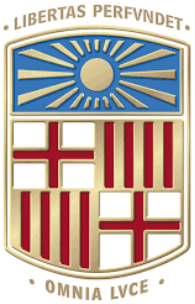
Musical analysis

Predicting the next hit

Pau Baldillou Salse
Pau Hernando Màrmol
Esther Ruano Hortoneda
Núria Torquet Luna

Marc Soler Bages



Bases de dades
Universitat de Barcelona
Maig de 2022

Contents

Abstract

This project aims at using statistics and a vast amount of information to find whether there is a relationship between the physical attributes of a song and its popularity. To do so, we make a thorough analysis of the available data and train a few machine learning and statistical models to predict a song's success.

In this paper, we present the work we have undertaken to accomplish our objectives and the conclusions we have extracted from the data. We include a description of the aforementioned objectives, which data we use and how we obtained it. Lastly, we include the statistical assessment performed, the models trained and their results.

1 Introducció

En aquest projecte farem servir els nostres coneixements en estadística i bases de dades per donar resposta a la hipòtesi:

"Els valors que descriuen físicament una cançó ens permeten predir el seu èxit". (1)

Per fer-ho, recollirem la informació dels atributs de milers de cançons i associarem, a cada una d'elles, tres mesures d'èxit; el primer pas, per tant, serà definir el que entenem per èxit. Organitzarem tota aquesta informació en una base de dades, i sobre ella farem diverses quèries que ens serviran per entrenar tres models de predicció.

A partir dels resultats obtinguts, extraurem conclusions que ens permetran acceptar o rebutjar la nostra hipòtesi.

2 Motivació

El motiu per triar aquest tema ha estat que és una afició comuna en tot l'equip, però de la qual tenim molt poc coneixement: la música. És per això que ens motiva a investigar i ens anima a creuar els coneixements de diverses àrees dels dos graus que cursem per aconseguir un projecte complet i transversal. En particular ens motiva especialment l'oportunitat d'utilitzar l'estadística que se'ns ha presentat de manera teòrica en un projecte real.

3 Objectius

Els objectius que ens hem marcat per aquest projecte són els següents:

i) Distingir els paràmetres que descriuen una cançó.

ii) Trobar eines per mesurar l'èxit d'una cançó segons diferents mètriques.

iii) Discernir quins paràmetres i en quina mesura influeixen en l'èxit d'una cançó segons cada mètrica.

iv) Fer un algorisme de predicció i posar-lo a prova.

v) Distingir si existeix una relació entre les diferents mètriques d'èxit.

4 Metodologia

Per aconseguir els objectius marcats farem el següent:

En primer lloc, donarem definicions per tots els conceptes que farem servir al llarg del treball.

Seguidament, obtindrem la informació sobre les cançons: farem servir el *Million Song DataSet*, un *dataset* generat per la Universitat de Columbia de 280 GB amb informació extensa sobre cançons produïdes entre els anys 1922 a 2011. Emmagatzema característiques de cada cançó i aquestes són precises, extenses i poc processades.

A continuació aconseguirem informació sobre l'èxit de les cançons de tres fonts diferents:

- Spotify; on mesurarem l'èxit pel nombre de reproduccions i com de recents són aquestes reproduccions en el temps.
- Llista Billboard *"Hot 100"*; on mesurarem l'èxit segons la posició màxima assolida, les setmanes transcorregudes en aquesta posició i la seva posició mitjana a la llista.
- Premis Grammy; considerarem exitoses les cançons guanyadores en les categories *"Best song"*, *"Best record"* dins el gènere *"General"* i les cançons *"Best song"* dins el gènere *"Dance/Electronic"*, *"Rock"*, *"R & B"*, *"Country"*, *"Rap"*, *"Improvised Jazz Solo"*, *"Gospel"*, *"Contemporary Christian"*, *"American Roots"*, *"Contemporary"*, *"Disco"* i *"Ethnic Or Traditional"*.

Compararem l'èxit de les cançons en les diverses fonts d'informació; i determinarem si hi ha alguna relació. Farem les següents operacions amb les tres fonts d'informació:

- Primerament, automatitzarem un programa que cerqui les 55.000 primeres cançons del Million Song Dataset a Spotify i extregui de cadascuna d'elles la seva popularitat. Amb aquesta popularitat assignarem l'èxit de Spotify. El subconjunt de cançons obtingut serà amb el que treballarem.

  També s'automatitzarà la cerca de les cançons dels 31 primers artistes que compleixin dues característiques claus: tenir moltes cançons a la base de dades i tenir un valor de popularitat elevat. Els motius de la tria els exposarem més endavant.
- Sobre aquestes cançons, un cop computat l'èxit a Spotify, també en calcularem l'èxit a la llista Billboard i als Grammy. Decidim fer-ho així perquè volem que totes les nostres cançons tinguin l'atribut d'èxit a Spotify, ja que és el més comú.
- Amb aquestes tres mesures d'èxit elaborarem, per separat, els tres models i els posarem a prova.
- Finalment extraurem conclusions de totes les proves realitzades i estructurarem els resultats.

El càlcul dels tres tipus d'èxits es computarà aplicant la mètrica definida a l'apartat 4.1 per assignar-los un valor numèric proporcional al seu èxit.

4.1 Definició de cançó exitosa

Definim tres mesures d'èxits diferenciades, que estudiarem i analitzarem per separat:

i) La popularitat a Spotify; hem modificat lleugerament el plantejament que havíem fet d'aquesta mesura d'èxit respecte la pràctica anterior: no hem pogut obtenir mitjançant l'*scraper* el nombre exacte de visualitzacions a Spotify; és per això que hem decidit fer servir un paràmetre que sí que podíem obtenir i que depèn directament del nombre de reproduccions: *popularity*. Aquest atribut es calcula a partir del nombre de reproduccions que té una cançó i com són de recents. Amb aquesta mesura pretenem tenir una visió general de com ha encaixat la cançó amb el públic a escala mundial.

ii) Llista Billboard *Hot 100* (*BH100* a partir d'ara); que ajunta les vendes (físiques i digitals), les reproduccions a la ràdio i a Internet als Estats Units. Aquesta ens donarà una visió més precisa en una demografia concreta.

iii) Posició obtinguda als premis Grammy: hem seleccionat els premis relatius a cançons, i hem ignorat aquells referents a l'artista, l'àlbum o el videoclip; han quedat les següents categories: *Record Of The Year, Song Of The Year, Best Dance/Electronic Recording, Best Rock Song, Best R&B Song, Best Rap Song, Best Country Song, Best Improvised Jazz Solo, Best Gospel Performance/Song, Best Gospel Song, Best Contemporary Christian Performance/Song, Best Contemporary Christian Song, Best American Roots Song, Best Contemporary Song, Best Disco Recording* i *Best Ethnic Or Traditional Folk Recording*. Representen valoracions per persones formades; contràriament als punts anteriors, que mostren l'opinió popular.

Definim una mètrica per cadascuna de les mesures d'éxit:

i) Èxit a Spotify: farem servir el valor de *popularity*, que està comprès entre 0 i 100. Hem canviat el plantejament respecte a l'entrega anterior a causa d'una consulta amb el Doctor Josep Vives, que ens va aconsellar utilitzar els diversos valors en lloc d'una mesura binària que representi èxit o no èxit.

ii) Èxit a la llista Billboard: definim la funció

$$\varphi(t) = (101 - m) + s \tag{2}$$

on t (de *track*) és una cançó donada, m és la màxima posició que la cançó t ha assolit a la llista BH100 i s el nombre de setmanes que la cançó ha passat a la llista. A la llista BH100 hi ha 100 posicions, per tant la idea de la fórmula 2 és invertir la màxima posició, de manera que com més alt a la llista, més alta sigui la puntuació i, seguidament, tenir en compte les setmanes que s'ha mantingut en la llista. Llavors catalogarem com a èxit les que superin el 60%, ja que considerem que el fet d'estar dins la llista ja comporta cert èxit.

iii) Èxit als premis Grammy: atorgarem un valor d'1 a guanyar un Grammy, i un valor de 0 a no guanyar-lo. Això es farà de forma acumulativa.

4.2 Model entitat-relació

En aquesta secció detallarem l'estructura que seguirà la nostra base de dades; que ha canviat lleugerament respecte a la primera entrega d'aquest treball, i justificarem aquestes modificacions. Els principals canvis que s'ha efectuat han sigut la reducció del nombre d'atributs en les taules Spotify i Million Songs. En realitzar la neteja de les taules per a reduir la seva mida s'ha acordat eliminar aquelles columnes que no fossin estrictament necessàries per al projecte. És per això que només han sobreviscut aquells atributs que aporten informació dels paràmetres físics de la cançó o permeten identificar-la. A continuació expliquem amb detall el paper de cada una de les entitats i el significat de cada un dels seus atributs:

1. Cançó: Hem considerat Cançó com a entitat principal de la qual hereten totes les altres. Això és degut a la manera com tractarem les dades; busquem poder relacionar fàcilment les quatre taules per a poder extreure les característiques físiques emmagatzemades a MillionSongs que presenten les cançons dins les taules Billboard, Grammy i Spotify. Per tant, hem considerat útil poder tenir característiques comunes de les cançons que comparteixin les quatre entitats. Aquests atributs, doncs, seran omesos en les entitats filles.

    - **titol:** Nom de la cançó.
    - **artista:** Intèrprets de la cançó. Una cançó pot estar interpretada per més d'un cantant, en conseqüència, l'atribut és multivaluat. La tupla (títol, artista) esdevindrà l'identificador únic de cada cançó. Aquesta serà la nostra clau principal.

El canvi efectuat respecte a l'entrega anterior ha sigut l'eliminació de l'atribut "id", corresponent a l'identificador de la cançó. Ens hem adonat que no ens resultava útil, ja que sovint teníem una mateixa cançó repetida en una taula (com és el cas de Grammy) i no li podíem assignar un id únic. És per això que s'ha decidit optar per la tupla (títol, autor), generalitzada a totes les taules.

2. Billboard: Aquesta entitat desarà la informació de les cançons que han estat a la llista Billboard, els atributs es corresponen als trobats al següent dataset, d'on hem llevat *artist* i *song*, ja que en el nostre cas els hereta de l'entitat Cançó.

   - **date:** Data en la qual es va publicar la cançó.
   - **rank:** Posició en la qual es trobava la cançó quan es va elaborar el *dataset* (06/11/2021).
   - **last-week:** Posició que va assolir la cançó l'última setmana en la llista.
   - **peak-week:** Posició que va aconseguir la cançó la setmana que es trobava més amunt en la llista.
   - **weeks-on-board:** Nombre de setmanes que va romandre en la llista.

