

---

# **Progetto Machine Learning**

---

## **SPOTIFY AWARD**

GIANLUCA GIUDICE - 830694

MARCO GRASSI - 829664

# Contents

|          |                                                           |           |
|----------|-----------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduzione</b>                                       | <b>3</b>  |
| 1.1      | Descrizione del problema . . . . .                        | 3         |
| 1.1.1    | Spotify . . . . .                                         | 3         |
| 1.1.2    | Premi per i singoli musicali . . . . .                    | 3         |
| 1.2      | Approccio al problema . . . . .                           | 4         |
| 1.3      | Struttura del codice . . . . .                            | 4         |
| <b>2</b> | <b>Dataset</b>                                            | <b>5</b>  |
| 2.1      | Acquisizione del dataset . . . . .                        | 5         |
| 2.1.1    | Dataset spotify - Kaggle . . . . .                        | 5         |
| 2.1.2    | Premi canzoni - Wikipedia . . . . .                       | 5         |
| 2.2      | Descrizione del dataset . . . . .                         | 7         |
| 2.2.1    | Spotify . . . . .                                         | 7         |
| 2.2.2    | Premi . . . . .                                           | 8         |
| 2.3      | Data integration . . . . .                                | 8         |
| 2.3.1    | Record linkage con MongoDB . . . . .                      | 8         |
| 2.4      | Analisi esplorativa . . . . .                             | 9         |
| 2.4.1    | Assunzioni sul dominio . . . . .                          | 9         |
| 2.4.2    | Creazione dataset bilanciato . . . . .                    | 13        |
| 2.4.3    | Preprocessing . . . . .                                   | 13        |
| 2.4.4    | Distribuzione dei valori . . . . .                        | 14        |
| 2.4.5    | Artisti nelle canzoni . . . . .                           | 17        |
| 2.4.6    | Correlazione tra features . . . . .                       | 19        |
| 2.4.7    | Principal component analysis . . . . .                    | 20        |
| 2.5      | Scelta delle features . . . . .                           | 21        |
| 2.5.1    | Variabili scartate . . . . .                              | 21        |
| 2.5.2    | Rappresentazione one hot encoding degli artisti . . . . . | 22        |
| 2.5.3    | Variabili categoriche . . . . .                           | 22        |
| 2.5.4    | Coordinate PCA . . . . .                                  | 23        |
| 2.5.5    | Riassunto feature finali utilizzate . . . . .             | 23        |
| <b>3</b> | <b>Campagna sperimentale</b>                              | <b>24</b> |
| 3.1      | Approccio . . . . .                                       | 24        |
| 3.1.1    | 10-folds cross validation . . . . .                       | 24        |
| 3.1.2    | Training set e test set . . . . .                         | 24        |
| 3.1.3    | Model selection . . . . .                                 | 24        |
| 3.2      | Misure di performance . . . . .                           | 24        |
| 3.2.1    | Accuracy . . . . .                                        | 24        |
| 3.2.2    | Precision, Recall e F-measure . . . . .                   | 24        |
| 3.2.3    | Curve ROC . . . . .                                       | 24        |
| 3.3      | Support Vector Machine . . . . .                          | 24        |

|                                   |           |
|-----------------------------------|-----------|
| CONTENTS                          | 2         |
| 3.3.1 Kernel . . . . .            | 24        |
| 3.4 Decision Tree . . . . .       | 24        |
| 3.5 Esperimenti . . . . .         | 24        |
| 3.5.1 Performance . . . . .       | 24        |
| 3.6 Modelli a confronto . . . . . | 24        |
| <b>4 Conclusioni</b>              | <b>25</b> |

# Chapter 1

## Introduzione

In questo capitolo viene introdotto il problema, il dominio di riferimento e l'approccio adottato per la risoluzione.

### 1.1 Descrizione del problema

In questo elaborato consideriamo i singoli musicali. Al giorno d'oggi le canzoni vengono spesso ascoltate dagli utenti tramite piattaforme di streaming musicale.

L'obiettivo del lavoro è quello di analizzare le features dei singoli musicali così da prevedere se una canzone diventerà o meno di successo. Nella sezione successiva verrà meglio specificato cosa si intende per **singolo musicale di successo** (subsection 1.1.2).

#### 1.1.1 Spotify

Spotify è un servizio di riproduzione digitale in streaming di musica, podcast e video, con accesso immediato a milioni di brani e altri contenuti di artisti provenienti da tutto il mondo. Questa piattaforma viene utilizzata da milioni di utente per ascoltare canzoni e nello specifico singoli musicali.

Da questo servizio è possibile ottenere migliaia di brani musicali, infatti Spotify mette a disposizione una API da cui è possibile scaricare informazioni su questi brani con associate alcune caratteristiche. Pertanto oltre alle classiche informazioni di un brano come "titolo" o "artisti" si avrà a disposizione una serie di caratteristiche specifiche del brano, ad esempio quanto una canzone è "energica" o "ballabile".

#### 1.1.2 Premi per i singoli musicali

Come vedremo in subsection 2.2.1 il dataset ottenuto da Spotify mette a disposizione un campo "popularity" che indica quanto può essere considerata popolare. Tuttavia questa caratteristica non ci sembra adeguata per identificare una canzone come di successo oppure no.

Per questo motivo consideriamo una canzone di successo in base alle certificazioni ottenute, rispettivamente "disco d'oro" o "disco di platino". Queste premi sono dei riconoscimenti vinti da un brano musicale e storicamente fanno riferimento al numero di copie vendute da un singolo. Tuttavia con la costante crescita dell'utilizzo di servizi per lo streaming di brani musicali, da qualche anno questi riconoscimenti vengono assegnati anche considerando il numero di streaming sulle diverse piattaforme, tra cui Spotify.

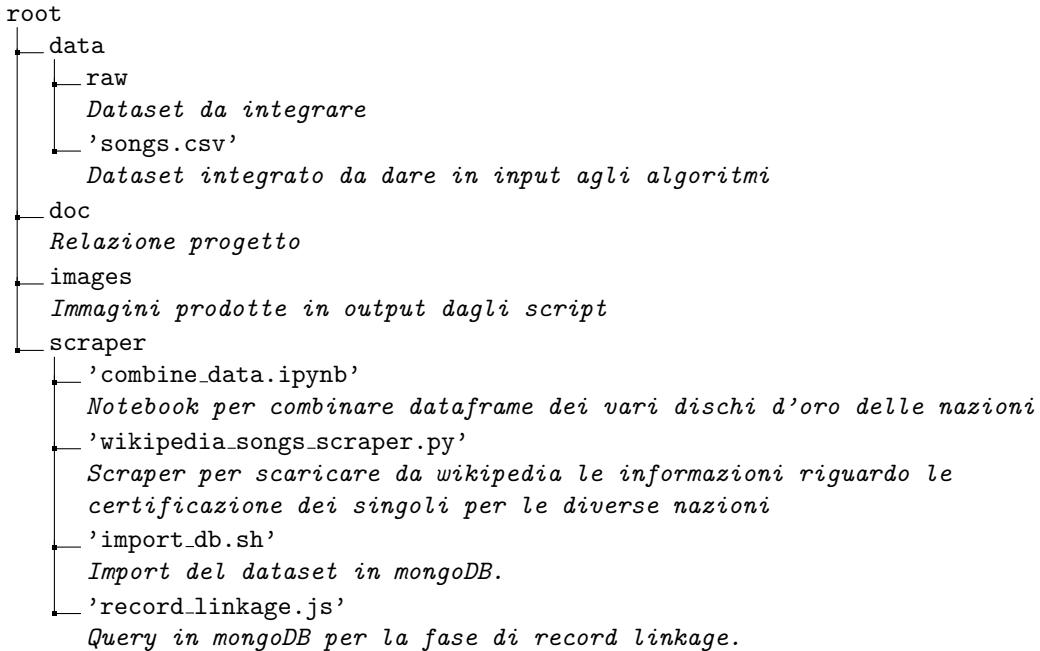
Riteniamo che questo riconoscimento sia una metrica oggettiva per considerare un singolo come di successo.

