Proiect Analiza și Exploatarea Datelor

Data Frame: Song data

Profesor coordonator:

Profesor universitar

Studenți:

Ștefan Amalia-Ioana

IE GRUPA 3

**PROIECT EXAMEN ANALIZA ȘI EXPLOATAREA DATELOR**

CONȚINUTUL

* Numărul de înregistrări
* Daca exista valori null + modul în care tratați aceste valori
* ?medii / cvartile/deviație standard / mediana pt var. numerice
* (<-- și ce sunt astea <-- )
* dacă exista var.categoriale + care sunt și de ce sunt așa
* care credeți că sunt cele mai importante variabile independente și de ce
* dacă transformați var. categoriale în var dummy
* corelații + explicații pentru corelații - între var independente și var. dependenta și doar între variabilele independente
* multicoliniaritate - dacă exista și ce se întâmpla cu variabilele respective
* regresie + interpretare

**INȚELEGEREA BAZEI DE DATE SONG\_DATA**

DANCEABILITY

Dansabilitatea descrie cât de potrivită este o piesă pentru dans pe baza unei combinații de elemente muzicale, inclusiv tempo, stabilitatea ritmului, puterea ritmului și regularitatea generală. O valoare de 0,0 este cel mai puțin dansabilă și 1,0 este cea mai dansabilă.

ENERGY

Energia este o măsură de la 0,0 la 1,0 și reprezintă o măsură perceptivă a intensității și activității. În mod obișnuit, piesele energetice se simt rapide, zgomotoase și gălăgioase. De exemplu, death metalul are o energie mare, în timp ce un preludiu Bach are scoruri scăzute pe scară. Caracteristicile perceptuale care contribuie la acest atribut includ intervalul dinamic, volumul perceput, timbrul, rata de debut și entropia generală.

KEY

Cheia generală estimată a piesei. Numerele întregi se mapează la înălțimi folosind notația standard Pitch Class. De exemplu. 0 = C, 1 = C?/D?, 2 = D și așa mai departe. Dacă nu a fost detectată nicio cheie, valoarea este -1.

LOUDNESS

Volumul total al unei piese în decibeli (dB). Valorile sonore sunt mediate pe întreaga piesă și sunt utile pentru a compara intensitatea relativă a pieselor. Loudness este calitatea unui sunet care este corelația psihologică primară a forței fizice (amplitudine). Valorile tipice variază între -60 și 0 db.

AUDIO MODE

Modul indică modalitatea (major sau minor) a unei piese, tipul de scară din care este derivat conținutul său melodic. Major este reprezentat de 1 și minor este 0.

SPEECHINESS

Speechiness detectează prezența cuvintelor rostite într-o piesă. Cu cât înregistrarea este mai exclusiv de tip vorbire (de exemplu, talk-show, carte audio, poezie), cu atât valoarea atributului este mai aproape de 1.0. Valorile de peste 0,66 descriu piese care sunt probabil formate în întregime din cuvinte rostite. Valorile între 0,33 și 0,66 descriu piese care pot conține atât muzică, cât și vorbire, fie în secțiuni, fie în straturi, inclusiv cazuri precum muzica rap. Valorile sub 0,33 reprezintă cel mai probabil muzică și alte melodii care nu sunt de tipul vorbirii.

ACOUSTICNESS:

O măsură de încredere de la 0,0 la 1,0 a faptului că pista este acustică. 1.0 reprezintă o încredere ridicată că pista este acustică. Distribuția valorilor pentru această caracteristică arată astfel:

INSTRUMENTALNESS

Prezice dacă o piesă nu conține voce. Sunetele „Ooh” și „aah” sunt tratate ca instrumentale în acest context. Melodiile rap sau vorbite sunt în mod clar „vocale”. Cu cât valoarea instrumentalității este mai aproape de 1,0, cu atât este mai mare probabilitatea ca piesa să nu conțină conținut vocal. Valorile de peste 0,5 sunt menite să reprezinte piese instrumentale, dar încrederea este mai mare pe măsură ce valoarea se apropie de 1,0. Distribuția valorilor pentru această caracteristică arată astfel:

LIVENESS

Detectează prezența unui public în înregistrare. Valorile mai mari ale intensității reprezintă o probabilitate crescută ca piesa să fie interpretată live. O valoare mai mare de 0,8 oferă o probabilitate mare ca piesa să fie live.

VALENCE

O măsură de la 0,0 la 1,0 care descrie pozitivitatea muzicală transmisă de o piesă. Piesele cu valență ridicată sună mai pozitiv (de exemplu, fericit, vesel, euforic), în timp ce piesele cu valență scăzută sună mai negativ (de exemplu, trist, deprimat, furios). tempo: tempo-ul total estimat al unei piese în bătăi pe minut (BPM). În terminologia muzicală, tempo-ul este viteza sau ritmul unei piese date și derivă direct din durata medie a ritmului.

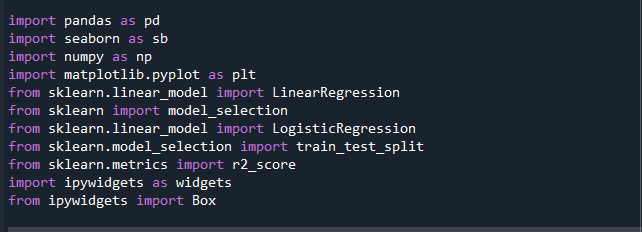
DURATION\_MS

Durata piesei în milisecunde.

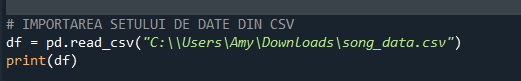
TIME\_SIGNATURE

O semnătură de timp totală estimată a unei piese. Semnătura de timp (metrul) este o convenție de notație pentru a specifica câte bătăi sunt în fiecare măsură (sau măsură).

Pentru început am importat bibliotecile relevante și am ales să folosim **pandas** și **sklearn** care sunt considerate biblioteci standard în astfel de scopuri, împreuna cu matplotlib.pyplot și sklearn, .liner\_model, .model\_selection și . metrics. De asemenea, am folosit **ipywidgets** pentru a oferi o interfață interactivă și elegantă pentru testarea predicțiilor modelelor.



Apoi, încărcarăm fișierul .xslx pe care l-am primit cu denumirea de „song data” într-un cadru de date și am rulat câteva statistici descriptive.

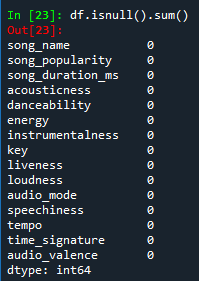




Cu comanda **df.head()** am afișat primele 5 înregistrări din setul de date, aceste înregistrări având un set de valori normale, fără să exista o anomalie în acestea.



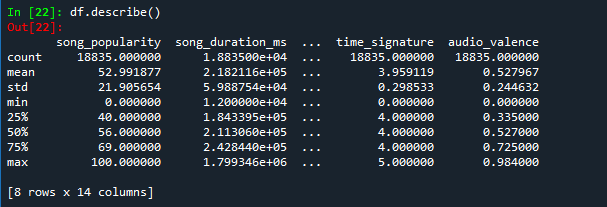
Prin comanda df.isnull(), specializata în căutarea valorilor din setul de date care nu au o valoare înregistrata, am verificat dacă exista în acest set de date valori null, în urma executării acesteia am observat că setul nostru de date este complet, fără valori null.



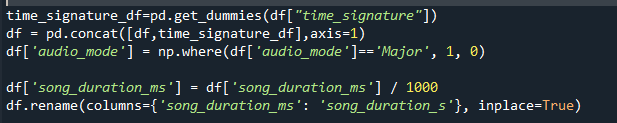


Pandas pd.isnull() returnează un set de date cu valorile booleene True și False spunând dacă valoarea lipsește. Apelarea sum() de două ori pe aceasta ne oferă un număr total al tuturor valorilor lipsă din setul de date. Dacă l-am folosi o singură dată, am obține o sumă de valori lipsă pentru fiecare coloană. Nu există valori lipsă într-un set de date format din peste 18.835 de rânduri. Înseamnă că nu este rău, iar concluziile noastre vor fi valabile.

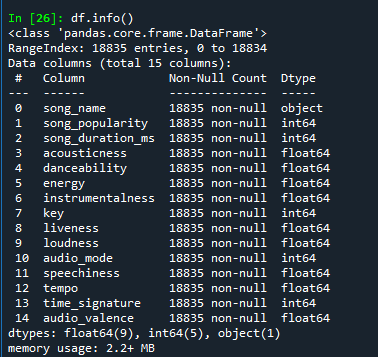
Statisticile descriptive, realizate prin comanda „df.describe()”, ne-au permis să detectam eventualele anomalii și datele lipsă. Am întâlnit și câteva modificări pe care am fost interesați să le efectuam, cum ar fi standardizarea duratei melodiilor de la milisecunde la secunde și ignorarea variabilelor independente cu un impact minim asupra studiului, în plus, am observat că nu exista cazuri de repetare a melodiilor.



