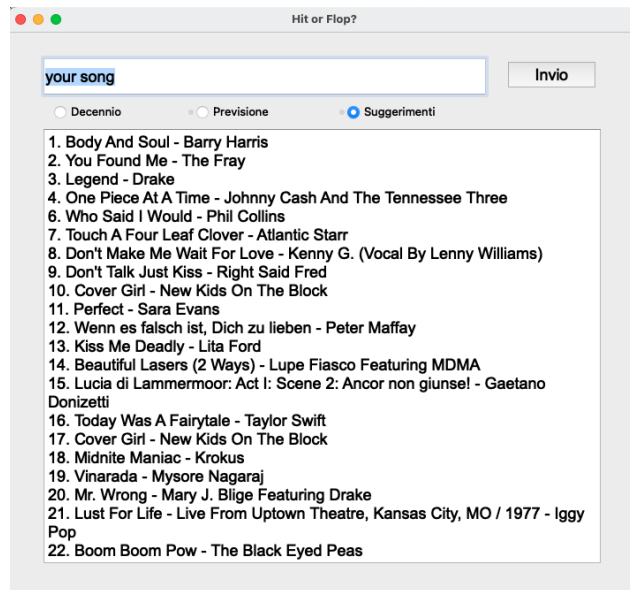
The screenshot shows a web application window titled "Hit or Flop?". It features a text input field labeled "your song" and a button labeled "Invio". Below the input field, there are three radio button options: "Decennio" (selected), "Previsione", and "Suggerimenti". The output area below the options displays the text: "La canzone è stata classificata nel seguente decennio: 1960.0".

2. Selezionare "Previsione" per **predire** se la canzone inserita dall'utente è stata un **flop (0)** oppure una **hit (1)**:

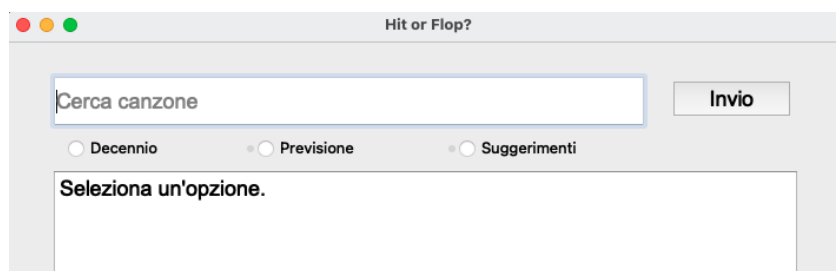


The screenshot shows the same web application window. The "Previsione" radio button is now selected. The output area displays the following text: "La canzone è un FLOP. Le previsioni sono: Logistic Regression: 1 K-Nearest Neighbors: 0 Decision Tree: 0 Neural Network: 0 Random Forest: 0 Gradient Boosting: 0".

3. Selezionare "Suggerimenti" per fornire all'utente dei **suggerimenti** con delle **canzoni simili** a quella presa in esame



**NB**: Se viene premuto il pulsante “Invio”, ma non viene selezionata nessuna opzione o se la barra di ricerca è vuota, verrà visualizzato il seguente messaggio:



## Informazioni su implementazione

Il dataset utilizzato è stato preso da <https://www.kaggle.com/theoverman/the-spotify-hit-predictor-dataset>; consiste in 6 file con estensione .csv in cui sono conservate le informazioni riguardanti le canzoni presenti nel database di Spotify. Ogni canzone presenta i seguenti attributi: *track*, *artist*, *uri*, *danceability*, *energy*, *key*, *loudness*, *mode*, *speechiness*, *acousticness*, *instrumentalness*, *liveness*, *valence*, *tempo*, *duration\_ms*, *time\_signature*, *chorus\_hit*, *sections*, *target*.

In totale, sommando tutte le entrate per ogni file, le canzoni prese in esame sono circa 41000.

## Acquisizione del dataset

Il dataset viene acquisito dai file .csv. Durante questa fase, viene aggiunto un attributo “*decade*”, che indica il decennio di appartenenza delle canzoni in base al nome del file in cui esse si trovano (ES: “*Misty Roses, Astrud Gilberto*” avrà in corrispondenza dell’attributo “*decade*” il valore 1990 perché si trova nel file dataset-of-90’s).

## Identificazione canzone tramite URI

Successivamente, viene chiesto all’utente di fornire in input una stringa che dovrà contenere almeno il titolo della canzone che si vuole analizzare (ES: “*your song elton john*” oppure “*your song*”), simulando una semplice ricerca che si farebbe nell’applicazione di Spotify poiché sono state sfruttate le API dell’applicazione attraverso il modulo *spotipy*.

