

Uvod. Vsakoletni glasbeni spektakel Evrovizija pobere kar nekaj zanimanja širše publike, tako mladih, kot tudi starih. Po tekmovanju je čas za glasovanje in praznovanje, kasneje pa tudi za analizo podatkov. Želeli si bomo spoznati, ali je glasovanje zmanipulirano med državami samimi res nepristransko, ali morda na le-to kaj vpliva. V nadaljevanju bomo spoznali pristop do tovrstnega problema, ekstrakcijo in filtriranje podatkov, način(e) primerjave držav, analizo rezultatov in njihovo razumevanje. Na poti se bomo srečali tudi s kar nekaj problemi, katere bomo poizkusili elegantno rešiti. Za le-to bomo uporabili različne pristope. Na koncu bomo tudi evalvirali dobljen rezultat in postopek pridobitve le-tega ter pokomentirali, kaj bi lahko še izboljšali v prihodnosti.

Podatki. Podatke za nalogo smo dobili že v obliki ene csv (comma separated values) datoteke. Zanje zato nismo potrebovali dodatne ekstrakcije (npr. webscraping) metode. Datoteko `eurovision-finals-1975-2019.csv` sestavlja 5 stolpcev/atributov:

- Year,
- Jury or Televoting,
- From country,
- To country,
- Points.

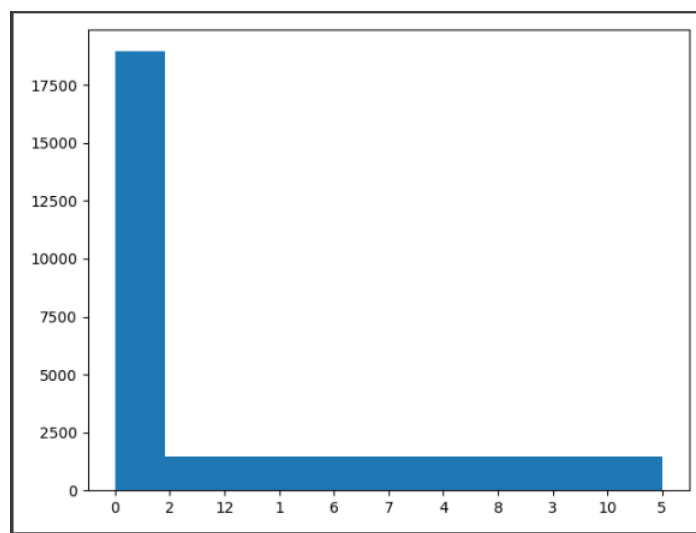
Relavantna polja za analizo so praktično vsa, razen polja "Jury or Televoting". Tip vrednosti relavantnih polj so celoštevilski, razen imena držav so tipa string.

Vseh podatkov je 33374, pri čimer z besedo podatek dejansko mislim le vrstico, ki nam pove podrobnosti nekega oddanega glasu. Možne vrednosti pri stolpcu 3 in 4 je 53 različnih držav tipa string. Države lahko trenutno še obstajajo, ali pa so že razpadle na nove države (Jugoslavija). Ker govorimo o Evroviziji, so države večinoma pripadnice Evrope. Polje Year oziroma leto glasovanja/tekmovanja se razteza od leta 1975, pa vse do lanskega leta (2019). Skupno imamo torej podatke o glasovanju za 44 let.

Zadnji stolpec, točke, zajemajo vrednosti od 0 (najslabše/najmanj točk) pa do 12 (maksimalno število, največ točk, najboljši glas). Ti podatki so lepo vidni na sliki 1.

Zanima nas, kako so te točke porazdeljene. Pomagal sem si s funkcijo za izris histograma `hist()` iz knjižnice `matplotlib`. Opazimo lahko, da je glasov z vrednostjo 0 občutno več, kot vseh ostalih. To je precej pričakovano, saj je na enem dogodku tekmovalo tudi do 53 držav, točke ($x > 0$) pa lahko dodeliš le 12 državam. Zato kar nekaj (večina) polj s točkami vsebuje vrednost 0. Po podrobnejšem pregledu ostalih vrednosti se lepo vidi tudi, da je ostalih točk enako. Tudi to je precej samoumevno - glasove vedno oddaš za vse možne točke (ne moreš se vzdržati na primer glasu za maksimalno število točk...).