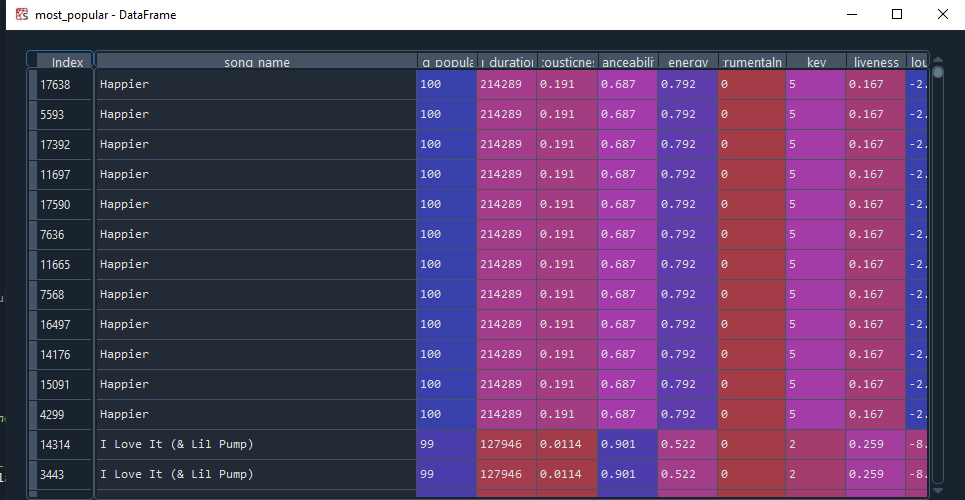
Această secvența de cod realizează următoarele transformări asupra unui dataframe numit "df" prin crearea unui nou data frame numit „time\_signature” în care se vor introduce variabilele dummies și care ulterior se vor concatena cu variabilele din data frame-ul „df”. Realizează conversia coloanei "audio\_mode" astfel încât valorile "Major" sa devina 1 si celelalte 0, ulterior înmulțind valorile din coloana "song\_duration\_ms" cu 1/1000 pentru a le transforma în secunde. După ce a realizat toate cele de mai sus acesta redenumește coloana "song\_duration\_ms" în "song\_duration\_s".

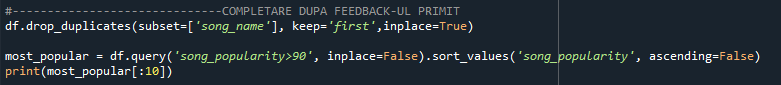


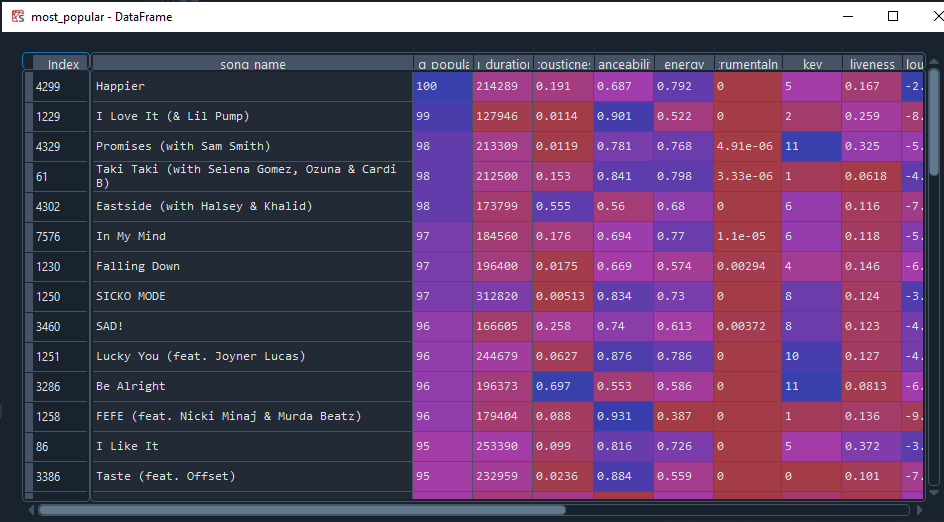
Prin **df.info()** se afișează tipul de date pentru fiecare variabila, această comanda fiind utila in realizarea proiectului .



Statisticile descriptive mi-au permis să detectez eventualele anomalii și datele lipsă. Am întâlnit și câteva modificări pe care am fost interesat să le efectuez, cum ar fi standardizarea duratei melodiilor de la milisecunde la secunde și ignorarea variabilelor inutile. În plus, am verificat dacă există mai multe cazuri ale unor melodii. Odată cu verificarea top celor mai ascultate melodii, am realizat că setul de date conține melodii dublicate, așa ca am încercat sa le eliminam.

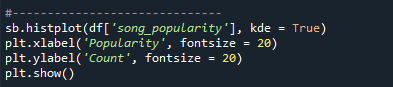






În urma procesului de curățare a datelor, numărul de melodii a fost redus la 13070. Sperăm că această modificare a redus un potențial părtinire care ar putea rezulta atunci când aceeași melodie apare atât în seturile de antrenament, cât și în seturile de testare. În topul celor mai ascultate melodii intră doar 55 de piese, cele care sunt cuprinse între 90 și 100 privind song\_popularity.

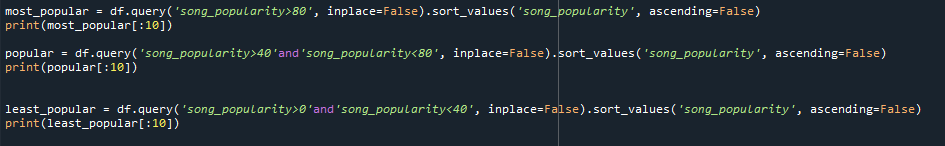
Care sunt cele mai populare melodii în acest moment? Pentru a verifica acest lucru, haideți să folosim o interogare grozavă a funcției pandas (). Aceasta este o funcție de filtrare care permite selecția și filtrează coloanele unui DataFrame cu o expresie booleană.

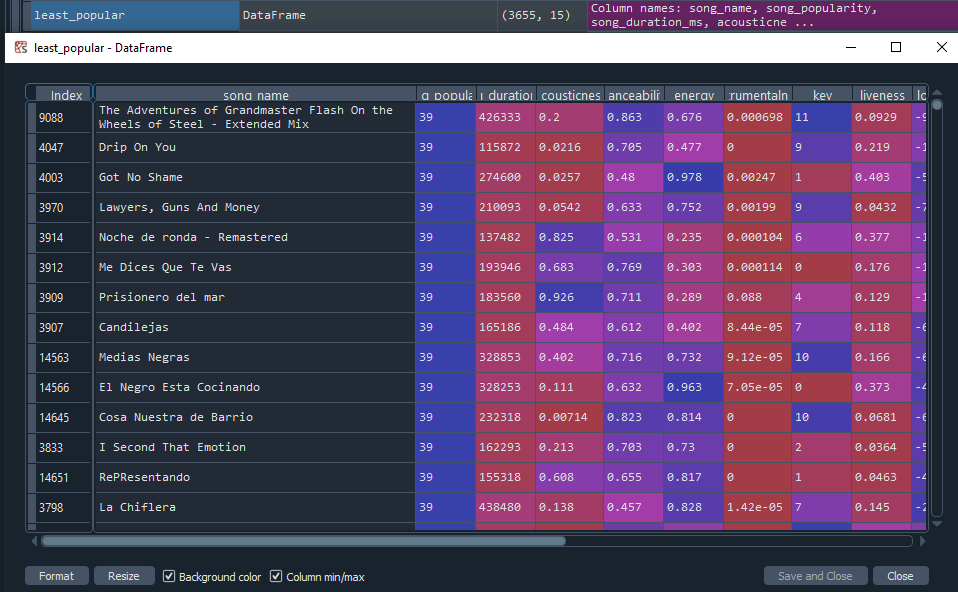
****

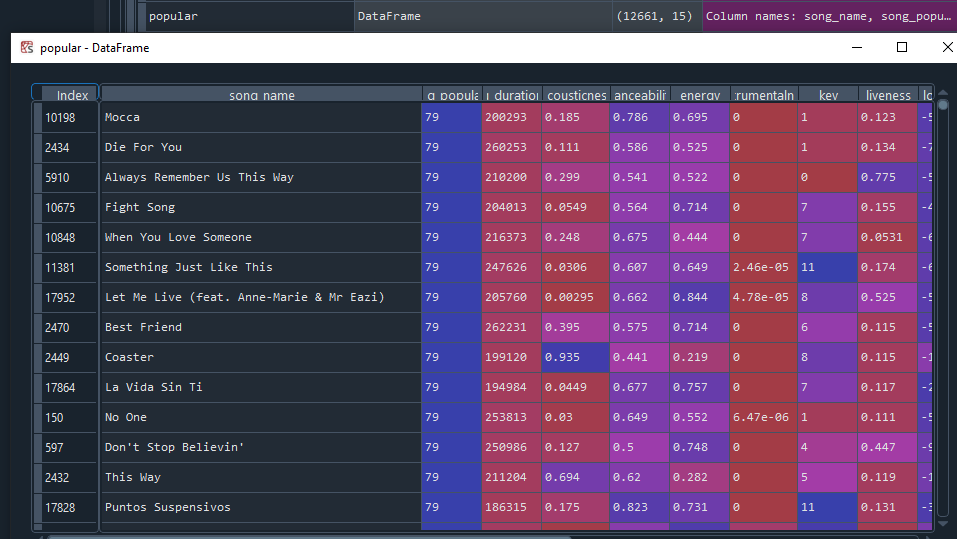
****

**Din grafic, reiese aranjarea melodiilor în funcție de popularitate. Astfel, putem observa faptul că melodiile se împart în 3 categorii**

