

Proiect Fundamente de Big Data

Influența factorilor interni asupra popularității unui cântec

Studenti: Malan Andrei-Octavian

Ionescu Alex-Daniel

Cuprins

Introducere.....	3
Setul de date	4
Metodologie	5
Rezultate și discuții	5
Metoda 1 - Arbori de clasificare	5
Metoda 1.1 – Bagging	7
Metoda 1.2 - Random Forest	9
Metoda 2 - Naive Bayes	10
Teste finale	11
Concluzii.....	12

Introducere

Domeniul muzical reprezintă o arie de interes pentru o gamă largă de oameni și diferite segmente de populație, acesta fiind o continuă sursă de divertisment și de generare de emoții care coincid cu trăirile umane. Indiferent de gen, sau segmentul țintit, fiecare melodie se poate remarca printr-un lucru, în mod principal: popularitatea acesteia. Acest concept trece bariera timpului, și face ca un cântec să fie prezent în posteritate dacă a atins sufletele a cât mai multor oameni.

Dat fiind faptul că toți oamenii ascultă muzică iar acest lucru nu este tocmai gratis de cele mai multe ori, cei implicați în domeniu au un interes major să dea totul pentru a crește cât mai mult popularitatea unui cântec, căci acest lucru poate însemna încasări mai mari, recunoaștere internațională sau cariere de succes. Astfel, tocmai pentru că am conștientizat acest aspect al lumii exterioare din domeniul muzical, ne propunem să aflăm cum influențează factorii interni ai unei melodii gradul de popularitate al acesteia sau cât de performant poate să fie un model care încadrează o melodie într-o clasă de popularitate pe baza factorilor muzicali care o compun.

Considerăm că analiza noastră poate fi de impact pentru o serie întreagă de segmente de populație, pe care le vedem interesate să afle dacă pot găsi înainte de producerea unei melodii o anumită rețetă prin care să-i crească popularitatea sau să o facă cunoscută la nivel național sau internațional. Știm că acest lucru poate genera sume imense de bani, faimă sau notorietate, și tocmai de aceea, putem enumera niște actori care ar fi interesați să citească rezultatele raportului nostru:

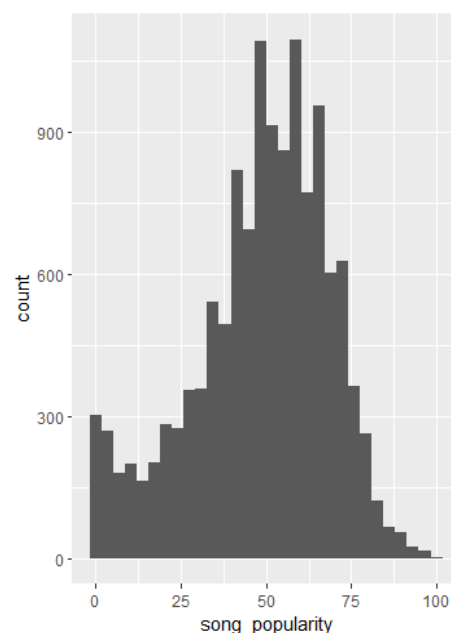
- ❖ casele de discuri- interesate să vadă pe ce factori interni ar trebui să se axeze pentru a putea maximiza șansele de încadrare a melodiei produse în clasa de popularitate dorită, astfel încât să maximizeze veniturile aduse din vânzarea cântecului;
- ❖ radiourile ar putea fi la rândul lor interesate să vadă ce melodii să difuzeze în vederea creșterii audienței, analizând factorii care sunt cei mai relevanți în măsurarea gradului de popularitate;
- ❖ nu în ultimul rând artiștii simpli care compun și vor să găsească faima și o punte de lansare într-o piață muzicală tot mai competitivă, și cu un public țintă tot mai pretențios și educat muzical.

