Vilniaus Universitetas

Dimensijos mažinimas

Darbą atliko:

Vainius Gataveckas, Matas Gaulia, Dovydas Martinkus

Duomenų Mokslas

3 kursas 2 gr.

Vilnius, 2022

**Turinys**

[1 Tikslas ir uždaviniai 3](#_Toc97985994)

[2 Duomenų aibė 4](#_Toc97985995)

[3 Atliktos analizės aprašymas 5](#_Toc97985996)

[3.1 Aprašomoji statistika 5](#_Toc97985997)

[3.2 PCA 8](#_Toc97985998)

[3.3 MDS 12](#_Toc97985999)

[3.4 t-SNE 15](#_Toc97986000)

[3.5 Metodų palyginimas 18](#_Toc97986001)

[4 Išvados 19](#_Toc97986002)

[Priedas 20](#_Toc97986003)

# Tikslas ir uždaviniai

Tikslas:

Panaudoti dimensijos mažinimo metodus daugiamačių duomenų vizualizavimui, ištirti metodų galimybes bei pateikti pasirinktos aibės vizualizavimo rezultatus ir gautų rezultatų interpretaciją

Uždaviniai:

Pateikti pasirinktos aibės aprašomąją statistiką, aprašyti duomenų aibės specifiką.

Sunormuoti duomenų aibę pagal vidurkį ir dispersiją.

Naudojant tris dimensijos mažinimo metodus sumažinti duomenų aibės dimensiją iki dim=2.

Vizualizuoti dimensijos mažinimo rezultatus ir ištirti, kaip keičiasi vizualizavimo rezultatai, keičiant algoritmų parametrus.

Įvertinti gautus rezultatus ir padaryti išvadas, kuris metodas geriau atvaizduoja rezultatą.

Įvardinti tirtų dimensijos mažinimo metodų privalumus ir trūkumus.

# Duomenų aibė

Spotify Past Decades Songs duomenų aibė

Duomenų aibės šaltinis: Kaggle

Nuoroda per internetą: <https://www.kaggle.com/cnic92/spotify-past-decades-songs-50s10s?select=1990.csv>

Duomenų aibę sudaro tokie požymiai:

„Number“ – (kategorinis, nominalusis) dainą identifikuojantis kodas

„Title“ – (kategorinis, nominalusis) dainos pavadinimas

„Artist“ – (kategorinis, nominalusis) atlikėjas arba grupė

„Top Genre“ – (kategorinis, nominalusis) dainos žanras

„Year“ – (kiekybinis, diskretusis, intervalinė skalė) išleidimo metai

„Decade“ – (kiekybinis, diskretusis, intervalinė skalė) išleidimo dešimtmetis

„BPM“ – (kiekybinis, tolydus, santykių skalė) dainos tempas

“Loudness (dB)“ - (kiekybinis, tolydus, intervalų skalė) dainos garsumas

„Duration“ – (kiekybinis, tolydus, santykių skalė) dainos trukmė

„Energy“ – (kiekybinis, tolydus, santykių skalė) dainos energija

„Danceability“ – (kiekybinis, tolydus, santykių skalė) lengvumas šokti pagal dainą

„Liveness“ – (kiekybinis, tolydus, santykių skalė) kaip tikėtina, kad daina yra gyvas įrašas

„Valence“ – (kiekybinis, tolydus, santykių skalė) dainos pozityvumas

„Acousticness“ – (kiekybinis, tolydus, santykių skalė) dainos akustiškumas

„Speechiness“ – (kiekybinis, tolydus, santykių skalė) kiek dainoje yra kalbama

„Popularity“ - (kiekybinis, tolydus, santykių skalė) dainos populiarumas pagal perklausų skaičių

# Atliktos analizės aprašymas

## Aprašomoji statistika

Duomenų aibę sudaro trys klasės pagal dainos išleidimo dešimtmetį (požymis „Decade“). Kiekvienai klasei priklausančių objektų kiekis pateiktas lentelėje (žr. 1 lentelė lentelė).

1 lentelė Objektų kiekis duomenų aibėje pagal dešimtmetį

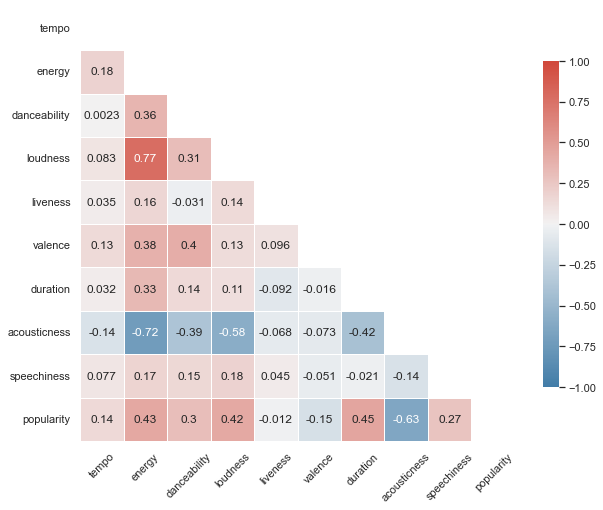
|  |  |
| --- | --- |
| Dešimtmetis | Skaičius |
| 2010-ieji | 100 |
| 1980-ieji | 105 |
| 1950-ieji | 73 |

Duomenų aibės skaitiniams požymiams apskaičiuotos pagrindinės aprašomosios statistikos charakteristikos (standartinis nuokrypis, vidurkis, mediana, mažiausia reikšmė (min), didžiausia reikšmė (max), pirmas ir trečias kvartiliai). Rezultatai pateikti lentelėje (žr. 2 lentelė). Papildomai pateiktos aprašomosios statistikos kiekvienam skaitiniam požymiui pagal dešimtmetį (žr. 1 priedas) Visi skaitiniai požymiai išskyrus „Tempo“, „Loudness“ ir „Duration“ matuoti skalėje nuo 0 iki 100.

2 lentelė Aprašomosios statistikos charakteristikos duomenų aibei

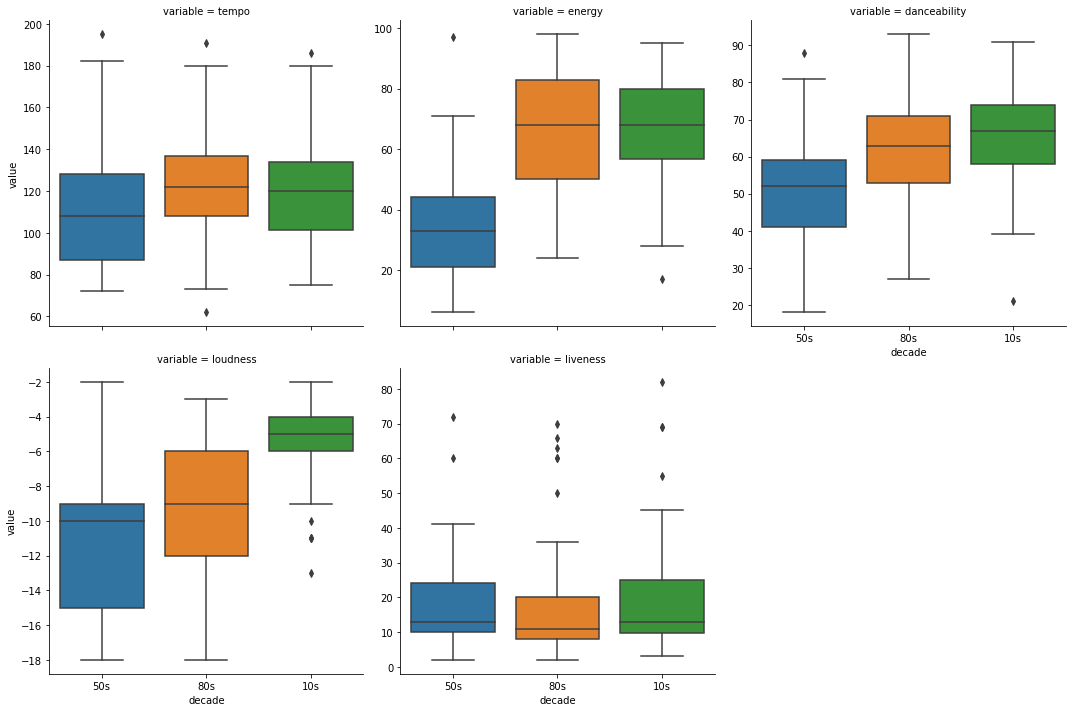
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | mean | std | min | 25% | 50% | 75% | max |
| tempo | 118.2 | 25.3 | 62.0 | 100.0 | 117.0 | 135.0 | 195.0 |
| energy | 58.1 | 22.8 | 6.0 | 41.2 | 60.5 | 78.8 | 98.0 |
| danceability | 60.4 | 14.3 | 18.0 | 51.0 | 62.0 | 71.0 | 93.0 |
| loudness | -8.5 | 4.0 | -18.0 | -11.0 | -8.0 | -5.0 | -2.0 |
| liveness | 17.4 | 13.4 | 2.0 | 9.0 | 13.0 | 22.0 | 82.0 |
| valence | 55.6 | 25.0 | 9.0 | 34.0 | 55.0 | 77.8 | 99.0 |
| duration | 212.3 | 56.5 | 98.0 | 174.2 | 210.0 | 245.0 | 433.0 |
| acousticness | 33.7 | 31.1 | 0.0 | 7.0 | 20.5 | 61.0 | 100.0 |
| speechiness | 5.8 | 5.4 | 2.0 | 3.0 | 4.0 | 6.0 | 46.0 |
| popularity | 63.6 | 16.6 | 26.0 | 54.0 | 68.0 | 76.0 | 94.0 |

