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

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.

2 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

3 Atliktos analizės aprašymas

3.1 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

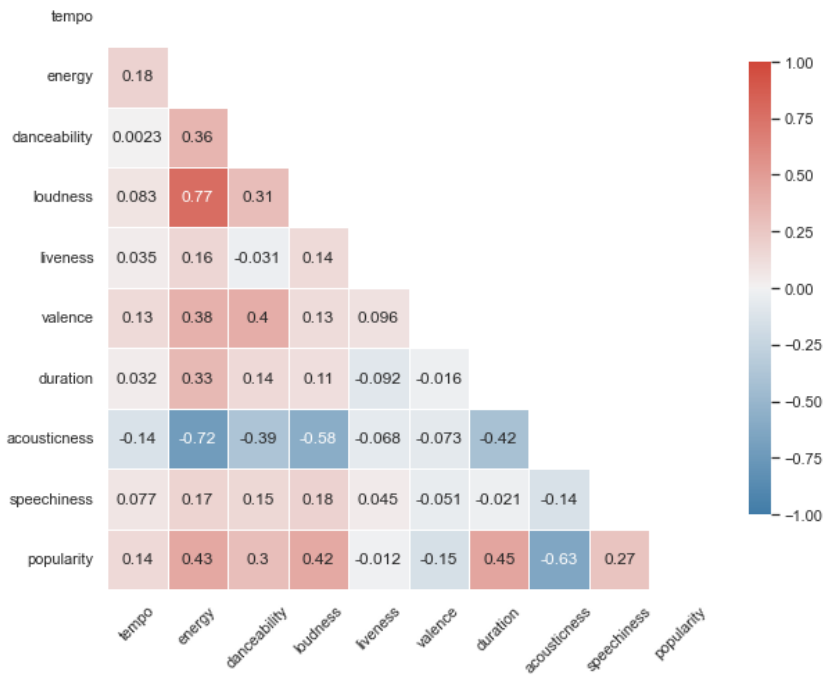
Dešimtmetis	Skaičius
2010-ieji	100
1980-ieji	105
1950-ieji	73

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

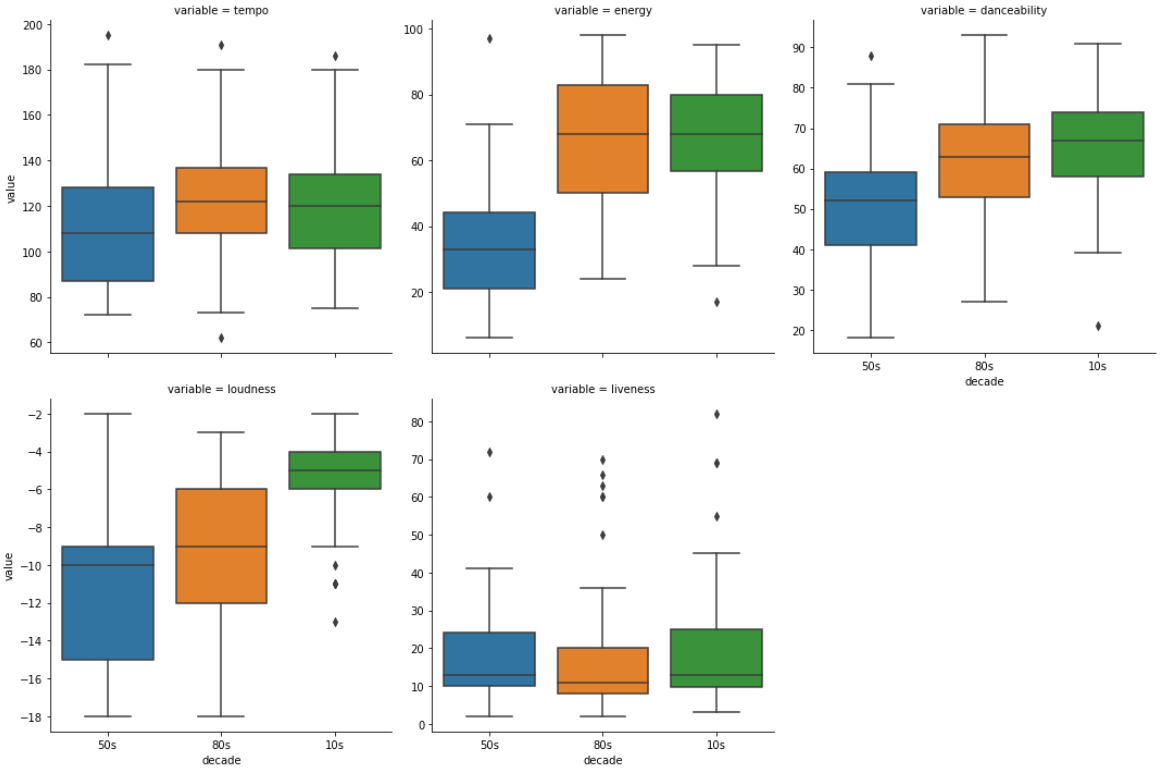
	mean	std	min	25%	50%	75%	max
tempo	118.2	25.3	62.0	100.0	117.0	135.0	195.0
energy	58.1	22.8	6.0	41.2	60.5	78.8	98.0
danceability	60.4	14.3	18.0	51.0	62.0	71.0	93.0
loudness	-8.5	4.0	-18.0	-11.0	-8.0	-5.0	-2.0
liveness	17.4	13.4	2.0	9.0	13.0	22.0	82.0
valence	55.6	25.0	9.0	34.0	55.0	77.8	99.0
duration	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.)

