

Úloha C

V tejto úlohe sa snažíme predikovať kvalitu vína, inšpirovaní prístupom Orleya Ashenfeltera k predikcii cien vína z Bordeaux.

Využívame dáta zo súboru A04wine.csv a aplikujeme modely L1 a L-inf z úlohy A, a využívame prístup rovnaký ako v B.

Na implementáciu formulovaných LP úloh využívame opäť:

- pandas - načítanie dát z .csv
- numpy - tvorenie matíc a vektorov
- scipy.optimize - implementovaný LP solver

Vyberieme z dát dané nezávislé premenné x a závislú premennú y

```
y = data['Price']
x = data[['WinterRain', 'AGST', 'HarvestRain', 'Age', 'FrancePop']]
```

Zistíme počet premenných (plus 1 pre intercept)

```
numberOfVariablesBeta = x.shape[1] + 1
```

Vytvoríme potrebné štruktúry pre zostavenie modelu normy l1

```
c = np.array([0]*numberOfVariablesBeta + [1] * len(x.values)) #
                                Objective function coefficients

ALeft = np.matrix([ [1] * len(x.values)]).transpose() # Coefficients for
                                beta0
ARigth = np.matrix(x.values) # Coefficients for other independent
                                variables beta
A = np.block([ALeft, ARigth]) # Concatenate coefficients of variables into
                                one matrix

I = np.identity(len(x.values)) # Identity matrix for intercept term
```

Naformulujeme problém a vyriešime pomocou scipy.optimize.linprog

```
A_ub = np.block([-A, -I], [A, -I])
b_ub = np.concatenate([-y, y])

solve = linprog(c, A_ub, b_ub, bounds = [(None, None)] *
                                numberOfVariablesBeta + [(0, None)])
```

Po vyriešení vyberieme z riešenia koeficienty

```
betas = solve.x[:numberOfVariablesBeta]
```

Čo nám dá:

$$\beta_0^{(1)} \approx -8.88014916e - 01, \beta_1^{(1)} \approx 1.57938208e - 03, \beta_2^{(1)} \approx 5.21297712e - 01$$

$$\beta_3^{(1)} \approx -4.51366553e - 03, \beta_4^{(1)} \approx 1.13002665e - 02, \beta_5^{(1)} \approx -2.21114476e - 05$$

Ďalej zostrojíme relevantné štruktúry a naformulujeme LP pre l - inf normu

```

c_inf = np.array([0]*numberOfVariablesBeta + [1]) # Koeficienty cielovej
                                                funkcie
A_inf = np.block([ALeft, ARigth]) # Koeficienty pre nezávislé premenné
                                   pre l_inf normu
i_inf = np.array([ [1] * len(x.values)]).transpose() # Koeficienty pre
                                                        intercept term pre l_inf normu

A_ub_inf = np.block([-A_inf, -i_inf], [A_inf, -i_inf])
b_ub_inf = np.concatenate([-y, y])

```

Vyriešime aj tento problém pomocou scipy pre l - inf normu a vyberieme beta-koeficienty

```

solve_inf = linprog(c_inf, A_ub_inf, b_ub_inf, bounds=[(None, None)]*
                                                        numberOfVariablesBeta + [(0, None)])
betas_inf = solve_inf.x[:numberOfVariablesBeta]

```

Po čom dostaneme:

$$\beta_0^{(2)} \approx 3.48412213e + 00, \beta_1^{(2)} \approx 8.33985438e - 04, \beta_2^{(2)} \approx 6.00266169e - 01$$

$$\beta_3^{(2)} \approx -3.34163738e - 03, \beta_4^{(2)} \approx -2.30363775e - 02, \beta_5^{(2)} \approx -1.19583600e - 04$$