Tarp skaitinių rodiklių apskaičiuotos Pirsono koreliacijos koeficientų reikšmės (angl. Pearson correlation coefficient). Rasta stipri teigiama koreliacija tarp dainos garsumo (požymis „Loudness“) ir energijos („Energy“) (r = 0.77). Dainos akustiškumas (požymis „Acousticness“) neigiamai koreliuoja su beveik visais kitais požymiais. Iš jų didžiausia neigiama koreliacija su požymiais „Energy“ (r=-0.72) „Popularity“ (r=-0.63) ir „Loudness“ (r=-0.58). Rezultatai pateikti koreliacijų diagrama (žr. 1 pav.)

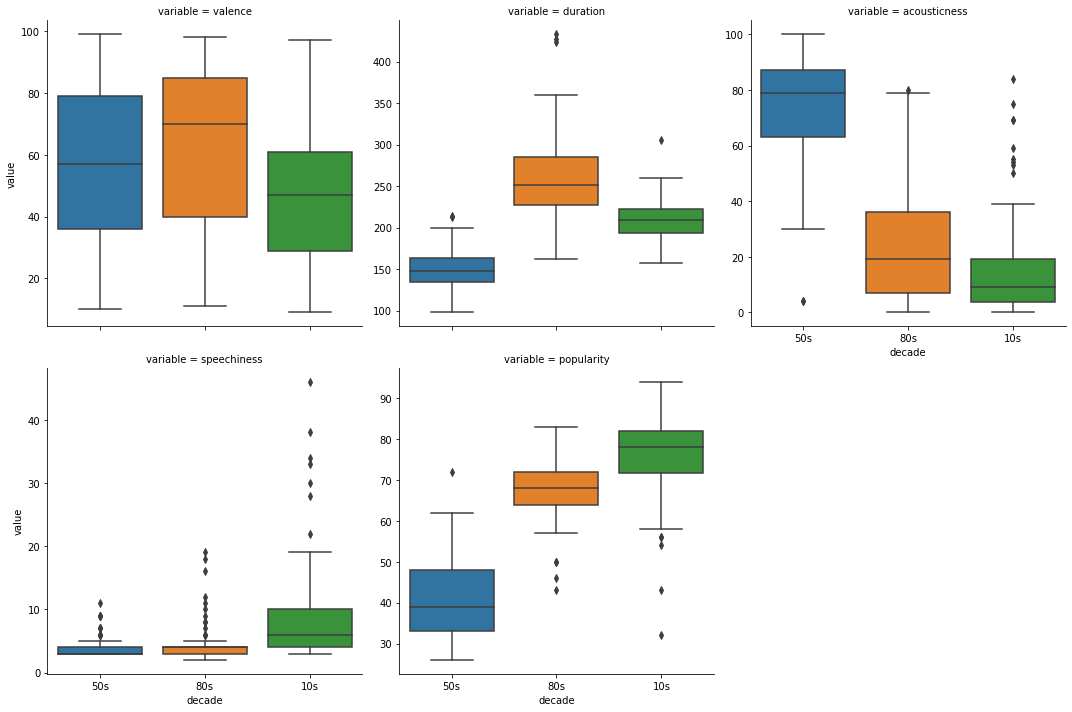


1 pav. Pirsono koreliacijos tarp požymių koeficientai

Kiekvieno skaitinio duomenų aibės požymio pasiskirstymas pagal dešimtmetį pavaizduotas stačiakampėmis diagramomis (žr. 2 pav. ir 3 pav.). Pastebėtas kad vėlesni dešimtmečiai pasižymi didėjančiomis garsumo, tinkamumo šokti, perklausų skaičiaus reikšmėmis, bet mažėjančiomis dainų akustiškumo reikšmėmis.



2 pav. Skaitinių požymių stačiakampės diagramos pagal dešimtmetį



3 pav. Skaitinių požymių stačiakampės diagramos pagal dešimtmetį (2 dalis)

Kaip matoma iš aprašomosios statistikos charakteristikų ir stačiakampių grafikų, požymiai „Duration“, „Loudness“ ir „Tempo“ matuoti kitokio dydžio skalėse negu likusieji skaitiniai duomenų aibės požymiai. Laikyta, kad šie skalių skirtumai neigiamai įtakos dimensijos mažinimo metodų rezultatus, todėl duomenys sunormuoti naudojant normavimą pagal vidurkį ir dispersiją (standartizavimas) , kur - požymio vidurkis požymio dispersija,   – požymio vidurkis.

## PCA

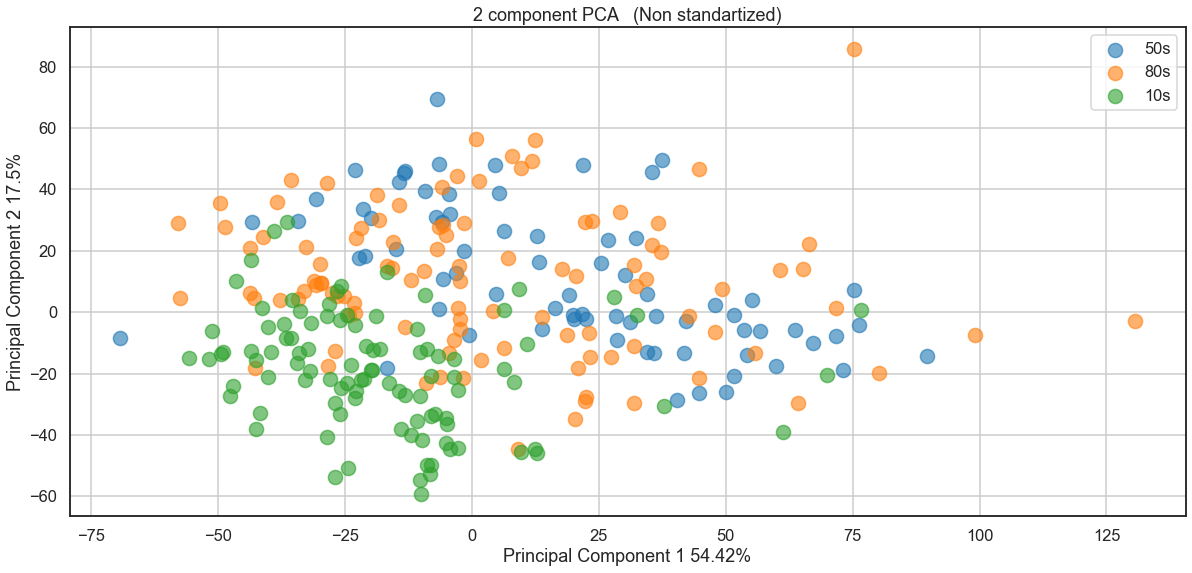
Pagrindinių komponenčių analizė (angl. Principal Component Analysis, toliau - PCA) yra tiesinis dimensijos mažinimo metodas, kuriame ieškoma krypties, kuria dispersija yra didžiausia (Principal Component, toliau - PC).