   Aquesta taula s'ha mantingut intacta respecte a la pràctica anterior.

3. Grammy: Aquesta entitat desarà la informació de les cançons que es troben a la llista Grammy, els atributs es corresponen als trobats al següent dataset, d'on hem llevat *id*, *title* i *artist*, ja que en el nostre cas els hereta de l'entitat Cançó. Aquesta entitat és lleugerament diferent de les altres que hereten de Cançó: té una clau composta. Això es deu al fet que una mateixa cançó pot aparèixer en diverses categories i, per tant, l'identificador (*id*) pot donar lloc a diverses entrades; per això el complementem amb la categoria corresponent. Cal recordar que els premis Grammy només premien les cançons que han estat produïdes al llarg de l'any i, com a resultat, la tupla (*títol, artista*, categoria) ens serveix d'identificador unívoc.

   - **year:** Any en el qual es va presentar candidat al Grammy.
   - **category:** Categoria en la qual va guanyar la cançó. Nosaltres només considerarem les categories mencionades anteriorment per, com hem exposat, només tenir en compte l'èxit de la cançó i no el del videoclip, àlbum, etc.
   - **workers:** Artistes secundaris que han col·laborat en la peça. Com s'ha comentat prèviament, hi ha cançons en les quals hi col·laboren diversos artistes. És per això que és un atribut multivaluat.

   Els canvis efectuats respecte a l'entrega anterior han sigut l'eliminació dels atributs "published_at" i "updated_at", corresponents a les dates de publicació i modificació de les files del dataset. També s'han eliminat les columnes "img", corresponent a la url de la imatge de la cançó en la pàgina web dels Grammy i "winner", booleà que sempre pren el valor "true". Tots aquests atributs no aportaven informació sobre les cançons guanyadores de Grammy's.

4. Spotify: Aquesta entitat desarà la informació de les 55.000 primeres cançons que es troben al dataset Million Songs Dataset. El procediment per obtenir els atributs de cada cançó s'especifica en l'apartat 4.3 del document. El *dataset* resultant és el següent, d'on hem llevat *id*, *title* i *all_artists*, ja que en el nostre cas els hereta de l'entitat Cançó.

   - **popularity:** La popularitat de la cançó en aquell instant. El valor estarà entre 0 i 100, sent 100 el més popular. S'identifica amb el nombre de reproduccions i com de recents han sigut en el temps.

   Respecte l'entrega anterior s'han eliminat tots els atributs que no fóssin "popularity", degut a que en aquesta pràctica ja tenim clar que tots els atributs que ens retorna l'API d'Spotify estan massa tractats i únicament la popularitat ens retorna una mesura aproximada del nombre de visualitzacions de la cançó.

5. MillionSongs: Aquesta entitat contindrà la major part de la informació que tenim sobre cada cançó, els atributs es corresponen als trobats al *Million Song Dataset*, d'on hem llevat *artistName* i *title*, ja que en el nostre cas els hereta de l'entitat Cançó.

   - **danceability:** Estimació algorítmica que prediu quant ballable és la cançó. Tipus de dada = Float.
   - **duration:** Duració en segons de la peça musical. Tipus de dada = Float.
   - **end of fade in:** Segons abans de l'inici de la cançó. Tipus de dada = Float.
   - **energy:** Energia que transmet la cançó des del punt de vista de l'escoltador. Tipus de dada = Float.
   - **key:** To amb la que està composta la cançó. Tipus de dada = Int.
   - **loudness:** Mesura del volum de la cançó en decibels. Tipus de dada = Float.
   - **mode:** Mode de la cançó, major o menor. Tipus de dada = Int.
   - **start of fade out:** Temps en segons on comença la disminució gradual del senyal d'àudio que marca el final de la cançó. Tipus de dada = Float.
   - **tempo:** Tempo estimat en BPM. Tipus de dada = Float.
   - **time signature:** Nombre estimat de pulsacions per compàs. Tipus de dada = Int.

   Respecte a la pràctica anterior s'han eliminat tots els atributs que no estiguessin relacionats directament amb els paràmetres físics d'una cançó.

Finalment, explicarem les relacions entre les entitats:

Tal com s'ha comentat en l'entitat Cançó, aquesta exerceix d'herència per a les entitats filles Billboard, Grammy, Spotify i MillionSongs; per tant, totes elles hereten els atributs de Cançó, fet que ens permetrà relacionar les entrades de les taules més fàcilment.
Spotify manté una relació d'inclusió 0,1 a 1 amb MillionSongs per la manera com ha estat creada; ja que la taula Spotify s'ha poblat amb les 55.000 primeres cançons de la MillionSongs, totes les entrades de l'entitat Spotify es trobaran també a la taula MillionSongs, però no totes les cançons de MillionSongs es troben a Spotify.
Per altra banda, les entitats Billboard i Grammy estableixen relacions de cardinalitats lleugerament diferents amb MillionSongs. Cada entrada de l'entitat Billboard pot estar parametritzada o no en MillionSongs. I cada entrada de MillionSongs pot haver entrat o no a la llista Billboard. Per tant, la cardinalitat és 0,1 a 0,1. En canvi, les cardinalitats canvien en la relació entre Grammy i MillionSongs. Una entrada de Grammy pot estar o no parametritzada en la MillionSongs. Ara bé, una cançó de MillionSongs pot guanyar més d'un Grammy. Per tant la cardinalitat de Grammy a MillionSongs és 0,N a 0,1.

Per tancar l'apartat, explicarem com hem elaborat els models entitat-relació. Hem fet servir les següents definicions per entitats i relacions:

Definition 4.1 (Weak entity)**.** *En una base de dades relacional, una entitat feble és una entitat que no pot ser identificada de forma única pels seus atributs propis; per tant, requereix la utilització d'una clau forana en conjunció amb els seus atributs per a crear la clau primària. La clau forana és usualment clau primària de l'entitat amb la qual es relaciona.*

Definition 4.2 (Identifying relationship)**.** *La relació que associa l'entitat feble amb l'entitat propietària de la clau forana emprada com a part de la clau primària s'anomena "relació identificativa".*

Definition 4.3 (Non-Identifying relationship)**.** *En una relació no-identificativa, els atributs que formen part de la clau primària de l'entitat pare no consten en la clau primària de l'entitat filla; ara bé, poden ser atributs (tot i no ser primaris) de l'entitat filla.*

Per tant, observem que són febles totes les entitats del nostre model a excepció de cançó, que les relacions d'herència són *Identifying* i les d'inclusió són *Non-Identifying*.

Per facilitar la lectura, hem decidit incloure les claus en totes les entitats, tot i que fossin foranes.

En la implementació amb *MySQL*, que detallarem més endavant 4.3.2, hem triat les classes dels atributs segons els arxius CSV dels quals extrèiem.

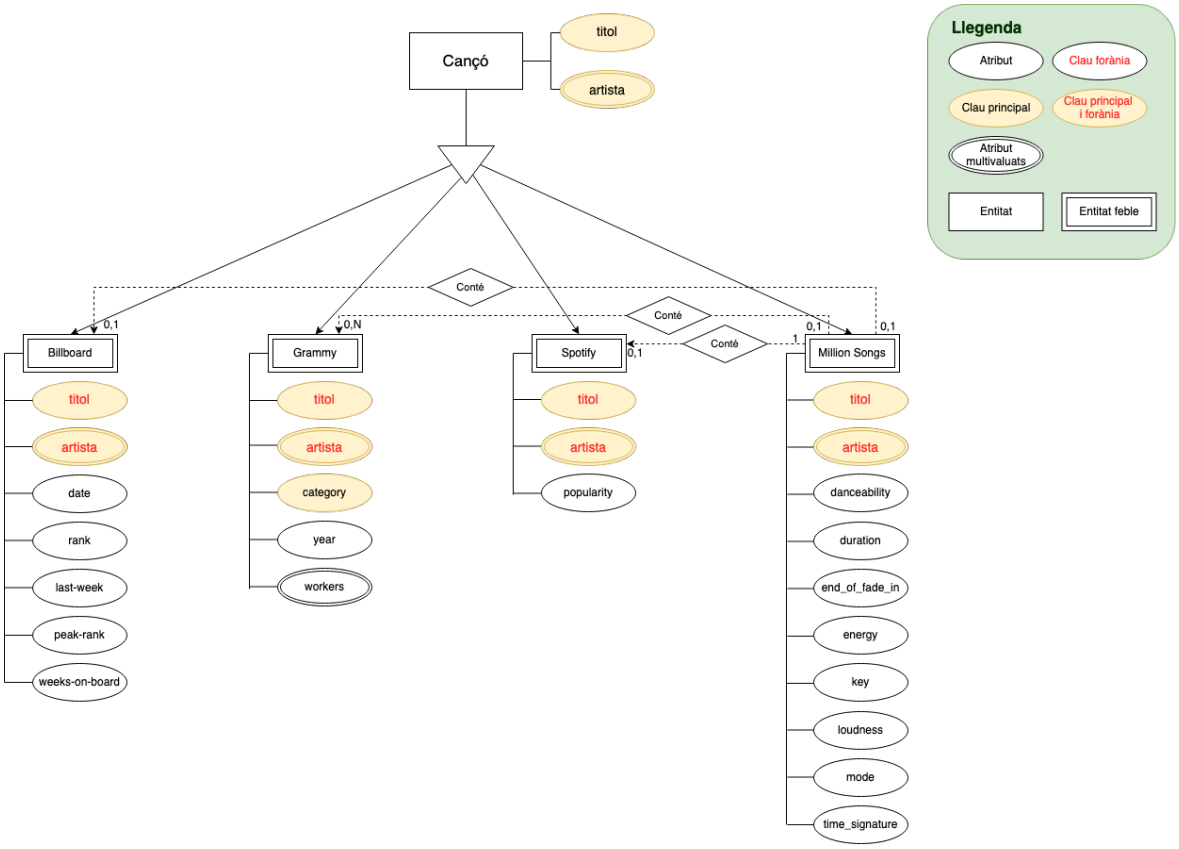


Figure 1: Model entitat relació

4.3 Dades

En aquest apartat, detallarem com hem obtingut les dades en primera instància, i com les hem modificat fent diverses tècniques per aconseguir arxius que ens permetessin manipular i entendre les dades a l'apartat 4.4.

