Algorytmy statystyki praktycznej 28.02.2019

Lista nr 1: Python - NumPy - Pandas - Regresja liniowa

Paweł Lorek

Na stronie
http://www.math.uni.wroc.pl/~lorek/alg_stat_prakt/python/
Jest m.in. plik `kc_house_data_csv.zip`
Plik ten (po rozpakowaniu) ma taka strukture:

- Pierwszy wiersz: wymienione nazwy pól po przecinku
- Kolejne wiersze = dane

Pierwsze 3 linijki

```
id,date,price,bedrooms,bathrooms,sqft_living,sqft_lot,floors,waterfront,view,condition,grade,sqft_above,sqft_basement,yr_built,yr_renovated,zipcode,lat,long,sqft_
"7129300520","20141013T000000",221900,3,1,1180,5650,"1",0,0,3,7,1180,0,1955,0,"98178",47.5112,-122.257,1340,5650
"6414100192","20141209T000000",538000,3,2.25,2570,7242,"2",0,0,3,7,2170,400,1951,1991,"98125",47.721,-122.319,1690,7639
```

Takie pliki .csv (w szczególności, gdy są bardzo duże) najwygodniej wczytać używając biblioteki `pandas`

```
1 import numpy as np
2 import pandas as pd
3
4 dane = pd.read_csv(kc_house_data_csv)
5 print(type(dane)) # jest to tzw. typ DataFrame
6 dane.head()
7 X=dane[['price','bedrooms']]
8 Xnp=X.values
```

Krotkie wyjasnienie:

- Linia 4: wczytanie danych , w linii 6 widzimy, iz typem jest `DataFrame`
- Linia 5: wyswietla skrotowo kilka pierwszych linii (dodaje "naglowki" oraz id wierszy (pierwsza kolumna)
- Linia 7: wydobywamy kolumny "price" oraz "bedrooms"
- Linia 8: Zamiana `X` typu `DataFrame` na macierz Numpy

ZADANIA

1. Podziel wszystkie dane na `treningowe` (zbiór A, ok. 70%) oraz `testowe` (zbiór B, ok 30%), najlepiej losowo.

2. Jak w opisanym przykładzie, wydobądź z danych pary (price,bedrooms): na danych treningowych przeprowadź regresję liniową, tj. wyznacz θ_0, θ_1 takie, które minimalizują błąd:

$$\frac{1}{|A|} \sum_{(price,bedrooms)\in A} (\theta_0 + bedrooms \cdot \theta_1 - price)^2$$

   Przy wyznaczonych θ_0, θ_1 oblicz ("jakość naszego dopasowania"):

$$R = \frac{1}{|B|} \sum_{(price,bedrooms)\in B} (\theta_0 + bedrooms \cdot \theta_1 - price)^2$$

3. Powtórz poprzedni punkt (przy tym samym A oraz B) wybierając różne pary, np. $(price, bathrooms)$, $(price, sqft\_living)$, itd. Która cecha daje najlepsze wyniki?

4. Przeprowadź regresję uwzględniając więcej parametrów, tj. model jest taki:

$$price = \theta_0 + bedrooms \cdot \theta_1 + bathrooms \cdot \theta_2 + sqft\_living\theta_3 + \ldots \theta_d,$$

   a zadaniem jest znalezienie optymalnych parametrów $(\theta_0, \ldots, \theta_d)$. Wylicz następnie analogicznie R i porównaj z poprzednimi.

5. Zapoznaj z innymi modelami liniowymi dostępnymni w bibliotece sklearn, użyj ich do powyższego przykładu. Szkic użycia np. modelu Lasso:

```
1  import numpy as np
2  from sklearn import linear_model
3
4  #zakladajac, ze xA to punkty (n x d), a yA to wartosci
5  #ze zb. treningowego A, a xB to punkty ze zbioru testowego
6
7  clfLasso = linear_model.Lasso(alpha = 1)
8  clfLasso.fit(xA,yA) #uczenie modelu
9
10 yB_predicted = clfLasso(xB)  #zaaplikowanie do danych testowych
```

   Lista (i opis) dostępnych metod:

   https://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model