PCA metodas neturi svarbių parametrų, kuriuos keičiant būtų gaunami skirtingi rezultatai.

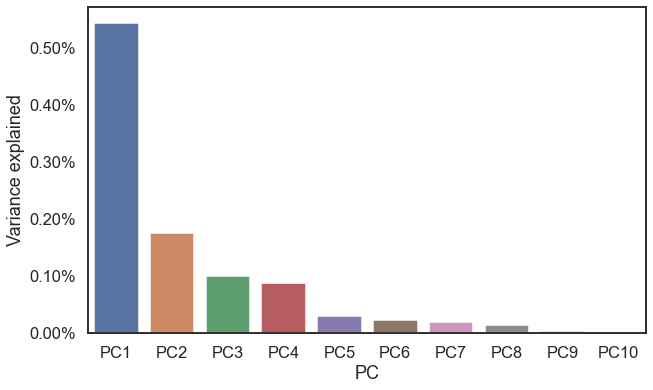
Pirmiausia PCA atlikta su nestandartizuota duomenų aibe (pirma pagrindine komponente PC1 paaiškinama 0.54 visos dispersijos, PCA2 - 0.17, žr. 6 pav.). Paliktos ir vizualizuotos pirmos dvi pagrindinės komponentės (žr. 5 pav.). Gautuose rezultatuose visos klasės persidengia. 2010-ųjų dainų klasė nuo likusiųjų atskiriama šiek tiek stipriau, tačiau 50-ųjų ir 80-ųjų dainos stipriai maišosi tarpusavyje.

PCA pakartotinai atlikta su standartizuota duomenų aibe (paaiškinama dalis dispersijos: PC1 - 0.34, PC2 - 0.14, žr. 7 pav.) Pastebimas rezultatų pagerėjimas – 50-ųjų ir 80-ųjų dainos mažiau maišosi tarpusavyje (žr. 7 pav.).

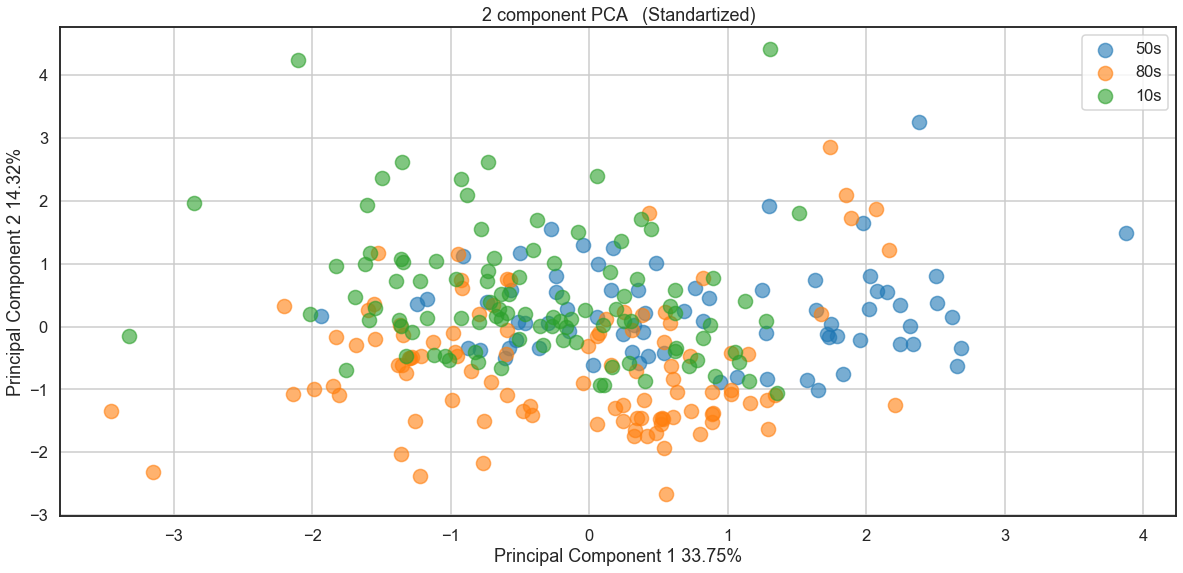
Palyginimui PCA atlikta paėmus požymių poaibį be požymių „Tempo“, „Liveness“ ir „Valence“, kurie pagal stačiakampes diagramas dešimtmečius atskiria mažiausiai (paaiškinama dalis dispersijos: PC1 - 0.47, PC2 - 0.15, žr. 10 pav.). Gautuose rezultatuose kiti dešimtmečiai stipriai nesimaišo su 80-ųjų dainomis, gautuose rezultatuose matomos susidariusios išskirtys (žr. 9 pav.).



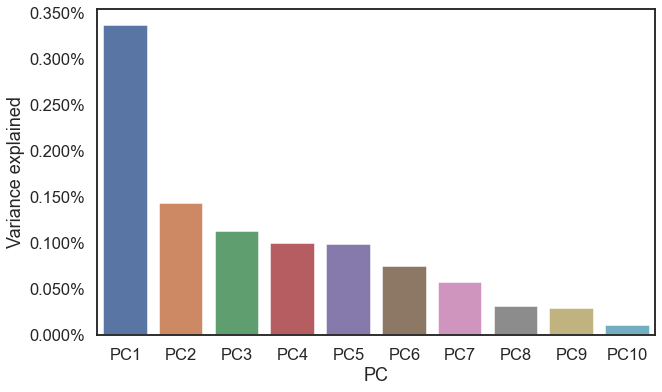
4 pav. PCA su nestandartizuotais duomenimis



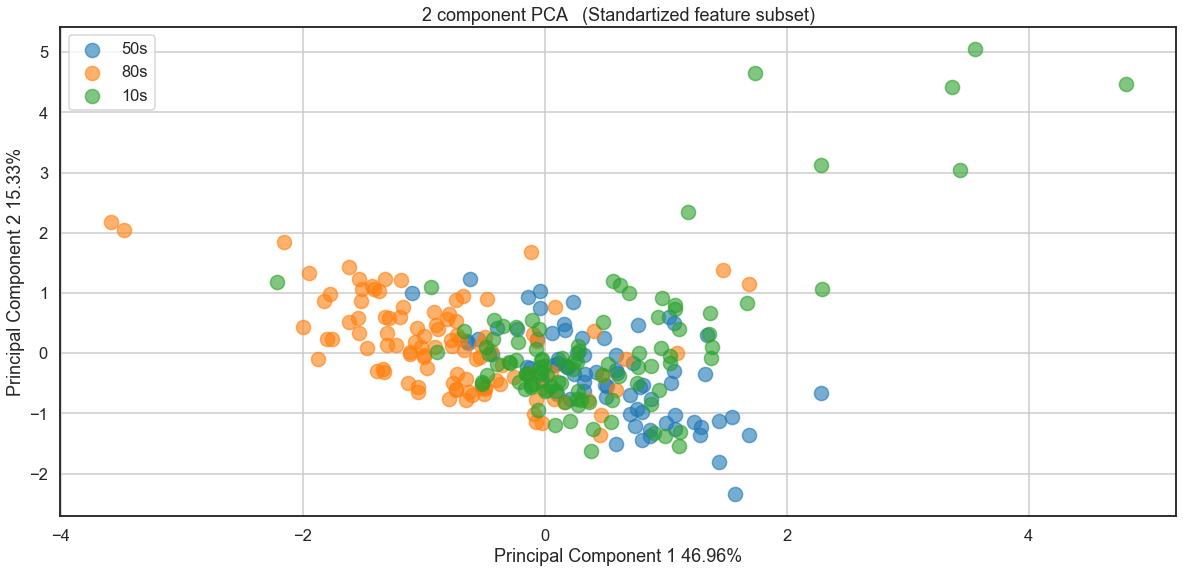
5 pav. PC paaiškinama dalis dispersijos nestandartizuotiems duomenims