Astfel, întrebările de cercetare la care încercăm să răspundem coincid cu răspunsurile pe care le caută și diferite entități implicate în domeniu, astfel încât să-și crească încasările, faima sau să se lanseze, putând spune că importanța studiului nostru este relevantă căci aduce atingere unor interese reale din zona muzicală națională sau chiar mondială.

Setul de date

Setul de date pe care l-am prelucrat se găsește pe platforma kaggle, la linkul <https://www.kaggle.com/datasets/yasserh/song-popularity-dataset>. Acest set de date conține în prima fază 15 coloane, începând cu numele cântecului, scorul de popularitate(numeric), și pe urmă urmează o serie de factori interni, aceștia fiind durata cântecului, nivelul de acustică, nivelul instrumentului, dansabilitatea, energia oferită, gama, nivelul de zgomot, vivacitatea, modul audio, tempo-ul, ritmul vorbirii, măsura notelor, și valența audio. Totodată, setul conține aproape 18000 de înregistrări, acest număr fiind alcătuit și din duplicate, lucru pe care l-am corectat eliminând cântecele unde apărea mai mult decât o înregistrare, ajungând în final să avem undeva la 13000 de înregistrări unice în set.

O altă prelucrare pe care am considerat necesară să o facem a fost modificarea tipului variabilei țintă din numeric în categoric. În forma inițială a setului de date, popularitatea cântecului era măsurată sub forma unui scor numeric, între 0 și 100, fapt ne-a determinat în primă fază să încercăm modelul regresiei liniare. Însă după primele vizualizări realizate, am observat că nu există dependențe clare care să definească niște relații specifice acestui model, astfel încât am decis să transformăm variabila țintă într-un tip categoric. Datorită faptului că măsurăm un sentiment uman, am ales să adoptăm varianta a 3 clase țintă, după cum urmează: Dacă scorul inițial era între 0 și 43, clasa de care va aparține cântecul se va numi Nepopular; între 44 și 60 vom avea clasa Neutru, iar de la 61 până la 100 vom avea clasa Popular. Această împărțire vine după o analiză a setului de date și a numărului de înregistrări aferente fiecărei clase. Astfel, pentru a putea analiza un set de date cât de cât echilibrat, am luat aceste intervale anterior menționate ca și intervale de referință.



Totodată, am eliminat denumirea cântecului din analiza noastră, preferând să ne axăm pe factori interni măsurați concret, care pot influența clasificarea unei melodii într-o anumită clasă de popularitate.

În final, ne-a rămas un set de date cu 13 factori interni, de tip numeric sau categoric, care acționează ca și variabile independente și o variabilă dependentă nominală, sub forma a 3 clase de popularitate(Nepopular, Neutru și Popular).

Metodologie

Pentru a putea respecta cerințele reale ale unui proiect de cercetare care presupune tehnici de Machine Learning, stabilirea unei metodologii clare este extrem de importantă.

Astfel, pentru a avea un demers unitar pe tot parcursul lucrării, am decis ca fiecare metodă pe care o abordăm să presupună următorul proces: fiecare set de date se va împărți în 3 secțiuni- setul de antrenament, într-un procent de 60%, setul de validare, care va presupune alocarea a 20% din date, iar în final setul de test, care va acoperi ultimii 20% din datele setului.

Antrenarea fiecărui set va loc pe setul de antrenament, iar măsurarea acurateții fiecărui model va fi stabilită pe setul de validare. În final, va urma compararea acurateților modelelor stabilite pe seturile de validare, iar de aici vom extrage cel mai bun model, asupra căruia vom determina acuratețea folosind setul de test.

Abia în acest punct vom putea trage o concluzie finală și clară asupra problematicii pe care ne-am propus să o cercetăm în demersul nostru academic.