Za lažjo analizo podatkov sem iz slovarja izločil vse države, ki so se na tekmovanju pojavile 5x ali manj. V tem primeru so bile to Moroko, Srbija in črna gora, Avstralija in Severna Makedonija. Slednja se je na tekmovanju pojavila samo enkrat, kar se mi ni zdelo smiselno oziroma omembe vredno. Zanimajo nas bolj "stare članice".



Slika 1: Prikaz porazdelitve točk vseh glasov med letoma 1975 in 2019.

Računanje razdalj. Pri gručenju držav, ki se med seboj favorizirajo, sem za njihovo podobnost glasovanja uporabil evklidsko razdaljo. Bistvene razlike med evklidsko in manhattansko razdaljo nisem opazil. Menim, da bi bili obe dovolj dobri. Pri računanju razdalje med dvema clusterjema sem se sprehodil čez vse možne kombinacije pripadnikov teh dveh gruč. To je sicer kar potratno, kar se tiče časovne kompleksnosti, o čimer bom več povedal na koncu.

Ko so posamezne gruče vsebovale po več držav, sem za izračun razdalje med gručami uporabljal povprečno razdaljo (average linkage).

Države gruč z najmanjšo razdaljo so bile združene v eno gručo. Ta postopek je potekal vse dokler nismo imeli le še ene gruče z vsemi državami.

Tu lako vidimo osrčje računanja povprečne dolžine med dvema skupinama držav. Algoritem se sprehodi skozi vse države znotraj obeh skupin in jih "primerja" med seboj. Primerja jih na način da kliče funkcijo `row_distance()`, ki kot argument sprejme imeni dveh držav in vrne evklidsko razdaljo njunih vektorjev glasovanja.

Listing 1: `merge_clusters(clusters, c1, c2)`

```
for c in c1_flat:
    for d in c2_flat:
        # distance between c & d
        dist += self.row_distance(c[0], d[0])
dist /= (len(c1_flat) * len(c2_flat))
```

Računanje evklidske razdalje dveh vektorjev p in q :

Listing 2: Evklidska razdalja

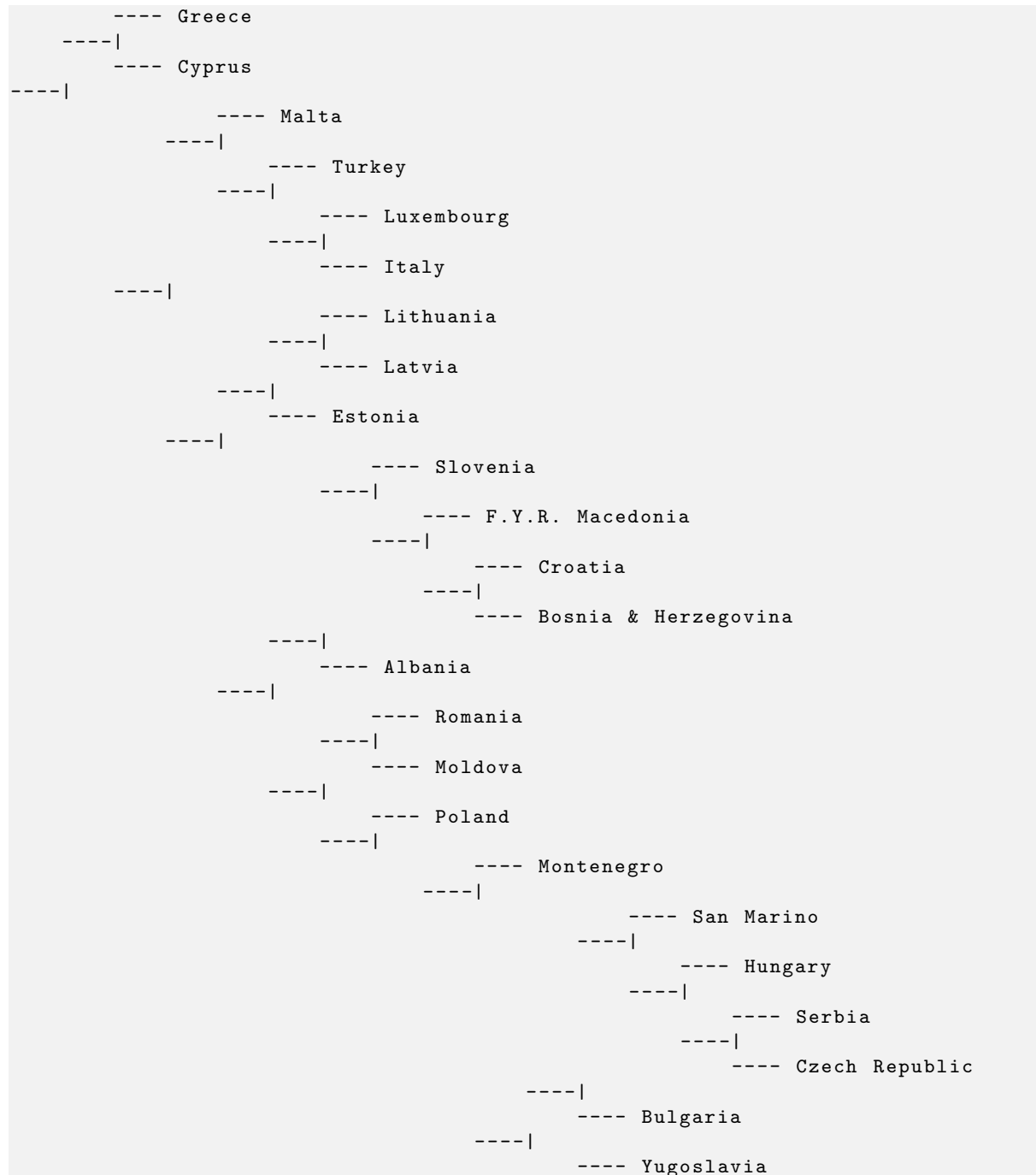
```
e = sum((p - q) ** 2 for p, q in zip(v1, v2)) ** .5
```