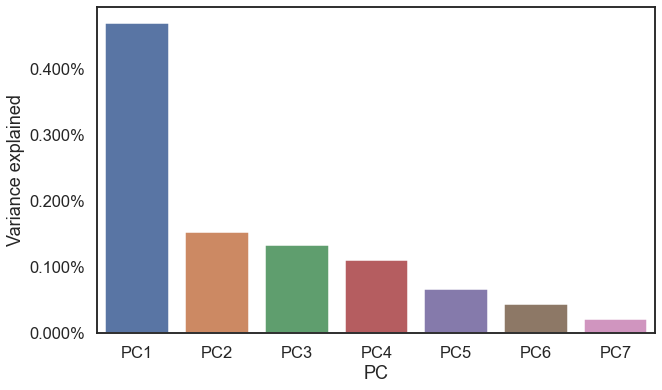
6 pav. PCA su standartizuotais duomenimis



7 pav. PC paaiškinama dalis dispersijos standartizuotiems duomenims



8 pav. PCA su standartizuotais duomenimis požymių poaibiui



9 pav. PC paaiškinama dalis dispersijos požymių poaibiui

## MDS

Daugiamatės skalės (angl. Multidimensional Scaling, toliau – MDS) yra netiesinis dimensijos mažinimo metodas. MDS ieškoma daugiamačių duomenų projekcijų mažesnės dimensijos erdvėje, siekiant išlaikyti atstumus tarp objektų.

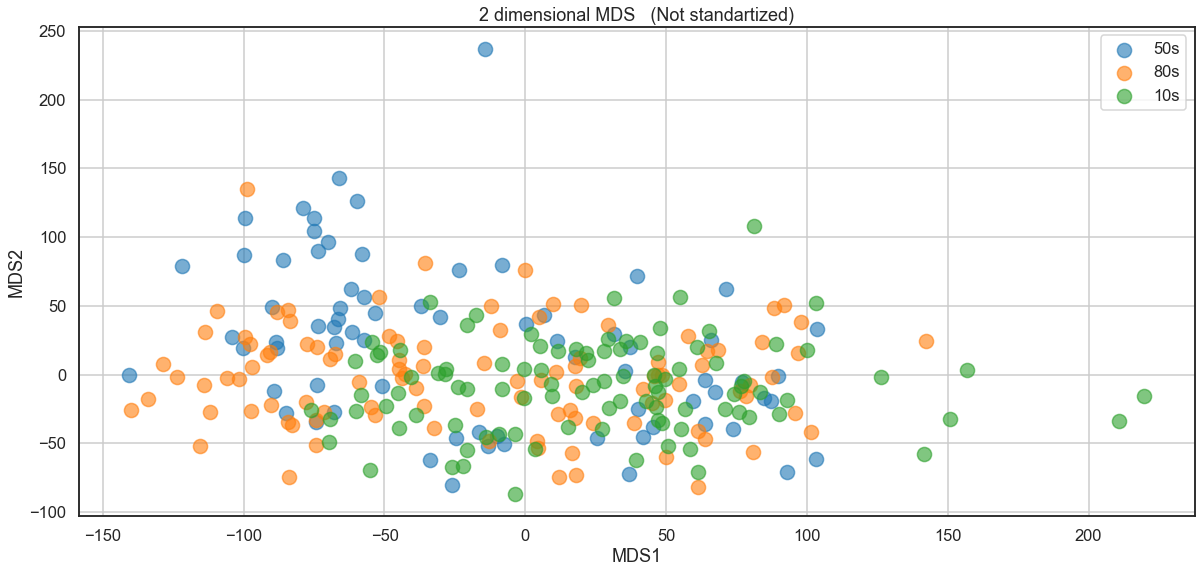
Pagrindinis parametras MDS yra pasirinkimas tarp metrikinės (angl. metric) ir nemetrikinės (angl. non-metric) MDS naudojimo.

Kaip ir naudojant PCA lyginami rezultatai naudojant nestandartizuotą, standartizuotą duomenų aibes, imant požymių poaibį. Dimensijos dydis sumažintas iki dim=2.

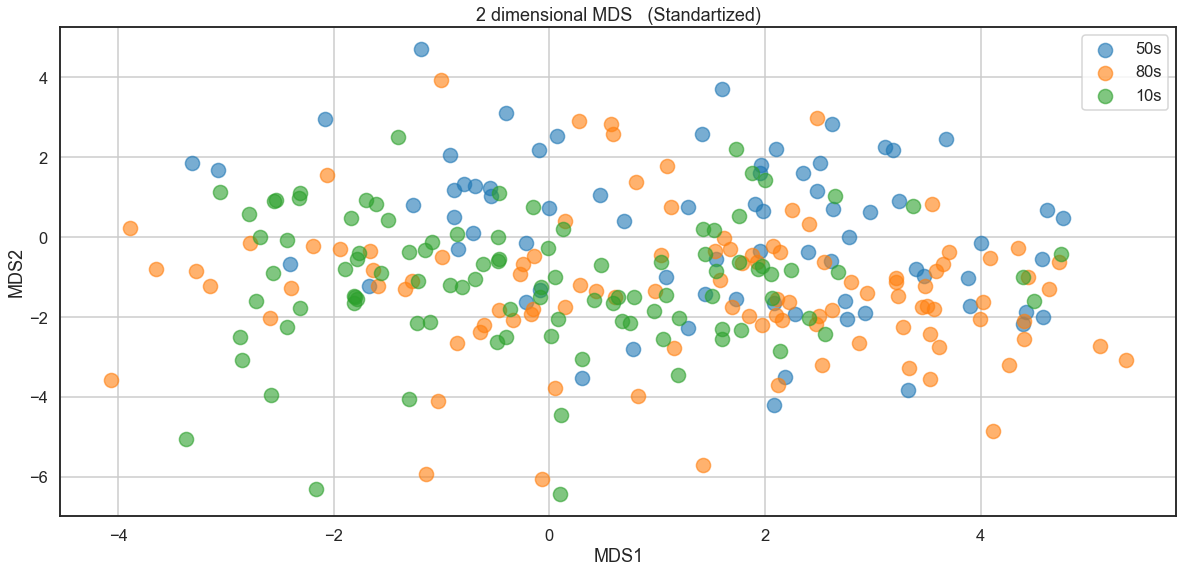
Naudojant metrikinę MDS su Euklidiniais atstumais tiek su nestandartizuota duomenų aibe (žr. 11), tiek su standartizuota (žr. 12) klasių tarpusavio persidengimas labai didelis.

Naudojant nemetrinę MDS gaunami stipriai prastesni rezultatai – užpildoma visa grafiko erdvė, klasės pasiskirsčiusios be jokios struktūros (žr. 13 pav.)