Rezultate și discuții

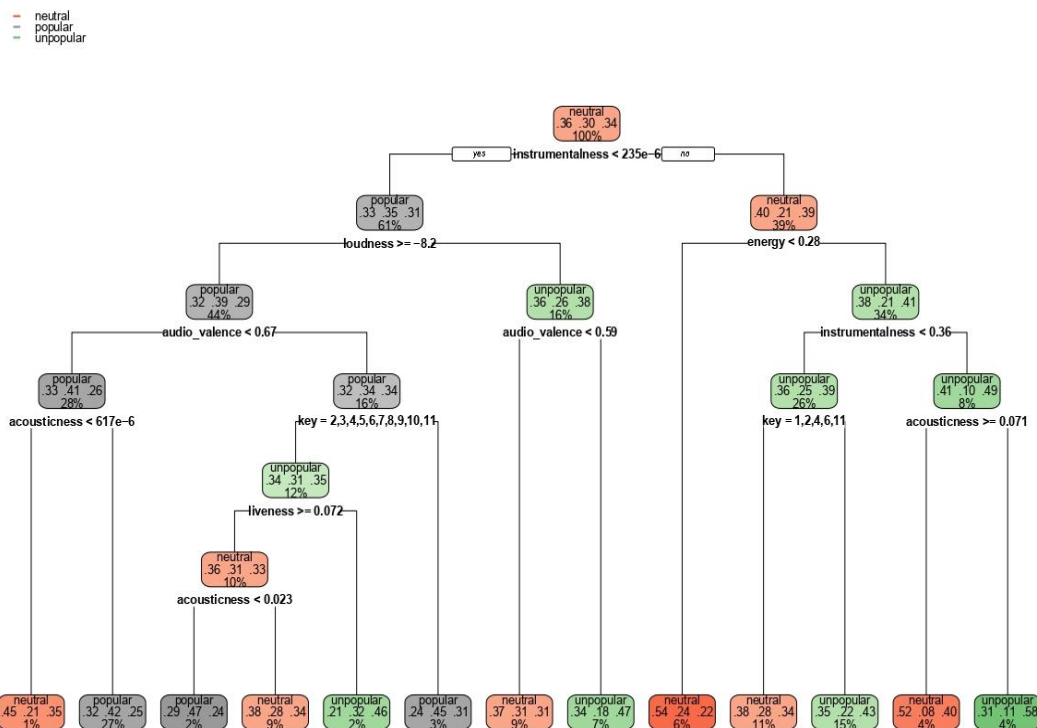
Metoda 1 - Arbori de clasificare

Utilizarea acestei metode se poate face datorită faptului că variabila noastră țintă, și anume popularitatea cântecului este împărțită în 3 clase(nepopular, neutru și popular).

Am început prin importarea bibliotecilor R necesare desfășurării demersului nostru: `rpart`(folosită pentru lucrul cu arbori), `rpart.plot`(pentru vizualizarea arborilor), `tidyverse`(pentru operațiuni de manipulare a datelor înainte să fie introduse în analiză), `rsample`(pentru împărțirea setului de date în cele dorite și menționate în metodologie), `caret`(pentru construirea modelelor pe care ulterior le analizăm) și `partykit`(o nouă librărie pentru vizualizarea arborilor, independentă de anterior menționată `rpart.plot`).

După aducerea setului de date sursă, am realizat câteva operațiuni pentru a putea crea premisele construirii modelului dorit: am selectat doar coloanele relevante pentru analiza noastră(am considerat numele cântecului ca fiind lipsit de relevanță), am transformat coloana cu durata cântecului din milisecunde în secunde, și ulterior am redenumit-o, iar coloanele cu valori nominale le-am transformat în factori(`key`, `audio_mode`, `time_signature`). Ultima și poate cel mai important pas, care a fost menționat și în capitolul legat de setul de date, este modificarea variabilei țintă, din numeric în categoric. Astfel, acum există toate premisele prin care modelul pe care dorim să-l avem să poată fi realizat și analizat.

O altă setare care ni s-a părut relevantă în analiza noastră este determinarea celui mai bun parametru de complexitate (cp). Pentru a nu hardcoda această valoare, am realizat un mini algoritm al cărui scop este căutarea celui mai bun parametru prin compararea performanțelor modelelor construite cu ajutorul acestui parametru de complexitate. Astfel, am comparat posibilele performanțe ale unor asemenea 100 de modele, printr-o variabilă numită tuneGrid, iar în final a rezultat că cea mai bună performanță a fost obținută cu un cp la valoarea de 0.0031.