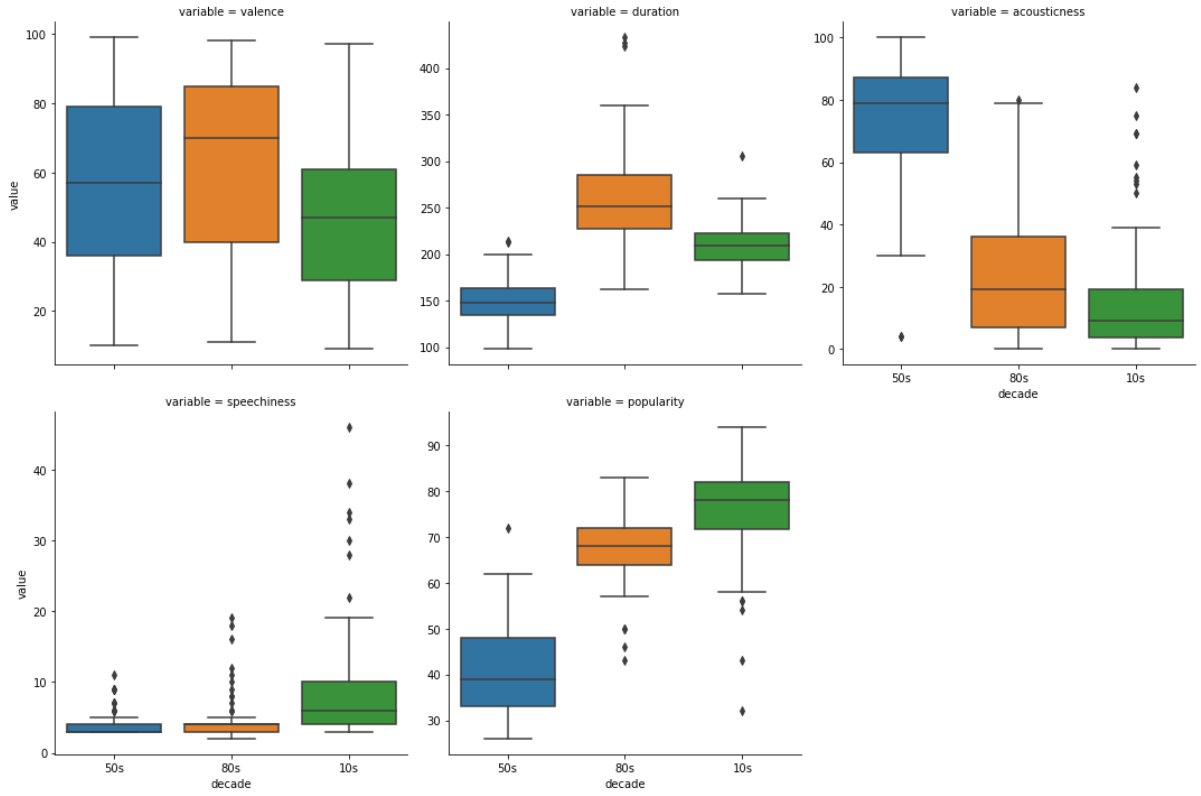


1 pav. Pirsono koreliacijos tarp požymių koeficientai

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) $x_{norm} = \frac{x-\bar{x}}{\sqrt{\sigma^2}}$, kur σ^2 - požymio vidurkis požymio dispersija, $\bar{x}$ – požymio vidurkis.

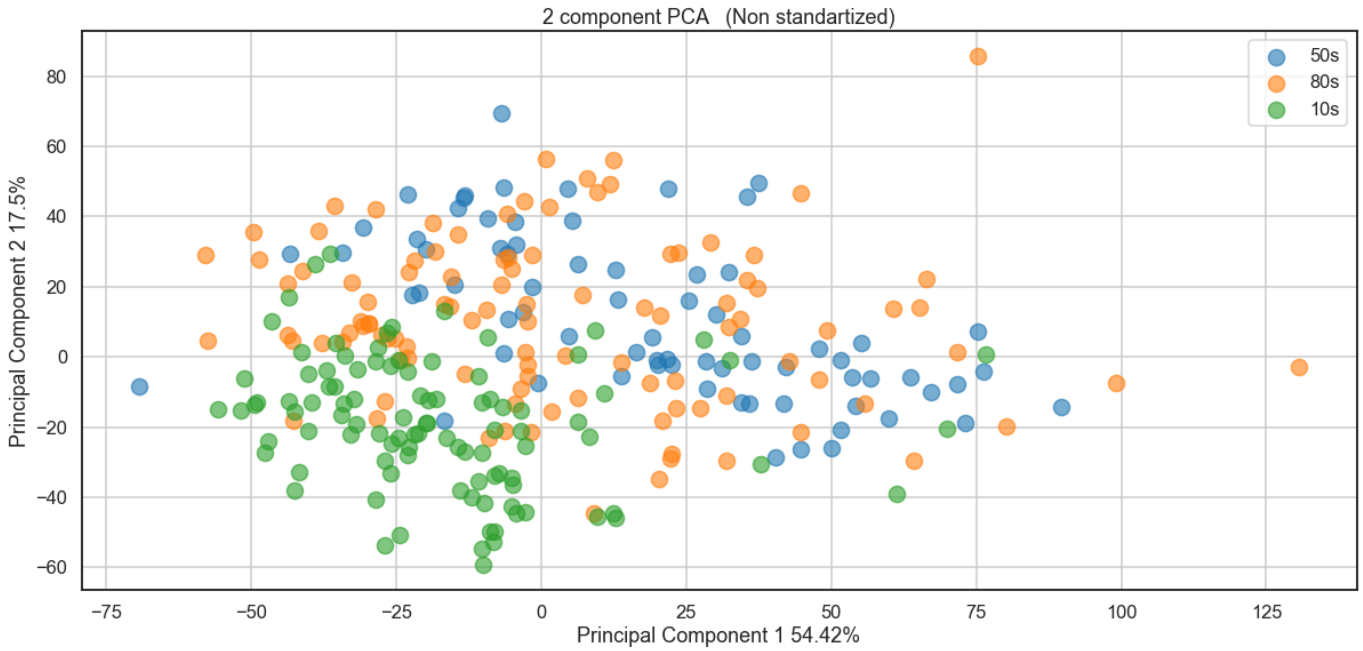
3.2 PCA

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). Kiekviena pagrindinė komponentė yra kažkokia pradinių duomenų aibės požymių tiesinė kombinacija. PCA metodas neturi svarbių parametrų, kuriuos keičiant būtų gaunami skirtingi rezultatai.

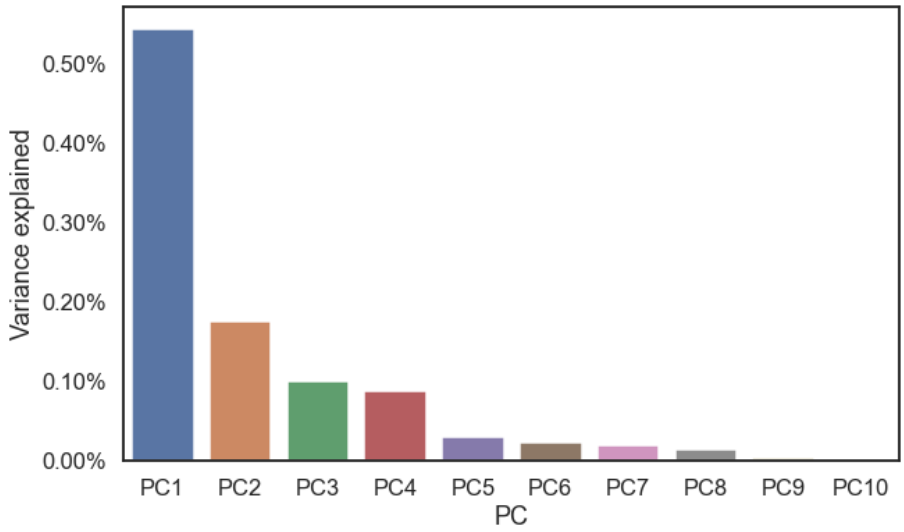
Pirmiausia PCA atlikta su nestandartizuota duomenų aibe (pirma pagrindine komponente PC1 paaiškinama 0.54 visos dispersijos, PCA2 - 0.17, žr. 5 pav.). Paliktos ir vizualizuotos pirmos dvi pagrindinės komponentės (žr. 4 pav.). Gautuose rezultatuose pastebimi susidarantys klasių klasteriai, tačiau visos klasės stipriai 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. 6 pav.).