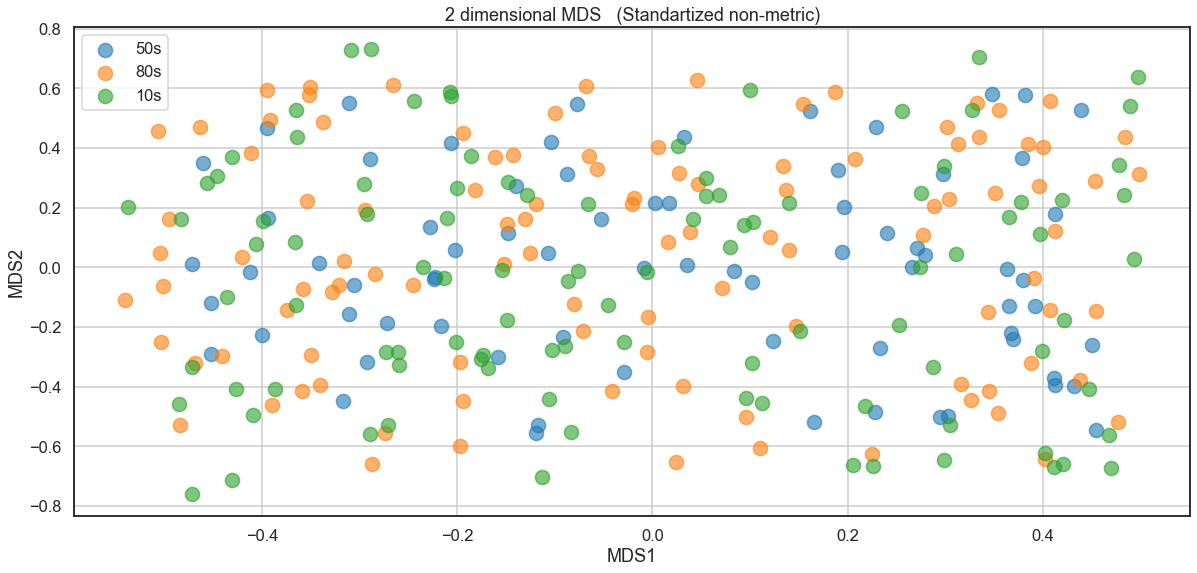
Atlikus metrikinę MDS su Euklidiniais atstumais standartizuotų požymių poaibiui, klasių atsiskyrimo prasme šiek tiek pagerinami rezultatai, gauti imant visą požymių aibę (žr. 14 pav.).



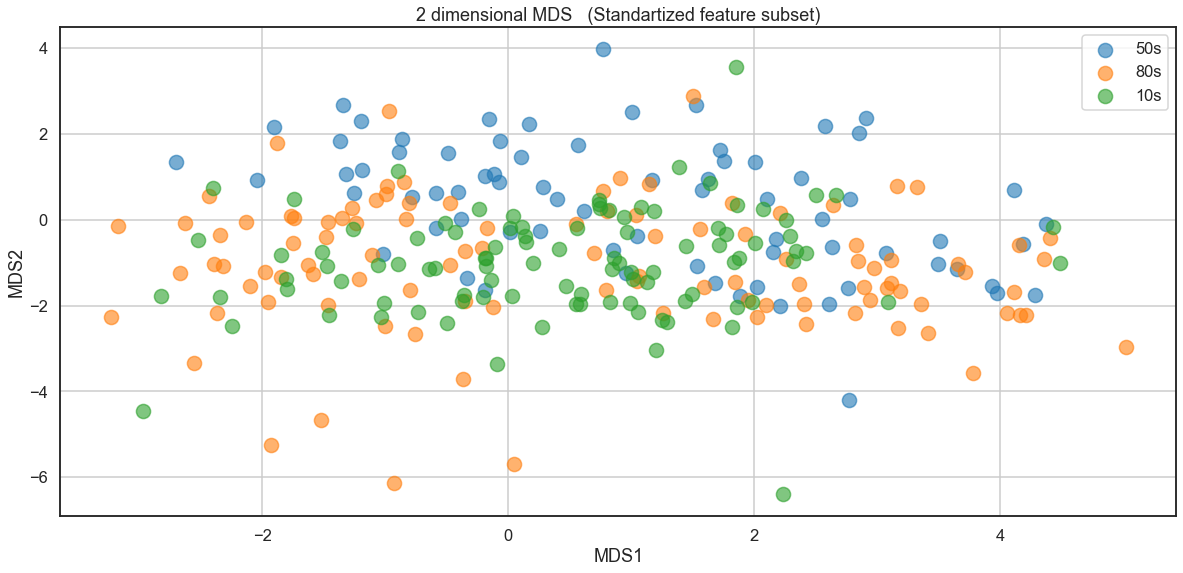
10 pav. MDS su nestandartizuotais duomenimis



11 pav. MDS su standartizuotais duomenimis



12 pav. Nemetrikinės MDS su standartizuotais duomenimis



13 pav. MDS su standartizuotais duomenis naudojant požymių poaibį

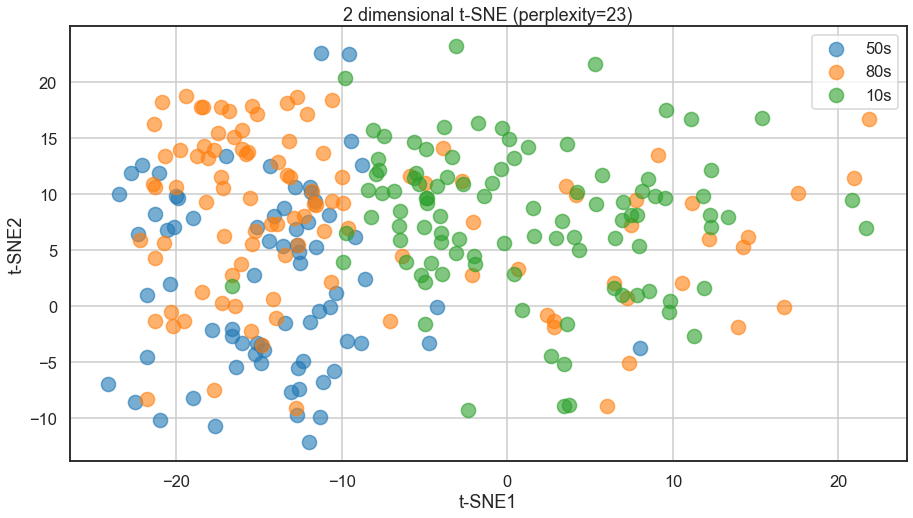
## t-SNE

t-SNE yra netiesinis dimensijos mažinimo metodas, kuris mažesnės dimensijos erdvėje išlaikyti kuo tikslesnį taškų pasiskirstymą atitinkantį daugiamatės erdvės taškų pasiskirstymą. t-SNE siekia išsaugoti kiekvieno taško kaimynus (orientuotas į vidinės struktūros išsaugojimą).

