Il successo di una canzone: un modello per scoprirlo

Shady Abd El Kader Abdel Mohssen Hossin, Stefano Aparo, Massimiliano Calani, Matteo Gaverini, Matteo Mazzola

Sommario
La scelta dell'ascolto di una canzone è influenzata da molti fattori, spesso personali, che variano da utente a utente; un indice super partes che influenza l'ascolto è il suo grado di successo: più un brano è ascoltato e trasmesso, maggiori sono le probabilità che diventi una hit. Cercare di prevedere la popolarità di una canzone può risultare importante sia per le piattaforme di streaming, sia per le etichette discografiche e sia per le emittenti radiofoniche, che in questo modo potrebbero compiere una selezione più mirata dei brani musicali da trasmettere. Il progetto nasce dalla volontà di prevedere il successo di una canzone in base ad alcune sue caratteristiche oggettive tra cui, ad esempio, il livello di acustica e l'energia che trasmette. Per far ciò si è addestrato un certo numero di classificatori per ogni categoria su un campione di dati; i modelli migliori (uno per ogni categoria) sono stati infine addestrati su un ulteriore campione per decretare il classificatore ottimale. I modelli sono stati validati su una collezione di dati storici relativi alle canzoni distribuite su Spotify a novembre 2018.

Indice

Introduzione

Spotify è un servizio musicale che offre lo streaming on demand di brani di varie case discografiche ed etichette indipendenti. Fondato nel 2008 in Svezia, il servizio ha oltre 75 milioni di utenti al giugno 2015 [1]. La sua sempre più crescente fama e la sua capacità di imporsi nel panorama musicale hanno permesso a questa piattaforma di diventare uno standard riconosciuto nella comunità dei musicisti, risultando oggi un attore principale nella dichiarazione dei dischi d'oro e nella certificazione FIMI. Spotify non solo ha rivoluzionato la fruibilità dei brani musicali, ma ha soprattutto cambiato il modo in cui gli artisti guadagnano dalla proprie canzoni. Un artista non riceve una quota fissa per ogni singolo ascolto ma viene pagato in relazione alla percentuale di stream rispetto al totale [2]. E' invece diverso il discorso riferito alle radio; considerando lo scenario italiano delle emittenti radiofoniche nazionali, esse devono fornire alla SIAE (Società Italiana degli Autori ed Editori) dei report indicando le canzoni passate in radio e, in base a dei coefficienti, vengono definiti i contributi dovuti; dopodichè i compensi che la SIAE incassa vengono distribuiti in proporzione alla durata del brano ed altri criteri alle case discografiche, artisti ed editori. Questo costo delle radio, però, potrebbe essere ammortizzato qualora si riuscisse a prevedere la popolarità di una canzone. Se esistesse un modello o un algoritmo che in base ad alcune caratteristiche oggettive di una traccia riuscisse con una certa probabilità a prevedere il suo successo o meno, le radio potrebbero trasmettere solo i brani che secondo l'algoritmo risultano delle hit. In questo modo le emittenti radiofoniche aumenterebbero gli ascolti e ridurrebbero i costi dovuti alla SIAE. Un'analisi preliminare basata sugli attributi descrittivi di ogni canzone potrebbe portare quindi vantaggi sia in termini di risparmio di tempo nella ricerca delle canzoni con un più appeal sul pubbli-

co, sia in termini di marketing, in quanto si indirizzerebbero più efficientemente le risorse verso un probabile successo. Lo scopo dell'analisi del dataset è quello di capire se è possibile appurare il successo o meno di una canzone dati i suoi attributi strutturali.