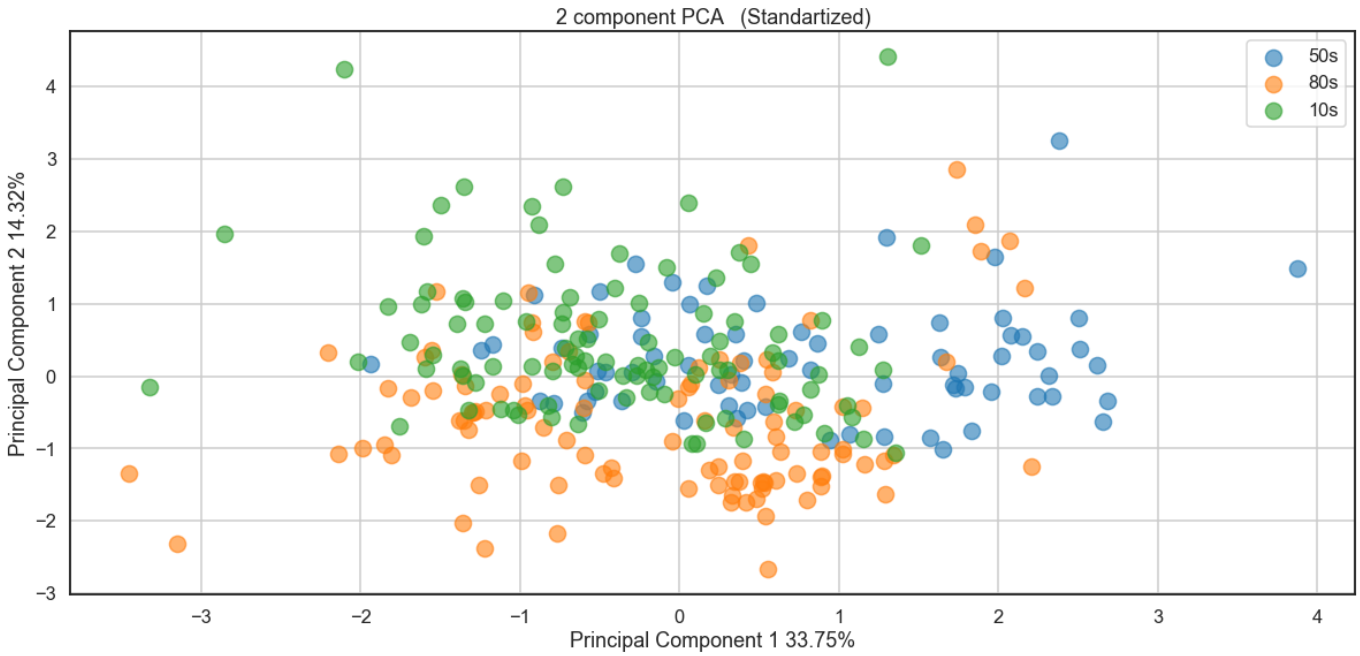
Palyginimui PCA atlikta paėmus požymių poaibį be požymių „Tempo“, „Liveness“ ir „Valence“, kurie pagal stačiakampes diagramas dešimtmečius atskiria mažiausiai (paaiškinama dalis dispersijos: PC1 - 0.47, PC2 - 0.15, žr. 9 pav.). Gautuose rezultatuose kiti dešimtmečiai mažiau maišosi su 80-ųjų dainomis, matomos susidariusios išskirtys (žr. 8 pav.).



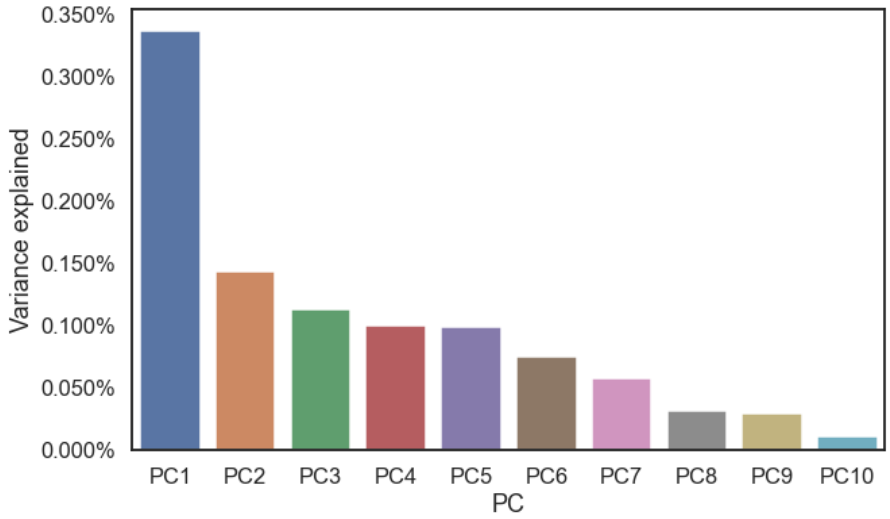
4 pav. PCA su nestandartizuotais duomenimis



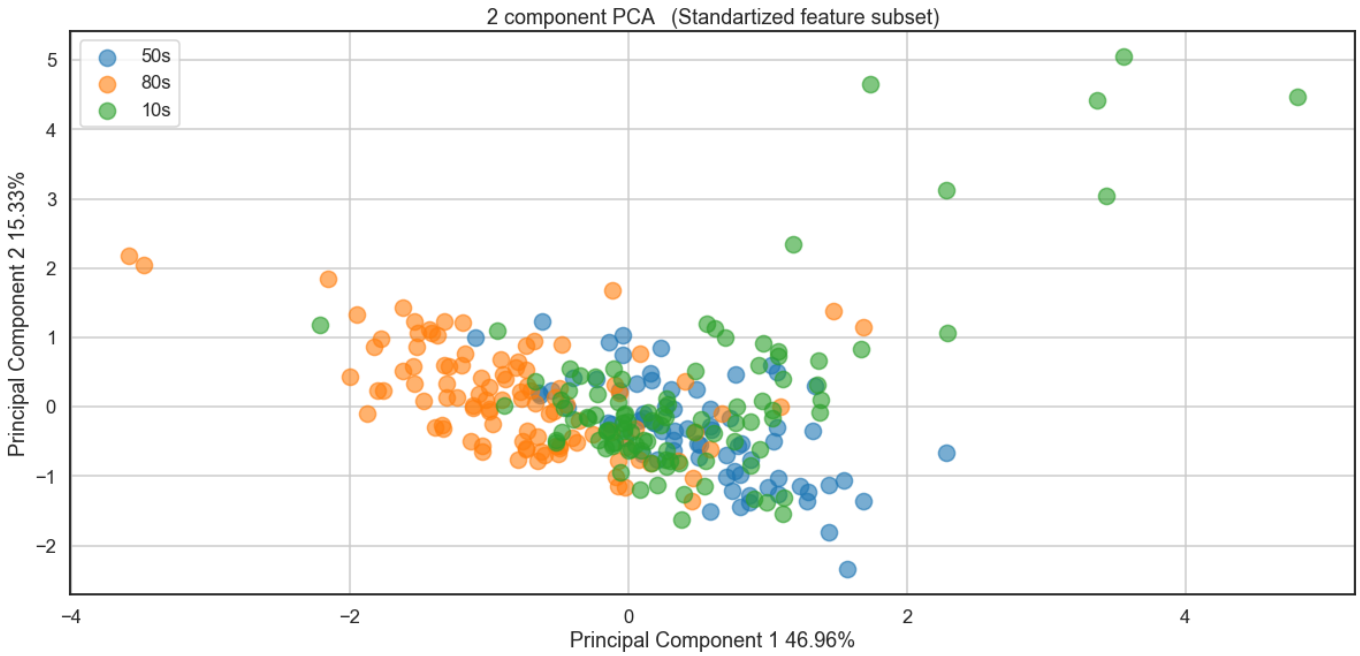
5 pav. PC paaiškinama dalis dispersijos nestandartizuotiems duomenims