È stato possibile risalire all'**URI** (*Uniform Resource Identifier*) per poter identificare il brano in maniera univoca. Grazie alla funzione `spotipy.audio_features()` è stato possibile ottenere le features della canzone (danceability, energy, key, ecc...).

### Classificatore bayesiano

Nella fase successiva, attraverso il classificatore bayesiano, viene eseguita la previsione della decade di appartenenza della canzone inserita in input dall'utente, in quanto questa informazione non è presente nelle features fornite da Spotify.

Il **classificatore bayesiano** è basato sull'applicazione del teorema di Bayes e richiede la conoscenza delle probabilità condizionali relative al problema al fine di assegnare la classe di appartenenza all'oggetto preso in considerazione.

### Apprendimento supervisionato

Nel modulo "*supervised.py*" sono state definite le funzioni necessarie per la fase di predizione. Durante questa fase dell'esecuzione vengono rimossi dal dataset gli attributi "track, artist e uri" in quanto non contengono informazioni utili all'apprendimento del mondo.

Successivamente, il dataset viene diviso in due ulteriori dataframe, X e y, che, rispettivamente rappresentano un insieme di training e uno di test. Questi due insiemi saranno poi "splittati" in modo tale che le feature di apprendimento siano separate da quelle di target.

Dunque l'insieme y conterrà solo la feature "target", mentre X tutte le altre. Affinché venga costruito un modello sulla base di dati confrontabili e ben pesati, viene svolta la *standardizzazione* dei valori.

Alla conclusione di questo step, vengono definiti i 6 modelli che verranno sfruttati per effettuare le predizioni: Regressione logistica, kNN, Alberi di decisione, Rete Neurale, Random Forest e Gradient Boosting.

- **Regressione logistica:** è un modello di regressione non lineare in cui vengono determinati i pesi di una funzione lineare appiattita dal sigmoide, minimizzando un errore sull'insieme di esempi.  
Il modello viene utilizzato per classificare le osservazioni in base alle proprie caratteristiche;
- **kNN:** Il classificatore kNN è usato per classificare gli oggetti basandosi sulle caratteristiche di quelli vicini a quello considerato.  
Il tutto viene implementato attraverso l'algoritmo *k-nearest neighbors* (k-NN) il cui input è costituito dagli esempi di addestramento, mentre l'output è l'appartenenza a una classe.
- **Alberi di decisione:** L'albero di decisione è un modello predittivo costituito da nodi interni, archi e nodi foglia. Esso viene costruito utilizzando tecniche di apprendimento a partire dall'insieme dei dati iniziali (data set), il quale viene diviso in due sottoinsiemi: il *training set* sulla base del quale si crea la struttura dell'albero e il *test set* che viene utilizzato per testare l'accuratezza del modello predittivo così creato.
- **Reti Neurali:** Sono state usate le reti neurali di tipo *MLPC*, quindi un classificatore a più strati che allena il modello secondo la back-propagation (*MLPC*) tramite la funzione `MLPClassifier()`.
- **Random Forest:** Classifica i dati in base all'addestramento fatto su più alberi in cui vengono fatte le varie predizioni che poi saranno aggregate.  
Le feature su cui vengono fatte le predizioni sono permutate a random.
- **Gradient Boosting:** Con questo algoritmo di classificazione viene restituita una predizione che viene migliorata ad ogni iterata, in modo tale da poter minimizzarne l'errore.

A questo punto, ci sarà la possibilità di poter predire la classe tramite dei modelli allenati precedentemente salvati su file .sav oppure allenarli nuovamente in caso in cui questi file non dovessero essere presenti nel percorso del progetto. Verrà dunque definita la classe predetta per ogni modello.

### **Apprendimento non supervisionato: clustering**

Infine, il sistema restituisce all'utente una lista di canzoni simili a quella inserita.

I risultati sono elaborati durante una fase di **clustering** basata sull'algoritmo di **KMeans**, in cui sono calcolati 4000 cluster.

Si tratta di apprendimento *non supervisionato* in cui, attraverso la definizione di un centroide, i dati sono divisi in gruppi che hanno caratteristiche simili.

Grazie a questa fase è possibile risalire al cluster di appartenenza della canzone inserita per restituire all'utente una lista di canzoni nel dataset che appartengono allo stesso cluster.