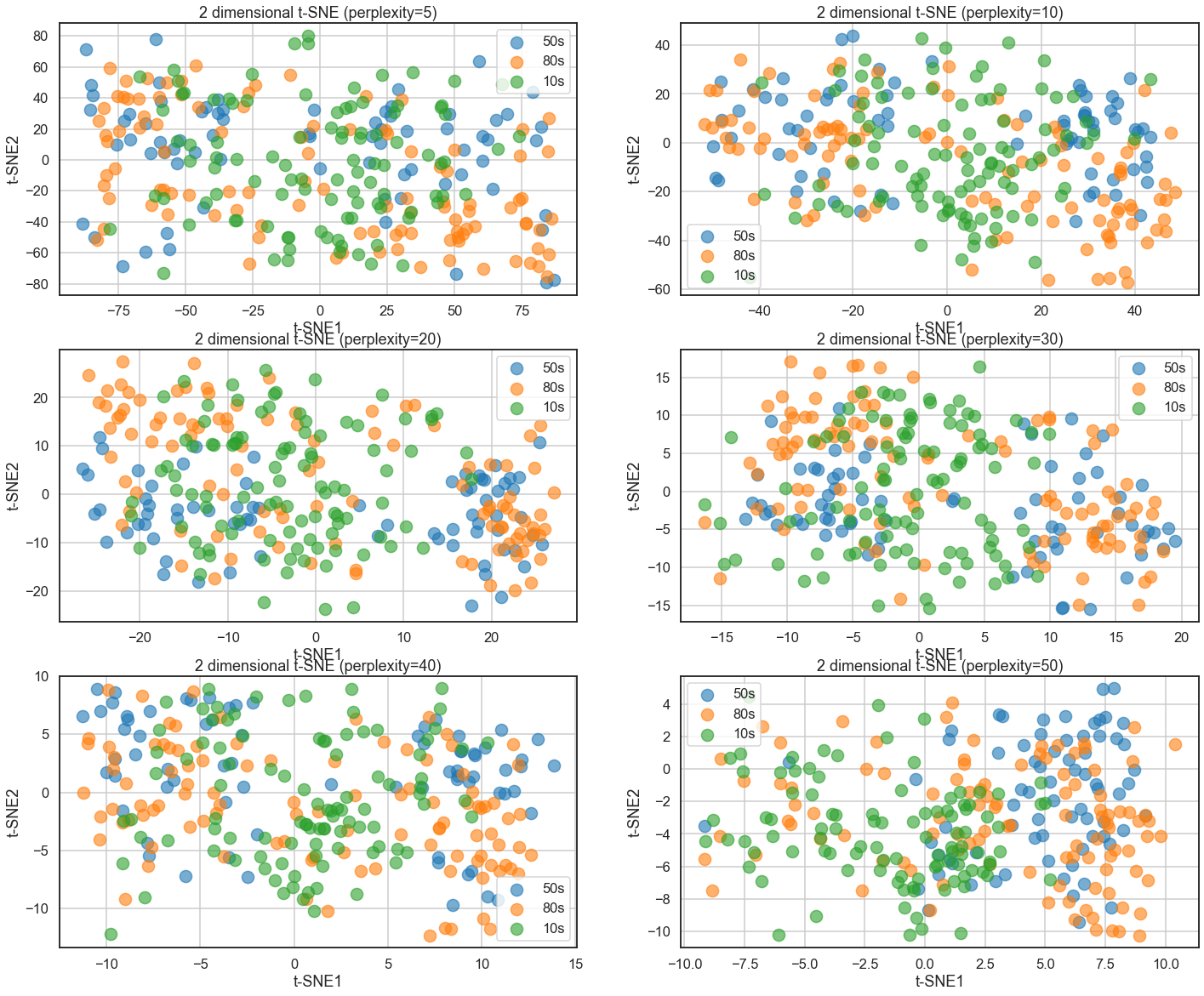
Pagrindinis metodo parametras vadinamas Perplexity ir yra susijęs su duomenų aibės objektų kaimynų skaičiumi. Su mažesnėmis parametro reikšmėmis didesnis dėmesys skiriamas vietinėms struktūroms, su didesnėmis – globalioms. Rekomenduojamos reikšmės nuo 5 iki 50. Didesnėms duomenų aibėms įprastai naudojamos didesnės reikšmės.

Naudojant t-SNE dimensijos dydis sumažintas iki dim=2. Standartizuotai duomenų aibei eksperimentiškai geriausi rezultatai rasti parametro reikšmėms esant 22-25 intervale (žr. 15 pav.). Su kitomis parametro reikšmėmis gaunamas mažesnis klasių atsiskyrimas arba jo negauta išvis (žr. 16 pav.).

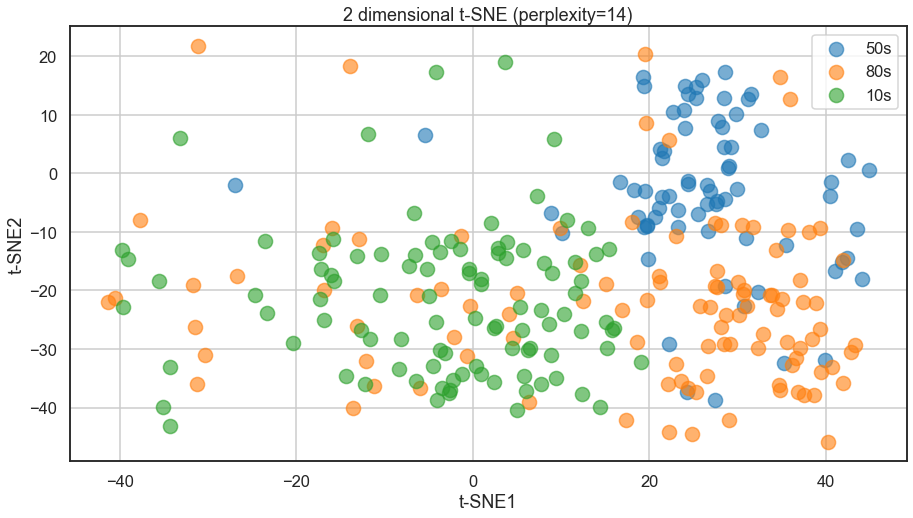
Naudojant požymių poaibį ir Perplexity reikšmes parinkus 12-14 intervale, klasių atsiskyrimo prasme pagerinamas geriausias visų požymių aibės rezultatas, tačiau klasės tarpusavyje vis tiek persikerta (žr. 17 pav.).



14 pav. t-SNE standartizuotiems duomenims



15 pav. t-SNE standartizuotiems duomenims su skirtingomis perplexity parametro reikšmėmis



16 p­­av. t-SNE požymių poaibiui

## Metodų palyginimas

PCA yra dažniausiai naudojamas dimensijos mažinimo metodas, su kurio yra lyginami kitų dimensijos mažinimo metodų rezultatai.

Jei tarp požymių yra tiesinė priklausomybė, tai taikant pagrindinių komponenčių metodą, duomenų matmenų skaičius mažinamas su nedidelėmis paklaidomis. Tačiau bendru atveju gali egzistuoti netiesinės priklausomybės, kurių PCA metodas negali įvertinti.

PCA stipriai įtakoja taškai atsiskyrėliai, nes jie įtakoja pagrindinėms koordinatėms gauti naudojamos kovariacinės matricos gaunamas reikšmes.

Turimiems duomenims atliekant PCA tiek naudojant visą standartizuotą požymių aibę, tiek jos poaibį gauti gana ryškūs, tačiau persidengiantys klasteriai. Kaip ir tikėtasi, su nestandartizuota duomenų aibe gauti prastesni rezultatai.

MDS gali išsaugoti netiesinę duomenų topologiją. Šis metodas nėra stipriai veikiamas taškų atsiskyrėlių kaip PCA.

Naudojant metrikinę MDS su Euklidiniais atstumais negauta gerų klasių atsiskyrimo rezultatų.

Nemetrikinė MDS gali būti naudojama bet kokiai nepanašumų matricai apskaičiuotai iš kokybinių, kiekybinių, ar maišyto tipo požymių rinkinio, tai pat situacijose kai turimi subjektyvūs objektų tarpusavio panašumo vertinimai, turimi tik nepanašumų rangai, kitose panašiose situacijose. Turimos duomenų aibės atveju neišnaudojami šios MDS versijos privalumai, be to siekiant išlaikyti tik atstumų rangus mažesnės dimensijos erdvėje prarandama didelė dalis informacijos, todėl šiuo metodu gaunami prasčiausi rezultatai klasių atsiskyrimo atžvilgiu.