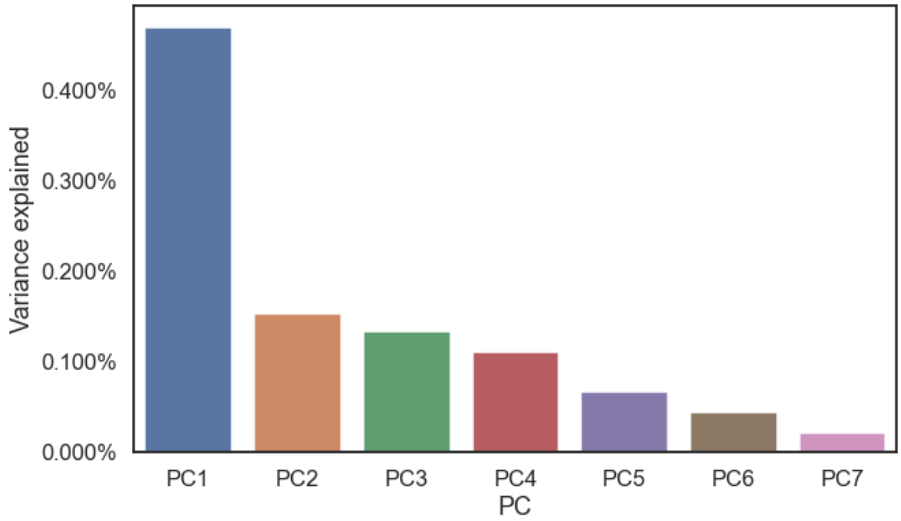
6 pav. PCA su standartizuotais duomenimis



7 pav. PC paaiškinama dalis dispersijos standartizuotiems duomenims



8 pav. PCA su standartizuotais duomenimis požymių poaibiui



9 pav. PC paaiškinama dalis dispersijos požymių poaibiui

3.3 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.

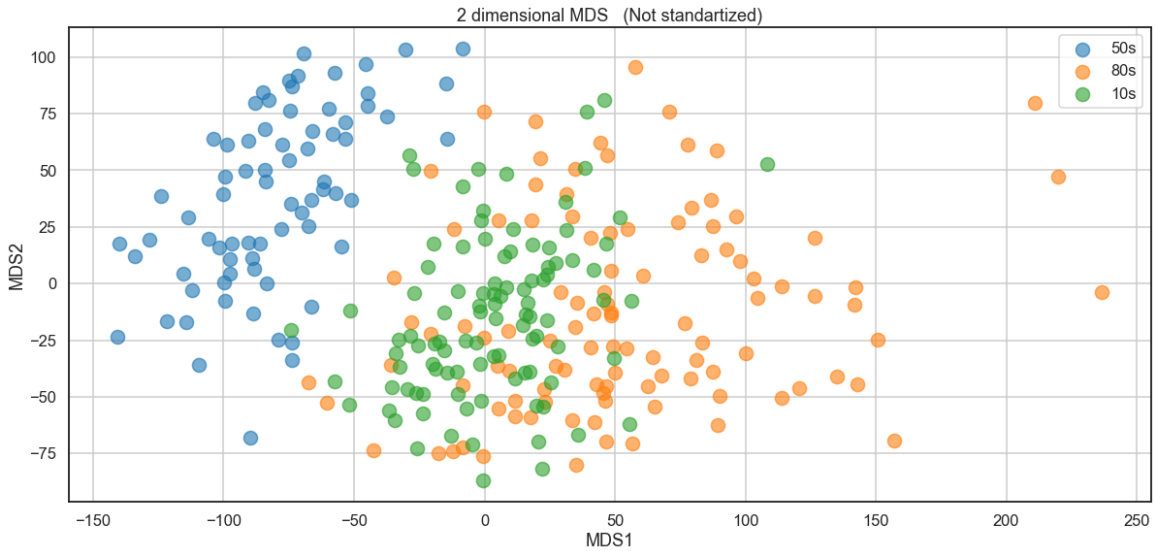
Pagrindinis parametras MDS yra pasirinkimas tarp metrikinės (angl. metric) ir nemetrikinės (angl. non-metric) MDS variantų naudojimo. MDS taip pat galimas pradinės taškų konfigūracijos žemesnės dimensijos erdvėje pasirinkimas.

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

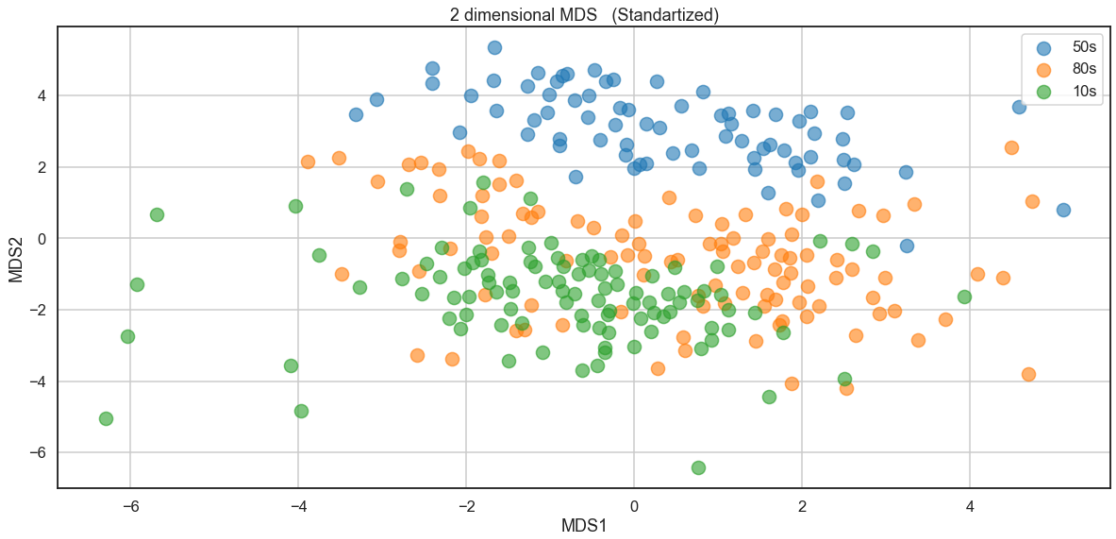
Naudojant metrikinę MDS su Euklidiniais atstumais tiek su nestandartizuota (žr. 10 pav.), tiek su standartizuota duomenų aibe (žr. 11 pav.) gautas mažesnis klasių persidengimas negu gautas naudojant PCA metodą, tačiau nė viena klasė pilnai neatsiskiria. Vietoje atsitiktinės pradinės taškų konfigūracijos, panaudota ir pradinė taškų konfigūracija, gauta PCA metodu (žr. 12 pav.). Visais šiais atvejais 50-ųjų dainos nuo likusių dešimtmečių atsiskiria labiau negu 80-ųjų dainos atsiskiria nuo 2010-ųjų. Naudojant standartizuotą duomenų aibę visoms klasėms gaunami tankesni klasių klasteriai.

Naudojant nemetrinę MDS gaunami stipriai prastesni rezultatai – užpildoma visa grafiko erdvė, klasės nesudaro jokių klasterių (žr. 13 pav.).