## 1.2 Approccio al problema

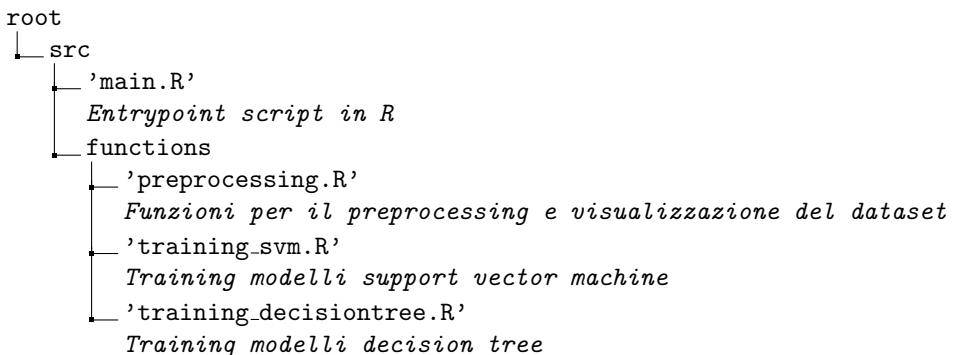
Il problema viene approcciato come un **task di classificazione binaria**. Dato un singolo musicale vogliamo prevedere se questo sarà di successo o no. Pertanto sviluppiamo modelli supervisionati di machine learning partendo da un dataset etichettato, in questo modo è possibile classificare i brani musicali.

## 1.3 Struttura del codice

Di seguito viene brevemente spiegata la struttura del codice, inoltre viene sotto indicata la working directory e l'entry point del programma.



In particolare gli **script in R** si trovano in:



**N.B.:** Per come è stato progettato il codice, la working directory è la `root/` e non la cartella `src/`. L'**entry point del programma** è lo script `src/main.R`

# Chapter 2

## Dataset

In questo capitolo viene analizzato il dataset. Viene quindi spiegato da dove è stato preso il dataset, descritte le covariate e viene effettuata un'analisi esplorativa.

### 2.1 Acquisizione del dataset

Il dataset completo contenente sia le informazioni sui singoli musicali che riguardo le varie certificazioni, ovvero le etichette del dataset, non è disponibile da una sola sorgente. Pertanto abbiamo ottenuto le informazioni necessarie da diversi sorgenti per poi integrare i dati.

#### 2.1.1 Dataset spotify - Kaggle

Per quanto riguarda i brani con le relative informazioni e caratteristiche del brano, Spotify mette a disposizione un'API da cui si possono ottenere questi dati.

Con questa tecnica sono stati ottenuti i dati relativi ai brani, un utente ha quindi caricato il dataset sul sito kaggle.com. Il dataset è disponibile su kaggle all'url indicato<sup>1</sup>.

Il dataset contenente queste informazioni è il file: `data/raw/to_integrate/data.csv`

#### 2.1.2 Premi canzoni - Wikipedia

Le informazioni riguardo i premi delle canzoni, ovvero le varie certificazioni vinte, sono disponibili su wikipedia.org. Questa informazione costituisce di fatto l'etichetta per classificare ogni brano musicale.

Nello specifico siamo interessati a:

- Singoli certificati oro
- Singoli certificati platino
  - 1. Singoli certificati 1 volta platino
  - 2. Singoli certificati 2 volte platino
  - 3. ...
  - 4. Singoli certificati N volte platino

Inoltre le varie certificazioni dei singoli vengono considerati in base al paese. I paesi da noi presi in considerazione sono:

---

<sup>1</sup>**Dataset Spotify:** <https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks>.

- Italia
- Australia
- Stati Uniti d'America
- Regno Unito
- Canda
- Danimarca

Si noti come una certificazione può essere consegnata in diversi paesi alla stessa canzone.

### Certificazioni disco d'oro

Per quanto riguarda i dischi d'oro, facciamo riferimento a questa pagina su wikipedia<sup>2</sup>. Fissato quindi uno stato (ad esempio l'Italia) è possibile visualizzare la pagina contenente la lista dei singoli che hanno vinto quel particolare premio. La lista è un elenco di url che puntano alla pagina wikipedia della canzone, un esempio a questo url<sup>3</sup>.

### Certificazioni disco di platino

Ragionamento analogo viene fatto per i dischi di platino, con l'unica differenza che la pagina è questa<sup>4</sup>, inoltre dal momento che i singoli posso vincere più volte un disco di platino, si considerano non solo i singoli che hanno vinto una volta il disch di platino ma anche quelli che l'hanno vinto N volte.

### Scraping

Ottenuti quindi i puntatori alle canzoni certificate, è possibile accedere alla pagina wikipedia del singolo, la quale contiene una tabella riassuntiva della canzone. Un esempio di tabella viene mostrato nella Figura 2.1.

---

<sup>2</sup>Disco d'oro per stato:

[https://it.wikipedia.org/wiki/Categoria:Singoli\\_certificati\\_oro\\_per\\_stato](https://it.wikipedia.org/wiki/Categoria:Singoli_certificati_oro_per_stato).

<sup>3</sup>Singoli certificati disco d'oro in Italia:

[https://it.wikipedia.org/wiki/Categoria:Singoli\\_certificati\\_disco\\_d%27oro\\_in\\_Italia](https://it.wikipedia.org/wiki/Categoria:Singoli_certificati_disco_d%27oro_in_Italia).

<sup>4</sup>Singoli certificati disco di platino per stato:

[https://it.wikipedia.org/wiki/Categoria:Singoli\\_certificati\\_platino\\_per\\_stato](https://it.wikipedia.org/wiki/Categoria:Singoli_certificati_platino_per_stato).

## Chico (singolo)

Da Wikipedia, l'enciclopedia libera.

**Indice [nascondi]**

- 1 Classifiche
  - 1.1 Classifiche settimanali
  - 1.2 Classifiche di fine anno
- 2 Note
- 3 Collegamenti esterni

**Classifiche** [ modifica | modifica wikitesto ]

| Classifiche settimanali           |                   | Classifiche di fine anno          |           |
|-----------------------------------|-------------------|-----------------------------------|-----------|
| [ modifica   modifica wikitesto ] |                   | [ modifica   modifica wikitesto ] |           |
| Classifica (2020)                 | Posizione massima | Classifica (2020)                 | Posizione |
| Italia <sup>[3]</sup>             | 5                 | Italia <sup>[4]</sup>             | 10        |

**Chico**

|                                 |                                                  |
|---------------------------------|--------------------------------------------------|
| Artista                         | Gué Pequeno                                      |
| Featuring                       | Rose Villain e Luchè                             |
| Tipo album                      | Singolo                                          |
| Pubblicazione                   | 31 luglio 2020                                   |
| Durata                          | 3:33                                             |
| Album di                        | <i>Mr. Fini</i>                                  |
| provenienza                     |                                                  |
| Genere                          | Pop rap                                          |
| Etichetta                       | Island                                           |
| Produttore                      | Sixpm                                            |
| Formati                         | Streaming                                        |
| <b>Certificazioni</b>           |                                                  |
| Dischi di platino               | ITALIA (3) <sup>[1]</sup><br>(vendite: 210 000+) |
| <b>Gué Pequeno - cronologia</b> |                                                  |
| Singolo precedente              | Singolo successivo                               |
| <i>Saigon</i><br>(2020)         | <i>Bla Bla</i><br>(2020)                         |
| <b>Luchè - cronologia</b>       |                                                  |
| Singolo precedente              | Singolo successivo                               |
| <i>Come me</i><br>(2019)        | <i>Maserati (Reloaded)</i><br>(2020)             |

Figure 2.1: Esempio di tabella per un singolo musicale su wikipedia

Viene quindi creato uno script per effettuare scraping delle tabelle, il codice è nella directory `scraping/wikipedia_songs_scraper.py`.

Successivamente si integrano le informazioni dei diversi stati, e tipi di certificazione vinti. Il risultato di questa operazione è il file `data/raw/to_integrate/awards_cleaned.csv`.

## 2.2 Descrizione del dataset

In questa sezione vengono elencate e descritte le features del dataset. Trattandosi di un problema supervisionato, ad ogni istanza viene associata la corretta etichetta, ovvero la classe positiva per i brani musicali che hanno vinto un premio e classe negativa per i brani che non hanno vinto un premio.

### 2.2.1 Spotify

Di seguito viene descritto il dataset proveniente da kaggle, ovvero quello contenente le informazioni dei brani.

| ATTRIBUTO        | DESCRIZIONE                                                       | TIPO                   |
|------------------|-------------------------------------------------------------------|------------------------|
| id               | Identificativo della canzone (generato da spotify)                | Intero                 |
| name             | Titolo della canzone                                              | Stringa                |
| artists          | Lista degli artisti che compaiono nel brano                       | Stringa                |
| year             | Anno del brano                                                    | Intero                 |
| duration_ms      | Durata della traccia in millisecondi                              | Intero                 |
| acousticness     | Metrica riguardante quanto un brano risulta "acustico"            | Float [0, 1]           |
| danceability     | Metrica riguardante quanto una traccia è ballabile                | Float [0, 1]           |
| energy           | Energia del brano                                                 | Float [0, 1]           |
| instrumentalness | Contenuto relativo di strumenti musicali nella traccia            | Float [0, 1]           |
| valence          | Metrica riguardante la "positività" della traccia                 | Intero                 |
| liveness         | Durata relativa della traccia suonata in una performance dal vivo | Float [0, 1]           |
| loudness         | Rumorosità della traccia in decibel (dB)                          | Float [-60, 0]         |
| release_date     | Anno di rilascio del brano                                        | Intero                 |
| speechiness      | Contenuto relativo di voce umana nella traccia                    | Float [0, 1]           |
| tempo            | BPM della traccia                                                 | Float                  |
| key              | Scala musicale utilizzata                                         | Factor {0, 1, ..., 11} |
| mode             | Indica se la traccia parte con una progressione armonica          | Booleano               |
| explicit         | Indica se la traccia è esplicita oppure no (linguaggio volgare)   | Booleano               |
| popularity       | Popolarità della traccia                                          | Float [0, 100]         |

### 2.2.2 Premi

Si noti come la distinzione tra tipo di premio vinto e lo stato in cui è stata ottenuta la certificazione per uno specifico brano, viene fatta solo a scopo dello scraping, in quanto i brani sono così rappresentati sul sito di wikipedia. Tuttavia da questo punto in poi non verrà più tenuto conto di questa informazione, infatti un singolo verrà considerato come **vincitore di un premio** (e quindi di successo) oppure come **non vincitore di un premio** (non di successo).