Il report è strutturato in 4 capitoli:

1. *Capitolo 1*: presenta il dataset e le operazioni di preprocessing effettuate
2. *Capitolo 2*: espone i modelli, le misure e i metodi di valutazione usati
3. *Capitolo 3*: presenta i risultati ottenuti
4. *Capitolo 4*: espone le conclusioni e sviluppi futuri

1. Data Exploration

1.1 Dataset

Il dataset utilizzato è composto da 116372 istanze e 17 attributi, che riportano una complessiva descrizione strutturale delle canzoni caricate su Spotify a novembre 2018. Il dataset comprende sia attributi numerici che attributi categoriali, nello specifico i quantitativi rappresentano le caratteristiche oggettive di una canzone mentre i qualitativi identificano una traccia musicale con id, titolo e nome artista. Di seguito sono riportati in dettaglio gli attributi del dataset:

1. *artist_name*: nome dell'artista
2. *track_id*: id univoco che identifica la traccia audio
3. *track_name*: titolo della traccia audio
4. *acousticness*: identifica da 0.0 a 1.0 la confidenza sul fatto che la traccia sia acustica
5. *danceability*: identifica da 0.0 a 1.0 la ballabilità della traccia
6. *duration*: durata della traccia in ms
7. *energy*: identifica da 0.0 a 1.0 l'energia della traccia
8. *instrumentalness*: identifica da 0.0 a 1.0 la confidenza che la traccia non contenga suoni vocali
9. *key*: stima della chiave musicale usata nella traccia
10. *liveness*: identifica da 0.0 a 1.0 la probabilità della presenza di un'audience al momento della registrazione, quindi che la traccia provenga da un concerto
11. *loudness*: identifica il volume di una traccia in dB
12. *mode*: identifica con un numero intero la modalità della traccia e il tipo di scala melodica derivata
13. *speechiness*: identifica da 0.0 a 1.0 la presenza di parole all'interno della traccia
14. *tempo*: tempo di una canzone misurato in BPM
15. *time_signature*: identifica con un numero intero la stima del numero di beats per misurazione
16. *valence*: identifica da 0.0 a 1.0 la positività di una traccia: un alto grado di valenza è associato ad emozioni positive (felicità) mentre un basso valore è associato ad emozioni negative (tristezza)
17. *popularity*: identifica da 0 a 100 la popolarità di una traccia

Per rispondere alla domanda di ricerca si è aggiunto un ulteriore attributo categorico binario nominato *level_popularity*. L'assegnamento dei due valori [y/n], che è avvenuto in modo arbitrario, è stato definito in questo modo: la categoria *n* è stata assegnata a tutte le canzoni che hanno un valore di *popularity* < 50, mentre la categoria *y* è stata assegnata a tutti gli altri casi, ovvero quando *popularity* ≥ 50; di conseguenza con *n* si identificano tutte le canzoni che non sono considerate affermate, mentre con *y* si identificano quelle che sono considerate un successo. Tra i due valori l'analisi si concentra principalmente sulla classe meno rappresentata del dataset, ma più importante, ovvero *y*.

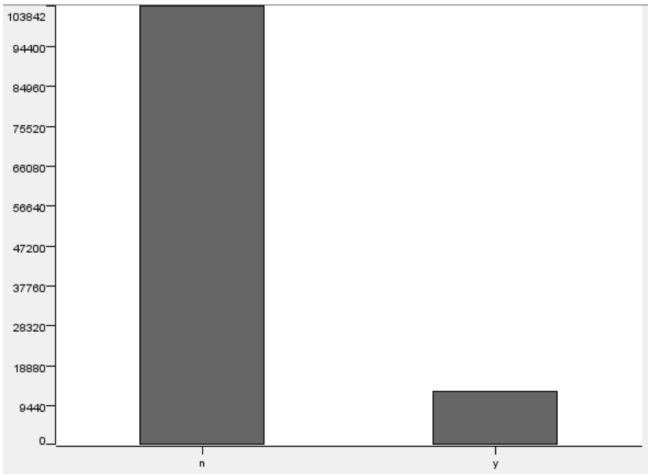

Figura 1. Distribuzione della classe *level_popularity*

Dal barchart si evince come il dataset si presenti molto sbilanciato; il 90% dei dati è infatti rappresentato dalla categoria *n*, mentre il restante 10% è rappresentato dalla categoria *y*: questa scomposizione è una rappresentazione reale dell'universo musicale, in quanto le canzoni famose, quelle che hanno un maggiore appeal sul pubblico, sono solo una piccola parte dell'insieme di tutte le canzoni prodotte. Poiché le proporzioni sono molto distanti tra di loro, per limitare problemi di overfitting dei modelli, si è considerato un sottoinsieme di dati, nello specifico si sono tenute in considerazione le canzoni che hanno una valore di *popularity* > 36, ovvero i brani a partire dal 75° percentile (Figura 2).