Ho separem en dues parts: recollida de dades i tractament. A la primera part descriurem com hem obtingut els diversos arxius csv amb els quals hem començat a treballar. A la segona part explicarem com, a partir d'ells hem extret la informació que necessitàvem (paràmetres físics i mesures d'èxit). Això ho hem fet de dues maneres: mitjançant Python i SQL. Això es deu al fet que, per una banda, ens semblava natural fer servir Python, ja que part de la recollida s'ha fet amb Python i la part d'anàlisi també es fa en Python. Per altra banda, però, volíem aplicar els coneixements assolits a l'assignatura i, per tant, volíem crear la base de dades fent servir la consola de *MySQL Workbench*. Per evitar redundància, només es detallaran les operacions fetes amb SQL. Tot i així, adjuntem els *notebooks*, els .csv obtinguts i un *Readme* explicatiu de totes les operacions fetes anàlogament amb Python. El codi i fitxers .csv resultants tant amb l'execució dels *notebooks* i els *scripts* en Python com en SQL es poden trobar en el següent enllaç.

4.3.1 CSV: recollida de dades i neteja dels datasets

En aquesta secció explicarem com s'han obtingut els quatre datasets amb els que treballarem i els criteris que s'han seguit per eliminar-ne files i columnes.

Hem utilitzat diferents *datasets*, alguns extrets per nosaltres i altres que hem trobat a internet, amb l'objectiu d'augmentar la diversitat de les mesures d'èxit.

Cal precisar que s'ha limitat la recollida de dades a les llistes Spotify, Billboard i als premis Grammy entre els anys 1922 i 2011, perquè disposem dels paràmetres físics d'aquestes cançons gràcies al *Million Songs Dataset*.

En el primer cas, les dades s'han extret mitjançant un *scraper* en Python sobre l'API que ofereix Spotify. S'han programat dos *scripts*, *playlistCreator* i *popularityExtractor* que recorren sistemàticament les files del Million Song Dataset. PlaylistCreator cerca cançons amb la mateixa clau primària (títol i artista) en el cercador d'Spotify i aplicant l'algoritme de mètriques de subcadenes Levenshtein emmagatzema en una llista de reproducció de Spotify el resultat de la cerca que més s'aproxima a la clau primària real. Com que les llistes de reproducció no poden excedir les 1.000 entrades, per cada 1.000 cançons avaluades es crea una nova llista de reproducció i s'emmagatzemen totes en un compte d'Spotify *developer*. Un cop afegides les cançons a llistes de reproducció, amb l'*script popularityExtractor* es recorren aquestes llistes, s'accedeix a la informació que Spotify emmagatzema de cada cançó, i s'extreuen els atributs *title*, *artists* i *popularity*. Aquests es guarden en un nou csv, *spotifyPopularity*, que contindrà el títol de cada cançó, els artistes que l'han interpretat i la popularitat segons Spotify. S'ha decidit extreure un total de 55.000 cançons, ja que el procés d'extracció era lent i certes funcions de l'API generaven errors si s'executaven massa vegades durant un curt lapse de temps.

Per altra banda, s'han usat datasets externs que recullen informació més general. El dataset Billboard recull les cançons que han aparegut a la llista Billboard Hot 100 entre els anys 1958 (l'any de la seva creació) i 2021. A l'hora d'estudiar les dades que contenia el dataset, es va observar que una mateixa cançó podia tindre diferents entrades a la taula corresponents a les setmanes que havia romàs a la llista Billboard. Això es traduïa en un gran problema: claus primàries repetides. La solució va ser senzilla. Mitjançant un script en Python utilitzant la llibreria Pandas es va modificar la base de dades Billboard. Es va decidir recórrer sistemàticament la taula per files i eliminar les entrades que tinguessin una clau primària que hagués aparegut anteriorment. Així no alteràvem aquelles cançons sense repetir, i deixàvem a la taula l'entrada més recent d'aquelles cançons repetides, aconseguint així que l'atribut *weeks-on-chart* estigués actualitzat.

Mentre que el dataset Grammy emmagatzema les cançons guanyadores de premis Grammy entre els anys 1958 i 2019. A l'estudiar el dataset es va observar que les següents columnes no aportaven informació rellevant per al nostre projecte:

- **published_at**: data en la qual s'ha publicat l'entrada al dataset.
- **updated_at**: data en la qual s'ha modificat l'entrada al dataset.
- **img**: url de la imatge de la cançó en la pàgina web dels Grammy.
- **winner**: booleà que sempre pren el valor *true*, corresponent al fet que la cançó ha guanyat un Grammy.

És per això que es va decidir prescindir d'aquestes columnes. A l'hora de tractar amb els èxits presents al dataset Grammy va succeir el mateix problema que amb la llista Billboard; teníem files amb claus primàries repetides, corresponents a cançons que havien guanyat més d'un Grammy. En aquest cas la solució fou crear una nova columna, anomenada num_grammy, que emmagatzemés el nombre de Grammy's que havia guanyat la cançó. Un cop afegida aquesta columna es va iterar per les files i es van eliminar les cançons repetides, deixant-ne només una per cada representant amb el nombre de Grammy's guanyats actualitzat.

Finalment, en l'etapa d'anàlisis es va requerir un nou dataset que contingués la *popularity* de Spotify i els paràmetres físics de la cançó del MillionSongsDataset dels artistes amb el valor *artist_hotness*

(atribut del MillionSongsDataset) més alt. Mitjançant un senzill *script* en Python es van obtenir els 31 artistes amb major puntuació en l'atribut *artist_hotness*, es van extreure totes les seves cançons presents en el MillionSongsDataset i es van guardar en un llistat csv. Executant els dos script esmentats a l'inici d'aquesta secció, *playlistCreator* i *popularityExtractor*, es van cercar les cançons a Spotify i es va obtenir la popularity d'aquestes cançons. En aquest punt teníem un llistat amb el títol de la cançó, els artistes que la interpretaven i la seva popularitat, però faltaven els paràmetres físics. Recorrent a un *inner_join* entre aquest llistat i el MillionSongDataset amb la clau (títol, artista) es va acabar aconseguint un nou llistat amb els atributs físics i la popularitat de les cançons dels artistes més populars de la MillionSongs.

4.3.2 SQL: creació i ús de la base de dades

En aquest apartat explicarem com hem pres la informació dels arxius csv originals i l'hem incorporat a la base de dades. A més, també documentarem la primera neteja de la informació, molt semblant a la que acabem de descriure, i els càlculs de les mesures d'èxit. També detallarem com hem aconseguit obtenir, d'aquesta base de dades, nous arxius csv per poder alimentar els models, que coincideixen amb el resultat de l'apartat anterior.

Els primers passos a seguir són la creació de la base de dades usant la següent sèrie de comandes:

DROP DATABASE IF EXISTS projecteBD;
CREATE DATABASE projecteBD;
USE projecteBD;

Ara, ja podem crear les nostres taules per la base de dades: la sintaxi exacta de les comandes executades es pot llegir a l' arxiu SQL que adjuntem a la pràctica. A continuació en detallem només un esquema:

En primer lloc, ens assegurarem d'esborrar qualsevol taula residual que hagi pogut quedar d'altres iteracions escrivint
DROP TABLE IF EXISTS nomTaula;

A continuació hem d'especificar els atributs que tindrà la nostra base de dades, seguits del tipus; hem hagut de tenir cura de no fer servir paraules protegides per aquests: per exemple, hem hagut de canviar *key* per clau.

Finalment, i de manera opcional, indicarem quins formen la clau primària.

Hi ha dos arxius CSV que contenen més informació de la que necessitem pel nostre model entitat-relació, i és per això que creem dues taules *million_songs_brut* i *grammy_brut*, on desarem la informació com està al CSV; també la desarem sense modificar a les taules spotify i billboard, perquè ja tenen exactament la informació que desitgem. A més d'aquestes quatre taules, en creem dues més: *million_songs* i *grammy*, on emmagatzemarem la informació que hem especificat al model entitat-relació.

El següent pas és incorporar a la base de dades la informació que tenim en els arxius csv i, per fer-ho, caldrà desar els arxius csv a l'adreça que retorni l'execució de la comanda:

SHOW VARIABLES LIKE "secure_file_priv";

Aquesta ens retorna el directori des del qual la nostra base de dades ens permet carregar la informació. Com a alternativa, podríem desactivar *secure-file-priv*, però ens ha semblat més prudent procedir de la primera manera.

Un cop fet això, podem carregar la informació fent servir la comanda LOAD DATA INFILE. L'arxiu que més dificultat ha presentat ha estat el corresponent al million song dataset (*million_songs_brut*), a causa dels següents motius:

- Alguns camps estaven buits: això és un problema si el camp és de tipus *float* i retorna un error DATA TRUNCATED. Ho hem solucionat llistant, al final del *load*, els noms dels atributs que importarem i posant un @ davant aquells que són float i poden ser buits. A continuació hem fet anar la comanda SET per donar-los valor null si estan buits.

- Alguns camps incorporaven comes i caràcters no imprimibles. En primer lloc ho vam solucionar eliminant les comes dels camps com la localització i també eliminant els caràcters no imprimibles. Després ens vam adonar que afegir ENCLOSED BY "'' eliminava el problema, i hem optat per aquesta solució, ja que evita que modifiquem l'arxiu csv.

Idènticament s'han omplert les taules *grammy_brut*, *spotify* i *billboard*, es pot consultar a l'arxiu sql adjunt.

Observem, arribat aquest punt, que hi ha ocasions en què no podem fer servir la parella (artista, títol) com a clau primària al million_song_dataset, ja que hi ha artistes que han publicat la mateixa cançó en diversos àlbums i, en conseqüència, té diverses entrades. És per això que el següent pas a seguir és eliminar aquestes repeticions. Ho hem fet mitjançant una CTE (*common table expression*), que emmagatzema, per cada entrada d'una cançó, l'ordinal d'aquesta aparició: és a dir, el primer cop que apareix li associa un 1, el segon cop un 2, etc. Finalment, hem fet servir la comanda DELETE per eliminar aquelles amb ordinal major que 1.

El següent pas a fer consisteix a omplir les taules million_songs i grammy, a partir de les seves versions"brutes"; fer-ho ha requerit només un SELECT dels atributs explicitats al model E-R.

L'últim pas que necessitem fer és donar a cada cançó la seva puntuació de popularitat. Per fer-ho de manera compacta, crearem una taula resultat, que inclourà el nom de l'artista i de la cançó, els paràmetres físics i tres columnes al final, una per a cada mesura d'èxit, que hem anomenat exitS, exitB i exitG per a les mesures de Spotify, Billboard i Grammy que hem descrit anteriorment.

Aquesta taula l'obtenim fent un INNER JOIN sobre Spotify, ja que, com hem indicat, volem que totes les nostres cançons tinguin aquesta mesura d'èxit; i sobre aquest hem fet LEFT JOIN per aportar la informació d'èxit a la llista Billboard i als premis Grammy.