Il risultato dello scraping della tabella di wikipedia è il seguente:

| ATTRIBUTO | DESCRIZIONE                          | TIPO                                     |
|-----------|--------------------------------------|------------------------------------------|
| title     | Titolo della canzone                 | Stringa                                  |
| artists   | Artisti presenti nella traccia       | Stringa                                  |
| date      | Data di rilascio della traccia       | Data                                     |
| genre     | Genere musicale del brano            | Stringa                                  |
| award     | Premio vinto dal singolo             | Stringa {Oro, 1-platino, 2-platino, ...} |
| nation    | Paese in cui è stato vinto il premio | Stringa (Sigla del paese)                |

## 2.3 Data integration

Date queste due sorgenti dati, è necessario integrare i dati allo scopo di avere un dataset etichettato, le label saranno appunto se un bravo ha vinto un premio (quindi è di successo) oppure no.

La strategia di entity resolution adottata è considerare un singolo musicale come una singola identità basandosi sul titolo e gli artisti di una canzone. Se il nome del brano musicale è lo stesso e gli artisti coincidono, allora il brano è il medesimo.

### 2.3.1 Record linkage con MongoDB

A questo scopo i due dataset vengono importati in mongoDB, ogni istanza è rappresentata da un documento. Per la fase di record linkage vengono effettuati i seguenti passi:

1. Import dei dataset in mongodb, inizialmente in due collezioni diverse.
2. Normalizzazione dei dati, i campi di join vengono trasformati in lowercase.
3. Creazione indici.

4. Entrambe le collezioni hanno il campo "artisti" il quale consiste in una stringa contenente la lista degli artisti separati dal carattere ", ". Viene quindi eseguito l'unfold del campo "artista" in entrambe le collezioni, costruendo una lista di per ogni documento facendo uno split sul carattere ", ".
5. Viene eseguite la join sul campo titolo considerando un match valido se l'intersezione tra gli insiemi di artisti dei documenti delle due collezioni non è vuota.
6. Viene ritrasformato il campo artista appiattendo la lista e rappresentando l'insieme degli artisti di un brano come una stringa, separando ogni artista con il carattere ", ".
7. Dump del database in un file .csv, questo è il dataset integrato e verrà usato per il training dei modelli.

## 2.4 Analisi esplorativa

In questa sezione vengono analizzate e discusse le covariate del dataset. Dopo la fase di record linkage le feature del dataset finale sono:

| Covariata         | <b>id</b>  | <b>name</b>         | <b>artists</b>  | <b>year</b>             | <b>duration_ms</b> | <b>acousticness</b> |
|-------------------|------------|---------------------|-----------------|-------------------------|--------------------|---------------------|
| <b>Utilizzata</b> | No         | No                  | Si              | No                      | Si                 | Si                  |
| <b>Covariata</b>  |            | <b>danceability</b> | <b>energy</b>   | <b>instrumentalness</b> | <b>valence</b>     |                     |
| <b>Utilizzata</b> |            | Si                  | Si              | Si                      | Si                 |                     |
| <b>Covariata</b>  |            | <b>liveness</b>     | <b>loudness</b> | <b>release_date</b>     | <b>speechiness</b> | <b>tempo</b>        |
| <b>Utilizzata</b> | Si         | Si                  |                 | No                      | Si                 | Si                  |
| <b>Covariata</b>  | <b>key</b> | <b>mode</b>         | <b>explicit</b> | <b>popularity</b>       | <b>award</b>       |                     |
| <b>Utilizzata</b> | Si         | Si                  | Si              | No                      | Label              |                     |

Table 2.1: Tabella riassuntiva di tutte le features del dataset.

Sotto viene riportata una tabella riassuntiva riguardo il numero di istanze nel dataset. Il significato dei threshold viene spiegato nella sezione successiva (subsection 2.4.1).

| DATASET                                                             | # ISTANZE |
|---------------------------------------------------------------------|-----------|
| Dataset completo                                                    | 151762    |
| Dataset ( $release\_date \geq 2005 \& popularity > 25$ )            | 26134     |
| Dataset bilanciato ( $release\_date \geq 2005 \& popularity > 25$ ) | 3670      |

Table 2.2: Numero di istanze nel dataset.

### 2.4.1 Assunzioni sul dominio

#### Anno di uscita singoli

Viene per prima cosa analizzata la distribuzione degli anni di uscita nel dataset.

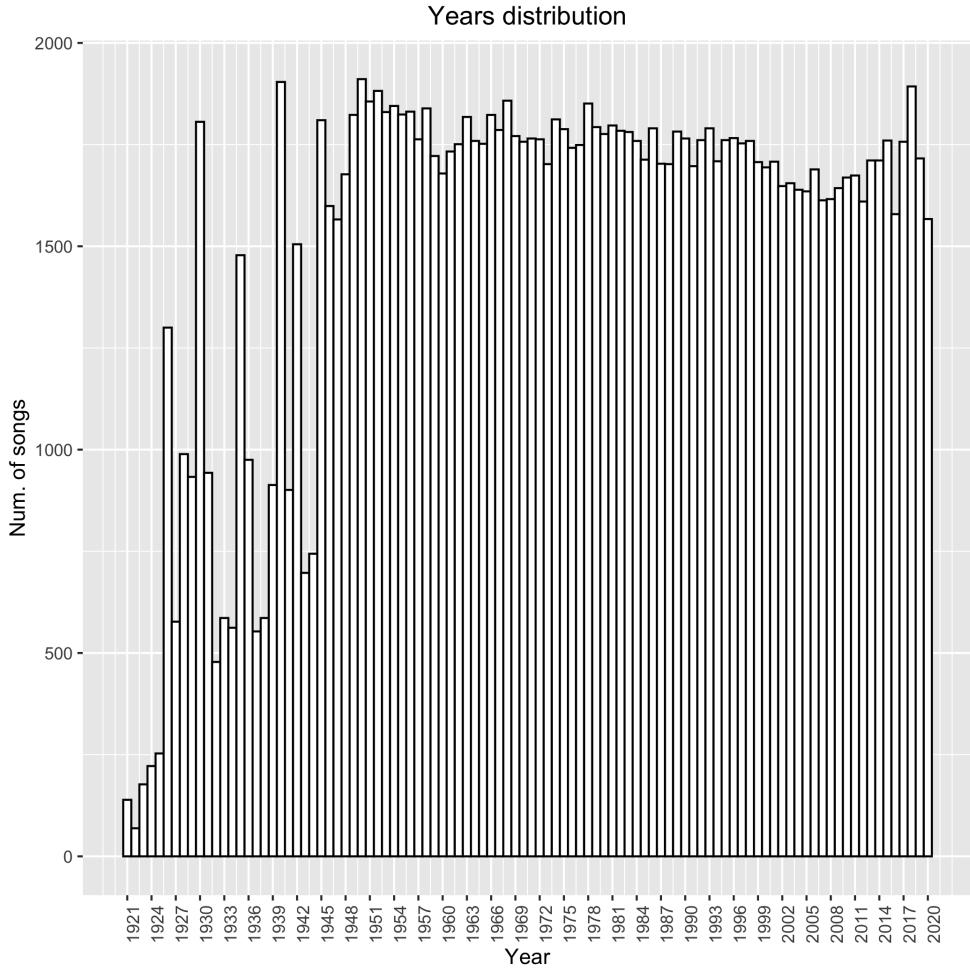


Figure 2.2: Distribuzione anno delle canozni considerando tutto il dataset.