La differenza delle proporzioni delle due categorie del dataset partizionato rimangono così differenti, ma risultano ridotte (rispettivamente 42% y e 58% n).

1.2 Preprocessing

Il dataset si presenta integro e non necessita di operazioni di missing replacement. Dei 17 attributi, 4 sono stati rimossi a priori per due motivi: *track_id*, *track_name* e *artist_name* perchè non rilevanti per il problema di classificazione, mentre *popularity* perchè utilizzato per la definizione dell'attributo di classe e quindi altamente predittivo. Le operazioni di preprocessing effettuate sono state feature selection e sampling.

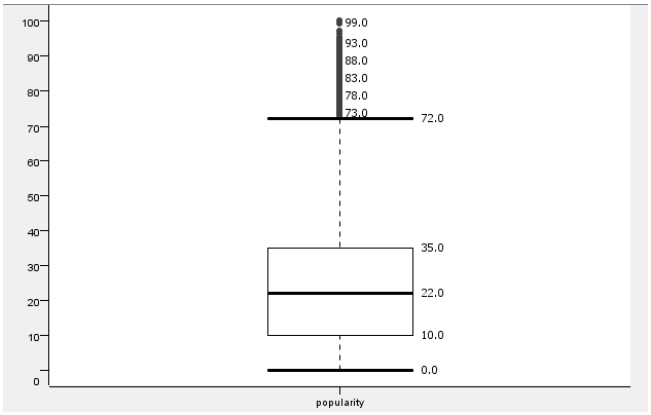

Figura 2. Distribuzione dell'attributo *popularity* nel dataset

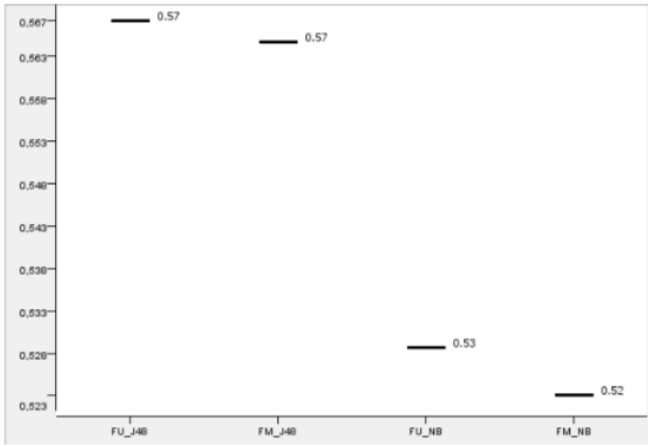

Figura 3. Accuratezza dei classificatori utilizzati nel filter (FU: Filtro Univariato, FM: Filtro Multivariato)

1.2.1 Feature Selection

Dato il numero di attributi a disposizione, è stato necessario stabilire quali tra questi potessero essere i più significativi e non ridondanti. A questo scopo è stata effettuata una feature selection, optando per il metodo filter sia univariato che multivariato:

- Filter Univariato:
  - Search: Ranker
  - Evaluator: InfoGainAttributeEval
  - Classificatori:
    - J48
    - NayveBayes
- Filter Multivariato:
  - Search: BestFirst
  - Evaluator: CfsSubsetEval
  - Classificatori:
    - J48
    - NayveBayes

Il modello con il migliore risultato in termini di accuratezza è stato **J48** associato al **filter univariato** (Figura 3), che ha emesso come attributi rilevanti *acousticness*, *danceability*, *duration_ms*, *instrumentalness*, *loudness* e *valence*. L'utilizzo del filtro non solo ha portato ad una riduzione dei tempi di elaborazione, ma ha anche tamponato il problema della course of dimensionality, che definisce come all'aumentare degli attributi in fase di classificazione, aumenta anche la sparsità implicita del dataset.