Finalment, vam fer una comanda tipus SELECT ... INTO OUTFILE per exportar la taula resultat.

A més, creiem que la variable "artista" influeix en l'èxit d'una cançó, i és per això que volíem fer un estudi local a alguns artistes; com detallem en l'apartat d'anàlisi. Per triar quins artistes analitzàvem, vam crear una taula artistesHotness que conté el nom de l'artista, el nombre de cançons de l'artista de les quals disposem, i els valors extrems que rep el seu atribut *artist _hotness*. Amb aquesta informació sabem, per una banda, de quins artistes tenim més cançons, cosa que ens pot donar millors resultats tenint en compte els models que fem servir, i, per altra banda, com és de popular l'artista. El motiu pel qual llistem els valors extrems és perquè la variable canvia amb el temps i, per tant, cada entrada on apareix l'artista pot tenir un valor diferent.

A la imatge següent es mostra una selecció de la taula, on ensenyem els artistes amb més de 80 cançons i una popularitat màxima major o igual a 0.8. Veient això hem decidit estudiar en detall Eminem i Michael Jackson. Els detalls d'aquesta tria es detallen a l'apartat d'anàlisi.

artist_name	numCancons	maxim	minim
Bruce Springsteen	132	0.838096	0.807587
Michael Jackson	112	0.911299	0.766545
U2	106	0.825639	0.357246
Green Day	104	0.812308	0.796139
Michael Bublé	98	0.840963	0.79299
T.I.	93	0.947858	0.865953
Daft Punk	88	1.02126	0.513054
Eminem	88	0.879237	0.80483
Coldplay	86	0.916053	0.872686
James Brown	86	0.911299	0.423621
Enrique Iglesias	85	0.814824	0.749062
Weezer	83	0.816313	0.776168
Nickelback	83	0.814335	0.725505
DJ Bobo	82	0.813005	0.393654

Figure 2: Llistat d'artistes amb més de 80 cançons i artist _hotness major que 0.8

4.4 Anàlisi de dades: estadística i models de predicció

Com es va descriure en la primera part del projecte, volíem reduir el problema a la classificació de cançons entre exitoses i no. Després d'aplicar els models que havíem plantejat, ens hem adonat que els resultats són bastant dolents i per això hem decidit realitzar els canvis que es descriuren a continuació.

Sóm conscients que hi ha artistes els quans produeixen cançons molt més exitoses que altres, per motius externs a la cançó. Les cançons menys populars de cantants molt famosos, acostumen a tenir més reproduccions i, en general, més èxit, que cançons de persones menys conegudes. Per això creiem que és necessari eliminar el "factor artista" del nostre anàlisi. Com s'ha comentat abans, arran del consell del Doctor Josep Vives, ens ha semblat que per aconseguir-ho el més adequat és buscar artistes molt prolífics i estudiar les diferències físiques entre les seves cançons més i menys exitoses.

Un altre canvi que hem realitzat, ha estat passar d'un problema de classificació a un de regressió. Creiem que els resultats binaris manquen precisió perquè hi ha diferents graus d'èxit. Això ens ha obligat a canviar alguns dels models. Hem mantingut Random Forest, però ara aplicant-lo a la predicció dels valors d'èxit en lloc de la seva classificació.

Per tal d'aplicar aquests canvis, ha estat necessari obtenir un nou *dataset* amb el màxim nombre de cançons possibles dels artistes escollits, així com les mesures d'èxit sense tractar. Això ha reduït dràsticament el nombre de dades amb les que podem treballar.

Finalment, hem decidit fer l'estudi i aplicar els models per a dos cantants: "Michael Jackson" i "Eminem". Per escollir aquests dos, els criteris utilitzats han estat:

1. Cantants amb més de 70 cançons en el nostre *dataset*.
2. Cantants de gèneres diferents.
3. Cantants destacats dels seus respectius gèneres.

Així doncs, escollint el "Rei del pop" i l'autor de "Rap god", la nostra intenció secundària és veure si les variables físiques de les que disposem afecten més un gènere que l'altre. Per aconseguir-lo, compararem els resultats que els models aconsegueixen. Tanmateix, som conscients que donada les noves restriccions en nombre de dades utilitzades i que només utilitzem un cantant per cada gènere, els resultats poden no ser significatius.

Finalment, estudiant les dades més a fons hem pogut veure que la mostra de cançons exitoses en els Grammy era massa petita. En els models aixo causava que sempre es predís que la cançó no seria exitosa amb un error molt baix. És per això que hem decidit eliminar els Grammys del nostre anàlisi.

4.4.1 Estadística i anàlisi prèvia a l'apliació dels models

En aquesta secció s'exposa com hem tractat les dades per tal d'extreure conclusions respecte a quines variables semblen més prometedores a l'hora de predir l'èxit d'una cançó. Hem separat el procés en els següents passos.

4.4.1.1 Estadístics bàsics i matriu de correlació Primerament, hem calculat els estadístics bàsics de cada variable: mitjana, desviació estàndard, mínim, primer, segon i tercer quartil, i màxim. Això ens ha permès eliminar dues columnes de la taula: *danceability* i *energy*, ja que aquestes dues columnes són idènticament 0 en totes les files. Aquests càlculs també ens han estat útils per obtenir una millor comprensió dels valors que pren cada variable. Els resultats es poden consultar a l'apèndix A, però no els comentarem, ja que no n'hem extret cap conclusió rellevant.

A continuació, hem creat la matriu de correlació, per tal de tenir una vista general de les relacions entre els diferents atributs. Hem observat que els atributs "*duration*" i "*start of fade out*" tenen una correlació de 0.9984. Una correlació tan elevada, significa que hi ha una relació lineal forta entre els

valors de les dues variables. Per tant, hem decidit eliminar *start of fade out*, que no ens aporta nova informació i així reduïm la dimensió de la nostra mostra. Reforçant la nostra decisió de no considerar els Grammys, podem veure a la matriu de correlació com l'èxit en aquests premis és el menys relacionat amb les altres variables entre els 3 èxits que hem considerat. Per tal de visualitzar aquests detalls, hem utilitzat un *heatmap*, que es mostra a la figura 3.

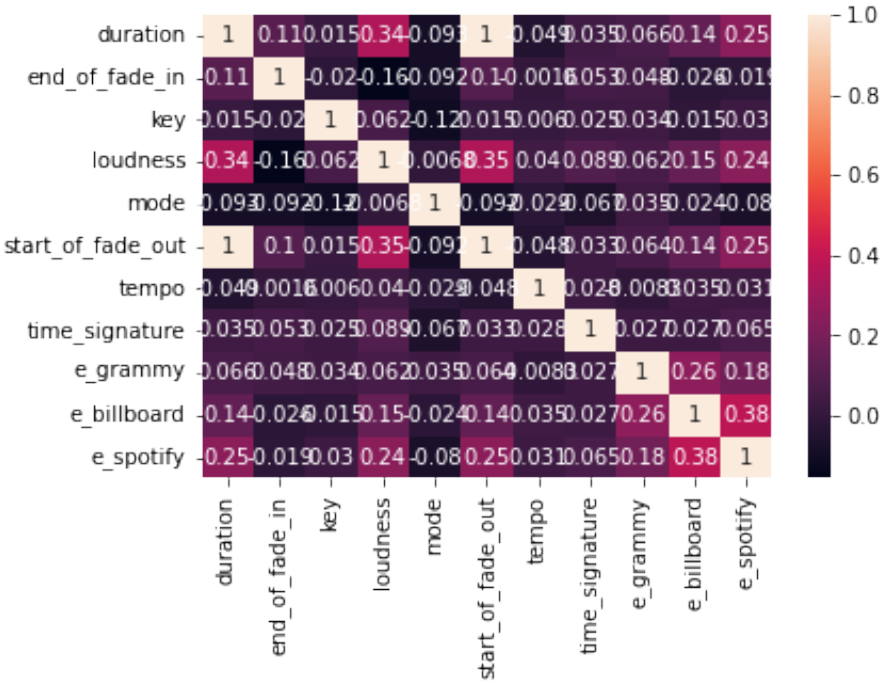


Figure 3: *Heatmap* generat a partir de la matriu de correlació que representa gràficament l'afinitat entre els parells de variables.

Aquestes dades les hem aconseguit pel *dataset* complet d'artistes coneguts. En l'apèndix B es poden trobar les dues matrius per Michael Jackson i Eminem. No hi ha cap diferència significativa respecte a la matriu de totes les dades, però volem fer les següents observacions:

- Tant en el cas de Michael Jackson com el d'Eminem, *duration* i *loudness* tenen una correlació elevada, de 0.54 i 0.6 respectivament. Això ens fa pensar que les dues variables estan més relacionades en aquests artistes que en general. Tot i això, no creiem que sigui una correlació prou elevada per a extreure aquesta variable dels models.
- Per Michael Jackson, sembla que la durada d'una cançó està lleument relacionada amb el seu èxit a Spotify.

4.4.1.2 Boxplots El aquest pas d'aquest procés, hem dibuixat boxplots de cada variable, i per cada tipus d'èxit. Per no interrompre el flux de lectura, aquí es farà una breu descripció dels resultats observats. A causa del gran nombre de boxplots realitzats, per facilitat del lector i comoditat de l'escriptor, s'adjunta a aquesta memòria un arxiu (boxplots.pdf) on es pot consultar el *notebook* utilitzat sense haver d'executar-lo.

Per cada variable, hem generat 7 boxplots. 3 parelles comparant les diferències entre els èxits i fracassos segons cada tipus d'èxit; i un boxplot de totes les dades. Això per cada un dels dos artistes seleccionats.

A causa que el nou *dataset* no conté els èxits categoritzats en èxit i fracàs, ha estat necessari generar una nova taula on sí que s'han guardat els èxits en mode binari, per tal de mantenir l'estructura d'un boxplot per descriure les cançons exitoses i un per les que no ho són.

Com es pot veure a la figura 4, i d'acord amb el que ja s'ha observat a la matriu de correlació, hi ha una diferència aparent entre la durada i la loudness de les cançons de Michael Jackson que són més exitoses i les que no ho són tant a Spotify.