Dal momento che Spotify è una piattaforma recente, si assume che gli utenti ascoltino maggiormente i brani usciti negli ultimi anni. Inoltre le diverse certificazioni come "Disco d'oro" e "Disco di platino" vengono rilasciate considerando il numero di streaming oltre che alle vendite solo da pochi anni. Inoltre con il passare del tempo i trend musicali cambiano, un fattore determinante per fare diventare una canzone di successo.

Con questa giustificazione si ritiene che sia meglio considerare solo i brani musicali dopo un certo anno di uscita. Prendiamo quindi in considerazione solo le canzoni dopo il 2005.

La distribuzione delle canzoni dal 2005 in poi è rappresentata in Figure 2.2. Viene di seguito mostrato il boxplot delle canzoni dopo quella data distinguendo tra classe positiva e negativa, così da vedere se esistono differenze.

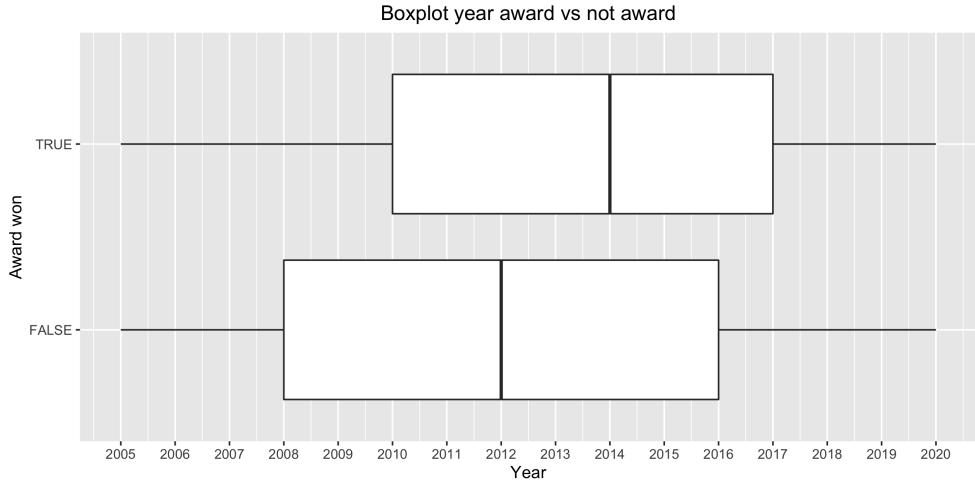


Figure 2.3: Boxplot anno di rilascio, distinguendo tra le classi.

### Popolarità

Un altro apsetto che viene analizzato a parte, riguardo il quale è necessario fare ulteriori assunzioni, è il campo popolarità.

Il dataset fornito da Spotify contiene un campo "popularity". Tuttavia come spiegato nella subsection 1.1.2, questo campo non viene utilizzato per etichettare una canzone come di successo, si utilizza invece l'informazione delle certificazioni vinte da una canzone (ottenute da wikipedia).

Il campo "popularity" non verrà usato nella fase di training dei modelli, proprio perché è un dato che non si conosce a priori nel momento in cui un singolo esce, ed è chiaramente influente nel determinare se una canzone vincerà o meno una certificazione e quindi se viene considerata di successo.

L'informazione sulla popolarità viene calcolata sul numero di streaming dopo un determinato lasso di tempo. Stimare questo valore, dal momento che non è conosciuto fin da subito, è di fatto un problema di regressione ed è molto simile al task di classificazione preso in esame, pertanto il campo viene scartato.

Anche se il campo non viene effettivamente utilizzato, è interessante analizzare la distribuzione dei valori di popolarità delle canzoni nel dataset. Inoltre assumiamo che per il problema trattato, si vogliono classificare delle canzoni che non sono completamente sconosciute. Sarebbe infatti irrealistico pensare che un singolo musicale del tutto sconosciuto vinca dal nulla una certificazione, risultando come brano di successo. Per questo motivo non si tratta di una assunzione molto restrittiva.

Si assume di voler utilizzare questi modelli per classificare brani che iniziano ad essere un minimo conosciuti o hanno almeno il potenziale di diventare popolari. Si noti come un brano di poco popolare non implica assolutamente che questo vinca un disco d'oro o di platino.

Di seguito viene analizzata la distribuzione della popolarità delle canzoni nel dataset.

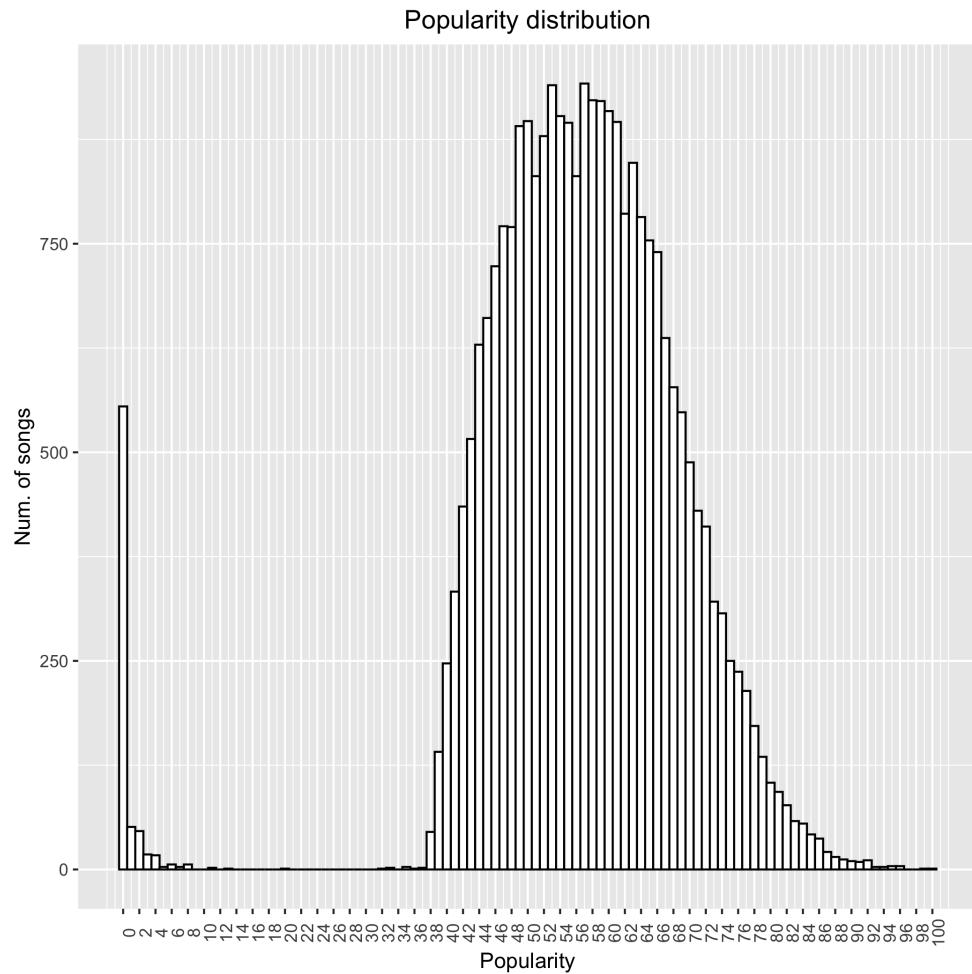


Figure 2.4: Distribuzione popolarità brani musicali.

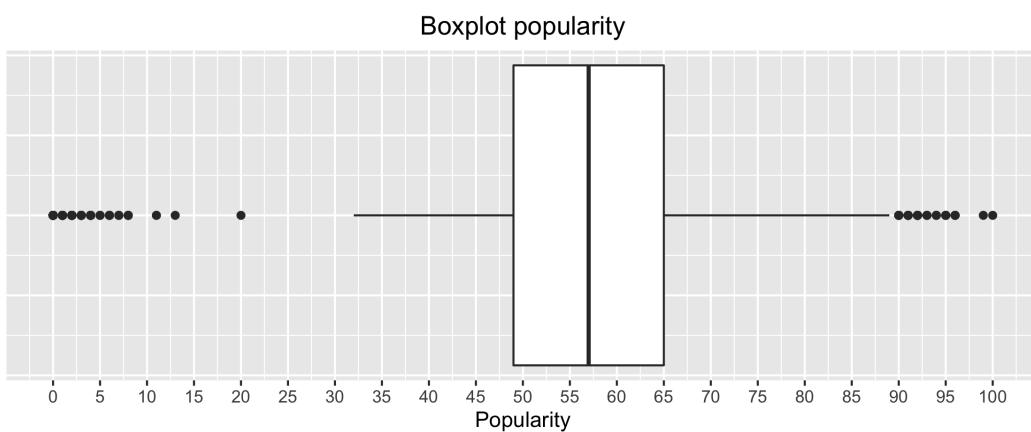


Figure 2.5: Boxplot popolarità dei brani musicali.