1.2.2 Sampling

Dopo aver eseguito la feature selection ed aver individuato gli attributi rilevanti, si è effettuato il sampling. Il dataset è stato partizionato in 2 subset (50% il primo, 50% il secondo) utilizzando la tecnica del campionamento stratificato (l'attributo considerato per eseguire questa tipologia di sampling è stato *level_popularity*). I campioni sono stati utilizzati in due fasi differenti; il primo è stato impiegato per individuare quale fosse il migliore classificatore per ogni categoria mentre il secondo è servito per decretare, tra tutti i classificatori migliori, quale fosse quello che consentiva di rispondere alla domanda di ricerca in maniera più efficace. I campioni utilizzati per addestrare i modelli sono stati a loro volta partizionati in due parti secondo le classiche proporzioni $\frac{2}{3}$ $\frac{1}{3}$, nello specifico il 67% per il training set e il restante 33% per il test set.

2. Modelli

2.1 Modelli utilizzati

Nel progetto sono state applicate diverse tecniche di classificazione per individuare quale fosse il classificatore migliore per rispondere alla domanda di ricerca. Nello specifico sono state usate 4 categorie differenti:

- **Modelli euristici**: si basano su classificazioni semplici e permettono di computare dataset molto ampi; utilizzano una base dati in training per imparare i pattern di classificazione e una di testing per misurare l'accuratezza della generalizzazione. Tra i classificatori presenti in questa categoria sono stati utilizzati *J48*, *Random forest* e il *Decision Tree*.
- **Modelli probabilistici**: si basano sull'ipotesi di indipendenza dei dati e sul teorema di Bayes; il modello assegna la classe che massimizza la probabilità a posteriori ad un record descritto dal suo vettore di funzionalità [3]. I classificatori utilizzati sono *Naive Bayes* e *Bayesian Network*.
- **Modelli di regressione**: basati sulla regressione logistica, il loro vantaggio è rappresentato dalla possibilità di considerare qualsiasi tipo di input, risultando così estremamente flessibili. I classificatori utilizzati sono *SimpleLogistic* e *Logistic*.
- **Modelli di separazione**: ne esistono di due tipi, SVM e Reti neurali. Il primo è un algoritmo supervisionato di machine learning che viene usato per classificare i dati. L'obiettivo di SVM è individuare un iperpiano

che divida al meglio i dati delle varie classi [4]. Le reti neurali invece sono un insieme di unità di elaborazioni semplici (neuroni) che comunicano inviando segnali tra loro attraverso delle connessioni pesate [5]. Per quanto riguarda SVM si è reso necessario utilizzare delle funzioni kernel in quanto i dati erano non separabili linearmente; le funzioni utilizzate sono *poly* e *rbf*. Per le reti neurali invece si sono utilizzati due modelli: *MultiLayerPerceptron* e *RProp MLP*; è importante evidenziare che per le reti neurali si è resa necessaria la normalizzazione dei dati prima di addestrare il modello per evitare la saturazione della rete.

2.2 Misure di valutazione

Per quanto riguarda le misure di valutazione si sono utilizzate F_1 measure, precision e recall; non si è considerata l'accuracy in quanto il problema di classificazione risulta essere sbilanciato e quindi quella misura di performance non fornirebbe utili indicazioni sulla puntualità della previsione della classe meno rappresentata (Rare class).

Precision
La precision (p) esprime il grado di fiducia che si ripone nel modello che classifica un record come positivo. Matematicamente è espressa come:

$$p = \frac{TP}{TP+FP}$$

dove TP indica i record positivi che il classificatore ha previsto correttamente mentre FP sono le istanze che il classificatore ha risposto come positive ma che in realtà sono negative.

Recall
La misura di recall (r), chiamata anche sensitivity, riporta in percentuale la frazione di positivi che sono stati effettivamente individuati rispetto al numero totale. Matematicamente è espressa come:

$$r = \frac{TP}{TP+FN}$$

dove TP indica i record positivi classificati correttamente mentre FN sono le istanze che il classificatore ha risposto come negative ma che in realtà sono positive.