Končen ASCII izris dendrograma sem implementiral z rekurzivno funkcijo, ki kot argument prejme en objekt tipa list. List vsebuje eno skupino (vseh) držav, ki se nato razdeli na dve podskupini, vsaka izmed njih še na dve in tako naprej vse dokler ne dobimo posameznih držav.

Listing 3: Rekurzivna funkcija izris()

```
def izris(arr, i):
    if len(arr) == 1:
        print("uuuu" * i, "----u", arr[0], sep='')
    else:
        i += 1
        izris(arr[0], i)
        print("uuuu" * (i - 1), "----|", sep='')
        izris(arr[1], i)
```

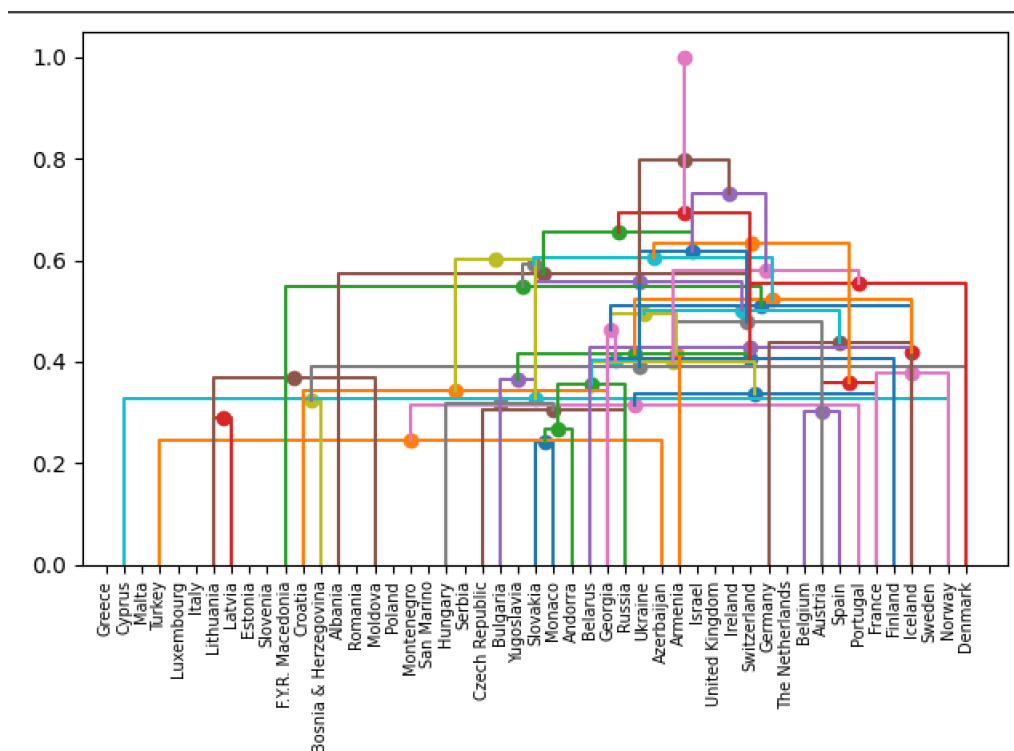
Dendrogram. ↓



```

-----|
                ---- Slovakia
                ----|
                ---- Monaco
                ----|
                ---- Andorra
        ----|
                ---- Belarus
                ----|
                ---- Georgia
                ----|
                ---- Russia
                ----|
                ---- Ukraine
                ----|
                ---- Azerbaijan
        ----|
        ---- Armenia
-----|
        ---- Israel
        ----|
        ---- United Kingdom
        ----|
        ---- Ireland
        ----|
        ---- Switzerland
        ----|
        ---- Germany
        ----|
        ---- The Netherlands
        ----|
        ---- Belgium
        ----|
        ---- Austria
        ----|
        ---- Spain
        ----|
        ---- Portugal
        ----|
        ---- France
        ----|
        ---- Finland
-----|
        ---- Iceland
        ----|
        ---- Sweden
        ----|
        ---- Norway
        ----|
        ---- Denmark