În acest fel, am ajuns în punctul de a construi primul model relevant în analiza noastră. Am antrenat modelul(m1) pe setul de antrenament, am decis ca variabila dependentă să țină cont de toate variabilele independente, să folosim metoda “class”, care sugerează alegerea unei clasificări în contextul de față, iar în final setarea parametrul de complexitate la valoarea determinată prin algoritmul menționat anterior. Ulterior, pentru măsurarea acurateții și a altor metrice importante, am folosit metoda predict, având la bază modelul antrenat m1 și setul de validare, urmând ca în final, folosind comanda de confusionMatrix, să aflăm toate informațiile care ne interesează.

La o primă analiză observăm că acuratețea este destul de mică, aproximativ 39%, ea aflându-se în cadrul intervalului de încredere(~37%-41%). Totuși, valoarea metricei P-Value este 0,00091, ceea ce rezultă în deducția faptului că acuratețea nu este determinată de șansă, iar specificitatea pentru fiecare clasă se află în intervalul 62%-74%, de unde putem interpreta de exemplu, dacă ar fi să luăm punctual o clasă, să spunem neutru, care are aproximativ 62% valoarea specificității, că 62% din instanțele negative din cadrul clasei de neutru au fost clasificate corect. Totodată, dacă ar fi să vedem pentru fiecare clasă luată separat rata de predicție pozitivă(pos pred value), am vedea că pentru clasa neutru avem 37,29%, pentru popular 38,72%, respectiv 41,2% pentru clasa nepopular, așadar o performanță destul de scăzută.

Entropia este acea măsură care indică impuritatea sau gradul de dezordine din cadrul unui set de observații. Pentru a putea calcula acest lucru, ne vom folosi de formula generală de calcul a entropiei: $-\sum_x x * \log(x)$, unde x reprezintă probabilitatea ca o instanță să aparțină unei clase. Astfel, pentru cele 3 clase pe care le avem(neutru,popular, nepopular), vom probabilitățile de 0.355, 0.295, respectiv 0,339, suma lor fiind 1. Aplicând formula, vom obține cu aproximație un procent al entropiei de 66%, de unde putem înțelege faptul că setul de date este destul de diversificat și imprevizibil.

Metoda 1.1 – Bagging

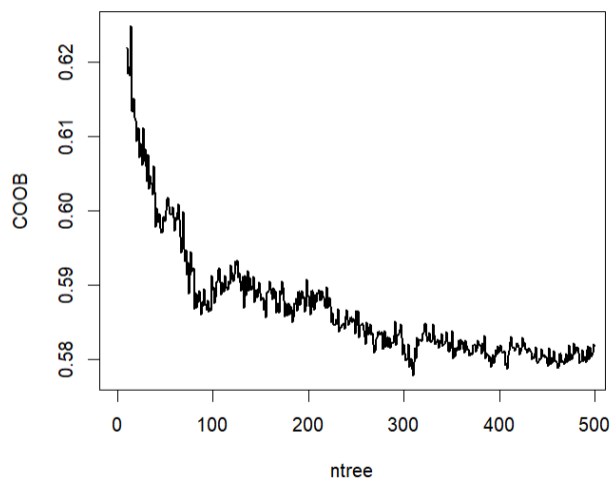
Bootstrap aggregating, numită și bagging, este o metodă de machine learning concepută pentru a îmbunătăți acuratețea modelelor și pentru a reduce overfitting-ul. Aceasta presupune împărțirea setului de antrenament în subseturi mai mici pe care se creează modele în mod independent pentru fiecare subset iar în final acuratețea modelului va fi dată de combinarea tuturor acurateților modelelor antrenate pe fiecare subset în parte.