F_1 Measure
La F_1 measure (F_1) rappresenta la media armonica delle due misure di performance citate precedentemente; assume valori compresi tra [0,1]. Questa misura si rivela fondamentale quando si mettono sullo stesso livello di importanza sia la precision che la recall; se il valore di F_1 measure è alto allora vuol dire che sia la recall che la precision assumono dei buoni valori. Matematicamente la F_1 measure è espressa come:

$$F_1 = \frac{2rp}{r+p}$$

Oltre a queste tre misure si è utilizzato anche un metodo di visualizzazione denominato **ROC curve** (Receiver Operating Characteristic curve). Essa è una curva rappresentata su un piano cartesiano dove l'ascissa identifica la percentuale dei falsi positivi (FPR), mentre l'ordinata identifica la percentuale dei veri positivi (TPR).

2.3 Metodi di valutazione

Per valutare le performance dei modelli sono stati utilizzati due metodi di valutazione per entrambi i campioni realizzati, ovvero *Holdout* e *Cross Validation*.

Holdout
Holdout è un metodo che consiste nel partizionare il dataset in due subset mutuamente esclusivi chiamati training set e test set [6]. Nel progetto si è deciso di realizzare un partizionamento stratificato utilizzando le classiche proporzioni ($\frac{2}{3}$ per il training set e $\frac{1}{3}$ per il test set). Le performance ottenute con holdout dipendono fortemente dalla scelta test set: il valore puntuale è quindi poco robusto.

Cross validation
La cross validation è un metodo che consiste nel partizionare il dataset D in k sottoinsiemi $D_1, D_2, ...D_K$ della stessa dimensione (chiamati fogli) [6]. L'inducer viene addestrato e testato $_K$ volte; ad ogni t iterazione $t \in \{_1, .., _K\}$ il classificatore viene addestrato su un training set $D_t = \{D_1, ..., D_{t-1}, D_{t+1}, ..., D_K\}$ e testato su un test set $D_{ts} = D_k$. Il valore della misura di performance considerata è uguale alla media delle misurazioni ottenute ad ogni iterazione. Nel progetto si è realizzata una cross validation con $_K$ che assume valore 10. Questo metodo migliora il valore puntuale che si ottiene da holdout perchè vengono effettuati molti test e assicura soprattutto che ogni osservazione sia inclusa una volta nel test set e un numero volte pari al numero di fogli nel training set. Per questo motivo il risultato di una cross validation viene preferito a quello di holdout.

3. Valutazione

3.1 Analisi dei risultati

Trovato l'indicatore di performance più adatto per decretare quale sia il migliore classificatore, si è passati al confronto vero e proprio dei modelli usati per ogni categoria. Per decretare il più preciso per il problema di classificazione si decide di utilizzare l'indicatore di performance F_1 measure calcolato sia su una procedura di holdout che su una cross validation.
Dai risultati del confronto emerge che per i modelli euristici il migliore è il *Decision Tree*, per i modelli logistici il *Logistic* , per i modelli probabilistici il *Naive Bayes* mentre per i modelli di separazione è il *Prop MLP*.
Si riscontra poi che la F_1 measure calcolata su SVM assume valore 0.0; questo perchè SVM etichetta tutti i record come negativi sia nell' Holdout che nella Cross validation e di conseguenza i valori TP e FP risultano essere 0.

3.2 Analisi delle performance dei migliori classificatori per categoria

Una volta individuati i migliori classificatori per ogni categoria si è effettuato il confronto di ognuno di essi. Questa operazione è avvenuta considerando tre diversi aspetti:

- Precision, Recall e F_1 measure
- ROC curve
- intervalli di confidenza degli errori tra due classificatori

Confrontando le misure delle performance dei classificatori (Figura 4) si possono effettuare una serie di riflessioni: si nota come non esista un classificatore migliore se si considerano contemporaneamente tutte e tre le misure di performance ed emerge che il Logistic assume valori più bassi sia di recall che di F_1 measure mentre assume un valore di precision più alto. Non è quindi possibile decretare a priori quale sia il classificatore ottimale, in quanto dipende dalla misura che si prende in considerazione.

