1 Predikcia kvality vína

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 L^1 a L^∞ z úlohy A. Budeme využívať podobný postup ako v úlohe B. Na implementáciu formulovaných LP úloh využívame:

- `pandas` - načítanie dát z `csv` súboru
- `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']]
# Calculate the number of variables (features)
k = x.shape[1]
```

Vytvoríme potrebné štruktúry pre zostavenie modelu normy L^1:

```
c = np.array([0]*(k + 1) + [1] * len(x.values)) # Objective function
                                    coefficients (plus 1 for the
                                    intercept term)

A = np.block([np.ones((len(x.values), 1)), np.array(x.values)]) #
                                    Concatenate coefficients of
                                    variables into one matrix
```

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)]*(k + 1) + [(0,
                                    None)] * len(x.values))
```

Po vyriešení vyberieme z riešenia koeficienty, čo nám dá:

$$\beta_0^{(1)} \approx -8.8801 \cdot 10^{-1},\ \beta_1^{(1)} \approx 1.5793 \cdot 10^{-3},\ \beta_2^{(1)} \approx 5.2130 \cdot 10^{-1}$$
$$\beta_3^{(1)} \approx -4.5137 \cdot 10^{-3},\ \beta_4^{(1)} \approx 1.1300 \cdot 10^{-2},\ \beta_5^{(1)} \approx -2.2111 \cdot 10^{-5}$$

Z týchto výsledkov môžeme usúdiť, že najviac pozitívne vplýva na cenu vína metrika *AGST - Average growing season temperature* a najsignifikantnejší negatívny vplyv má *dážď počas zberu*.

Ďalej zostrojíme relevantné štruktúry a naformulujeme LP pre L^∞ normu:

```
c_inf = np.array([0]*(k+1) + [1]) # Objective function coefficients

A_inf = np.block([np.ones((len(x.values),1)), np.array(x.values)]) #
                                    Coefficients for independent
                                    variables for l-inf norm

i_inf = np.ones((len(x.values), 1)) # Coefficients for values of vector
                                    t
# inequality constraints for l-inf norm
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.optimize.linprog()` pre L^∞ normu a vyberieme β koeficienty:

```
solve_inf = linprog(c_inf, A_ub_inf, b_ub_inf, bounds=[(None, None)]*(k
                                                + 1) + [(0, None)])
```

$$\beta_0^{(\infty)} \approx 3.4841,\ \beta_1^{(\infty)} \approx 8.3399 \cdot 10^{-4},\ \beta_2^{(\infty)} \approx 6.0027 \cdot 10^{-1}$$
$$\beta_3^{(\infty)} \approx -3.3416 \cdot 10^{-3},\ \beta_4^{(\infty)} \approx -2.3036 \cdot 10^{-2},\ \beta_5^{(\infty)} \approx -1.1958 \cdot 10^{-4}$$

Vidíme, že aj lineárna regresia pomocou L^∞ normy odhaduje najväčší pozitívny vplyv metriky *AGST* a najväčší negatívny vplyv *dažďu počas zberu*. Zmenil sa však vplyv premennej *vek* (oproti prechádzajúcemu modelu) z pozitívneho na negatívny.