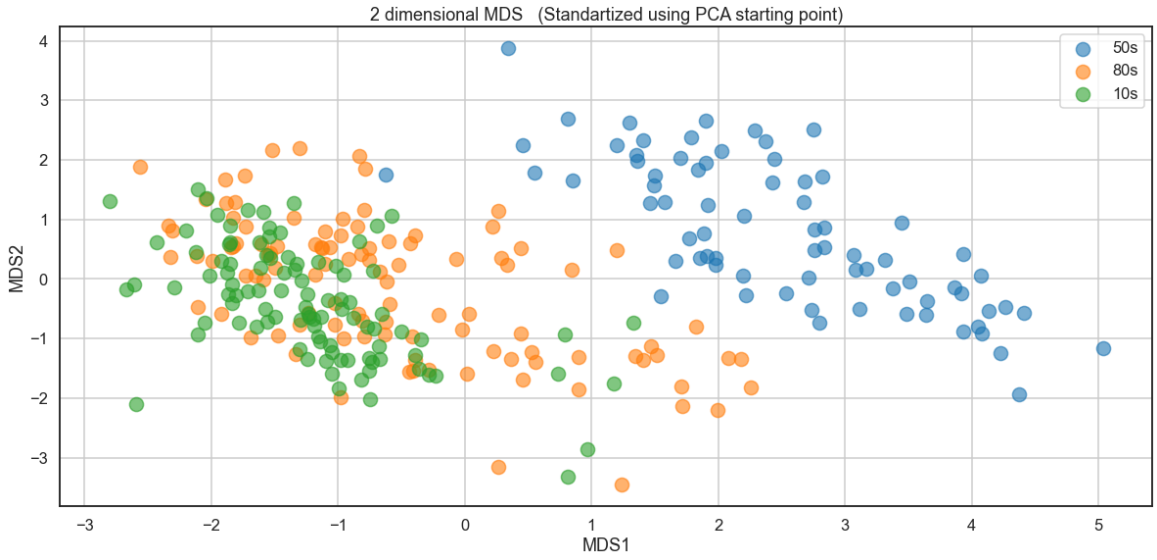
Atlikus metrikinę MDS su Euklidiniais atstumais standartizuotų požymių poaibiui, klasių atsiskyrimo prasme pagerinami rezultatai, gauti imant visą požymių aibę: 50-ųjų dainų klasė beveik pilnai atsiskyrusi nuo likusiųjų (žr. 14 pav.).



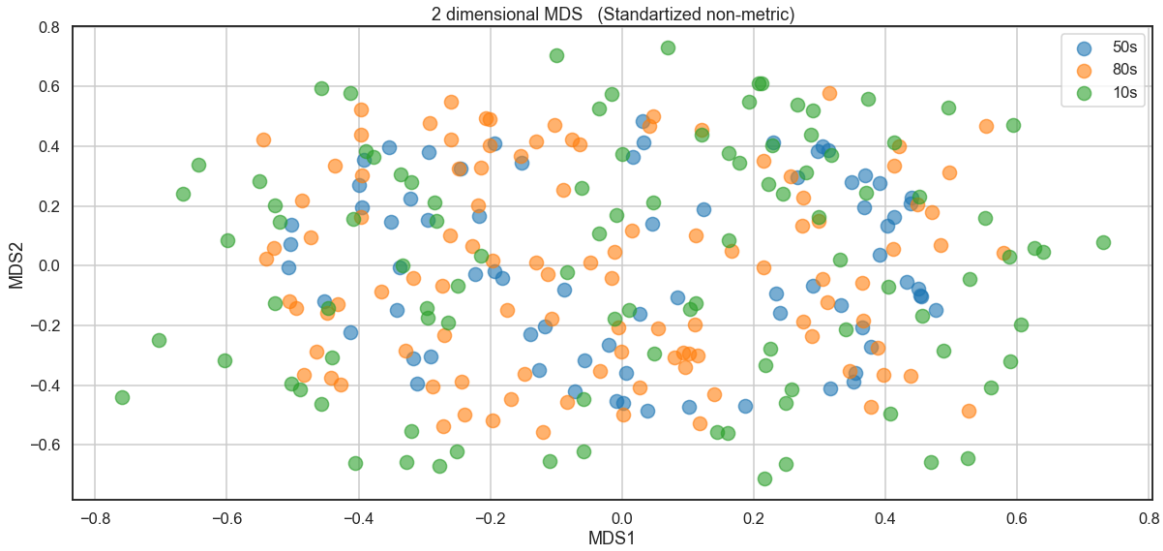
10 pav. Metrikinė MDS su nestandartizuotais duomenimis



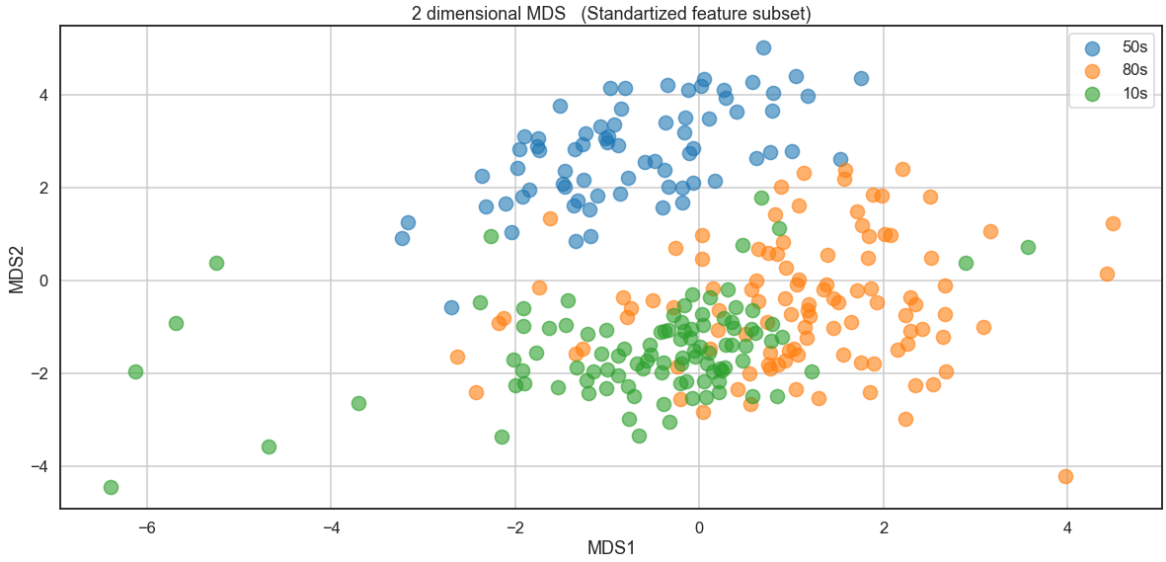
11 pav. Metrikinė MDS standartizuotiems duomenimis



12 pav. Metrikinė MDS standartizuotiems duomenimis naudojant PCA gautą pradinę konfiguraciją



13 pav. Nemetrikinė MDS standartizuotiems duomenimis



14 pav. Metrikinė MDS standartizuotiems duomenis naudojant požymių poaibį

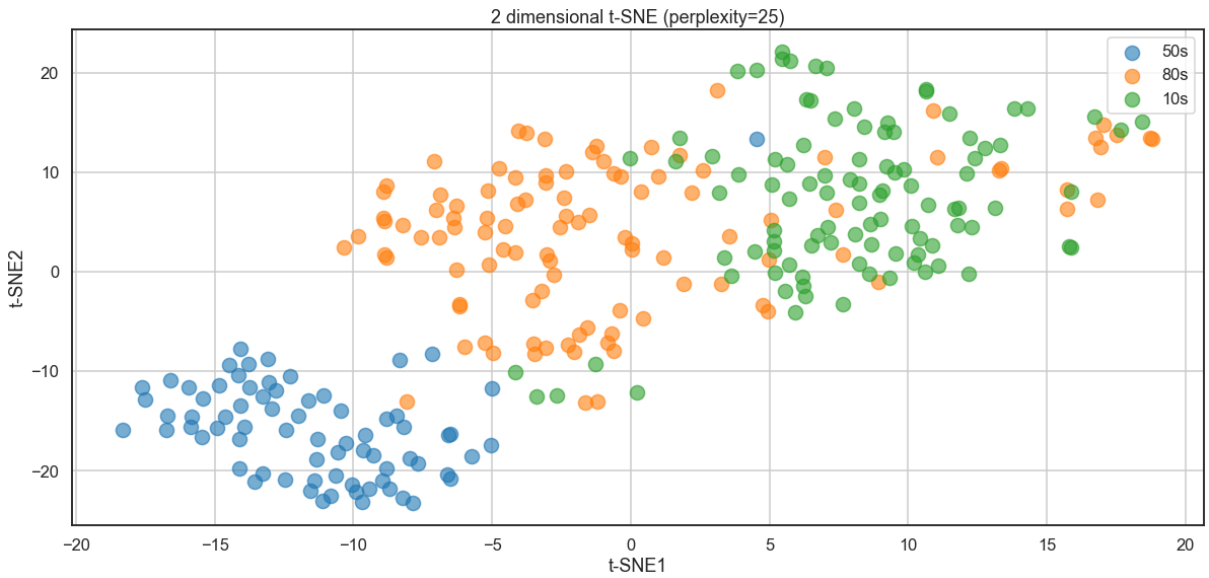
3.4 t-SNE

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).