Analizzando la distribuzione dei valori della popolarità si considerano il primo secondo e terzo quartile, rispettivamente:  $q_{1/4} = 49$ ;  $q_{2/4} = 57$ ;  $q_{3/4} = 65$ . Inoltre i valori di media e

deviazione standard sono:  $\mu = 57.83$ ;  $\sigma = 10.12$ .

Analizzando la distribuzione dei valori della popolarità viene scelto 25 come threshold, valore 3 volte più estremo della deviazione standard. Si considerano quindi solo i singoli con il valore "popularity" maggiore del threshold. In questo modo il dataset rispecchia l'assunzione sopra spiegata riguardo la popolarità minima, tuttavia si scartano solo i valori davvero estremi per non rendere questa assunzione troppo restrittiva.

Dopo queste operazioni di riduzione del numero di istanze del dataset in base alle soglie, il numero di istanze diventa 26134.

### 2.4.2 Creazione dataset bilanciato

Dopo aver ridotto il dataset in base alle diverse soglie si vuole vedere il numero di istanze appartenente alla classe positiva e negativa. Per classe positiva si intendono i brani che hanno vinto un premio e sono quindi di successo, viceversa per la classe negativa.

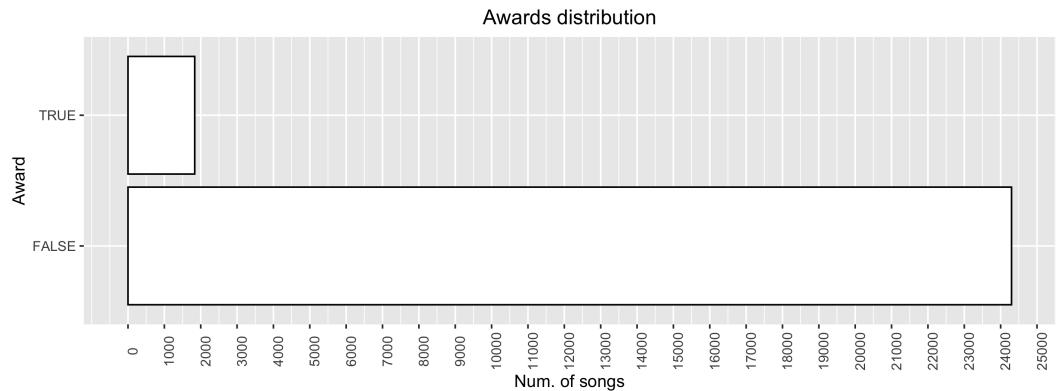


Figure 2.6: Istanze positive e negative.

#### Undersampling

Dalla Figura 2.6 emerge che il dataset è sbilanciato, il rapporto tra classe negativa e positiva è circa 14 : 1. Dal momento che questo può creare problemi in fasi di training, si sceglie di costruire un dataset bilanciato a partire da tutte le istanze. Come strategia per la costruzione di questo nuovo dataset vengono scelte casualmente un numero di istanze negative pari al numero di istanze negative.

### 2.4.3 Preprocessing

Per quanto riguarda le variabili numeriche queste vengono standardizzare, ovvero per ogni covariata si trasformano gli individui in modo da ottenere dei valori tali che  $\mu = 0$ ;  $\sigma = 1$ . Questa è sicuramente una best practice ma è cruciale per quanto riguarda la PCA, tecnica che viene utilizzata e spiegata in subsection 2.4.7.

Inoltre nella fase di preprocessing tutti gli artisti vengono trasformati in lowercase. Questo è necessario nel momento in cui si vuole tenere traccia degli artisti nelle canzoni con una rappresentazione one hot encoding (subsection 2.5.2).

#### 2.4.4 Distribuzione dei valori

##### Variabili numeriche

Di seguito viene mostrata la distribuzione delle covariate, distinguendo tra singoli che hanno vinto un premio (Classe positiva) e quelli che non hanno vinto un premio (Classe negativa).

Dal pairplot della Figure 2.7 possiamo notare come non esiste una netta distinzione nei dati tra brani di successo e brani non di successo. Questo sarà sicuramente un problema in fase di classificazione e potrebbe portare a basse performance dei modelli. Riteniamo quindi che sia necessario considerare altre features per ben distinguere brani musicali di successo da quelli non di successo.

Il campo "popularity" ben distingue le due classi, tuttavia come già discusso in section 2.4.1, questa covariata non verrà utilizzata per la creazione dei modelli.

Inoltre esiste correlazione tra alcune covariate, come ad esempio tra "energy" e "loudness", questo rispecchia intuitivamente l'idea di come una canzone più energica è anche più rumorosa. La correlazione tra le variabili viene meglio analizzata in subsection 2.4.6.

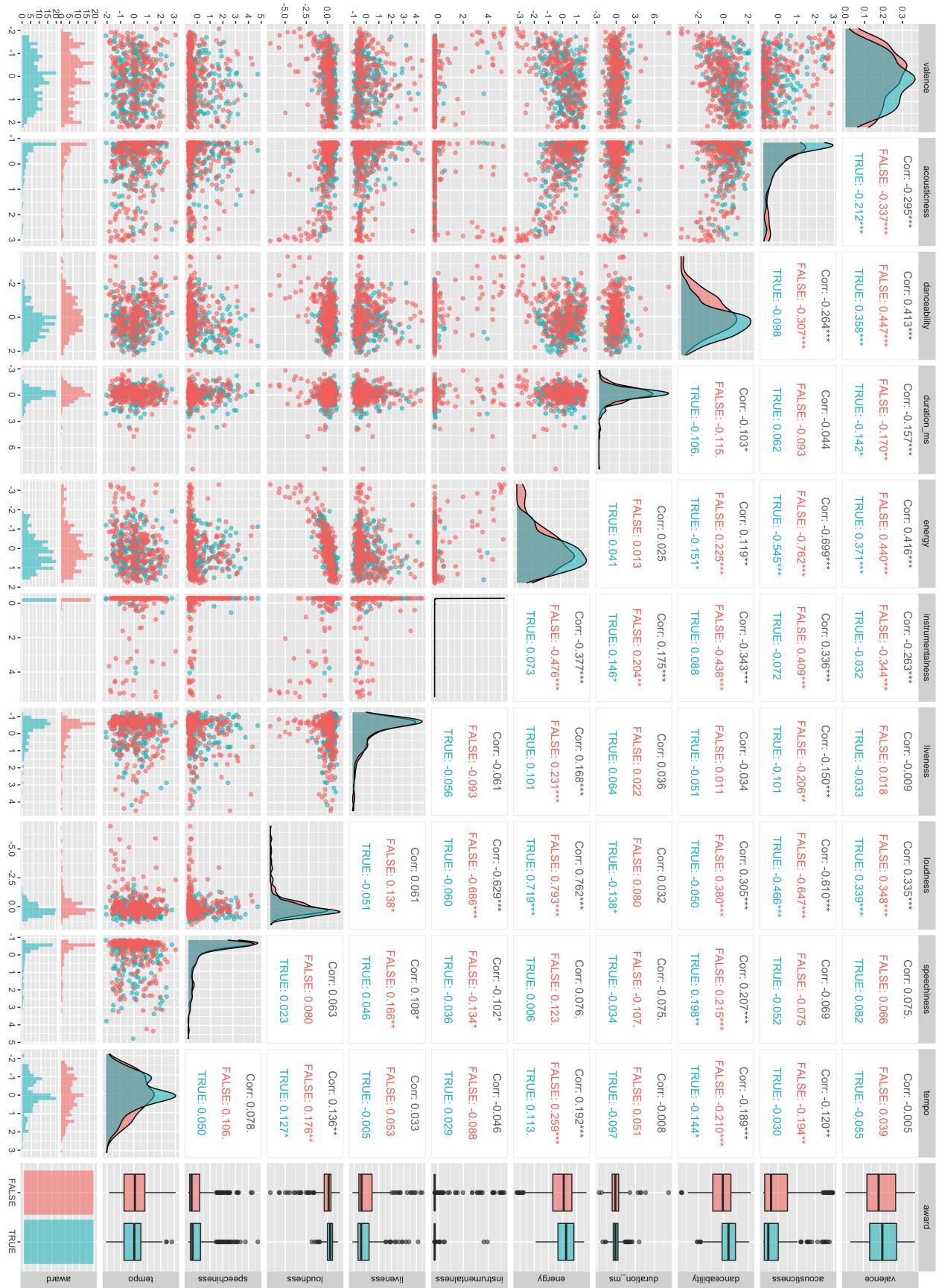
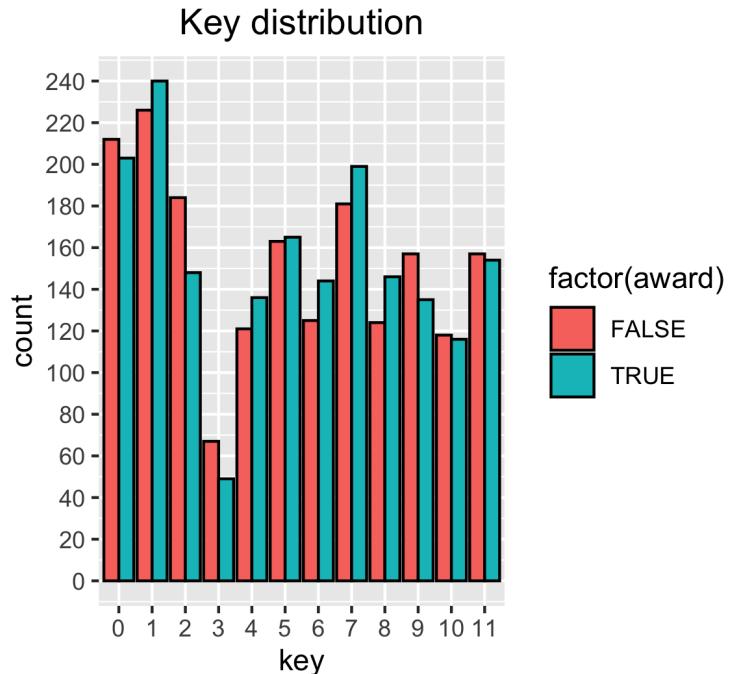