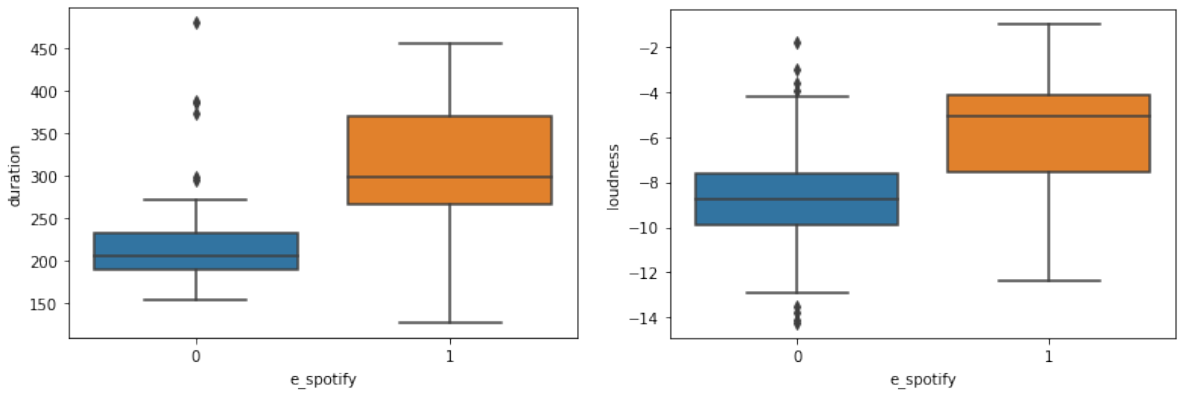


Figure 4: Boxplots de la durada i la loudness de les cançons de Michael Jackson segons si són èxits (1) o no (0).

Com es pot observar, la mitjana de les cançons classificades com a èxit és més elevada que la del conjunt de la resta de cançons per les dues variables. Això ens fa pensar que aquesta és una variable explicativa rellevant. Tanmateix, com es pot veure no forma una partició (hi ha cançons no exitoses de gran durada i cançons exitoses molt curtes), de manera que no és suficient per classificar les cançons de l'artista.

Com es pot veure a la figura 5, aquesta relació vista en el cas de Michael Jackson, no es replica per les cançons d'Eminem. El que sí que podem observar és com, en aquest cas, les cançons de menys durada podem dir amb certa seguretat que són no-èxits.

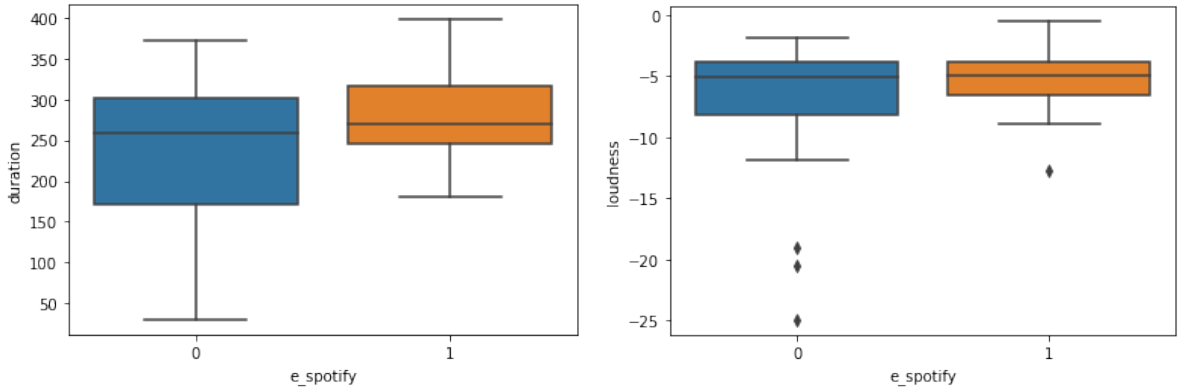


Figure 5: Boxplots de la durada i la loudness de les cançons d'Eminem segons si són èxits (1) o no (0).

A part del ja comentat, no hem detectat grans diferències entre les dues classes de cançons. L'únic que ens agradaria comentar, és que, en general, el rang de valors i la desviació típica de les cançons no exitoses són més elevades que per les que sí que ho són. Això es pot explicar pel fet que la mostra és més gran i, per tant, és d'esperar que hi hagi més variació.

En últim lloc, hem detectat que en aquelles variables on hi ha diferències, aquestes són més pronunciades pels èxits a Spotify que en els altres dos. És per aquest motiu, que d'aquí en endavant estudiarem més a fons només el comportament de les variables en els èxits de Spotify.

4.4.1.3 Visualització de les dades Finalment, per tal d'entendre millor la relació individual entre les variables explicatives i el que intentem predir, l'èxit de la cançó, hem dibuixat *scatter plots* amb totes les variables i hem realitzat regressió lineal com a mesura de relació entre les diferents variables i l'èxit a Spotify. Aquesta relació la quantifiquem amb l'error R square, que es detalla més endavant a la secció 4.4.2.1.

En la present secció comentarem només aquells resultats que considerem interessant, la resta de gràfics i el coeficient de determinació de la regressió lineal associada a cada variable, es poden trobar a l'apèndix C.

Els únics resultats que creiem rellevants són, de nou, la durada i loudness per Michael Jackson. En aquest cas els coeficients de correlació per la resta de variables han estat molt baixos. Per tant, les variables aïllades no són bones predient l'èxit de les cançons.

Com es pot veure a la figura 8, hi ha una lleu relació positiva entre la durada i la loudness d'una cançó amb el seu nivell d'èxit. Observem que la durada, que té un coeficient de determinació proper al 0.4 apunta a que com més duri una cançó sigui, més èxit tindrà. En canvi, la loudness es comporta de la manera contrària i com més sorollosa (decibels prenen valors més negatius), menys èxit.

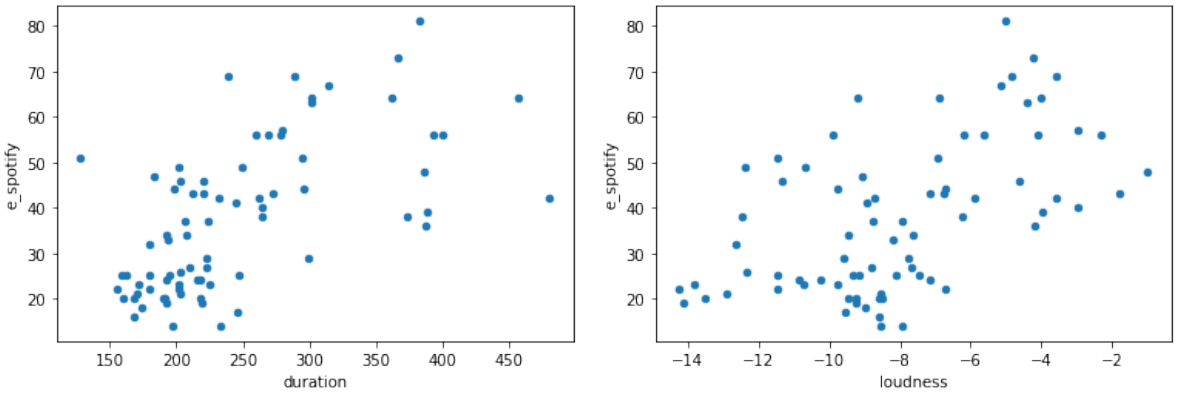


Figure 6: Scatter plots de duration i loudness contra l'èxit a Spotify per les cançons de Michael Jackson. Coeficient de determinació: 0.376775264254405 i 0.2988733762475402 respectivament.

Pensem que té sentit que les variables durada i loudness destaquin alhora perquè, com hem vist a la secció 4.4.1.1 hi ha una relació moderadament forta entre les dues. És per aquest motiu, que si una afecta l'èxit d'una cançó, l'altra també. De fet, també a la matriu de correlació es podia apreciar que les variables duration i loudness estàn relacionades amb l'èxit a Spotify, com mostra la figura 9.

4.4.2 Models de predicció

Un cop ens hem fet una idea genèrica de la mostra, el següent pas ha estat utilitzar aquesta per a entrenar els diferents models que havíem esmentat en l'anterior entrega. Hem utilitzat un total de 3 models per a cada mètrica d'èxit, tots ells programats en python:[1]

1. Support Vector Machine (SVM):

   Es tracta d'un algorisme que troba l'hiperplà que separa de millor manera dos tipus de dades diferents. En el nostre cas separarà les cançons exitoses de les que no ho són.

2. K-Nearest Neighbours (KNN):

   És un algorisme simple que es pot aplicar a problemes de classificació. És un model no paramètric que ens serveix per aconseguir la funció de densitat de les variables predictores (*features*) que pertanyin a una classe.

[1] *Github on està el codi.* `https://github.com/pbaldisa/musical-analysis`.

La idea bàsica és que si una entrada es troba rodejada per cançons exitoses, serà exitosa. Si, en canvi, està envoltada de cançons que no ho són, es determinarà que ella tampoc ho és.

k fa referència al nombre de veïns més propers que es tenen en compte a l'hora de determinar la classe d'un *input*.

3. Random Forest(RF):

   Utilitza un gran nombre d'arbres de decisions. Cada arbre de decisió dona una sèrie de prediccions. La predicció que surt més vegades és la que agafem com a bona. La clau és que els arbres tinguin molt poca correlació entre si, el que fa que el resultat sigui un model fiable. Per exemple, en el nostre cas, es podrien fer diferents arbres de decisió en funcions de paràmetres diferents (freqüència, duració, tempo...) i prendre la predicció més concurrent com a bona.

Per a entrenar i testejar aquests models hem fet servir la llibreria de Python "scikit-learn". Hem decidit entrenar els models amb el 75% de la mostra total i testejar-lo amb el 25% restant. En una primera iteració aplicàvem els models de **classificació** a datasets d'unes 15.000 cançons de diferents artistes, amb mesures d'èxit binàries. Els resultats han estat molt dolents, i hem hagut de replantejar l'aplicació d'aquests. Hem decidit utilitzar models de **regressió**, ja que considerem diversos tipus d'èxits. A més, aplicarem els models a cançons d'un mateix artista, perquè hem observat que el "factor artista" és una variable que, tot i no ser física, resulta molt important a l'hora de determinar l'èxit d'una cançó. A causa d'aquests canvis, hem hagut d'ignorar el model logístic, perquè utilitzava la funció sigmoid, que només serveix per a models de classificació.

4.4.2.1 Mètriques

Per a estudiar la qualitat dels models hem calculat tres mètriques:

1. **R square:** Serveix per a mesurar com de bé les variables dependents poden ser explicades pel model. Si el model treballa amb molts paràmetres pot ser que doni molt bons resultats per al subset d'entrenament però molt dolents per al de testeig. Aquesta mètrica ha de prendre valors entre 0 i 1, si dona un resultat negatiu significa que el model no és bo. L'equació per calcular aquest coeficient és:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \widehat{y_i})^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2}$$

2. **Mean Square Error (MSE)/Root Mean Square Error (RMSE):** És una mesura absoluta sobre com de bones són les prediccions. La fórmula per calcular-ho és

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \widehat{y_i})^2$$

   Per altra banda, el RMSE és l'arrel de MSE. S'utilitza més sovint que l'anterior perquè el valor s'ajusta millor a les mostres.