1. **cele nepopulare între 0 și 40**
2. **cele mai puțin populare între 40 și 80**
3. **cele populare între 80 și 100**

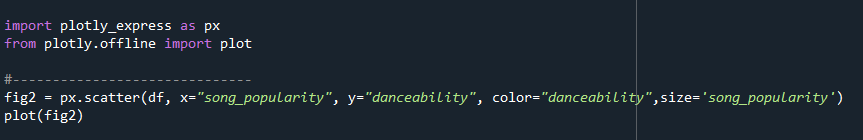
****

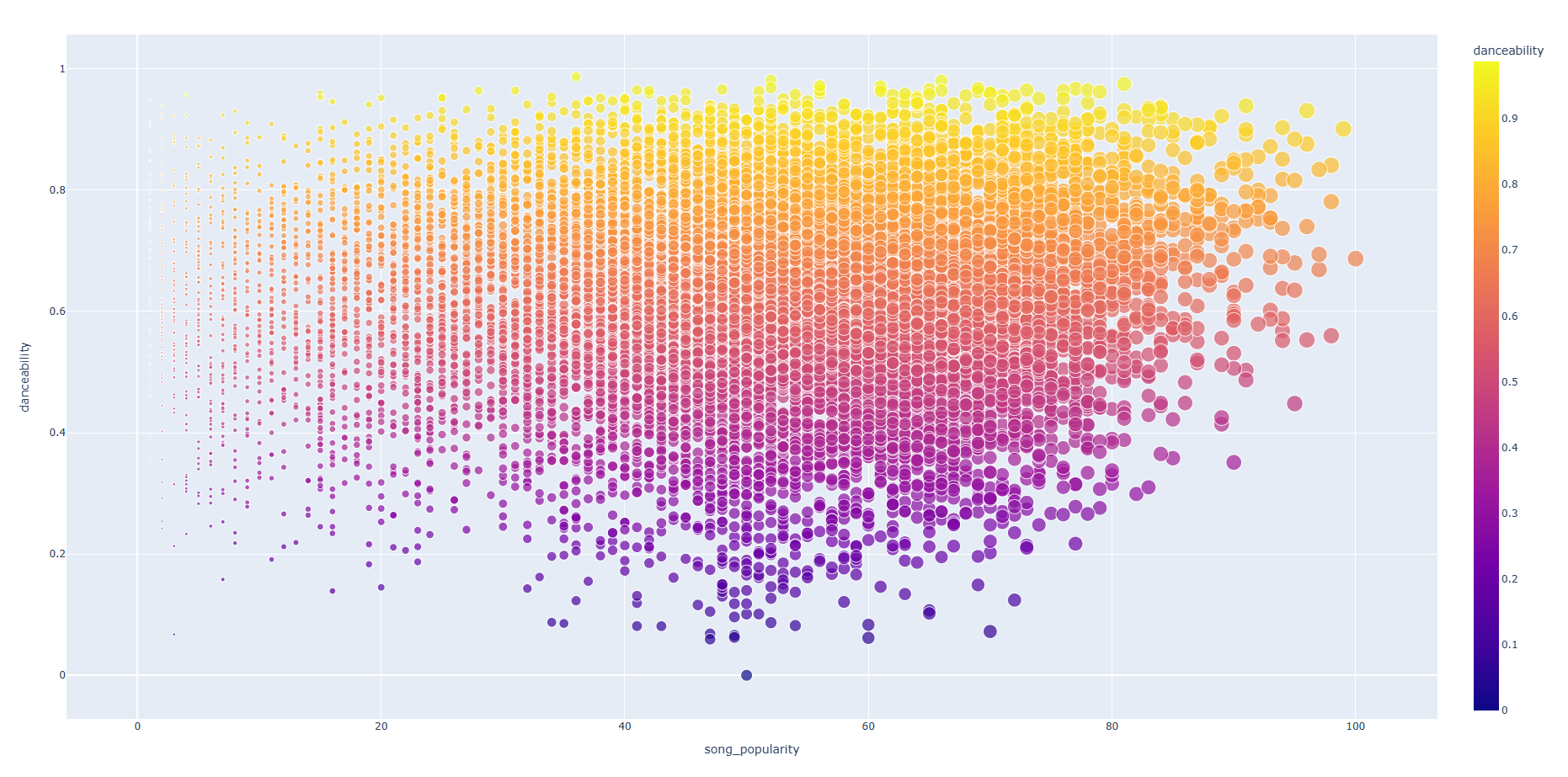
****

****

****

**GRAFICE PE BAZA DE DATE CURĂȚATĂ**

****

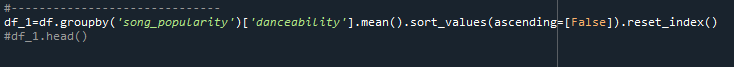
****

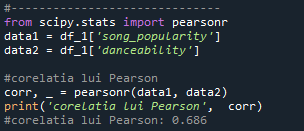
Am trasat un grafic de dispersie. În acest caz, deoarece zona cercurilor corespunde scorului de dansabilitate, putem numi această diagramă și Grafică cu bule. Acest lucru implică faptul că, cu cât scorul de popularitate este mai mare, cu atât mai mare va fi zona corespunzătoare acelei bule și invers. Graficul în sine este interactiv și, printr-o privire rapidă asupra lui, lansăm că „popularitatea” și „capacitatea de dans” sunt corelate pozitiv, ceea ce implică că, pe măsură ce popularitatea cântecului crește, și scorul de dansabilitate pentru acea melodie crește. Nu este întotdeauna nevoie să trasăm un grafic pentru a verifica corelația dintre două caracteristici. Același lucru poate fi realizat și cu câteva coduri simple prezentate mai jos. Pentru aceasta vom folosi un modul numit Scipy.Stats. Acest modul conține un număr mare de distribuții de probabilitate, precum și o bibliotecă în creștere de funcții statistice. Apoi, importăm funcția pearsonr din acest modul, care ne ajută să calculăm constanta de corelație a lui Pearson „r” pentru două caracteristici diferite.

Următoarele sunt cele trei condiții pentru coeficientul de corelație al lui Pearson „r”:

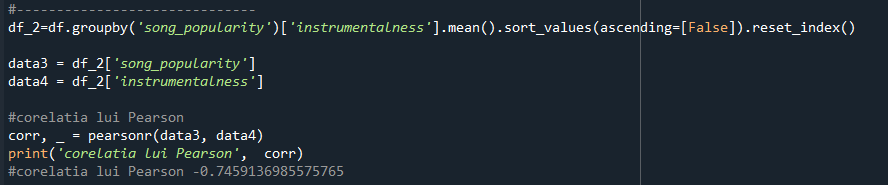
* r>0, implică, corelație pozitivă
* r=0, implică nicio corelație
* r<0, implică, corelație negativă.