```



Slika 2: Grafični prikaz ASCII dendrograma (z ustreznimi normaliziranimi dolžinami) zgoraj.

Skupine in njihove preferenčne izbire. Države sem po analizi rezultatov lahko sestavil v skupno 6 skupin. Pri le-teh lahko vidimo neke vzorce povezovanja.

Najbolj očiten vzorec za nastanek skupin bi bila geografska lokacija posameznih držav. Hitro lahko vidimo, da se dobro povezujejo države sosedje. Drugi vzrok povezovanja je tudi jezikovna skupina članic skupin. Seveda imamo nekaj izjem (npr. Izrael v skupini Združenega kraljestva) vendar načeloma vse države, bolj ali manj, sledijo zgornjima pogojem.

V zadnjem stolpcu so pri vsaki skupini našteje skupine, za katere le-ta ne glasuje. Odločitev za to je oddaljenost gruč lepo vidna na ASCII izpisu zgoraj. Bolj, kot sta bili skupini narazen, manj točk sta si na tekmovanjih tekom let podeljevali.

Komentar:

Za boljše oziroma natančnejše rezultate bi evklidske razdalje lahko računali le med državami, ki se določeno leto HKRATI pojavijo na tekmovanju. Za boljšo optimizacijo in posledično hitrejši algoritem pa bi lahko vrednosti razdalj shranjevali v nekakšno podatkovno obliko podobno matriki razdalj.

Tabela 1: Preferenčne skupine držav in njihove članice

Skupina držav	Države pripadnice	Skupine držav, za katere (skupina) ne glasuje
Pribaltsko - sredozemska	Greece, Cyprus, Malta, Turkey, Luxembourg, Italy, Lithuania, Latvia, Estonia	Skandinavska, Romanska, Germanska
Slovanska	Slovenia, F. Y . R . Macedonia, Croatia, Bosnia & Herzegovina, Albania, Romania Moldova, Poland, Montenegro, San Marino, Hungary, Serbia, Czech Republic, Bulgaria, Yugoslavia, Slovakia, Monaco, Andorra, Belarus, Georgia, Russia, Ukraine, Azerbaijan, Armenia	Skandinavska, Romanska
Združeno Kraljestvo	Israel, United Kingdom, Ireland, Switzerland,	Pribaltsko - sredozemska Skandinavska
Germanska	Germany, The Netherlands, Belgium, Austria	Pribaltsko - sredozemska
Romanska	Spain, Portugal, France	Pribaltsko - sredozemska, Slovanska
Skandinavska	Finland, Iceland, Sweden, Norway, Denmark	Pribaltsko - sredozemska, Slovanska, Združeno Kraljestvo