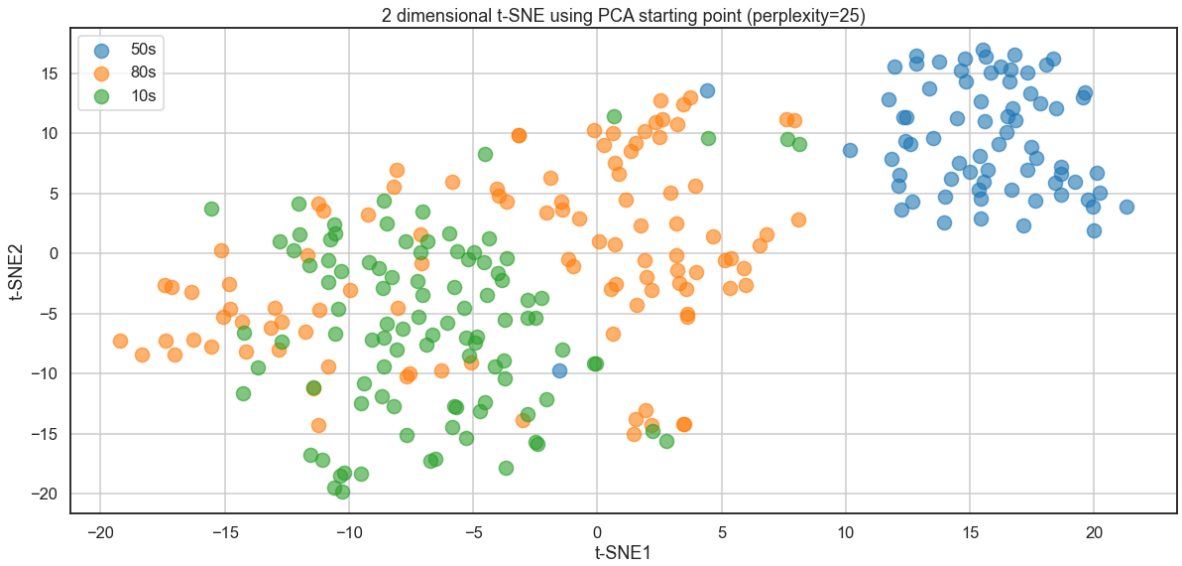
Pagrindinis metodo parametras vadinamas Perplexity ir yra susijęs su duomenų aibės objektų kaimynų skaičiumi. Su mažesnėmis parametro reikšmėmis didesnis dėmesys skiriamas vietinėms struktūroms, su didesnėmis – globalioms. Rekomenduojamos reikšmės nuo 5 iki 50. Didesnėms duomenų aibėms įprastai naudojamos didesnės parametro reikšmės. t-SNE taip pat galimas pradinės taškų konfigūracijos žemesnės dimensijos erdvėje pasirinkimas.

Naudojant t-SNE dimensijos dydis sumažintas iki dim=2. Standartizuotai duomenų aibei eksperimentiškai geriausi rezultatai rasti parametro reikšmėms esant 25-35 intervale (žr. 15 pav.). Vietoje atsitiktinės konfigūracijos taip pat panaudota ir pradinė taškų konfigūracija gauta PCA metodu (žr. 16 pav.). Gautuose rezultatuose gautas mažas klasių persidengimas. Be to kaip ir taikant MDS metodą, pastebimas didesnis 50-ųjų dainų atsiskyrimas. Mažinant parametro reikšmės išlaikomas mažas klasių persidengimas, tačiau Perplexity reikšmei esant kuo mažesnei, sudaromas tuo didesnis kiekis vietine struktūra pagrįstų papildomų klasterių (žr. 17 pav.). Imant didesnes parametro reikšmes negaunami ryškūs pokyčiai gautuose rezultatuose (žr. 18 pav.).

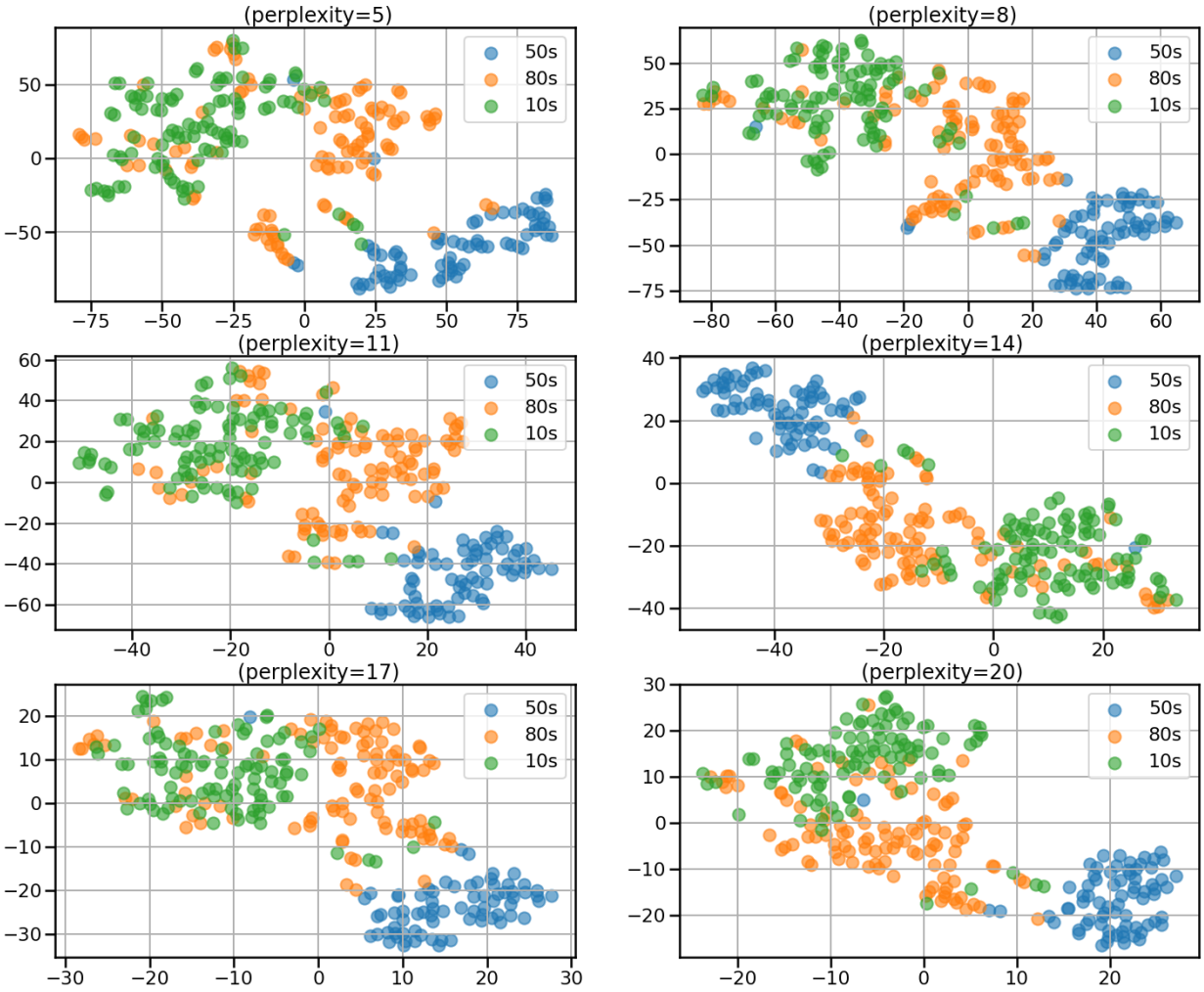
Požymių poaibiui geriausi rezultatai gaunami naudojant panašias Perplexity parametro reikšmes kaip ir visai požymių aibei (žr. 19 pav.). Gauti ryškūs klasių klasteriai su minimaliu persidengimu.