3. **Mean Absolute Error (MAE):** És molt similar al MSE però en comptes de sumar l'error al quadrat suma el valor absolut de l'error. És una representació més directe però més difícil de tractar. La seva fórmula és:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \widehat{y_i}|$$

La llibreria de Python utilitzada per a realitzar aquests càlculs ha estat sklearn.metrics .

Com s'ha esmentat anteriorment hem escollit a Eminem (69 cançons) com a representant del gènere rap i Michael Jackson (76 cançons) com a representant del pop. Hem aplicat tots els models per als èxits spotify i billboard, perquè la freqüència absoluta dels èxits seguint la mètrica dels grammys era

molt baixa. Així per a cada model, hem obtingut una gràfica on es mostra la relació entre el valor d'èxit real i l'obtingut després de la predicció, juntament amb el càlcul de les mètriques anteriorment esmentades.

Un exemple seria el següent:

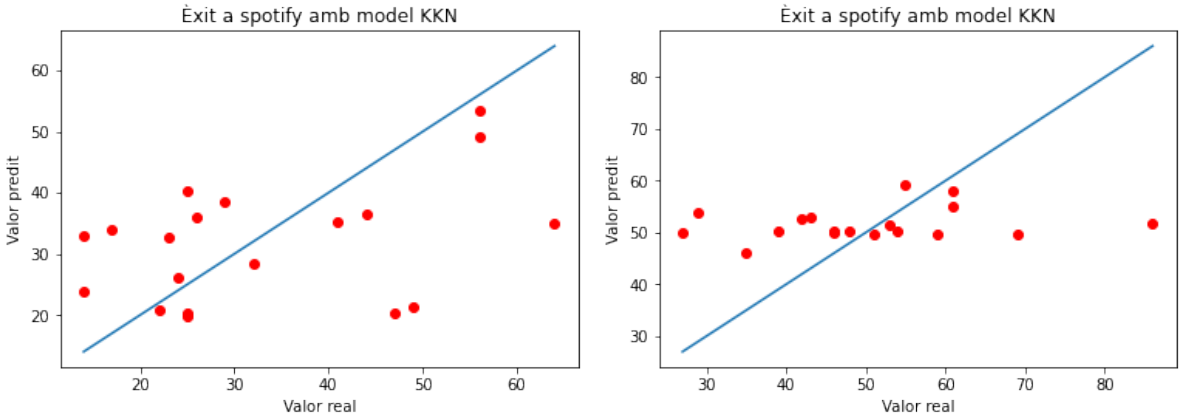


Figure 7: Representació gràfica de la relació entre èxits reals i predits d'Spotify per les cançons de Michael Jackson i Eminem respectivament.

Clarament, com més a prop els punts de la línia blava millor. Tot i així, els resultats no han estat gaire bons, ja que les mètriques són aquestes:

	Michael Jackson	**Eminem**
R^2 score	0.093429	0.03766
RSME	14.14295	13.66967
MAE	11.23157	10.21111

Si aquests resultats no són favorables, els de l'èxit a billboard ho són encara menys. Això es deu al fet que hi ha menys èxits sota aquesta mesura. Aplicant els mateixos models de regressió en ambdós casos s'han obtingut els següents resultats:

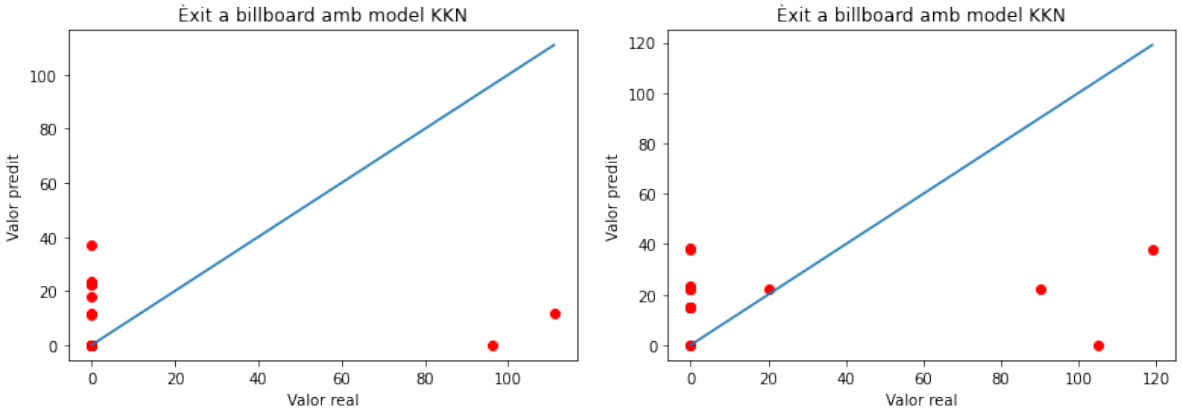


Figure 8: Representació gràfica de la relació entre èxits reals i predits de billboard per les cançons de Michael Jackson i Eminem respectivament.

Amb métriques:

	Michael Jackson	**Eminem**
R^2 score	-0.17265	-0.046607
RSME	34.49701	39.97888
MAE	17.96842	28.84444

4.4.3 Eines utilitzades

Tot el que s'explica en aquesta secció ha estat programat en Python. Hem utilitzat diverses llibreries per a tasques concretes, que es declaren a continuació.

Al llarg de tota l'anàlisi de dades, hem usat Pandas, que permet gestionar les dades guardant-les en *Dataframes*, que són taules. També permet algunes operacions que hem fet servir, com el càlcul de la matriu de correlació o el càlcul dels estadístics principals.

Per dibuixar la matriu de correlació hem fet ús Seaborn i Matplotlib. Aquestes dues llibreries també les hem fet servir per calcular i dibuixar els boxplots.

Per acabar, tots els models i les seves mesures d'èxit han estat implementats amb Sklearn.

5 Conclusions

Aquest projecte partia d'una sèrie d'objectius que hem intentat dur a terme. El principal d'ells era donar una resposta a la pregunta: "És possible determinar l'èxit d'una cançó a partir dels seus atributs físics?".

Al llarg del desenvolupament del treball, els propòsits han hagut de modificar-se per adaptar-se a allò que observàvem pel camí. Per començar, el primer objectiu era determinar quins paràmetres descriuen una cançó. A causa de la limitació en les dades de les quals disposàvem, hem estat forçats a treballar amb molt pocs paràmetres, de manera que hem hagut de treballar amb tots ells i no s'ha pogut comparar quins són els més idonis per la tasca en mà. En contraposició amb això, hem trobat eines per mesurar l'èxit de les cançons.

Un dels resultats més positius ha estat poder discernir quins paràmetres (dels disponibles) afecten l'èxit d'una cançó. Gràcies a l'estudi individualitzat que hem fet dels paràmetres per les cançons de Michael Jackson i Eminem, hem pogut veure que en els dos casos, tot i que en major mesura en el cas de Jackson, la variable que més afecta l'èxit (de mode lineal) és la durada de les cançons.

Per donar una resposta a la qüestió principal, hem partit d'una hipòtesi nul·la: H_0 : "els paràmetres físics no afecten l'èxit d'una cançó" i hem intentat refutar-la. És una pràctica comuna[2] que la hipòtesi nul·la sigui el contrari d'allò que realment volem demostrar (1). En el nostre cas, suposem que els paràmetres físics no impacten en l'èxit d'una cançó i s'intenta trobar proves per refutar-ho. Si això s'aconsegueix, es dona per cert que els paràmetres mencionats sí que afecten la popularitat de les cançons. Per tal de dur a terme aquest procés, hem entrenat 3 models estadístics i de machine learning, i hem utilitzat els seus resultats per decidir si disposem de suficient evidència per rebutjar la hipòtesi nul·la.

Malauradament, els resultats no han estat els esperats i, amb les dades de les quals disposem, no hem estat capaços de refutar la hipòtesi nul·la. Hem arribat a aquesta conclusió perquè els errors en tots els models entrenats han estat elevats i, per tant, s'adeqüen H_0. Això significa que com que no tenim prou proves per refutar la hipòtesi nul·la, no podem rebutjar-la i com a conseqüència no es pot afirmar que els paràmetres físics d'una cançó afectin el seu èxit. És important afegir que això no vol dir que

[2]Larry Wasserman. *All of statistics : a concise course in statistical inference*. New York: Springer, 2010, pp. 149–150. ISBN: 9781441923226 1441923225. URL: https://archive.org/details/springer_10.1007-978-0-387-21736-9/page/n383/mode/2up.

s'hagi demostrat que no afecten en absolut, sinó que no tenim proves per donar suport a la hipòtesi original.

A pesar d'això, sí que ha estat possible fer un anàlisi dels resultats obtinguts. De les tres mesures d'èxit que ens vam plantejar, Spotify ha estat la que millor funciona; i de tots els models utilitzats, K-Nearest Neighbours és el que ha donat millors resultats. També, l'èxit de les cançons de Michael Jackson ha estat més fàcil de predir, tot i que per una diferència marginal. Per això, i arran del fet que només hem realitzat l'estudi per un artista de cada gènere, no podem afirmar que el pop sigui més senzill de predir que el rap.

Finalment i en l'àmbit de coneixements personals que aquesta entrega ens ha permès assolir, creiem que ha estat una molt bona manera d'aprendre a treballar amb un gran nombre de dades. Això ho hem fet amb MySQL i Python, de manera que ha estat l'oportunitat perfecta per entrenar-nos amb aquestes tecnologies. A causa de les operacions que són necessàries per tractar les dades, pensem que hem pogut dur a la pràctica el que hem après al llarg del curs de bases de dades. Però no només això sinó que també hem tingut l'oportunitat de treballar amb conceptes estadístics més enllà de l'abast de l'assignatura d'estadística matemàtica que estem cursant paral·lelament.