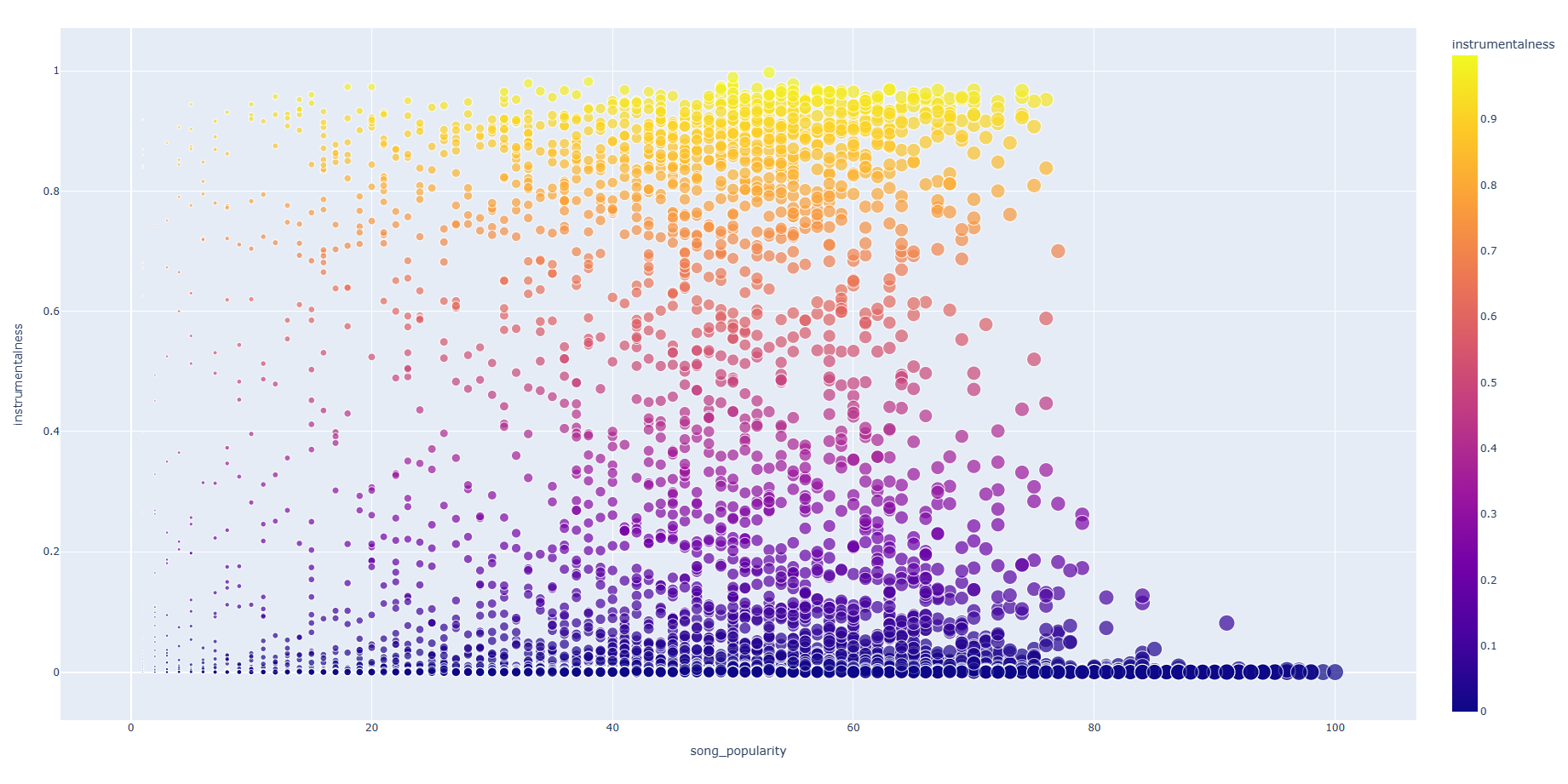
Vedem că, din moment ce valoarea coeficientului de corelație r =0,68 (>0), cele două caracteristici sunt corelate pozitiv, sau putem spune că o creștere a unei caracteristici va avea o creștere a celeilalte caracteristici și invers.

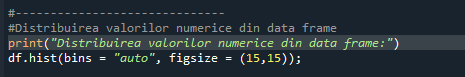


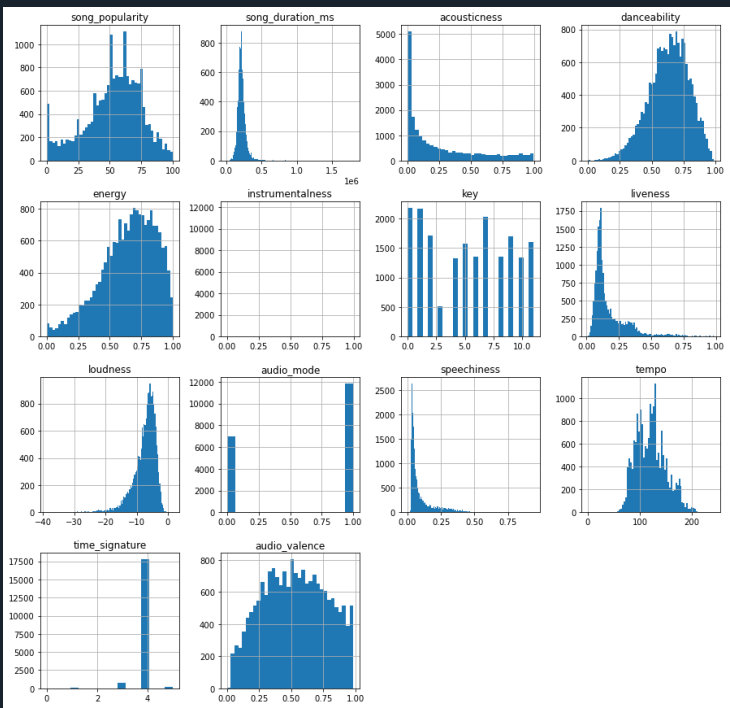


Instrumentalitatea valorii 1 indică faptul că nu există deloc cuvinte, cu cât valoarea este mai mică, cu atât mai multe cuvinte conține cântecul. Urmând o procedură similară și trasând un grafic cu bule pentru popularitate și instrumentalitate. Privind graficul, se observă cu ușurință că cele două caracteristici sunt corelate negativ, ceea ce presupune că, o creștere a uneia duce la o scădere a altuia și invers.

****

****





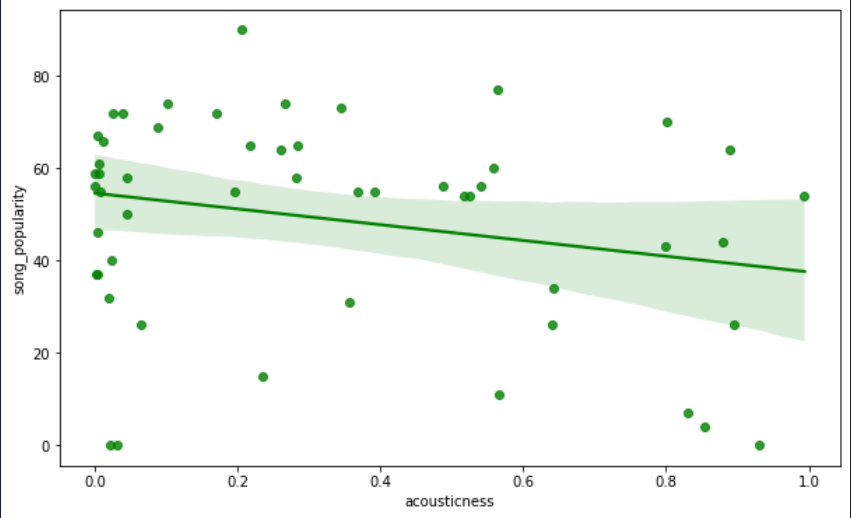
Aceste histograme afișează cum sunt distribuite datele și oferă o perspectivă asupra naturii (continue vs. categorice) fiecărei variabile. Volumul și intensitatea au distribuții distorsionate, iar cele mai multe melodii au o valoare a dansabilitatii de aproximativ 0,6, aproximativ 0,7. În plus, eliminarea duplicatelor a avut un efect asupra distribuției variabilelor țintă, deoarece aceasta nu mai este distribuită uniform. În cele din urmă, majoritatea melodiilor sunt scrise la scară majoră, iar distribuția energiei ne face să știm că majoritatea melodiilor au o valoare energetică ridicată.