După ce am importat librăria ipred, am realizat primul model folosindu-ne de setul de antrenare și aplicând următoarele setări: variabila țintă va depinde de toate celelalte variabile și COOB va avea valoarea True. COOB, o prescurtare pentru Cost out of bag, este folosit pentru a estima eroarea fiecărui model construit fără a folosi un alt set de validare deoarece, în construirea modelelor, nu se cuprind toate observațiile din setul de date, unele dintre

acestea rămânând în afara modelelor, adică “out-of-bag”. Deoarece nu am specificat numărul de bags în momentul creării primului model, implicit se vor face 25 de bags, ulterior specificând numărul acestora pentru a îmbunătăți acuratețea modelului creat. În cadrul primului model creat, rata erorii COOB este de 0.608, ceea ce înseamnă că aproximativ 60.8% din observațiile prezente în afara modelelor create sunt clasificate greșit, indicând faptul că modelul creat mai poate fi îmbunătățit.

Făcând predicția pe setul de validare, acuratețea obținută este de 39.35%, situându-se în intervalul de încredere (37,46% - 41,26%). Valoarea lui P-Value este foarte mică, 0.000158, acest lucru însemnând că acuratețea nu este conferită de șansă. Nivelul de specificitate se situează pentru cele 3 clase în intervalul (65,02%-74,09%), în timp ce rata de detecție pozitivă se află în intervalul (35,40%-41,83%), arătând o performanță scăzută.

După crearea primului model, am decis să facem 500 de bags pentru a observa numărul optim de bags astfel încât modelul să fie cât mai bun posibil. Pentru a face acest lucru, am inițializat un vector pe care l-am numit *ntree* cu valori între 10 și 500 pe care îl vom folosi pentru a crea numărul specificat de arbori. Am creat apoi un alt vector neinițializat ce va avea lungimea egală cu cea a vectorului *ntree* pe care l-am numit *err* și care va stoca eroarea fiecărui model creat. Cu ajutorul unui *for* care parcurge întreg vectorul *ntree* cu ajutorul unei variabile *i*, am creat modele în care numărul de bags al fiecăruia va fi dat de elementul din vectorul *ntree* de pe poziția *i*, iar pentru fiecare model creat, am salvat eroarea acestuia în vectorul *err*. Ulterior am realizat o reprezentare vizuală a datelor obținute cu ajutorul funcției *plot* în care am pus pe axa x vectorul *ntree* iar pe axa y am pus vectorul *err*. Din reprezentarea vizuală obținută putem observa că numărul optim de bags este în apropiere de 500 de bags.



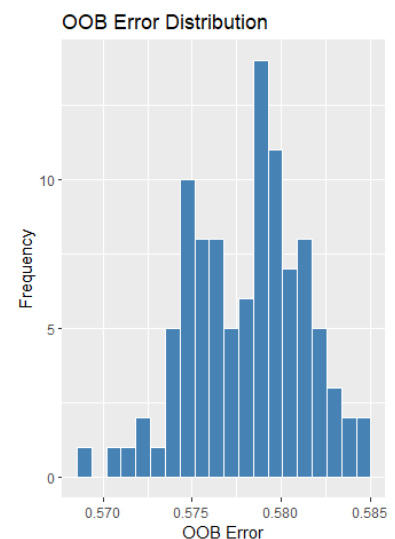
Am realizat astfel un model nou cu 500 bags, obținând o acuratețe de 41,71%. Celelalte metrice pe care le-am mai menționat în lucrare iau pe rând următoarele valori: intervalul de încredere este (39,8% - 43,63%), P-Value este 6.835e-10, specificitatea se situează în intervalul (67,55%-75,03%), iar rata predicției pozitive se află între valorile (40,83%-42,57%), interpretările acestor valori fiind similare cu cele descrise mai sus.

Metoda 1.2 - Random Forest

Random Forest este o metodă de machine learning prin care putem crea noi modele pe baza cărora să tragem concluzii legate de setul nostru de date. În comparație cu bagging, ea prezintă o diferență majoră: în timp ce bagging-ul creează modele unde subseturile sunt diferite, dar respectă aceleași variabile independente(pe coloane), random forest-ul creează modele unde subseturile sunt diferite atât ca eșantionare, cât și a variabilelor analizate. Merită menționat faptul că fiecare model este un arbore de decizie, iar acuratețea se măsoară ca o medie a acurateților fiecărui arbore rezultat.