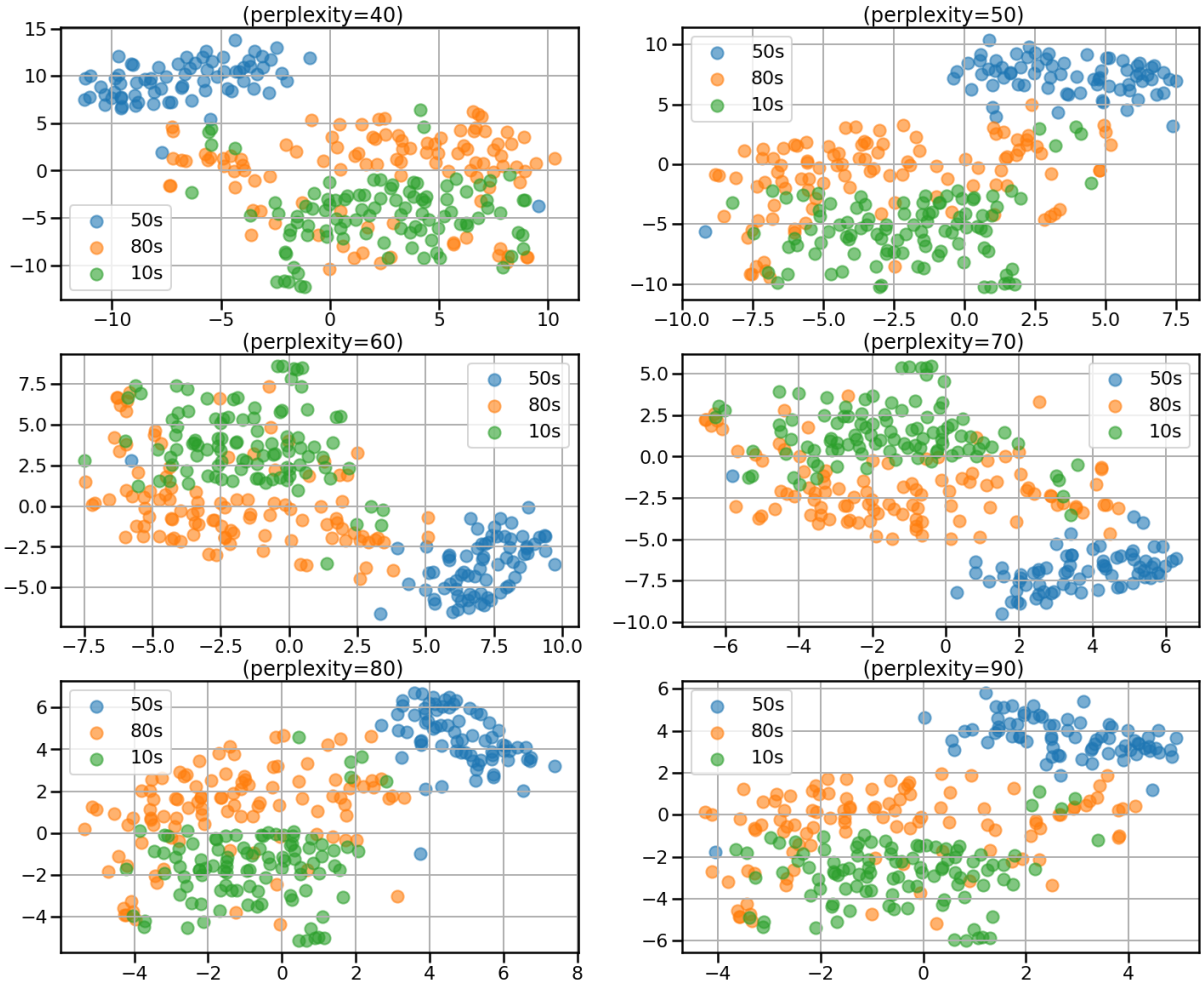
15 pav. t-SNE standartizuotiems duomenims



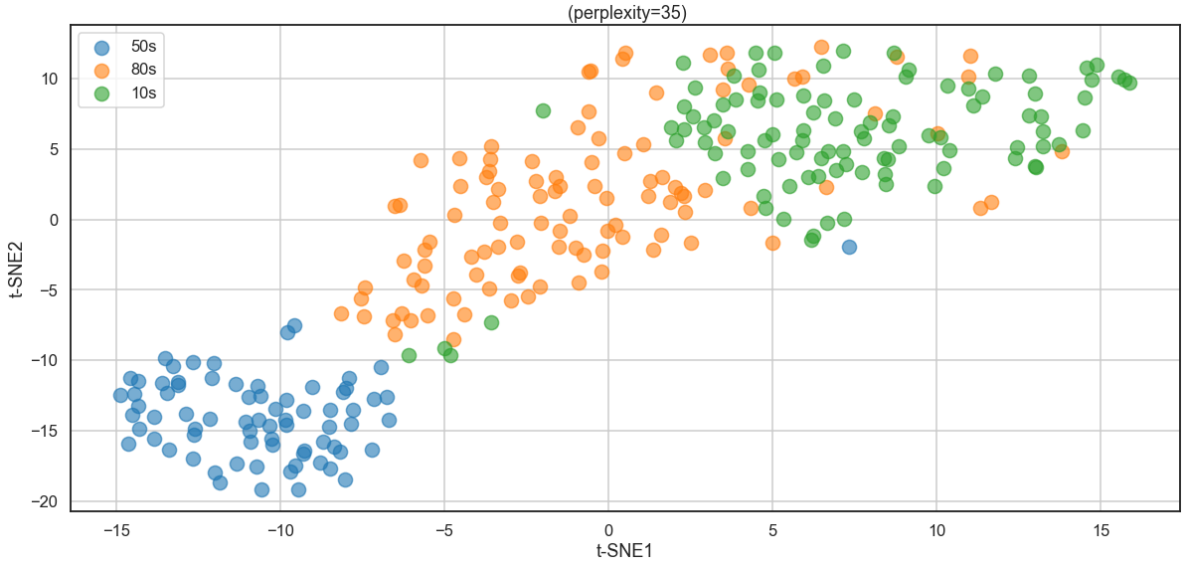
16 pav. t-SNE standartizuotiems duomenims naudojant PCA gautą pradinę konfigūraciją



17 pav. t-SNE standartizuotiems duomenims su skirtingomis Perplexity parametro reikšmėmis (pradedant nuo 5 ir didėjant po 3)



18 pav. t-SNE standartizuotiems duomenims su skirtingomis Perplexity parametro reikšmėmis (pradedant nuo 40 ir didėjant po 10)



19 pav. t-SNE požymių poaibiui

3.5 Metodų palyginimas

PCA yra dažniausiai naudojamas dimensijos mažinimo metodas, su kurio dažnai yra lyginami kitų dimensijos mažinimo metodų rezultatai. Šiuo metodu gautos PC turi aiškią interpretaciją – jos yra pradinių požymių tiesinės kombinacijos.