Figure 2.7: Pairplot delle covariate.

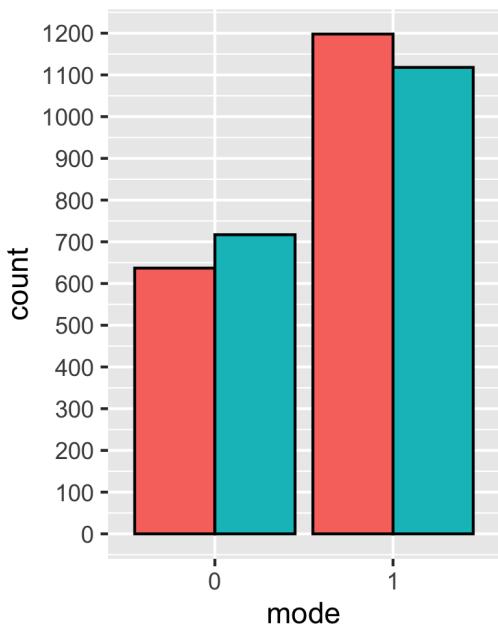
### Variabili categoriche

Di seguito si esplorano le variabili categoriche del dataset, distinguendo tra classe positiva e negativa.



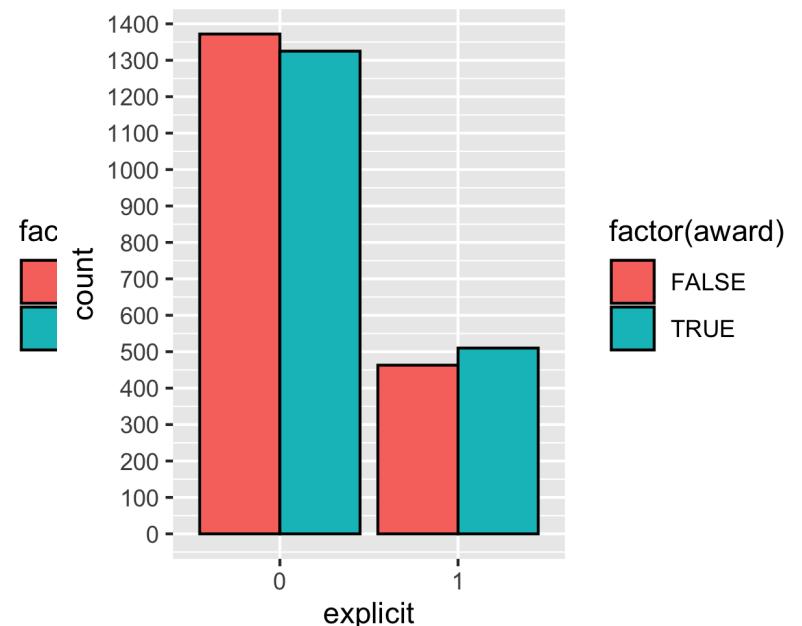
(a) Variabile key.

Mode distribution



(b) Variabile mode.

Explicit distribution



(c) Variabile explicit.

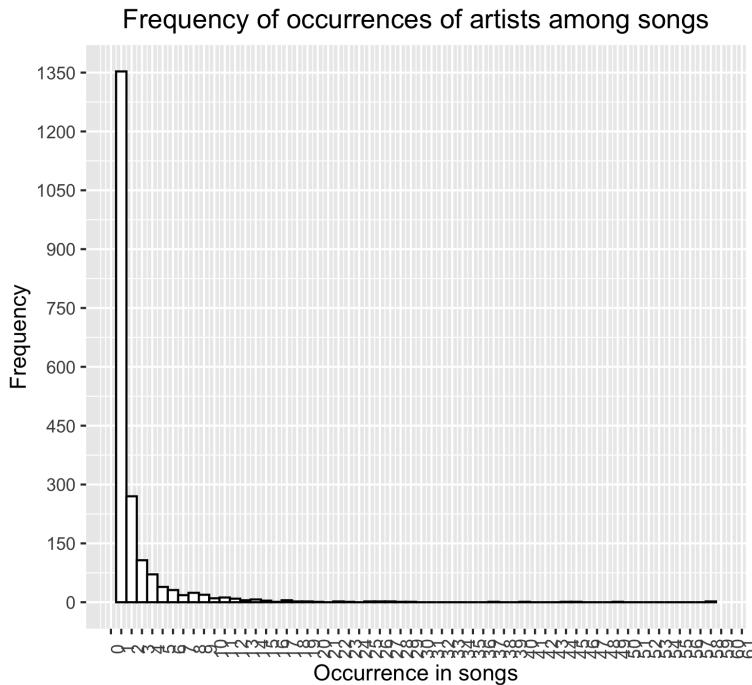
Figure 2.8: Distribuzione delle variabili categoriche.

### 2.4.5 Artisti nelle canzoni

Un'altra caratteristica che si ritiene importante per riconoscere una canzone come di successo, è quali artisti sono presenti in una canzone.

#### Frequenza artisti

Vine ora considerato il numero di occorrenze di ogni artista tra tutte le canzoni, si analizza quindi la frequenza delle occorrenze.



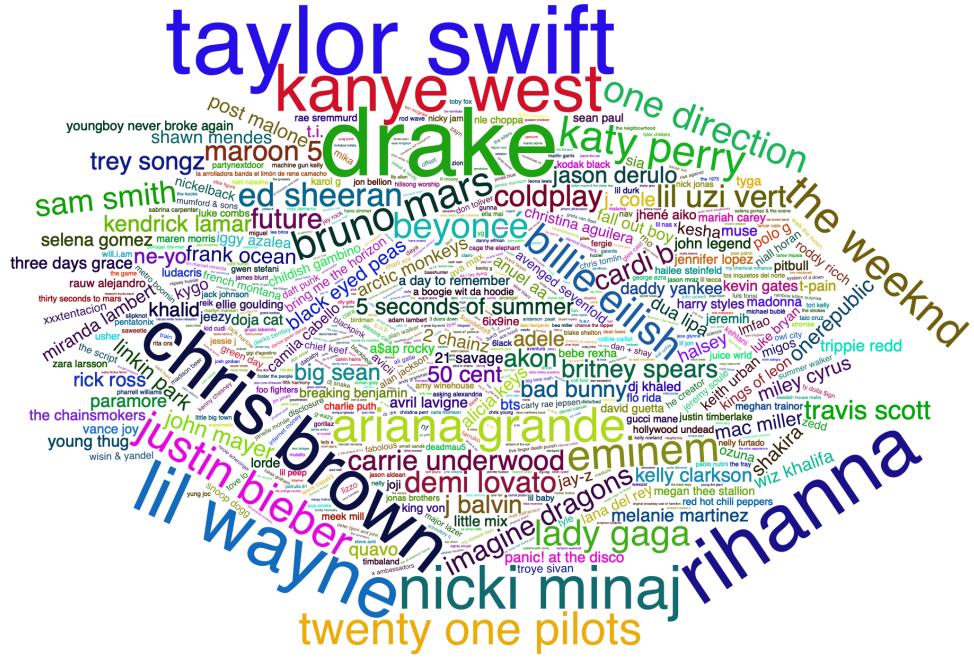


Figure 2.10: Wordcloud artisti considerando entrambe le classi.

