I-SUNS: Zadanie č.1

Spracovanie dát, neurónové siete I.

Vypracoval: Jakub Šíp

AIS ID: 91419

Na implementáciu tohto zadania som si zvolil programovací jazyk Python 3.8 a knižnicu sklearn. Pre spracovanie dát som použil knižnicu pandas, ktorá uľahčuje prácu s dátovými štruktúrami.

Úloha 1:

- Dáta som načítaval do pandas DataFrame-u, ktorý je plne kompatibilný s sklearn
- Vďaka knižnici pandas sa s dátami pracuje ako s tabuľkou a jednotlivé stlpce vieme vyberať/meniť aj na základe ich názvu v headeri

Úloha 2:

- Z dát som odstránil stĺpec ID, ktorý je pre nás nepodstatný a všetky riadky, ktorým chýbala niektorá z hodnôt
- Stĺpec AGE som previedol z dní na roky a zvyšné číselné údaje skontroloval a vyradil nezmyselné (napríklad vek 5000+ rokov)
- Prekódoval som dáta zadané slovne do číselnej podoby a to nasledovne normal = 0, above normal = 0.5, well above normal = 1 a dáta s pohlavím do binárneho stavu man = 0, woman = 1
- Do takto vyfiltrovaných dát som pridal nový stĺpec obsahujúci BMI
- Tieto dáta som uložil do druhého csv s názvom ‚tip_top.csv' kvôli časovej náročnosti tohto kroku
- Normalizácia prebieha pred každým trénovaním neurónovej siete za pomoci funkcie StandardScaler(), ktorá upraví hodnoty na rozsah +/- 1

Úloha 3:

- Najmocnejší ukazovateľ je pre nás asi krvný tlak
- Na ďalšiu analýzu je však vhodné poslať aj dáta o životnom štýle (alco, smoke, active) + cholesterol, glucose, gender, age.

Úloha 4:

- Na túto úlohu som použil funkciu MLPClassifier()
- Neurónovú sieť som trénoval s týmito parametrami hidden_layer_sizes=(10, 10, 10), max_iter=1000, learning_rate='adaptive'
- Dáta som rozdelil tak, že 80% dát sa využilo na trénovanie a 20% na testovanie
- výsledná presnosť klasifikácie 73%
- Maximálny proces iterácií v tomto prípade 1000 nikdy nenastane, lebo učenie sa zastaví, keď sa presnosť predpovede nezlepší 10 nasledujúcich iterácií

- Ani po viac násobnom skúšaní so zmenou parametrov sa mi však nepodarilo zvýšiť úspešnosť nad 73-74%. Tá sa zvýšila, len v prípade, že na testovanie bolo použitých len 5% dát, čo je zas veľmi malé množstvo.
- Táto neurónová sieť je klasifikačná, takže jej výstup sú v našom prípade len 2 hodnoty a to 1 alebo 0

	precision	recall	f1-score	support
0	0.71	0.78	0.75	6909
1	0.75	0.68	0.71	6792
accuracy			0.73	13701
macro avg	0.73	0.73	0.73	13701
weighted avg	0.73	0.73	0.73	13701

Úloha 5:

- Na túto úlohu som využil funkcie MLPRegressor() a LinearRegression()
- Neurónovú sieť som učil s rovnakými parametrami ako klasifikátor teda 1000 iterácií na učenie, hidden_layers=(10,10,10) atď.
- Z dát som samozrejme vylúčil výšku a váhu, inak by tento krok nemal zmysel
- Z nami nameraných dát môžeme vidieť, že na odhad BMI je z nami použitých metód lepšia práve lineárna regresia

```
---------------MLPRegressor-----------------
Mean Squared Error: 1.2692864054623735e-06
Root Mean Squared Error: -3.7639402400948523
---------------LinearRegressor---------------
Mean Squared Error: 2.356951721164690 2e-07
Root Mean Squared Error: 0.11537875927009278
```

Nepovinné úlohy:

- Pri oboch neurónový sieťach som aj so zmenou parametrov dostal veľmi podobné výsledky. Najväčší vplyv na ich funkcionalitu mala zmena rozdelenia pomeru dát na testovanie a trénovanie. V tomto prípade si klasifikátor viedol rovnako ale pri regresii

to malo veľký vplyv na jej presnosť. Kde pri trénovaní len na 40% dát bol výsledok takýto:

```
---------------MLPRegressor-----------------
Mean Squared Error: 0.00019473033671627636
Root Mean Squared Error: -724.1439967656576
---------------LinearRegressor-----------------
Mean Squared Error: 2.3775272898282645e-07
Root Mean Squared Error: 0.11464763506394815
```

- a koeficient determinácie dosiahol hodnoty až -724.143