**Eșantionarea a 4% din valoarea toata a întregului set**

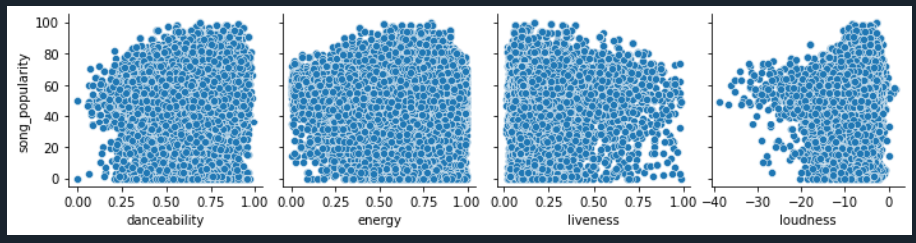


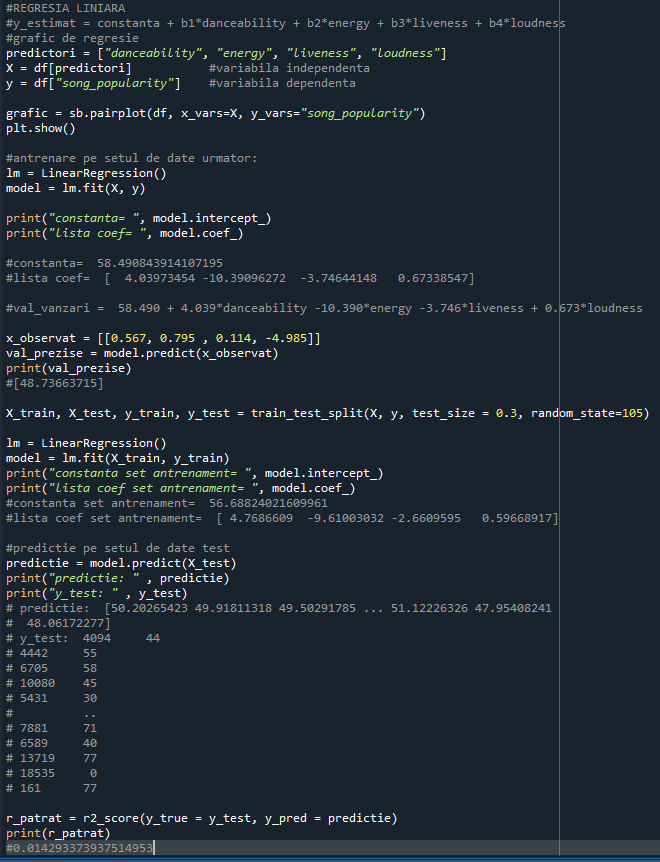
Pentru a realiza graficul de mai jos, se folosește următoarea secvența de cod ce extrage un subset aleator din dataframe utilizând funcția sample împreuna cu argumentul int(0.004\*len(df)), acest argument care reprezintă 4% din lungimea sa. Ulterior creează o figura de dimensiunea 10\*6 prin plt.figure(figsize = [10,6]) în care se va desena graficul de regresie între cele doua coloane selectate pentru acousticness și y pentru song\_populatity.

Linia verde din acest grafic, care este în scadere, reprezintă acustica și popularitatea melodiilor .Dacă acustica va crește atunci popularitatea va scadea, iar dacă popularitatea crește sau rămâne stabilă, atunci acustica va scădea.



**REGRESIA LINIARA PE UN SEGMENT DE DATE**

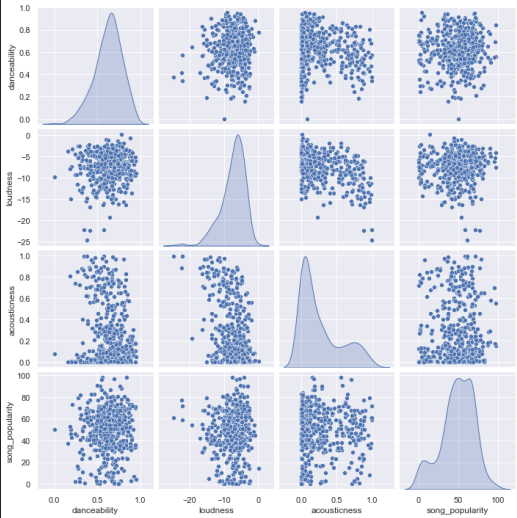




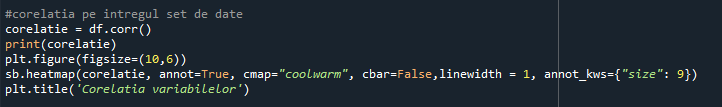
În această parte am estimat rata de succes a modelelor în prezicerea cântecelor. Am folosit funcția scor()/r\_patrat care calculează coeficientul de determinare. Un scor mai aproape de 1 înseamnă că regresorul este mai precis.

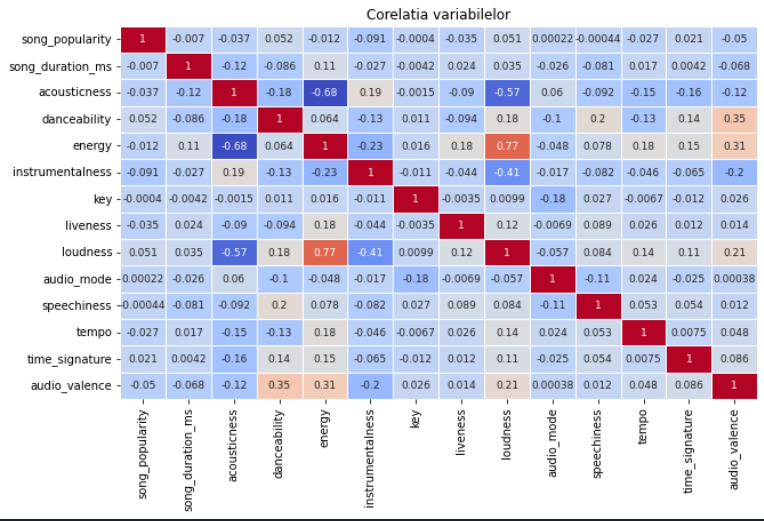






**CORELAȚIA PE ÎNTREGUL SET DE DATE**



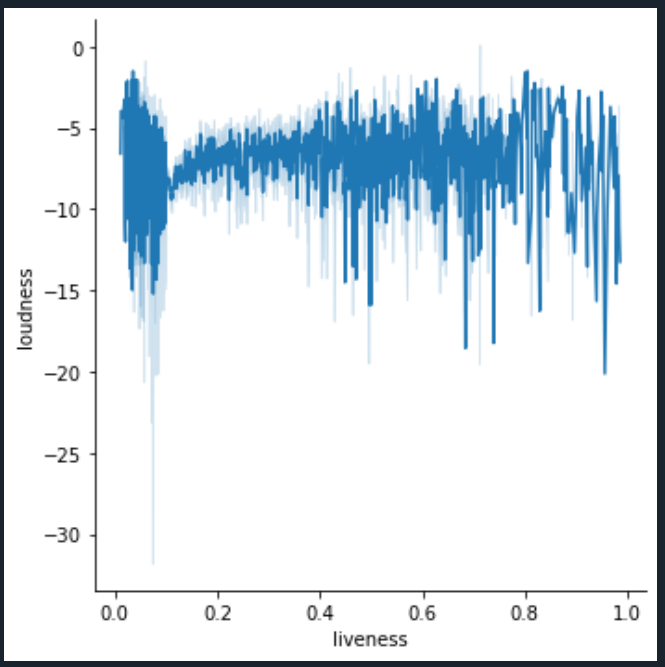


Corelație pozitivă puternică: se regasește între „loudness” și „energy”

Corelație pozitivă slabă: se regasește între „loudness” și „liveness”

**Relația dintre „loudness” și „liveness”**

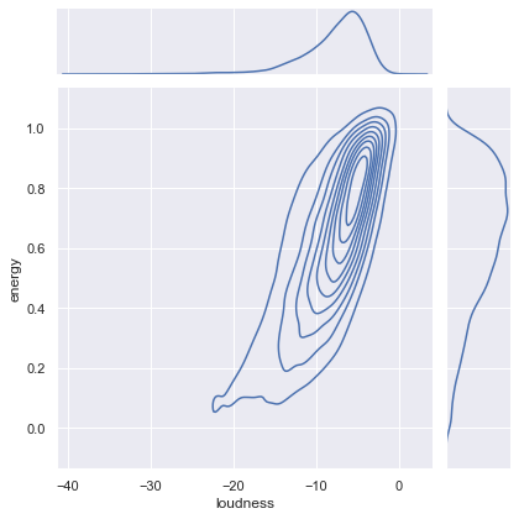




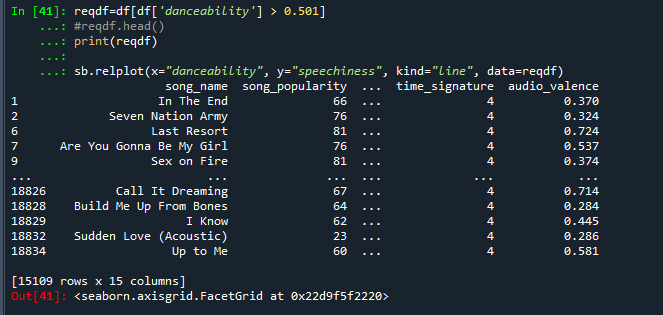
Î - Liveness este afectat de Loudness?

R - De deasupra Heatmap-ului și a graficului de linii între cele două coloane de date putem vedea că o relație liniară nu există între cele două coloane. Acest lucru ne spune că utilizarea unei funcții nu o va afecta pe cealaltă.