Row ID	precision	recall	f1
DT	0.476	0.481	0.478
LOGISTIC	0.514	0.242	0.328
PROPMLP	0.502	0.45	0.475
NB	0.491	0.707	0.579

Figura 4. Misure delle performance dei migliori classificatori per ogni categoria

La ROC curve (Figura 5) mostra come tutti i modelli migliori, per quanto riguarda il rapporto tra recall e FPR, offrono prestazioni discrete in quanto le aree sottese a esse non sono sensibilmente maggiori di quella sottesa dal modello casuale (rappresentata dalla linea retta grigia). In particolare il *Decision Tree* risulta essere il modello peggiore, mentre il *Naive Bayes* risulta essere il classificatore migliore fino a una percentuale di FPR del 35%, mentre per percentuali maggiori del 35% di FPR il modello Prop MLP risulta essere il migliore.

La verifica del miglior classificatore è avvenuta utilizzando l'intervallo di confidenza degli errori tra i due modelli migliori, ovvero *NaiveBayes* e *Prop MLP*, utilizzando una T-Student con 9 gradi di libertà (Figura 6). Con un livello di confidenza di 95%, il modello *Prop MLP* risulta essere statisticamente differente da *Naive Bayes*; questo perché l'estremo superiore dell'intervallo di confidenza assume valore > 0. Dopo aver verificato la significatività statistica tra i due classificatori si sono creati altri intervalli di confidenza (confrontando sempre *Prop MLP* con gli altri due modelli) ed è emerso che *Prop MLP* risulta essere in ogni caso il classificatore migliore.

3.3 Miglioramenti eventuali

Una volta valutati i modelli e constatato che i risultati non sono pienamente soddisfacenti, si è deciso di migliorarne le

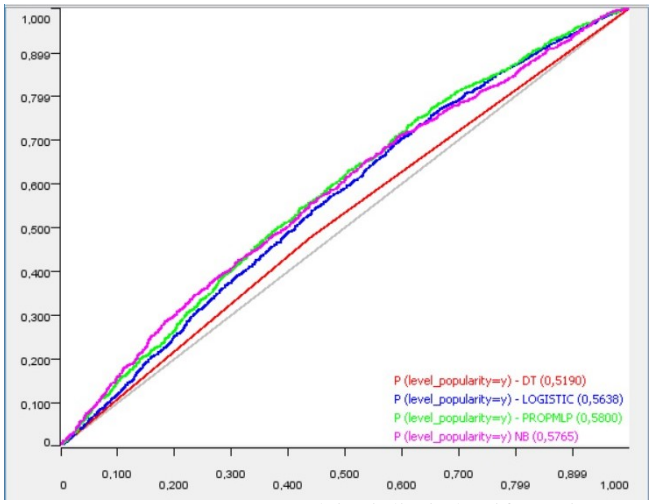

Figura 5. ROC curve dei migliori classificatori

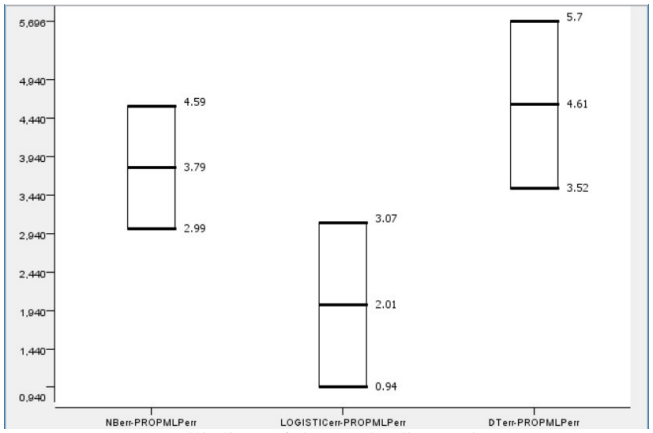

Figura 6. Intervalli di confidenza degli errori tra *Naive Bayes* e *Prop MLP*

prestazioni effettuando una discretization sul campione. Si è dapprima calcolato il range, ottenendo i seguenti risultati:

- $range = [0;1]$ per *acousticness*, *danceability*, *instrumentalness* e *valence*
- $range = [-51.42;0.612]$ per *loudness*