Apskritai naudojant MDS gauti prasčiausi klasių atsiskyrimo rezultatai iš naudotų metodų.

t-SNE siekiama išsaugoti taškų kaimynus, tačiau galimas informacijos praradimas globaliose struktūrose, pavyzdžiui gavus gerai atsiskyrusius klasterius atstumų tarp jų dydžiai neturi interpretuojamos prasmės.

t-SNE yra stipriai lankstus metodas, gebantis rasti struktūrą ten, kur to nesugeba padaryti kiti metodai. Tačiau gaunami rezultatai stipriai priklauso nuo pasirinktos Perplexity parametro reikšmės. Parinkus žemą Perplexity reikšmę, pradedami sudaryti mažo dydžio klasteriai, pavyzdžiui paprastas triukšmas duomenyse gali būti atvaizduojamas kaip turintis struktūrą.

Turimai standartizuotai duomenų aibei mažesnis klasių persidengimas gautas naudojant Perplexity reikšmes 22-25 intervale, o imant tik standartizuotų duomenų požymių poaibį – intervale 12-14. Naudojant kitas parametro reikšmes gauti prasti rezultatai.

# Išvados

Duomenų aibę sudaro duomenys apie 100 2010-ųjų, 105 1980-ųjų ir 73 1950-ųjų dainas. Šio požymio reikšmės laikytos klasėmis duomenų aibėje.

Rasta stipri teigiama koreliacija tarp dainos garsumo (požymis „Loudness“) ir energijos („Energy“) (r = 0.77). Taip pat pastebėta, kad dainos akustiškumas (požymis „Acousticness“) neigiamai koreliuoja su beveik visais kitais požymiais. Iš jų didžiausia neigiama koreliacija su požymiais „Energy“ (r=-0.72) „Popularity“ (r=-0.63) ir „Loudness“ (r=-0.58).

Kadangi požymiai „Duration“, „Loudness“ ir „Tempo“ matuoti kitokio dydžio skalėse negu likusieji skaitiniai duomenų aibės požymiai, duomenų aibė normuota naudojant standartizavimo metodą.

Palyginus pasiskirstymą pagal dešimtmetį rasta, kad požymių „Tempo“, „Liveness“ ir „Valence“ pasiskirstymai tarp klasių skiriasi mažai, todėl pasirinkta papildomai atlikti dimensijos mažinimą naudojant požymių poaibį be šių požymių.

Skirtingais dimensijos mažinimo metodais požymių aibę sumažinta iki dim=2 ir gauti rezultatai vizualizuoti. Su visais naudotais metodais negautas pilnas klasių atsiskyrimas nė vienai klasių porai.

PCA atlikus standartizuotai duomenų aibei (paaiškinta dalis variacijos: PC1 - 0.34, PC2 - 0.14), standartizuotam prieš tai minėtam požymių poaibiui (paaiškinta dalis variacijos: PC1 - 0.47, PC2 - 0.15) ir rezultatus vizualizavus, abejais atvejais klasės neatsiskiria, tačiau pastebimi gana ryškūs susidarę klasteriai. Naudojant nestandartizuotą duomenų aibę gauti prastesni rezultatai kaip ir tikėtasi.

MDS metodu požymius sumažinus iki dvimačių ir juos pavaizdavus grafiškai klasės tarpusavyje atskiriamos prasčiausiai iš naudotų metodų.

Naudojant t-SNE metodą eksperimentiniu būdu rasta, kad Perplexity reikšmės intervale 22-25 geriausiai atskiria klases standartizuotai duomenų aibei, reikšmės intervale 12-14 - imant standartizuotų duomenų požymių poaibį. Su šiomis parametro reikšmėmis gautas klasių atsiskyrimas prilygsta (potencialiai pagerina) rezultatams, gautiems PCA atveju.

# Priedas

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | decade | mean | std | min | 25% | 50% | 75% | max |
| tempo | 10s | 118.7 | 22.4 | 75.0 | 101.5 | 120.0 | 134.0 | 186.0 |
| tempo | 50s | 111.2 | 28.0 | 72.0 | 87.0 | 108.0 | 128.0 | 195.0 |
| tempo | 80s | 122.6 | 25.1 | 62.0 | 108.0 | 122.0 | 137.0 | 191.0 |
| energy | 10s | 68.0 | 16.3 | 17.0 | 56.8 | 68.0 | 80.0 | 95.0 |
| energy | 50s | 34.9 | 17.4 | 6.0 | 21.0 | 33.0 | 44.0 | 97.0 |
| energy | 80s | 64.9 | 20.2 | 24.0 | 50.0 | 68.0 | 83.0 | 98.0 |
| danceability | 10s | 65.4 | 11.9 | 21.0 | 58.0 | 67.0 | 74.0 | 91.0 |
| danceability | 50s | 51.0 | 14.5 | 18.0 | 41.0 | 52.0 | 59.0 | 88.0 |
| danceability | 80s | 62.3 | 13.4 | 27.0 | 53.0 | 63.0 | 71.0 | 93.0 |
| loudness | 10s | -5.5 | 2.0 | -13.0 | -6.0 | -5.0 | -4.0 | -2.0 |
| loudness | 50s | -11.6 | 3.4 | -18.0 | -15.0 | -10.0 | -9.0 | -2.0 |
| loudness | 80s | -9.1 | 3.7 | -18.0 | -12.0 | -9.0 | -6.0 | -3.0 |
| liveness | 10s | 17.9 | 13.8 | 3.0 | 9.8 | 13.0 | 25.0 | 82.0 |
| liveness | 50s | 18.2 | 12.0 | 2.0 | 10.0 | 13.0 | 24.0 | 72.0 |
| liveness | 80s | 16.4 | 13.9 | 2.0 | 8.0 | 11.0 | 20.0 | 70.0 |
| valence | 10s | 46.4 | 20.9 | 9.0 | 29.0 | 47.0 | 61.0 | 97.0 |
| valence | 50s | 57.6 | 25.1 | 10.0 | 36.0 | 57.0 | 79.0 | 99.0 |
| valence | 80s | 63.1 | 25.9 | 11.0 | 40.0 | 70.0 | 85.0 | 98.0 |
| duration | 10s | 209.2 | 24.1 | 157.0 | 194.0 | 209.0 | 222.0 | 306.0 |
| duration | 50s | 149.5 | 22.5 | 98.0 | 135.0 | 148.0 | 163.0 | 214.0 |
| duration | 80s | 258.9 | 51.2 | 162.0 | 227.0 | 251.0 | 285.0 | 433.0 |
| acousticness | 10s | 14.7 | 17.9 | 0.0 | 3.8 | 9.0 | 19.0 | 84.0 |
| acousticness | 50s | 73.0 | 20.1 | 4.0 | 63.0 | 79.0 | 87.0 | 100.0 |
| acousticness | 80s | 24.6 | 21.7 | 0.0 | 7.0 | 19.0 | 36.0 | 80.0 |
| speechiness | 10s | 8.4 | 7.8 | 3.0 | 4.0 | 6.0 | 10.0 | 46.0 |
| speechiness | 50s | 4.2 | 1.8 | 3.0 | 3.0 | 3.0 | 4.0 | 11.0 |
| speechiness | 80s | 4.4 | 2.8 | 2.0 | 3.0 | 4.0 | 4.0 | 19.0 |
| popularity | 10s | 76.0 | 9.2 | 32.0 | 71.8 | 78.0 | 82.0 | 94.0 |
| popularity | 50s | 40.9 | 10.4 | 26.0 | 33.0 | 39.0 | 48.0 | 72.0 |
| popularity | 80s | 67.7 | 7.3 | 43.0 | 64.0 | 68.0 | 72.0 | 83.0 |

1 priedas Aprašomosios statistikos charakteristikos duomenų aibei pagal dešimtmetį

Žemiau pateiktas naudotas programinis kodas:

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import seaborn as sns

def read\_clean\_data(filename):

d = pd.read\_csv(filename)[['title','artist','year','bpm', 'nrgy', 'dnce', 'dB','live', 'val', 'dur','acous', 'spch','pop']]

d = d.rename({'bpm':'tempo','nrgy':'energy','dnce':'danceability','dB':'loudness','live':'liveness',