De asemenea, putem reprezenta un grafic bivariat pentru o anumita pereche de variabile:

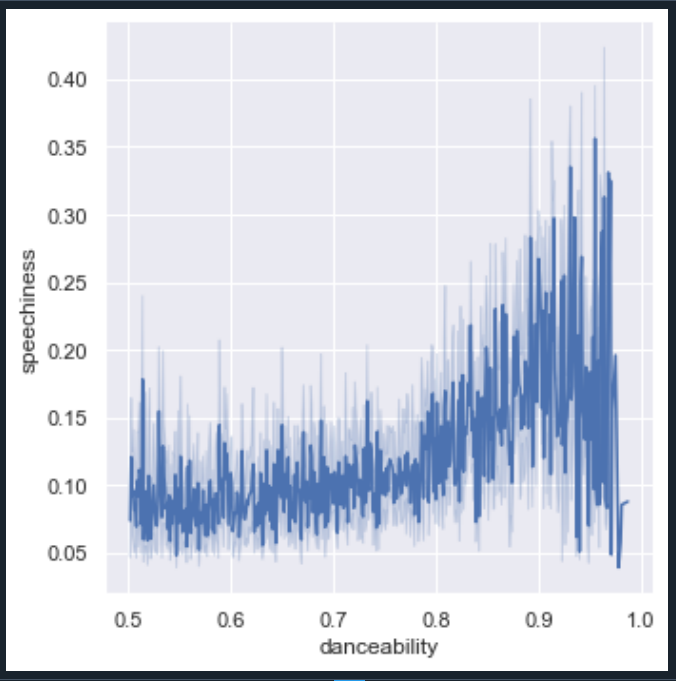


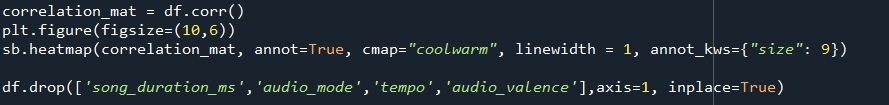
Codul utilizat pentru a afișa graficul bivariat pentru o anumită pereche de variabile, loudness și energy, se folosește de biblioteca seaborn pentru a crea un plot de tip „joiny plot” pentru datele din DataFrame-ul denumit”df”. Valorile de pe axa x sunt preluate din coloana „loudness” și cele de pe axa y sunt din coloana”energy”.



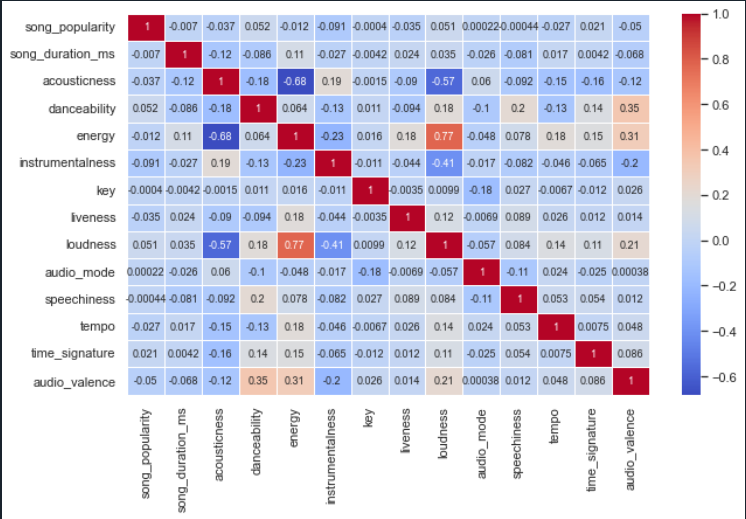
Prima linie creeaza un nou data frame numit ”reqdf” care va conține rândurile numite „danceability” din data frame ul „df”, unde valoarea lor este mai mare decat 0.501. Acest nou data frame „reqdf” este afișat prin apelarea comenzii print.

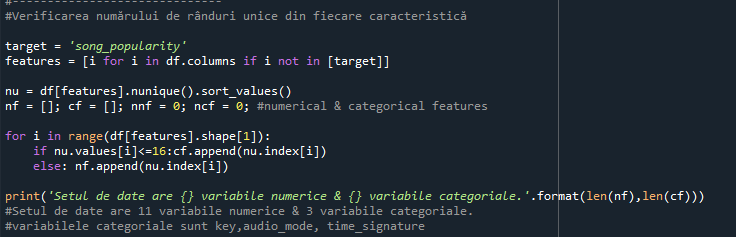
Graficul este unul de tip „line plot” pentru înregistrarile din „redqf” unde axa x reprezintă valorile preluate din coloana „danceability” și valorile de pe axa y sunt din coloana „speechiness”



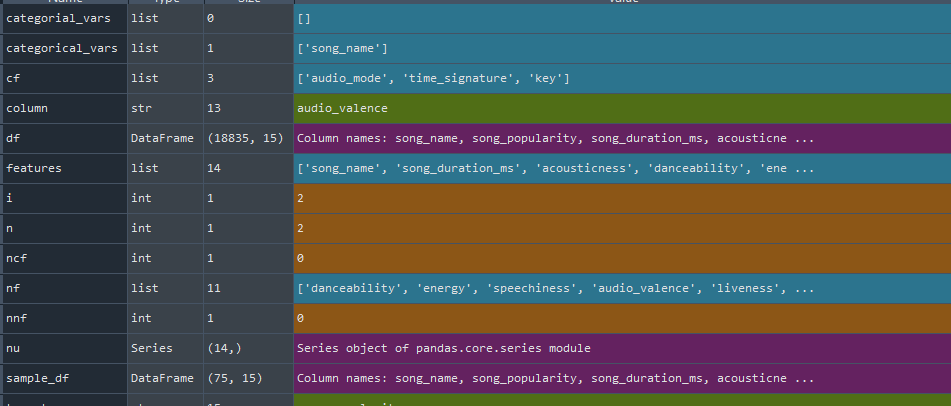


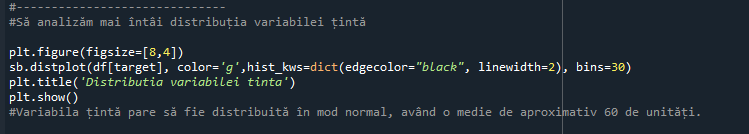
Pentru a realiza acest heatmap am calculat matricea de corelație între toate coloanele din dataframe-ul „df”, urmat de a ii specifica dimensiunea imaginii generate prin comanda „figsize=(10,6)”. Setarea parametrilor a fost realizata cu scopul de a ușura citirea heatmap-ului. În final au fost șterse coloanele ”song\_duration\_ms”, „audio\_mode”, „tempo”, „audio\_valence” din DataFrame-ul „df” deoarece acestea nu afectează analiza în cautarea corelației pentru melodia perfectă.



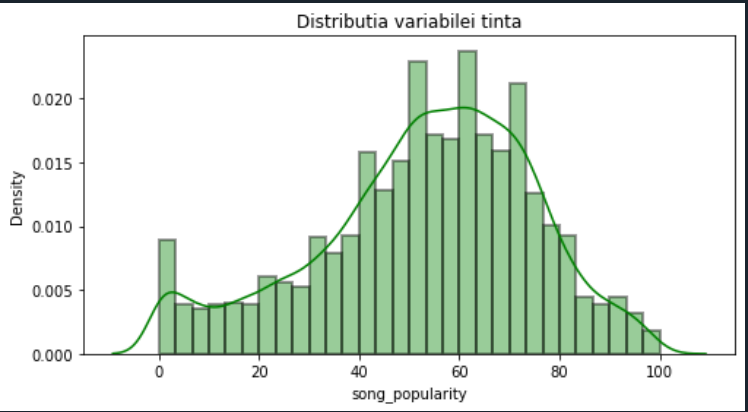


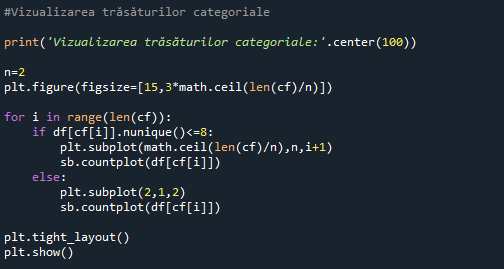
Prin utilizarea acestui cod se poate segmenta setul de date în doua liste distincte. Prima lista denumita „cf” se creeaza utilizând criteriul de selectie al variabilelor, dacă acesta are valoarea mai mica sau egala cu 16 sunt considerate variabile categorice și sunt adăugate în lista, dacă valoarea variabilei este mai mare decat 16 aceasta se adaugă în lista „nf” unde se vor afla toate variabilele numerice. Dupa ce au fost sortate toate înregistrarile acestea sunt afișate.