După importul librăriei randomForest, disponibilă în R, am conturat primul model folosindu-ne de setul de antrenare, și având următoarele setări: vom avea un număr de 500 de arbori, respectiv fiecare arbore va avea un număr de 3 variabile analize. Din acest punct, se observă o primă măsură a performanței modelului, și anume OOB_Error(Out of Bag Error), care în situația curentă va fi aproape 58%. Pentru a putea detalia și mai bine variația acestei metrici, ne creăm un parametru de hyper_grid, folosindu-ne și de librăria ranger(specifică lucrului cu modele de Random Forest), unde vom observa cele 240 de valori OOB pentru toate combinațiile din hyper_grid. Variația este una destul de mică, căci cu metodele arrange și top_n putem vedea cele mai bune valori respectiv cele mai puțin bune valori. Astfel, valoarea OOB în toate cazurile se învârtă în jurul a 57-58%, cu 2 extreme: 56,95% respectiv 58,75%.

În continuare, am dorit determinarea unui model optim pentru a putea măsura o acuratețe generală care să ne poată da o idee despre cât de bun este acest model. Ne folosim din nou de metoda ranger din cadrul aceleiași librării pentru a putea crea un model unde să avem următoarele setări: variabila țintă(popularitatea cântecului) depinde de toți factorii, facem antrenarea pe setul de antrenare, dorim să avem 500 de arbori, vom ține cont de 2 variabile la fiecare subset analizat, totodată fiecare arbore va conține o distribuție de aproximativ 70% din setul de antrenament, și în final nu vom dori să calculăm importanța variabilelor(aceste valori au fost determinate ca urmare a unui algoritm prin care am determinat cei mai optimi parametrii ca urmare a analizei performanței arborilor creați). Alăturat, se poate vedea o histogramă a valorilor OOB obținute anterior.



În final, facem predicția pe setul de validare, astfel încât prima măsură a acurateții obținute să respecte metodologie impusă, iar prin matricea de confuzie vom afla datele finale care ne interesează în mod special. Acuratețea modelului este de 42,48%, această valoare aflându-se în cadrul intervalului de încredere(40,56%-44,41%). Valoarea lui P-Value este foarte mică, 3.406e-12, acest lucru însemnând că acuratețea nu este conferită de șansă. Nivelul de specificate se situează pentru cele 3 clase în intervalul (67,25%-75,69%), în timp ce rata de

deteție pozitivă se află în intervalul (41,06%-44,44%), așadar o performanță în continuare scăzută, dar ceva mai bună față de modelele anterioare.

Dacă ar fi să tragem o concluzie având la bază modelele analizare pe arborii de decizie, respectiv cifrele acestora, cel mai bun model este cel realizat cu Random Forest.

Metoda 2 - Naive Bayes

Metoda de clasificare Naive Bayes se bazează pe teorema lui Bayes și consideră că variabilele predictoare sunt independente între ele. Această metodă este folosită pentru clasificarea binară dar și pentru clasificarea mai multor clase, lucru pe care l-am folosit în această lucrare de cercetare.

Am început această metodă prin importarea librăriilor necesare: tidyverse(pentru operațiuni de manipulare a datelor înainte să fie introduse în analiză), rsample(pentru împărțirea setului de date în cele dorite și menționate în metodologie), caret(pentru construirea modelelor pe care ulterior le analizăm) și corrplot(pentru realizarea reprezentărilor vizuale).

La fel ca în cazul metodei prezentate anterior, am efectuat câteva operațiuni necesare: am selectat doar coloanele relevante pentru analiza noastră, am transformat coloana cu durata cântecului în secunde, și am redenumit-o, iar coloanele “key”, “audio_mode” și coloana “time_signature”, toate cu valori nominale, le-am transformat în factori. Ultimul pas pe care l-am făcut a fost modificarea variabilei țintă, din numerică în categorică.