Viene quindi mostrato il worcloud degli artisti andando a considerare solo le canzoni di successo:

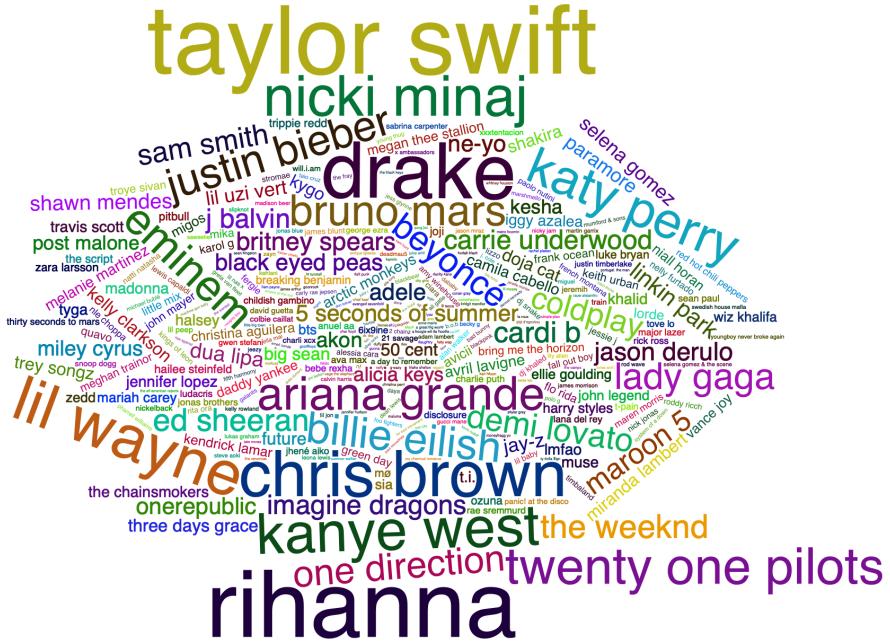


Figure 2.11: Wordcloud artisti considerando solo la classe positiva.

Analogamente il worcloud degli artisti considerando solo le canzoni non di successo:

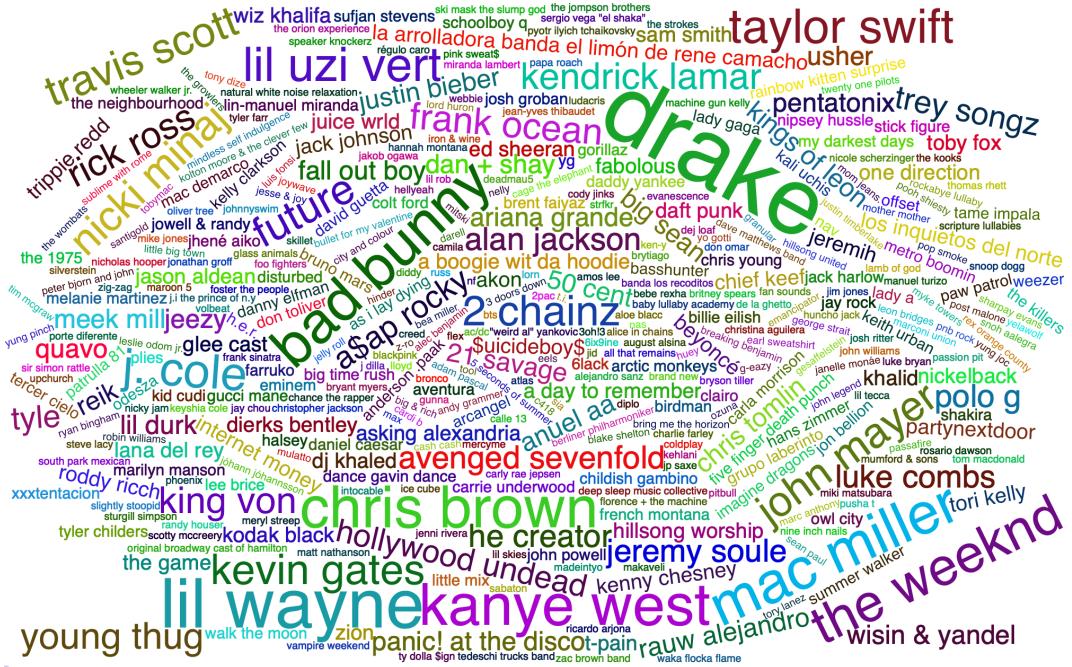


Figure 2.12: Wordcloud artisti considerando solo la classe negativa.

Dai grafici emerge che molti artisti compaiono sia nella classe positiva che in quella negativa. Tuttavia ci sono alcuni artisti e.g. "Taylor Swift" o "Rihanna" che spiccano nella classe di successo mentre sono meno frequenti nei brani non di successo. A fronte di questa analisi riteniamo che considerare gli artisti all'interno di un brano possa essere utile per distinguere le due classi. Viene quindi utilizzata una rappresentazione one hot encoding (spiegato in subsection 2.5.2) per utilizzare questa informazione durante la fase di training dei modelli.

#### 2.4.6 Correlazione tra features

Si considera ora la correlazione tra le faatures numeriche. A questo scopo calcoliamo la matrice di correlazione e la rappresentiamo qui sotto per mezzo di una heatmap:

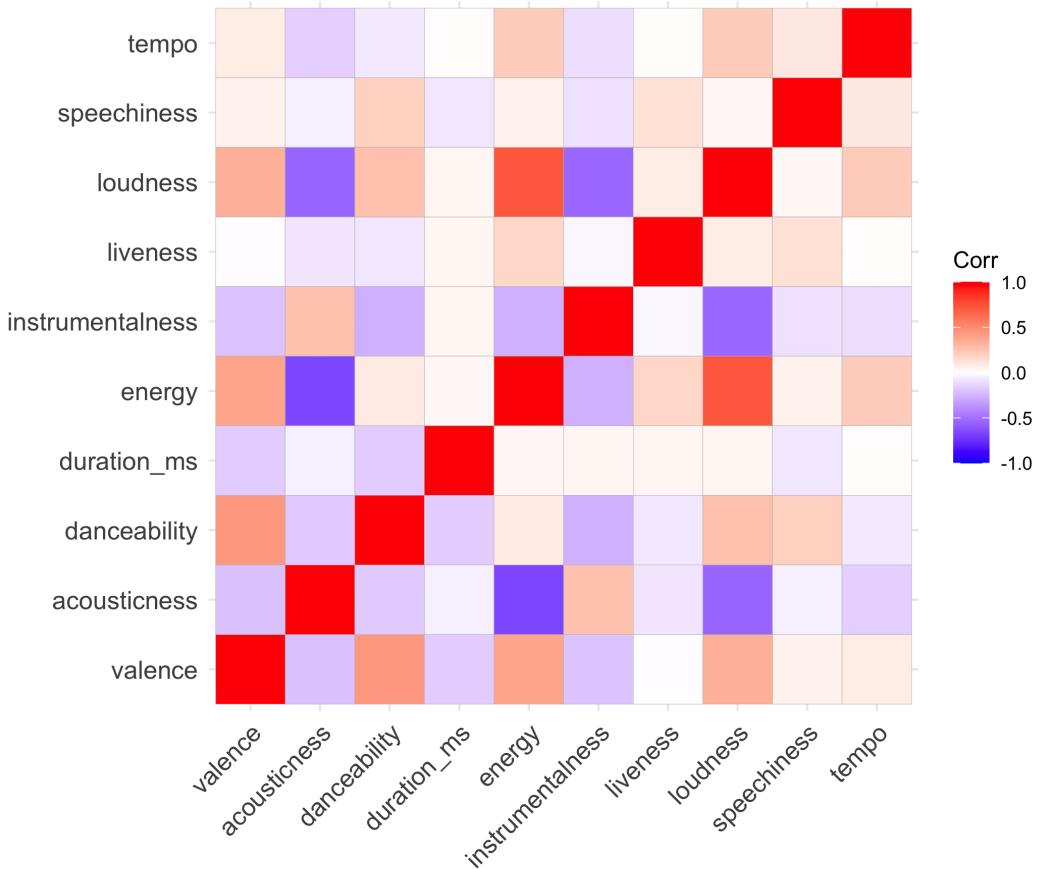


Figure 2.13: Matrice di correlazione.

Da questa immagine si può notare come alcune covariate, ad esempio "energy" e "loudness", sono correlate positivamente, mentre altre covariate come "acousticness" e "energy" oppure "acousticness" e "loudness" sono correlate negativamente. Questo rispecchia il senso comune di energia di una canzone, e ci si aspetta che al crescere di questa aumenta anche la loudness di un brano.

#### 2.4.7 Principal component analysis

Dal momento che esiste correlazione tra le covariate, viene utilizzata la tecnica PCA con lo scopo di ridurre la dimensione del dataset spiegando la maggior parte della varianza.