'val':'valence','dur':'duration','acous':'acousticness','spch':'speechiness','pop':'popularity'},

axis = 1)

d['decade'] = filename[2:4] + 's'

return d

filenames = ['1950.csv','1980.csv','2010.csv']

df = pd.concat([read\_clean\_data(i) for i in filenames]).reset\_index()

df\_id = df[["title","artist"]]

df = df.iloc[:,4:]

df.groupby('decade')['tempo'].count()

sns.set\_context("talk")

sns.catplot(x="decade",kind="count",data=df)

sns.set\_theme(style="white")

corr = df\_features.corr()

mask = np.triu(np.ones\_like(corr, dtype=bool))

f, ax = plt.subplots(figsize=(10,8))

cmap = sns.diverging\_palette(240,15,as\_cmap=True)

plot=sns.heatmap(corr,mask=mask,vmax=1,vmin=-1,center=0,cmap=cmap,square=True,cbar\_kws={"shrink":.5},

linewidth=1,ax=ax,annot=True)

plot.tick\_params(axis='x', rotation=45)

df\_long = df.iloc[:,:5]

df\_long["decade"] = df.iloc[:,-1]

df\_long = df\_long.melt("decade")

sns.catplot(x="decade",y="value",data=df\_long,col="variable",kind="box",col\_wrap=3,sharey=False)

df\_long = df.iloc[:,5:]

df\_long = df\_long.melt("decade")

sns.catplot(x="decade",y="value",data=df\_long,col="variable",kind="box",col\_wrap=3,sharey=False)

df.describe().T.drop("count",axis=1)

summaries = df.groupby("decade").describe().unstack()

summaries = summaries.unstack(-2).reset\_index(1).drop("count",axis=1)

# ## PCA

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

def plot\_pca(x\_pca, exvar,title):

pc1 = str(round(100\*exvar[0], 2))

pc2 = str(round(100\*exvar[1], 2))

fig = plt.figure(figsize=(20, 9))

ax = fig.add\_subplot(1,1,1)

ax.set\_xlabel("Principal Component 1 " + pc1 + "%")

ax.set\_ylabel("Principal Component 2 " + pc2 + "%")

ax.set\_title("2 component PCA" + f" ({title})")

targets = ['50s', '80s', '10s']

colors = ['tab:blue', 'tab:orange', 'tab:green',]

for target, color in zip(targets,colors):

indicesToKeep = x\_pca['key'] == target

ax.scatter(x\_pca.loc[indicesToKeep, 'PC1']

, x\_pca.loc[indicesToKeep, 'PC2']

, c = color

, s = 200

, alpha = 0.6)

ax.legend(targets)

ax.grid()

def do\_pca(df, standartize = False,title=""):

x = df.loc[:, df.columns[:-1]].values

y = df.loc[:,['decade']].values

if standartize:

x = StandardScaler().fit\_transform(x)

x = pd.DataFrame(x)

pca = PCA()

x\_pca = pca.fit\_transform(x)

x\_pca = pd.DataFrame(x\_pca)

exvar = pca.explained\_variance\_ratio\_

explained = pd.DataFrame(

{"explained":exvar, "PC":["PC"+str(i) for i in range(1, len(exvar)+1)]})

print(explained)

fig = plt.figure(figsize=(10, 6))

ax = fig.add\_subplot(1,1,1)

ax=sns.barplot(x="PC",y="explained",data=explained,ax=ax)

ax.set\_ylabel("Variance explained")

ax.yaxis.set\_major\_formatter(mpl.ticker.PercentFormatter())

x\_pca['key']= y

names = ["PC" + str(i) for i in range(len(x\_pca.columns)-1)]

names.append("key")

x\_pca.columns = names

plot\_pca(x\_pca, exvar,title)

df\_small = df.drop(columns=["tempo","liveness","valence"])

# #### not standartized

do\_pca(df, False,"Non standartized")

# #### standartized

do\_pca(df, True, "Standartized")

do\_pca(df\_small,True, "Standartized feature subset")

# ## MDS

import plotly.express as px

from sklearn.manifold import MDS

from sklearn.metrics import euclidean\_distances

def plot\_mds(df,title):

fig = plt.figure(figsize=(20, 9))

ax = fig.add\_subplot(1,1,1)

ax.set\_xlabel("MDS1")

ax.set\_ylabel("MDS2")

ax.set\_title("2 dimensional MDS" + f" ({title})")

targets = ['50s', '80s', '10s']

colors = ['tab:blue', 'tab:orange', 'tab:green',]

for target, color in zip(targets,colors):

indicesToKeep = df['key'] == target

ax.scatter(df.loc[indicesToKeep, 'PC1']

, df.loc[indicesToKeep, 'PC2']

, c = color

, s = 200

, alpha = 0.6)

ax.legend(targets)

ax.grid()

def do\_mds(df, standartize = False,metric=True,title=""):

model2d=MDS(n\_components=2,

metric=metric,

n\_init=4,

max\_iter=1000,

verbose=0,

eps=0.00001,

n\_jobs=None,

random\_state = 1000,

dissimilarity='euclidean')

x = df.loc[:, df.columns[:-1]].values

y = df.loc[:,['decade']].values

if standartize:

x = StandardScaler().fit\_transform(x)

x = pd.DataFrame(x)

X\_trans = model2d.fit\_transform(x)

stress = round(model2d.stress\_,0)

print('Iterations: ', model2d.n\_iter\_)

print('Stress: ', stress)

X\_trans = X\_trans.reshape(2, -1)

y = y.flatten()

toplot = pd.DataFrame({"PC1":X\_trans[0],"PC2":X\_trans[1],"key":y})

plot\_mds(toplot,title)

# #### not standartized

do\_mds(df, False, title = "Not standartized")

# #### standartized

do\_mds(df, True, title = "Standartized")

do\_mds(df, True, False, title = "Standartized non-metric")

do\_mds(df\_small, True, title = "Standartized feature subset")

# ## t-SNE

from sklearn.manifold import TSNE

def plot\_tsne(df,perplexity,ax = None):

if ax is None:

fig = plt.figure(figsize=(15, 8))

ax = fig.add\_subplot(1,1,1)

ax.set\_xlabel("t-SNE1")

ax.set\_ylabel("t-SNE2")

ax.set\_title("2 dimensional t-SNE" + f" (perplexity={perplexity})")

targets = ['50s', '80s', '10s']

colors = ['tab:blue', 'tab:orange', 'tab:green',]

for target, color in zip(targets,colors):

indicesToKeep = df['key'] == target

ax.scatter(df.loc[indicesToKeep, 'PC1']

, df.loc[indicesToKeep, 'PC2']

, c = color

, s = 200

, alpha = 0.6)

ax.legend(targets)

ax.grid()

def do\_tsne(df, standartize = False,perplexity=30,n\_iter=1000,ax=None):

model2d=TSNE(n\_components=2,

perplexity = perplexity,

n\_iter=n\_iter,

random\_state = 1000,

verbose=0)

x = df.loc[:, df.columns[:-1]].values

y = df.loc[:,['decade']].values

if standartize:

x = StandardScaler().fit\_transform(x)

x = pd.DataFrame(x)

X\_trans = model2d.fit\_transform(x)

print('Iterations: ', model2d.n\_iter\_)

print('kl divergence: ', model2d.kl\_divergence\_)

X\_trans = X\_trans.reshape(2, -1)

y = y.flatten()

toplot = pd.DataFrame({"PC1":X\_trans[0],"PC2":X\_trans[1],"key":y})

plot\_tsne(toplot,perplexity,ax)

for i in [12,13,14]:

do\_tsne(df\_small, True,i,8000)

for i in [23]: # 23-25

do\_tsne(df, True,i,8000)

fig, ax = plt.subplots(3,2,figsize=(24, 20))

# somehow good again in the 50s range

ax = ax.flatten()

for i,j in enumerate([5,10,20,30,40,50]):

do\_tsne(df, True,j,8000,ax[i])