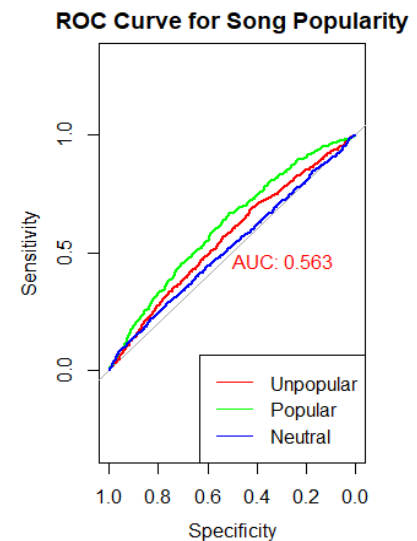
Următorul pas pe care l-am făcut a fost împărțirea setului de date în 3: set de antrenare, set de validare și set de testare, proporțiile fiind de 60%, respectiv 20% pentru ultimele 2 seturi. Ulterior am început construirea modelului. Într-o variabilă denumită features am stocat numele coloanelor din setul de date eliminând numele coloanei țintă “song_popularity” cu ajutorul funcției setdiff urmând ca să stocăm coloanele specificate în features într-o variabilă numită x, iar în altă variabilă numită y am stocat variabila țintă - “song_popularity”. Folosind funcția train, am creat primul model având ca parametri: x - variabilele predictoare, y - variabila țintă și method - “nb”, specificând că metoda de clasificare folosită este Naive Bayes.

Primul model are o acuratețe mică de 37,15% situându-se în intervalul de încredere (35,29 %, 39,05%). Valoarea lui P-Value este de 0.0988, acest lucru însemnând că acuratețea modelului poate fi dată de întâmplare. Nivelul de semnificație se situează pentru cele 3 clase în intervalul (41,47%-87.51%), în timp ce rata de dețție pozitivă se află în intervalul (34.76%-42.89%), arătând o performanță scăzută.

Pentru a îmbunătăți modelul creat, am adăugat un nou parametru funcției train numit tuneGrid, pentru care am creat un data frame în care sunt 3 parametri: usekernel, fL și adjust. În momentul antrenării modelului se vor face combinații între cei 3 parametri rezultând ulterior un model cu cea mai bună combinație. Astfel, modelul obținut prin combinarea celor 3 parametri are o acuratețe mică, de 37,19% situându-se în intervalul de încredere (35,33 %, 39,08%). Valoarea lui P-Value este de 0.0919, acest lucru însemnând că acuratețea modelului poate fi conferită de întâmplare.

Următorul pas pe care l-am făcut este vizualizarea curbei ROC pentru cele 3 clase:

Valoarea AUC(Area Under The ROC Curve) este de 0.563 ceea ce înseamnă că abilitatea modelului de a clasifica pentru cele 3 clase este mică. Valoarea maximă a acestui index este 1, însemnând că modelul poate clasifica perfect cele 3 clase, iar valoarea de 0.5 înseamnă că acuratețea modelului poate fi dată de întâmplare.



Teste finale

Metodă	Acuratețe	Interval încredere	P-Value	Specificitate			Rata predicției pozitive		
				U	N	P	U	N	P
Arbori de decizie	38,89%	37,01%- 40,8%	0.0009161	62,43 %	71,45 %	74,13 %	37,29 %	38,72 %	41,2%
Bagging	41,71%	39,8% - 43,63%	6.835e-10	69,62 %	67,55 %	75,03 %	42,57 %	40,83 %	41,72%
Random Forest	42,48%	40,56%- 44,41%	3.406e-12	70,38 %-	67.25 %	75,69 %	44,44 %	41,06 %	41,84%
Naive Bayes	37,19%	35,33 %- 39,08%	0.0988	41,47 %	78.81 %	87,56 %	43,01 %	40.13 %	34,80%

Legendă: N - clasa neutru, U - clasa nepopular, P - clasa popular

Așa cum se poate observa, metoda prin care am obținut cele mai bune și constante valori este Random Forest. Conform metodologiei stabilite în această lucrare, urmează să aplicăm această metodă pe setul de test, și de acolo să tragem concluziile finale.