L'attributo *duration_ms* non è stato considerato dato che è una variabile intera.
Visti i risultati ottenuti si sceglie di procedere con le seguenti discretizzazioni:

- le variabili che hanno ottenuto un valore di range pari a 1 vengono discretizzate in 5 binner
- la variabile *loudness* viene discretizzata in 10 binner
- si decide di non discretizzare la variabile *duration_ms*

Nello specifico si è deciso di realizzare una discretizzazione non supervisionata di tipo "equal frequency"; non si è considerata l'altra tipologia, ovvero "equal width", in quanto alcuni attributi del campione presentano outlier. Una volta discretizzato il campione, è stato nuovamente condotto l'addestramento dei modelli migliori, *Prop MLP* escluso (questo

perché le reti neurali non accettano attributi nominali come variabili esplicative).
Analizzando la ROC curve (Figura 7) si osserva che i classificatori *Naive Bayes* e *Logistic* hanno migliorato le loro prestazioni, misurate in termini di AIUC, passando rispettivamente da 0.576 a 0.591 e da 0.563 a 0.577.

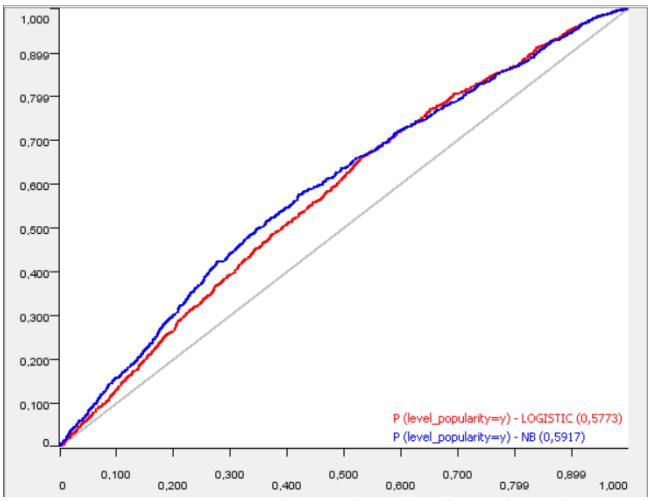

Figura 7. ROC curve in seguito alla discretizzazione

Riguardo gli indicatori di prestazione, si decide di visualizzare solamente i modelli migliori (Figura 8); analizzando $F_1 measure$ si evidenziano un miglioramento del modello *Logistic* e un peggioramento del modello *Naive Bayes*.

Row ID	D precision	D recall	D f1
LOGISTIC	0.529	0.356	0.425
NB	0.508	0.599	0.549

Figura 8. Misure delle performance dopo la discretizzazione

4. Conclusioni

Prevedere il successo di una canzone è una sfida che va oltre l'analisi degli elementi che compongono una traccia musicale: anche se tracce 'commerciali' condividono attributi comuni, è difficile riuscire a definire la ricetta giusta per sfondare nel mondo della musica. I risultati riscontrati, in ogni caso, non sono del tutto negativi; considerando le misure di performance dei due classificatori migliori (*Prop MLP* e *Naive Bayes*) si può notare che sia la precision che la recall non assumono valori eccessivamente bassi. Nello specifico, il primo classificatore assume un valore di precision 0.50 e una recall di 0.45 mentre il secondo classificatore come precision assume valore 0.49 mentre come recall 0.70. Il dataset lascia spazio a successive analisi che permetterebbero di identificare variabili d'interesse sulla popolarità di un brano, una su tutte l'artista: risulterebbe infatti interessante analizzare gli artisti con una base di supporters simile al fine di valutare, a parità di condizione iniziale, come viene influenzata la popolarità di un brano.

Riferimenti bibliografici

[1] Wikipedia. Spotify.

[2] Denis Rizzoli. Spotify: ecco quanto incassano davvero gli artisti.

[3] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York, 2001.

[4] Ray Sunil. Understanding support vector machine algorithm from examples (along with code).

[5] Ben Kröse, Ben Krose, Patrick van der Smagt, and Patrick Smagt. An introduction to neural networks. 1993.

[6] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.