A Estadístics

	duration	end_of_fade_in	key	loudness	start_of_fade_out	tempo	time_signature
count	1211.000000	1211.000000	1211.000000	1211.000000	1211.000000	1211.000000	1211.000000
mean	234.941919	1.148684	5.224608	-9.704753	225.665208	122.233003	3.697770
std	92.735189	2.318206	3.610923	5.513450	91.567689	32.333065	1.087641
min	10.605260	0.000000	0.000000	-34.198000	10.605000	0.000000	0.000000
25%	180.897505	0.089000	2.000000	-12.037500	173.212000	100.083500	4.000000
50%	229.067300	0.229000	5.000000	-8.238000	218.860000	119.945000	4.000000
75%	273.122810	0.877000	8.000000	-5.935000	262.313000	139.783000	4.000000
max	1400.267300	30.674000	11.000000	-0.558000	1389.592000	243.049000	7.000000

B Matrius de correlació

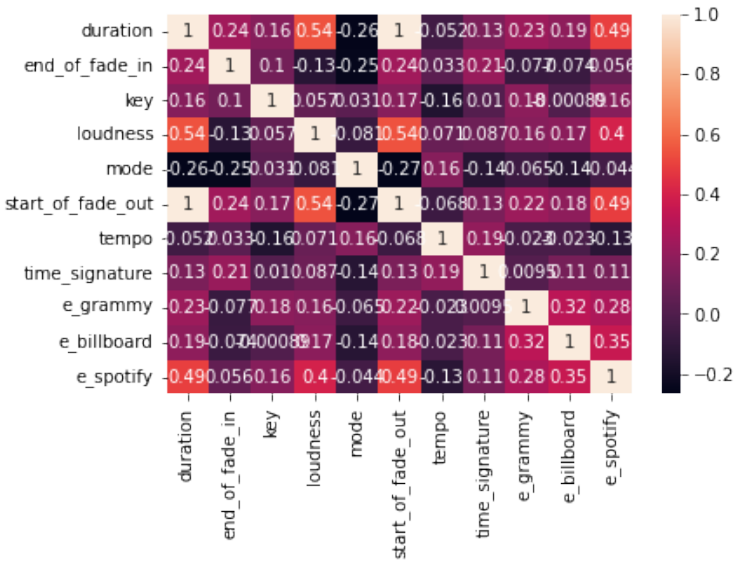


Figure 9: Matriu de correlació per les cançons de Michael Jackson

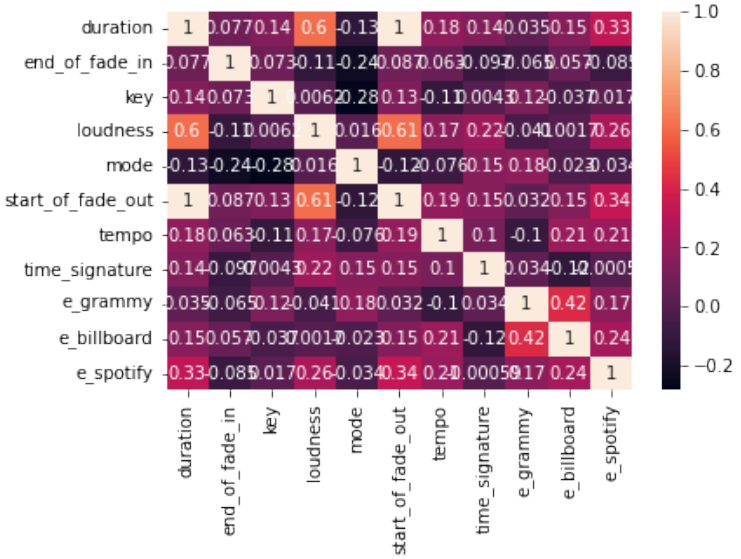


Figure 10: Matriu de correlació per les cançons d'Eminem

C Visualització de dades: scatter plots

C.1 Michael Jackson

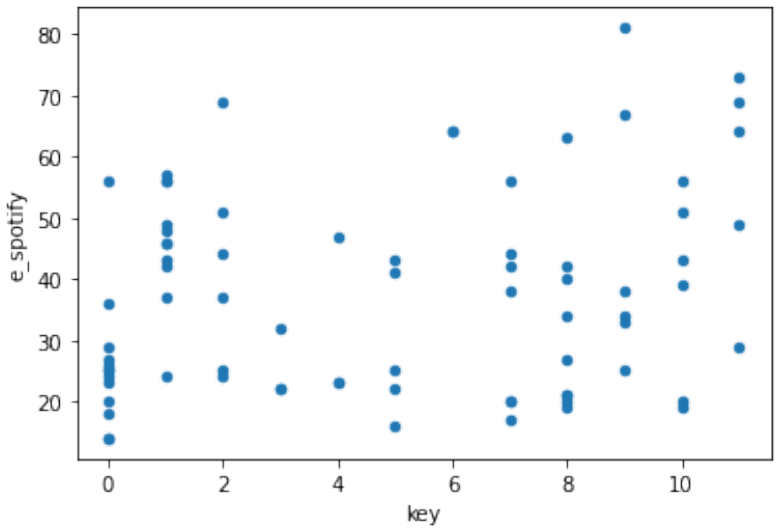


Figure 11: Scatter plot de key contra l'èxit a Spotify per les cançons de Michael Jackson. Coeficient de determinació: 0.04967052034446373.

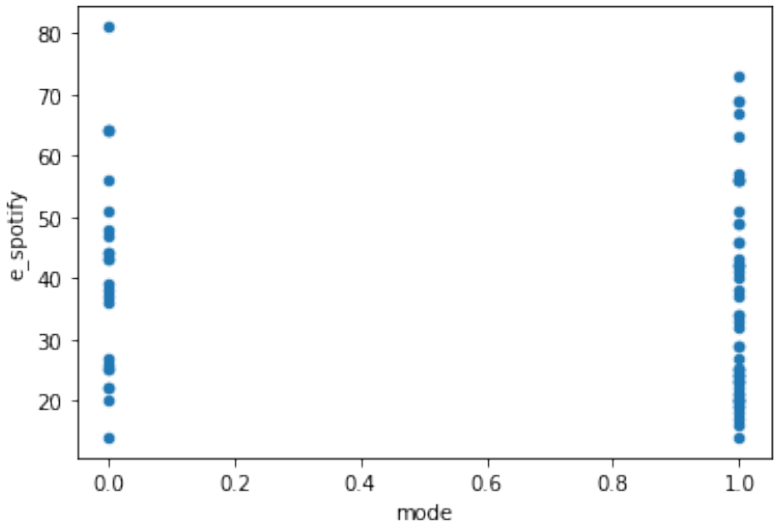


Figure 12: Scatter plot de mode contra l'èxit a Spotify per les cançons de Michael Jackson. Coeficient de determinació: 0.01919130578983752.

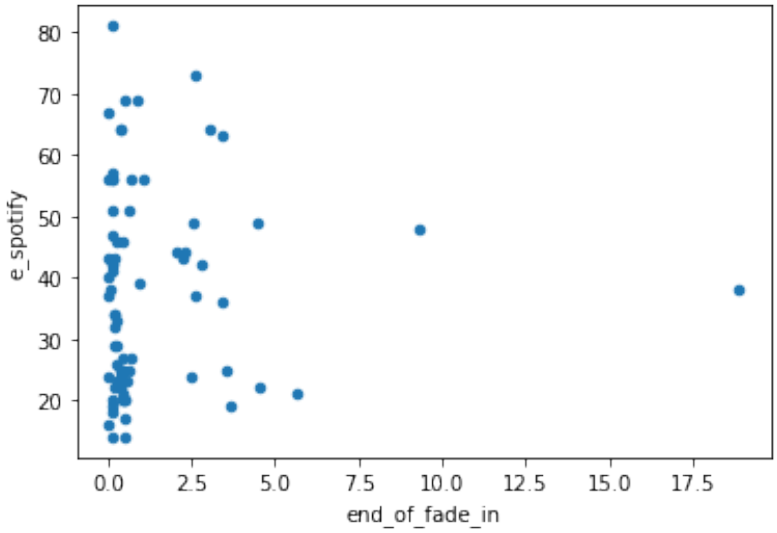


Figure 13: Scatter plot de end of fade in contra l'èxit a Spotify per les cançons de Michael Jackson. Coeficient de determinació: 0.0027205758047210704.

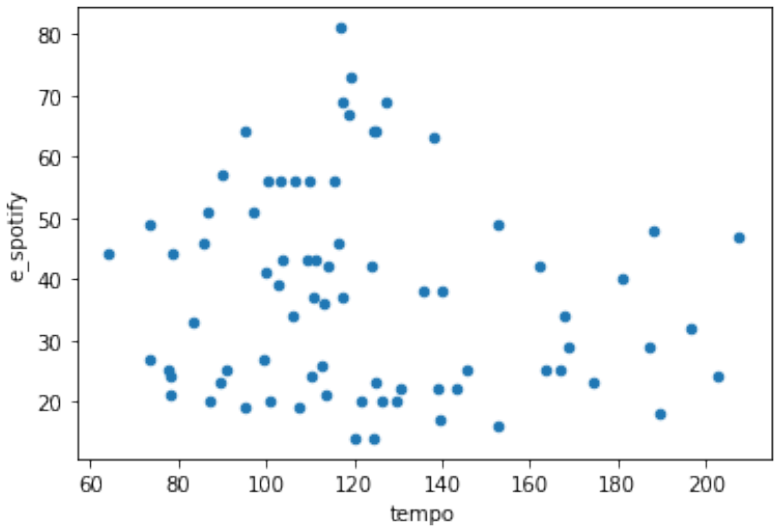


Figure 14: Scatter plot de tempo contra l'èxit a Spotify per les cançons de Michael Jackson. Coeficient de determinació: 0.016359895928788837.

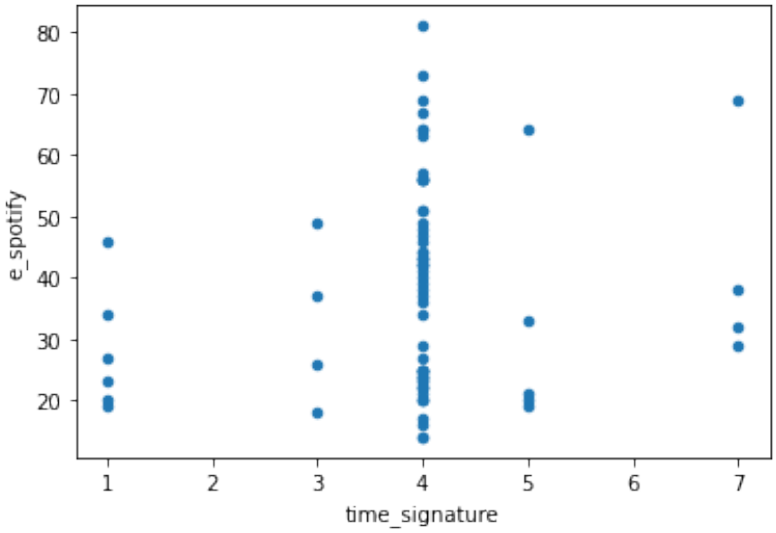


Figure 15: Scatter plot de time signature contra l'èxit a Spotify per les cançons de Michael Jackson. Coeficient de determinació: 0.022134229508029124.