Pentru acest demers, ne vom crea un model separat, numit `optimal_model`, unde vom parcurge următoarele setări: popularitatea cântecului va putea depinde de toate variabilele dependente, vom folosi 500 de arbori care vor ține cont pe rând de câte 2 astfel de variabile, iar eșantionul ales pentru mărimea subseturilor va fi 70% din mărimea setului pe care se realizează modelul (setări rezultate din găsirea parametrilor optimi la pasul de Random Forest). În final, realizăm cu funcția `predict` având în vedere modelul nou creat, și setul de test, iar în final, vom realiza matricea de confuzie de unde ne vom extrage concluziile dorite.

Astfel, acuratețea pe setul de testare este în final aproximativ 42% (42,16% mai exact), aceasta regăsindu-se în cadrul intervalului de încredere (40,25% - 44,09%), dar nefiind satisfăcător de mare. Valoarea P-Value este destul de bună, $3,133e-11$, ceea ce sugerează că acuratețea modelului este dată în mică proporție de șansă, iar specificitatea se situează într-un interval de aproximativ 66,5% - 76,5% pentru cele 3 clase, ceea ce ar putea crea impresia unui model mai performant. Dacă este să ne uităm la rata predicției pozitive, vedem un interval destul de restrâns de aproximativ 40% - 43% ceea ce nu prezintă un procent foarte bun în cazul nostru de clasificare și indică o oarecare slăbiciune a performanței generale.

Dacă până acum am discutat metricele pe care le-am mai dezbătut pe parcursul lucrării și pentru celelalte modele, merită să vedem și câteva alte metrice din care să putem trage concluzii. Senzitivitatea, măsură care indică capacitatea modelului creat de a indica corect exemplele pozitive variază între 40,3% și 44%, indicându-se o performanță relativ scăzută pe predicțiile pozitive. Coeficientul Kappa, cel care ilustrează cât de departe ne aflăm de matricea de confuzie, și deci e de dorit să fie cât mai apropiat de 1, este la nivelul de 0.1293, care sugerează că poate exista o discrepanță semnificativă între predicțiile modelului și valorile reale. Când vine vorba de acuratețe, am văzut nivelul general situându-se în jurul nivelului de 44%, însă dacă e să privim la un nivel al acurateții echilibrat (balanced accuracy) pentru fiecare clasă, nivelul crește până în intervalul 53,5% - 59% pentru fiecare clasă luată individual, acest nivel fiind în continuare nesatisfăcător astfel încât să considerăm ca suficient de performant modelul final.

Concluzii

În concluzie, putem spune că după ce am analizat toate metodele și modelele aferente, considerăm că la nivelul datelor pe care le-am avut la dispoziție, un model care analizează popularitatea unui cântec în funcție de factorii săi interni nu este suficient de performant astfel încât să vedem o legătură clară între cele 2 tipuri de variabile analizate: țintă și predictoare. Influența factorilor obiectivi s-a dovedit în urma raportului nostru ca având un aport insuficient de mare pentru a contribui la popularitatea unei melodii.

Este de menționat faptul că o posibilitate pe care o avem în vedere este ca popularitatea unei melodii să fie influențată și de factori subiectivi, cum ar fi prestația live sau carisma artistului, care în corelație cu factorii obiectivi să crească performanța unor modele ce investighează problema de față, însă în cadrul setului nostru de date, nu am avut acces la astfel de măsurători, de altfel greu de cuantificat. Așadar, luați singuri, factorii obiectivi analizați nu au o influență așa de mare asupra popularității, ci aceștia ar putea funcționa, pe model simbiotic, cu niște factori ce țin mai mult de artist, decât de natura și structura cântecului.