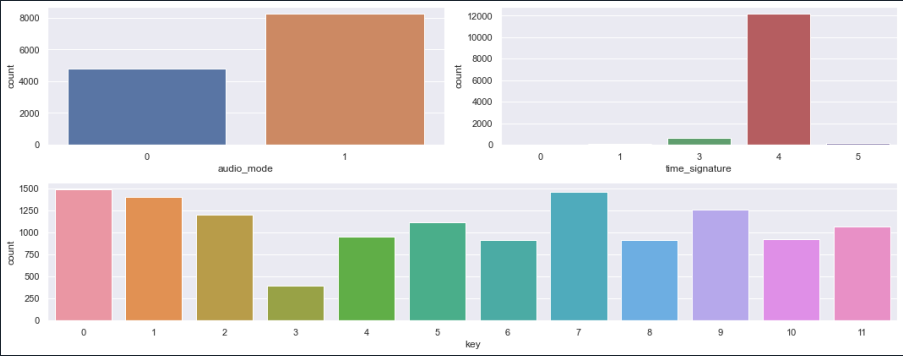


Pentru a realiza graficul ce conține “Distribuția variabilei ținta”, se utilizează librariile Matplotlib și Seaborn pentru a trasa distribuția variabilelor Density și Song\_popularity . Dimensiunea este specificată de figsize=[8,4], iar trasarea distribuției variabilei ținta este specificată ulterior prin definirea atributelor necesare trasării.

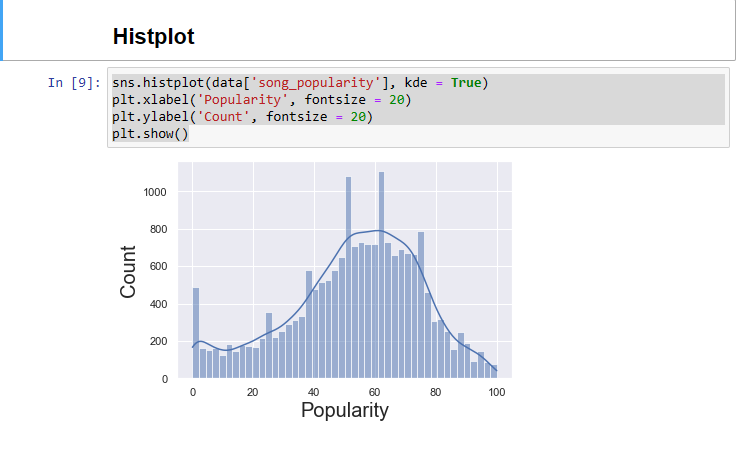




În realizarea acestor grafice variabila n a fost inițializată cu valoare de 2 și este folosită cu scopul de a determina numărul de grafice per line în figura ce urmează a fi generată. Funcția plt.figure specifică dimensiunea figurii și aceasta calculează în funcție de numărul de grafice determinat prin împarțirea lungimii listei ”cf” la “n”, astfel încat fiecare graphic să aibă spațiu suficient. În buclă se va itera prin fiecare element din lista ”cf”, verificând dacă numarul unic de valori din coloana corespunzătoare din dataframe-ul “df” este mai mic sau egal cu 8. Dacă această condiție este îndeplinită atunci se folosește funcția “plt.sublot” pentru definirea poziției graficului și ulterior “sb.countplot” pentru a îl genera. Prin plt.tight\_layout se optimizează afișarea graficului în figura.

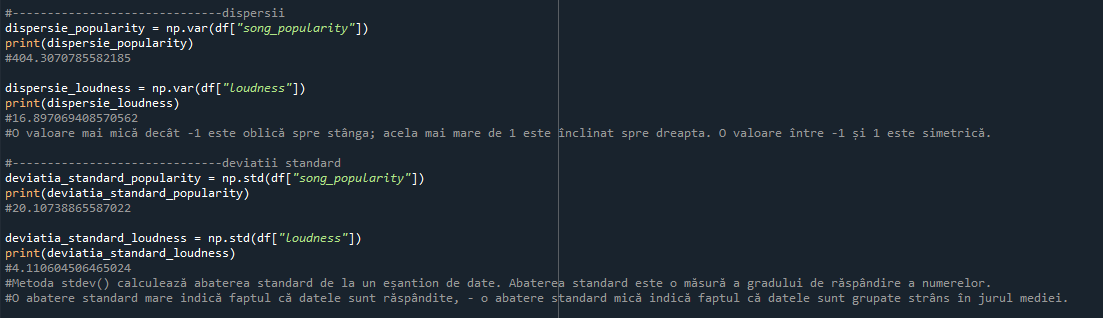


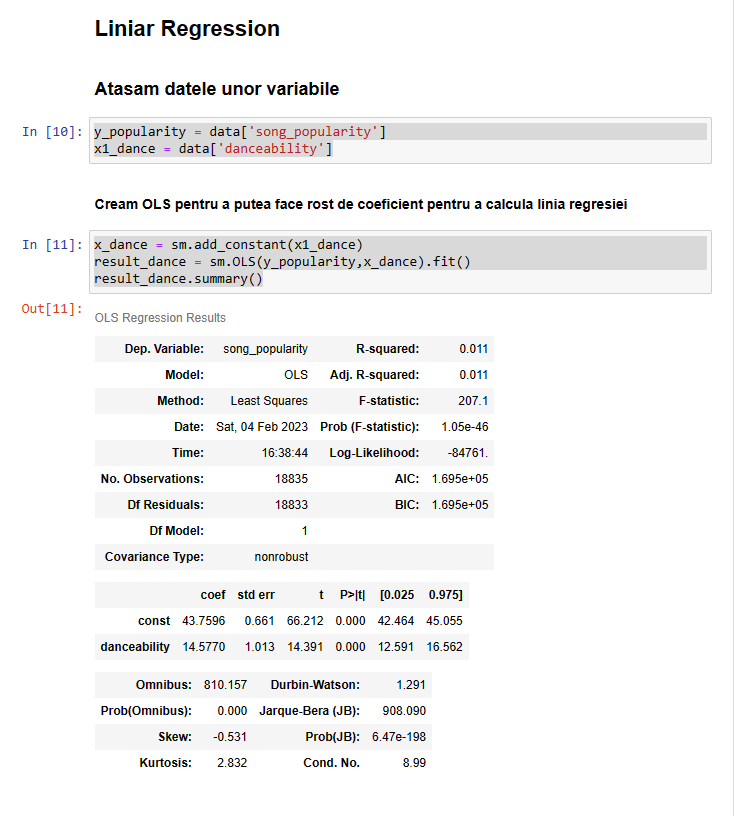
**JUPITER NOTEBOOK**



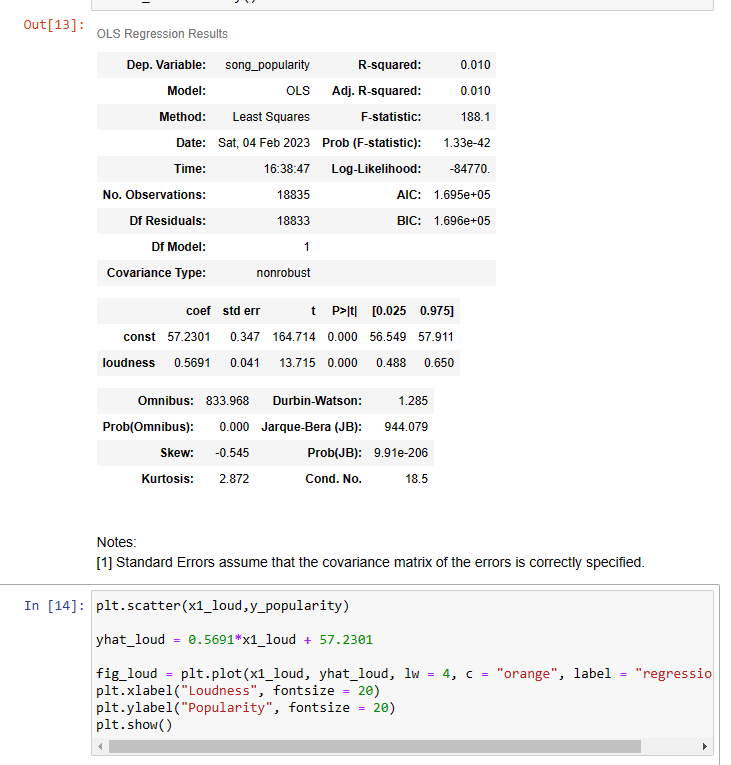
Această histogramă a popularitații melodiilor din setul de date este realizată prin utilizarea bibliotecii Seaborn și a unui set de date numit ”data”. Prin sns.hisplot se generează o histogramă pentru seria de date specificată, în acest caz data[“song\_popularity”]. Setarea parametului “kde” pe true specifică generarea și a unei curbe a densitații peste histogramă. Plt.xlabel si plt.ylabel specifică etichetele pe cele doua axe, x și y, și dimensiunea fontului care este setată pe 20.