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, dvimatėje erdvėje formuojasi klasteriai, tačiau klasių gaunamas stiprus klasių persidengimas. 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 pagerintas klasių atsiskyrimo rezultatas gautas naudojant PCA.

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 nemetrikine MDS gauti prasčiausi rezultatai klasių atsiskyrimo atžvilgiu iš visų naudotų metodų.

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

t-SNE yra 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 mažiausias klasių persidengimas gautas naudojant Perplexity reikšmes 25-35 intervale. Standartizuotos duomenų aibės požymių poaibiui geriausi rezultatai rasti esant panašioms parametro reikšmėms. Naudojant šias reikšmes apskritai gauti geriausi klasių atsiskyrimo dvimatėje erdvėje rezultatai iš naudotų metodų. Naudojant kitas parametro reikšmes gauti prastesni rezultatai.

4 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.

Naudojant dimensijos mažinimo metodus, dainos pagal dešimtmetį dvimatėje erdvėje išsidėsto ne užimdamos visą grafiko plotą, bet sudaro klasterius. Naudojant daugelį metodų, pastebimas didesnis 50-ųjų dainų skirtumas tiek nuo 80-ųjų, tiek nuo 2010-ųjų dainų, negu gautas skirtumas tarp 80-ųjų ir 2010-ųjų dainų. Nepaisant to, su visais naudotais metodais negautas pilnas klasių atsiskyrimas dvimatėje erdvėje nė vienai klasių porai.

Iš prieš tai esančių teiginių galima kelti atitinkamas hipotezes apie duomenų aibę:
Skirtingų dešimtmečių dainoms yra būdingi tam tikri bruožai, kurie per tris dešimtmečius pasikeičia.
Nuo 50-ųjų iki 80-ųjų muzikos tendencijose įvyko didesni pokyčiai negu nuo 80-ųjų iki 2010-ųjų.
Egzistuoja dainų bruožai, kurie išlieka tam tikrose ribose visais dešimtmečiais.

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, pastebimi susidarę klasių klasteriai, tačiau klasės stipriai persidengia. Naudojant nestandartizuotą duomenų aibę gauti prastesni rezultatai kaip ir tikėtasi.

MDS metodu požymius sumažinus iki dvimačių ir juos pavaizdavus grafiškai gauti geresni klasių atsiskyrimo rezultatai už PCA, išskyrus naudojant nemetrikinę MDS.

Naudojant t-SNE metodą eksperimentiniu būdu rasta, kad Perplexity reikšmės intervale 25-35 geriausiai atskiria klases tiek standartizuotai duomenų aibei, tiek imant standartizuotų duomenų požymių poaibį. Su šiomis parametro reikšmėmis gautas geriausias klasių atsiskyrimas iš visų trijų metodų.

Rasta, kad visiems trims naudotiems metodais, juos pritaikius tik požymių poaibiui be požymių, kurių empirinis pasiskirstymas tarp klasių stipriai nesiskyrė, klasių atsiskyrimas visada pagerina arba bent prilygsta rezultatams gautiems su visa požymių aibe.

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.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
from matplotlib.ticker import PercentFormatter

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(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)


# #### not standartized


do_pca(df, False,"Non standartized")


# #### standartized
```

```
do_pca(df, True, "Standartized")

do_pca(df_small,True, "Standartized feature subset")


# ## MDS

from sklearn.manifold import MDS
from sklearn.metrics import euclidean_distances
from sklearn.metrics import pairwise_distances
from sklearn.manifold import smacof

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, 'MDS1']
                   , df.loc[indicesToKeep, 'MDS2']
                   , c = color
                   , s = 200
                   , alpha = 0.6)
    ax.legend(targets)
    ax.grid()

def do_mds(df, standartize = False,metric=True,precomputed=False,title=""):
    mds=MDS(n_components=2,
          metric=metric,
          n_init=4,
          max_iter=1000,
          verbose=0,
          eps=10e-6,
          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)
    if not precomputed:
        x_mds = mds.fit_transform(x)
        stress = round(mds.stress_,0)
        n_iter = mds.n_iter_
    else:
        pca = PCA(n_components = 2)
        init_pca = pca.fit_transform(x)
        dist = pairwise_distances(init_pca,metric="euclidean")
        x_mds, stress, n_iter = smacof(dist, n_components = 2, init=init_pca,
                                       return_n_iter=True, eps=10e-06, max_iter=1000)

    x_mds = pd.DataFrame(x_mds,columns=["MDS1","MDS2"])
```

```
    print('Iterations: ',n_iter)
    print('Stress: ', stress)

    x_mds["key"] = y

    plot_mds(x_mds,title)

# #### not standartized


do_mds(df, False, title = "Not standartized")


# #### standartized


do_mds(df, True, title = "Standartized")

do_mds(df, True, precomputed=True, do_pca=True, title = "Standartized using PCA starting point")

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,title=""):
    if ax is None:
        fig = plt.figure(figsize=(20, 9))
        ax = fig.add_subplot(1,1,1)
    ax.set_xlabel("t-SNE1")
    ax.set_ylabel("t-SNE2")
    ax.set_title(title + " "  + 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, 't-SNE1']
                   , df.loc[indicesToKeep, 't-SNE2']
                   , c = color
                   , s = 200
                   , alpha = 0.6)
    ax.legend(targets)
    ax.grid()


def do_tsne(df, standartize = False,perplexity=30,n_iter=1000,init="random",**kwargs):
    tsne=TSNE(n_components=2,
          perplexity = perplexity,
          n_iter=n_iter,
          random_state = 1000,
          verbose=0,
          init = init)

    x = df.loc[:, df.columns[:-1]].values
    y = df.loc[:,['decade']].values
```

```
    if standartize:
        x = StandardScaler().fit_transform(x)
    x = pd.DataFrame(x)

    x_tsne = tsne.fit_transform(x)
    x_tsne = pd.DataFrame(x_tsne)

    print('Iterations: ', tsne.n_iter_)
    print('kl divergence: ', tsne.kl_divergence_)

    x_tsne.columns = ["t-SNE1","t-SNE2"]
    x_tsne["key"] = y

    plot_tsne(x_tsne,perplexity,**kwargs)

sns.set_context("poster")

for i in [25]:
    do_tsne(df, True,i,8000,title="2 dimensional t-SNE")
    do_tsne(df, True,i,8000, init="pca", title="2 dimensional t-SNE using PCA starting point")


fig, ax = plt.subplots(3,2,figsize=(24, 20))
ax = ax.flatten()
for i,j in enumerate([40,50,60,70,80,90]):
    do_tsne(df, True,j,8000,ax=ax[i])
    ax[i].set_xlabel("")
    ax[i].set_ylabel("")


sns.set_context("talk")
for i in [25,30,35]:
    do_tsne(df_small, True,i,8000)
```