C.2 Eminem

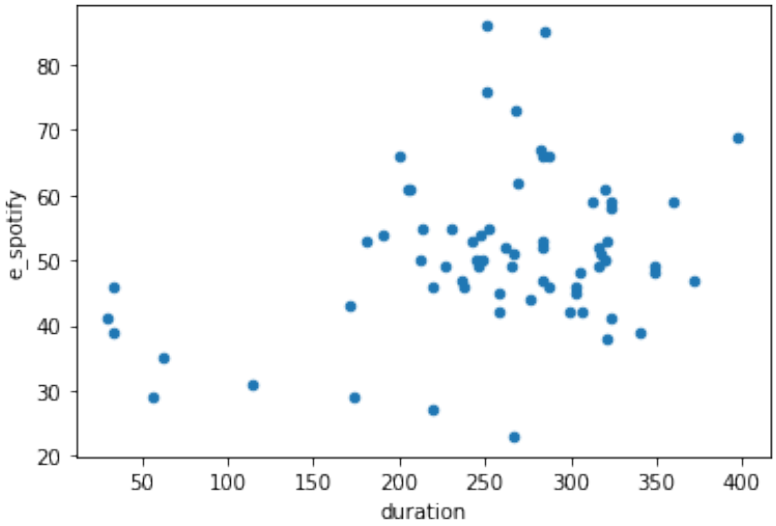


Figure 16: Scatter plot de duration contra l'èxit a Spotify per les cançons de Eminem. Coeficient de determinació: 0.09550063238110562.

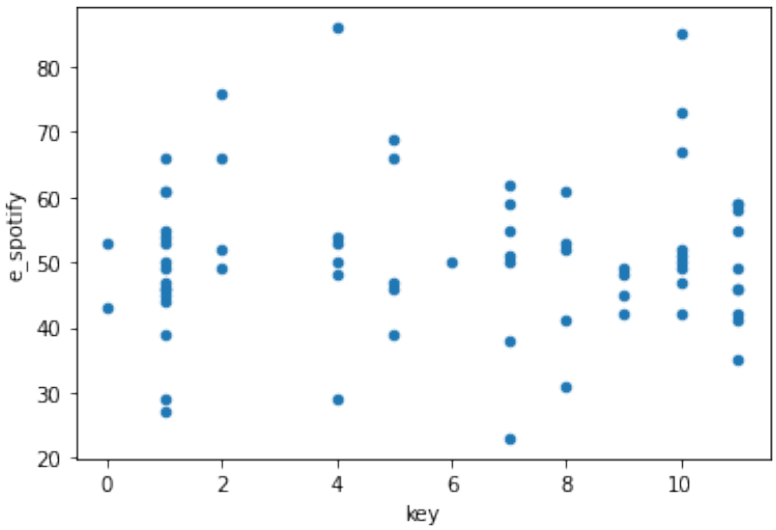


Figure 17: Scatter plot de key contra l'èxit a Spotify per les cançons de Eminem. Coeficient de determinació: 0.0005529240009249925.

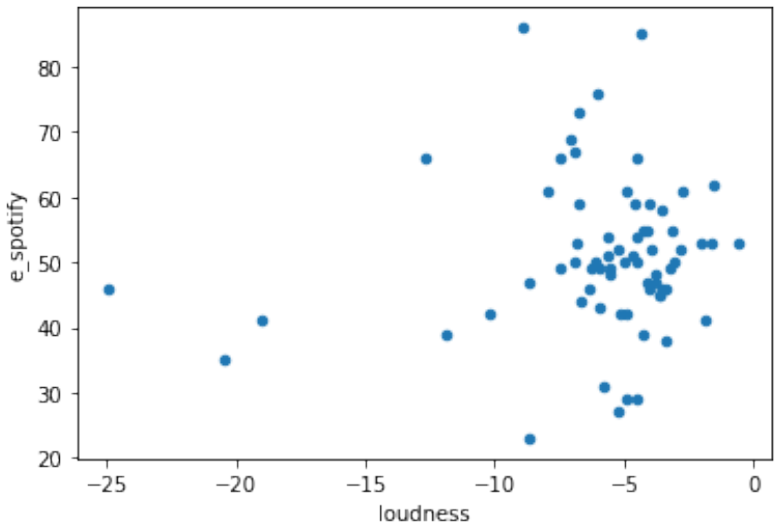


Figure 18: Scatter plot de loudness contra l'èxit a Spotify per les cançons de Eminem. Coeficient de determinació: 0.01107556778136487.

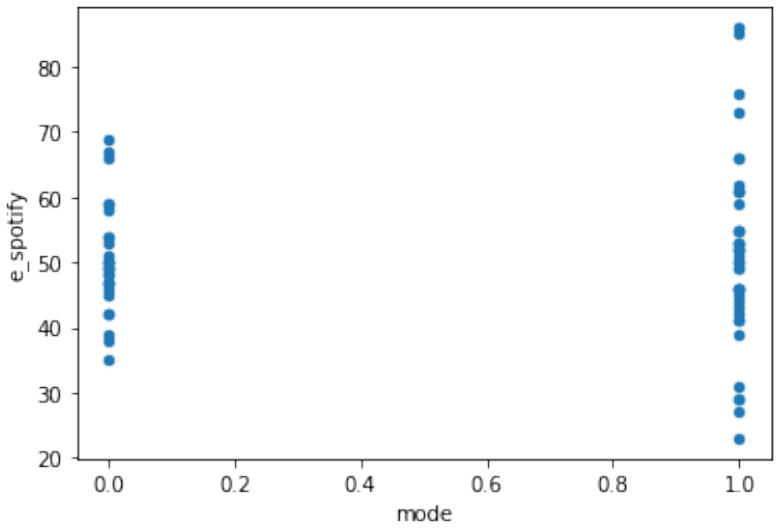


Figure 19: Scatter plot de mode contra l'èxit a Spotify per les cançons de Eminem. Coeficient de determinació: 0.0013779077523514704.

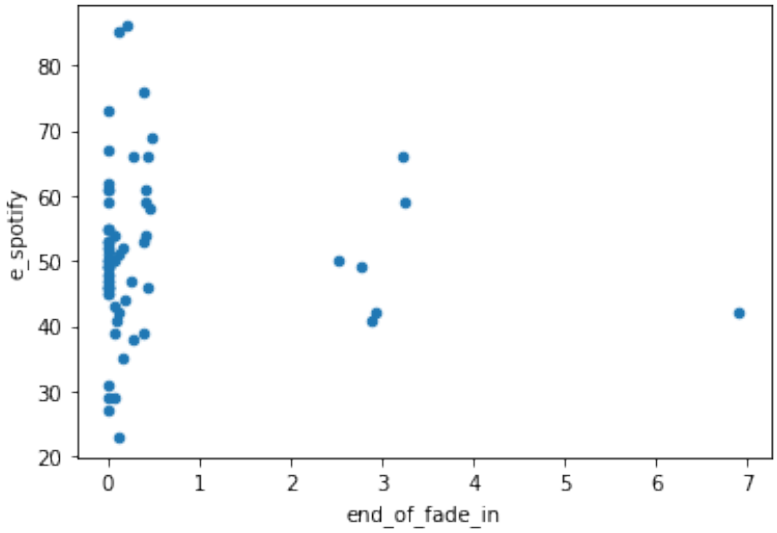


Figure 20: Scatter plot de end of fade in contra l'èxit a Spotify per les cançons de Eminem. Coeficient de determinació: 0.000415045236148992.

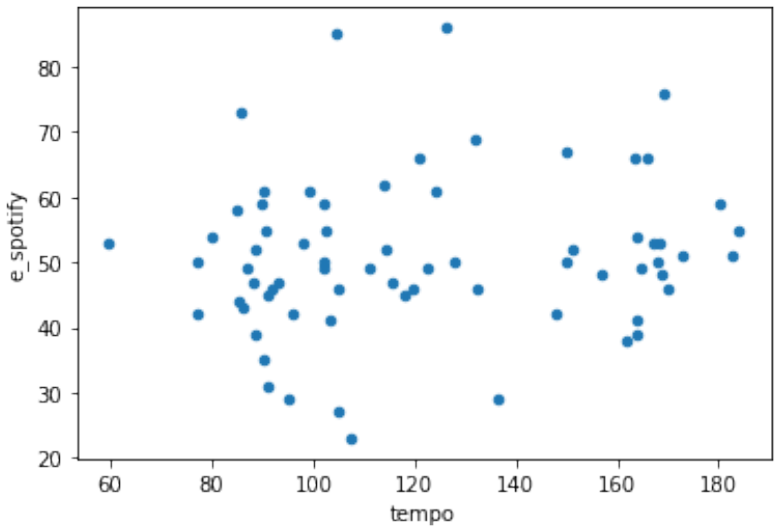


Figure 21: Scatter plot de tempo contra l'èxit a Spotify per les cançons de Eminem. Coeficient de determinació: 0.017139995721213053.

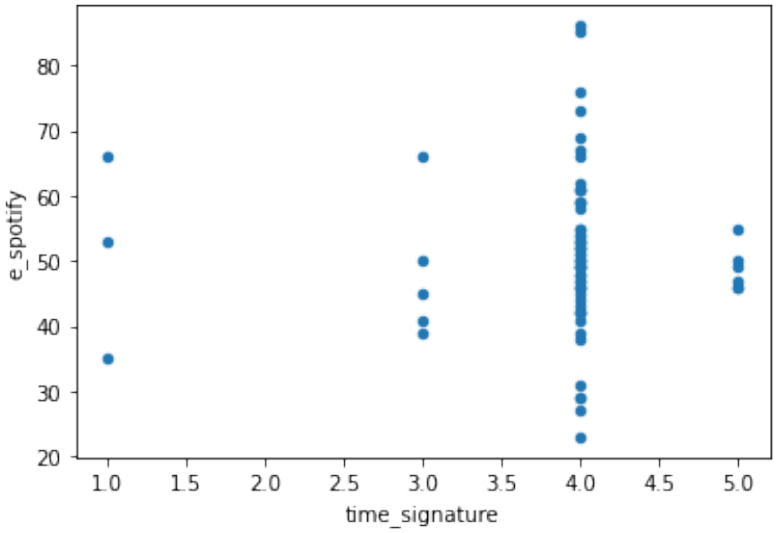


Figure 22: Scatter plot de time signature contra l'èxit a Spotify per les cançons de Eminem. Coeficient de determinació: 1.8361191200577665e-05.