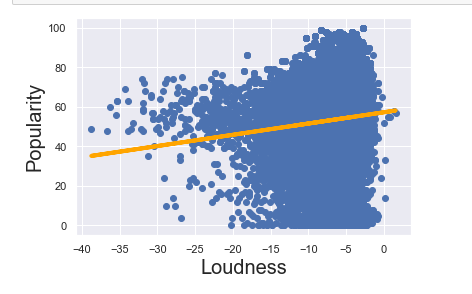


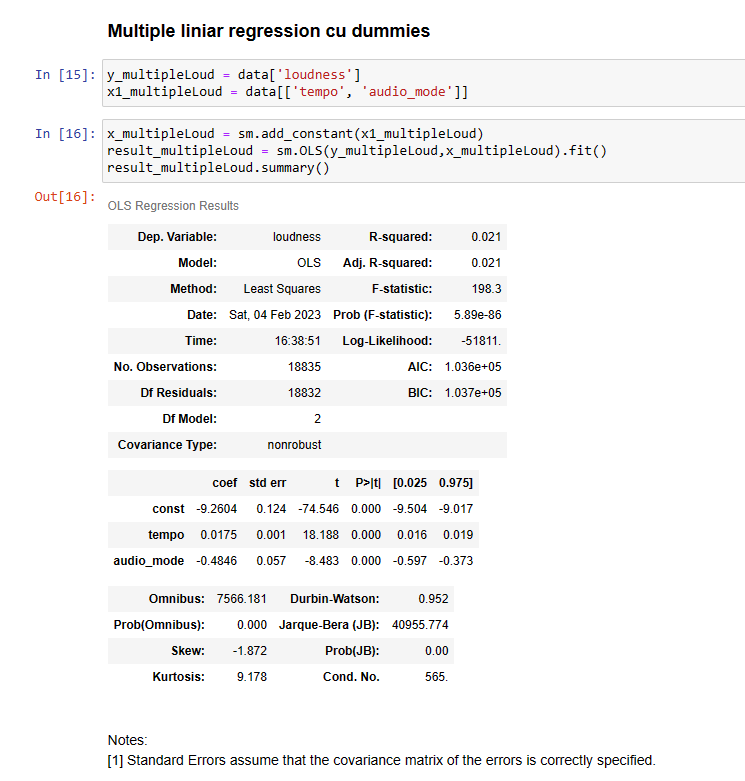


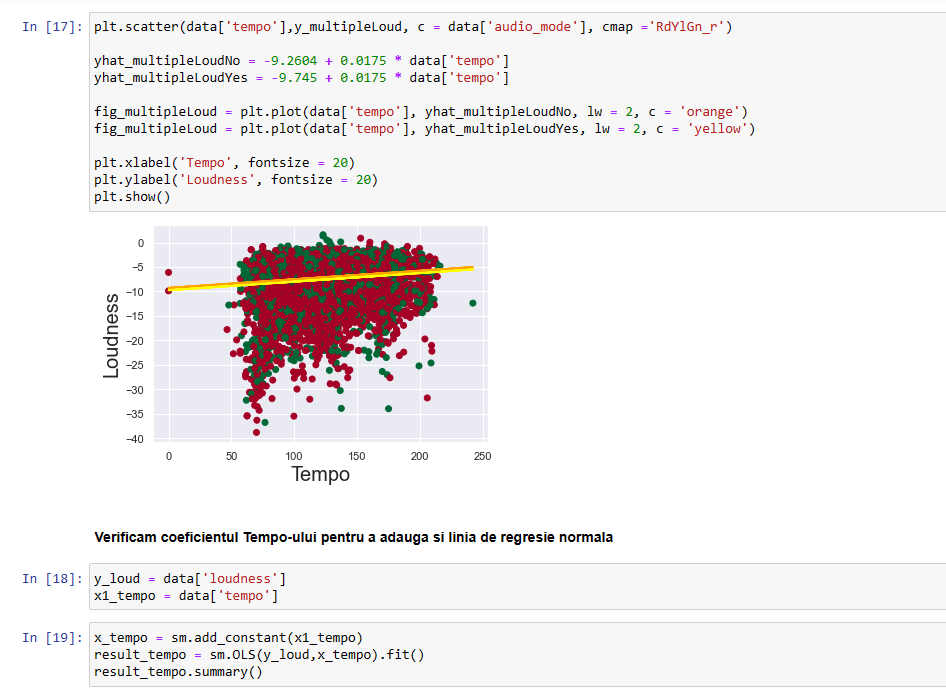


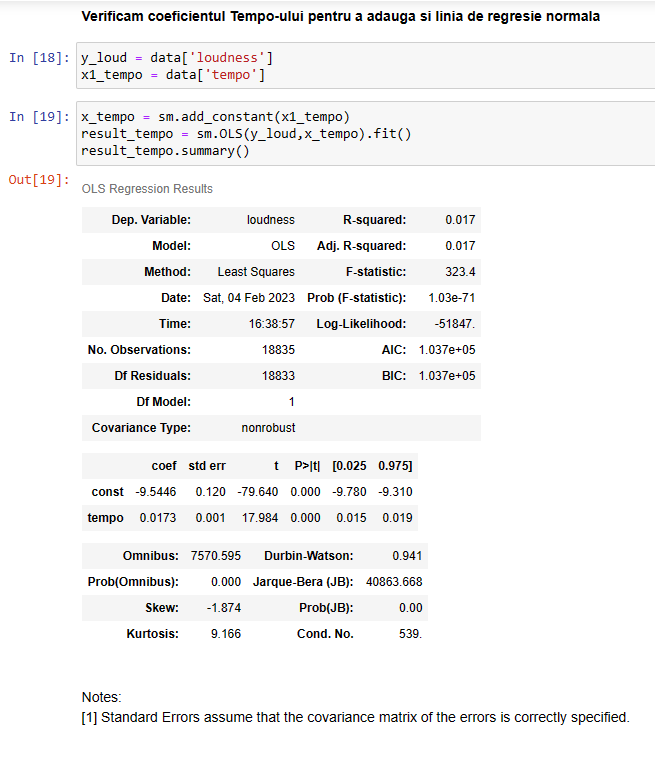




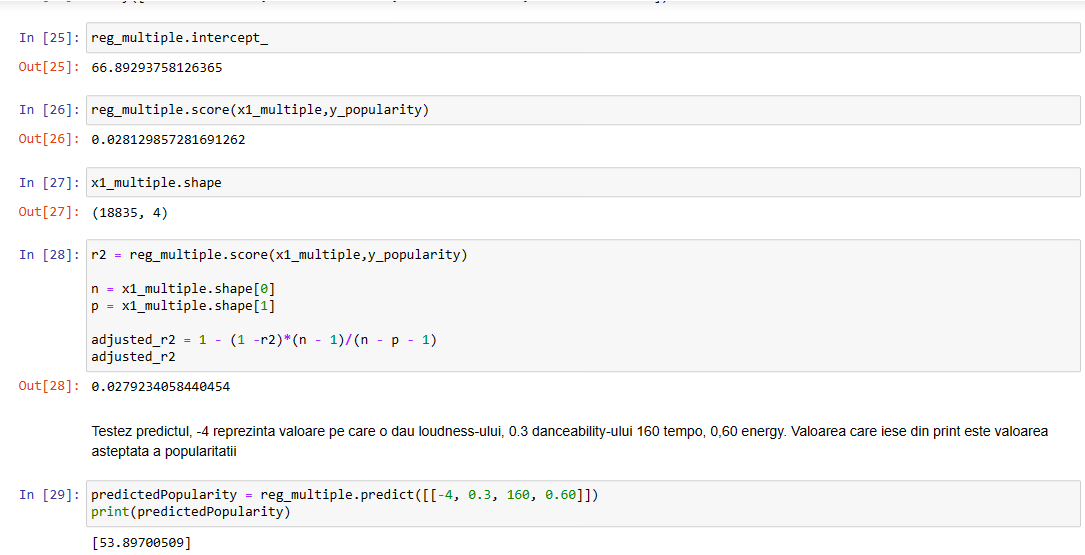














Clustering-ul este un set de tehnici folosit pentru a identifica grupuri de obiecte similare în seturi de date cu două sau mai multe variabile. În practică, aceste date pot fi colectate din baze de date de marketing, biomedicale sau geospațiale, printre multe alte locuri. Clustering poate fi utilizat pentru o varietate de scopuri, cum ar fi:

* Segmentarea clienților pentru scopuri de marketing,
* Segmentarea imaginilor pentru recunoașterea obiectelor,
* Detectarea anomaliei pentru identificarea punctelor de date neobișnuite,
* Sisteme de recomandare pentru gruparea obiectelor similar.

În graficul de mai jos arată că există 3 grupuri formate pe caracteristică . Aceste trei grupuri implică faptul că există 3 modele de muzică bazate pe danceability și energy.