La PCA viene effettuata solo sulle features numeriche, si noti inoltre come il dataset è stato precedentemente standardizzato, in questo modo i valori numerici hanno media 0 e deviazione standard 1. Questa è una condizione necessaria per applicare correttamente la tecnica PCA.

A partire dalla matrice di covarianza viene effettuata la decomposizione agli autovalori, si ottengono quindi gli autovettori con gli autovalori associati. Ordinando gli autovalori in

ordine decrescente viene mostrata la varianza spiegata da ogni componente principale, oltre che alla varianza spiegata cumulata.

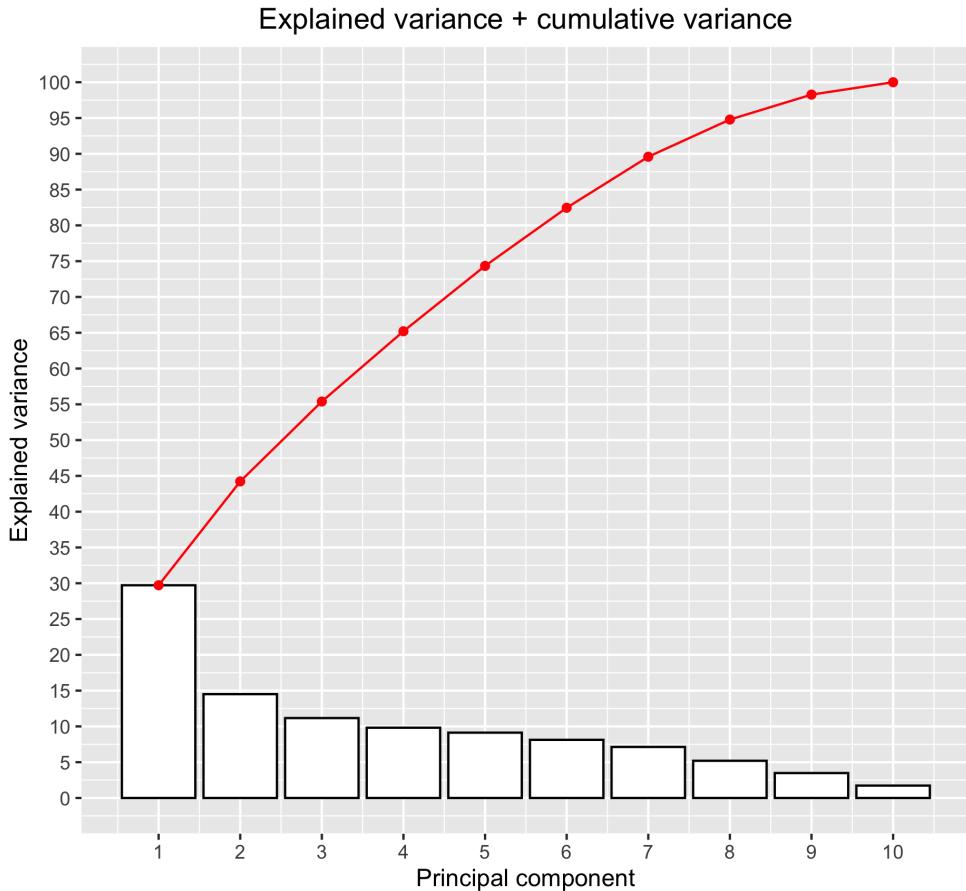


Figure 2.14: Varianza spiegata dalle componenti principali.

In tabella viene riportata la varianza spiegata cumulata:

| PC1   | PC2   | PC3   | PC4   | PC5   | PC6   | PC7   | PC8   | PC9   | PC10   |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| 29.72 | 44.22 | 55.39 | 65.20 | 74.33 | 82.45 | 89.58 | 94.78 | 98.26 | 100.00 |

Sulla base di questi dati è possibile notare come le prime 7 componenti principali spieghano circa il 90% della varianza totale. Per questo motivo si sceglie di proiettare il dataset dallo spazio originale nel nuovo sottospazio trovato con la PCA. Questo nuovo spazio ha dimensione pari a 7, ovvero il numero di componenti principali che si è scelto di utilizzare.

## 2.5 Scelta delle features

In questa sezione viene spiegata e giustificata la scelta di ogni feature.

### 2.5.1 Variabili scartate

Come si può notare dalla Table 2.1 non tutti le features vengono utilizzate. Le variabili scartate sono le seguenti:

- **id:** ID assegnato da spotify a ogni brano musicale. Questo campo non è rilevante per distinguere le due classi
- **name:** Titolo del brano musicale. Questa variabile non è utilizzata. Si potrebbero usare tecniche di NLP per tenere conto del sentimento del titolo brano e usare questa feature aggiuntiva.
- **release\_data:** L'anno di rilascio del brano viene utilizzato solo per la fase iniziale per scartare i brani troppo vecchi. Questa covariata non viene utilizzata per il training in quanto si ritiene che non sia influente per distinguere le due classi (Figure 2.3). Inoltre si vogliono costruire modelli che siano indipendenti dalla data di rilascio.
- **year:** Campo uguale a "release\_date". Nel dataset questa informazione è duplicata.
- **popularity:** Questo campo non viene utilizzato per i motivi discussi in section 2.4.1

### 2.5.2 Rappresentazione one hot encoding degli artisti

Dopo aver analizzato i wordcloud degli artisti emerge che questa informazione è utile per distinguere le due classi. Si procede quindi costruendo l'insieme degli artisti tra tutte le canzoni. Per ogni artista viene tenuto traccia del numero di brani musicali in cui l'artista compare.

La frequenza è necessaria perchè si vogliono tenere solo gli artisti che hanno una frequenza maggiore o uguale a 2. Questa operazione viene effettuata per non contare gli artisti completamente sconosciuti, in quanto si ritengono non influenti ai fine della classificazione. Inoltre a causa dell'undersampling è possibile avere diverse canzoni cantate da artisti "sconosciuti" e non avendo scelto una particolare strategia per l'undersampling adottiamo a questo punto l'utilizzo di un threshold.

Si procede quindi costruendo una matrice dove le colonne rappresentano ogni artista nell'insieme degli artisti, le righe invece gli individui nel dataset. L'entrata  $i, j$  della matrice può contenere il valore 1 o 0. Nel caso in cui il  $j$ -esimo artista ha cantato nella  $i$ -esima canzone allora l'entrata della matrice ha il valore 1, 0 altrimenti.

La matrice costruita costituisce la rappresentazione degli artisti nelle canzoni.

### 2.5.3 Variabili categoriche

Tutte le variabili categoriche vengono utilizzate ai fini della classificazione in quanto dalla descrizione della Table 2.1 risultano essere importanti nella distinzione delle classi.

#### Variabile key

La variabile key è di tipo factor i cui valori sono numeri da 1 a 11. Questa covariata indica la scala musicale utilizzata nella canzone. Dal momento che esiste una relazione d'ordine totale tra le note musicali, infatti si parla di scala musicale, per questa variabile viene effettuata l'operazione di cast a intero. Come secondo passo i valori vengono scalati in modo tale da essere compresi tra 0 e 1.

#### Variabile mode

La variabile mode è binaria, viene utilizzata impostando a 1 se vero 0 altrimenti.

#### Variabile explicit

Per la variabile explicit si procede analogamente a come fatto per la covariata mode.

### 2.5.4 Coordinate PCA

Le variabili numeriche non vengono direttamente usate per il training, piuttosto si utilizza il nuovo spazio ottenuto dalla PCA. Questo nuovo spazio ha dimensione inferiore rispetto al numero di variabili numeriche iniziale, ed è stato costruito proiettando gli individui del dataset di partenza nel nuovo spazio che ha come basi le 7 componenti principali.

### 2.5.5 Riassunto feature finali utilizzate

| FEATURE                  | # COVARIATE |
|--------------------------|-------------|
| One-hot-encoding artisti | 655         |
| Componenti principali    | 7           |
| Variabili categoriche    | 3           |
| Label                    | 1           |
|                          | 666         |

Table 2.3: Tabella riassuntiva features utilizzate.

## **Chapter 3**

# **Campagna sperimentale**

### **3.1 Approccio**

3.1.1 10-folds cross validation

3.1.2 Training set e test set

3.1.3 Model selection

Ottimizzazione iperparametri

Grid search

### **3.2 Misure di performance**

3.2.1 Accuracy

3.2.2 Precision, Recall e F-measure

3.2.3 Curve ROC

### **3.3 Support Vector Machine**

3.3.1 Kernel

### **3.4 Decision Tree**

### **3.5 Esperimenti**

3.5.1 Performance

### **3.6 Modelli a confronto**

## **Chapter 4**

# **Conclusioni**

Perchè le performance sono basee?

Spotify genera le caratteristiche con un algorimto quindi sicuramente un po' approssimato

E' effittivamente difficile capire se una canzone vincerà un premio, dipende da molti fattori