## SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE FAKULTA ELEKTROTECHNIKY A INFORMATIKY

**STROJOVÉ UCˇ ENIE A NEURÓNOVÉ SIETE ZADANIE 1**

## 2019 Dinh Nguyen Tien

## Úvod

Našim cieľom bolo spracovať a pripraviť dáta tak, aby boli vhodné na neskoršiu manipuláciu s nimi. V tomto zadaní sme pracovali s dátami, ktoré obsahujú 51 ukazovateľov pre 145 krajín. Rozhodli sme sa pre využitie programovacieho jazyka Python. Toto zadanie je rozdelené na 4 úlohy, ktoré sú popísané v ďalších kapitolách.

## Úloha 1

**Zadanie úlohy:**

Načítajte dáta a pripravte ich na spracovanie. Načítajte dáta z .csv do vášho programu – vyberte vhodnú štruktúru, aby s imi vedeli vami zvolené algoritmy/ modely/ knižnice ďalej pracovať. Máte možnosť vybrať si podmnožinu dát (do rozumnej miery). Nezabudnite dáta normalizovyvať (inak výsledky nebudú výpovedné).

**Vypracovanie:**

Na spracovanie dát sme využili už vstavanú knižnicu csv. Pomocou nej sme ďalej implementovali podmnožinu krajín, ktoré budeme brať do úvahy. V prípade dát, ktoré sú reprezentované -, ~0.0 alebo - ~0.0, sme ich nahradili číselnou hodnotou 0.  
Na normalizáciu sme využili knižnicu sklear. Z nej sme volali funkciu MinMaxScaler, ktorá nám znormalizovala dáta. Tieto dáta sme aj vyexportovali do súboru normal.csv.

## Úloha 2

**Zadanie úlohy:**

Oboznámte sa so základnými používanými metrikami. Vyberte si metriku (naprl. L1, L2) a pomocou nej zostrojte maticu podobností krajín (stačí podmnožina krajín a jej ukazovateľov). Potešíme sa, ak si vyberiete takú množinu ukazovateľov, aby ste dostali čo najzaujímavejšie výsledky.

**Vypracovanie:**

Keďže knižnica poskytuje aj euklidovú vzdialenosť, tak sme sa ju rozhodli využiť. V tejto úlohe bolo potrebné využiť dáta z úlohy 1, ktoré sme normalizovali. Ak by sme to nespravili normalizáciu, nevyšla by nám euklidova vzdialenosť (veľké číselné rozdiely medzi dátami). Výsledok sme aj vyexportovali do súboru s názvom euclidAllData.csv.

## Úloha 3

**Zadanie úlohy:**

Využite algoritmy zhlukovania na nájdenie skupín podobných krajín. Vašim cieľom by malo byť, aby váš model rozdelil krajiny podľa ich geografickej polohy (stĺpec Region v originálnych dátach) -> podľa tohto vyberajte modely, parametre a vstupné dáta. Vyskúšajte aspoň jeden zhlukovací algoritmus s dopredu uvedeným počtom zhlukov (napr. k-means) a s dopredu neznámym počtom zhlukov (napr. DBSCAN)

**Vypracovanie:**

Ako prvé sme využili K-means algoritmus z knižice sklearn. V tomto algoritme bolo potrebné aby sme nastavili počet klastrov. V našom prípade sme si ich určili 10, keďže aj regiónov je 10. Dáta, sme vyexportovali d súboru K-means.csv (K-means.xls, konvertovane online editorom).

Ďalší algoritmus, ktorý sme využili je DBSCAN z sklearn. Epsilón (vzdialenosť) sme nastavili na 1.68 a minimálny počet bodov je 2.

## Úloha 4

**Zadanie úlohy:**

Analyzujte získané výsledky. Vedeli ste sa priblížiť k geografickému rozloženiu? Ak nie, prečo? Čo ste v rámci zadania vyskúšali, aby tomu tak bolo? Analyzujte zhluky pre aspoň dve rôzne spustenia/nastavenia – aké sú veľké, čo je v nich, ktorá krajina je vhodným reprezentantom (je najbližšie k centroidu).

**Vypracovanie:**

**K-means**Našich 30 krajín rozdelilo do 10 klastrov. Ako môžete vidieť v súbore K-means.xls alebo nižšie na obrázku, v jednotlivých klastroch sú aj rovnaké regióny ale aj rozdielne. Domnievame sa, že z dôvodu podobnosti dát ich pridalo do rovnakého klastru, aj keď nepatria do rovnakého regiónu.

|  |  |
| --- | --- |
| Kluster | Počet |
| 0 | 6 |
| 1 | 3 |
| 2 | 6 |
| 3 | 1 |
| 4 | 3 |
| 5 | 1 |
| 6 | 4 |
| 7 | 1 |
| 8 | 3 |
| 9 | 2 |

****

## 

## 

## *Obrázok 1 – K-means*

## DBSCAN

## V tomto prípade nám rozdelilo krajiny do 8 klastrov. Predpokladáme, že z podobného dôvodu ako v K-means – podobnosť dát. Avšak Mohlo to takisto nastať aj kvôli nastavením hodnotám eps a min\_sample (vzdialenosť a minimálny počet bodov).

|  |  |
| --- | --- |
| Kluster | Počet |
| 0 | 9 |
| 1 | 13 |
| 2 | 3 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |
| 6 | 1 |
| 7 | 1 |

## 

## 

## 

## *Obrázok 2 – DBscan*

## SKlearn

Na vypracovanie úloh sme používali funkcie z knižnice sklear. Celá dokumentácia je dostupná na webovej stránke https://scikit-learn.org/stable/index.html.

Využité funkcie:

**MinMaxScaler** – normalizuje dáta do určitého rozsahu

**euclidean\_distances** – berie do úvahy riadky X (poprípadne Y = X) ako vekotry a vypočíta maticovú vzdialenosť medzi dvojicou vektorov

**KMeans** – na začiatku určíme počet clustrov (zvolený náhodný cenetroid). Každý bod sa priradí ku tomu zhluku, ku ktorému je najbližšie. Po pridaní bodu do clustra sa vypočíta nový centroid (do konvergencie alebo zastavujúcej podmienky)

**DBSCAN** – Vyberie sa bod, ktorý nebol ešte navštívený. Ak ma vo svojom okolí (eps) dostatočný počet bodov (min\_samples) stane sa z neho počiatok clustra. Ďalej sa pokračije na nenavštívený bod v klustri, pridáme body z jeho okolia a označíme ho ako navštívený (do konvergencie alebo zastavujúcej podmienky).