Formiranje listi i matrica korišćenjem *NumPy* biblioteke

Autor: dr Miloš Todorov

Novi Sad, april 2023. godine

Sadržaj

Python liste

- Liste u Pythonu mogu da sadrže stringove, brojeve, ili druge liste. One služe za skladištenje drugih objekata

- Elementima liste pristupamo pomoću indeksa, sa tim da u Pythonu indeksi kreću od nule

- Liste u Pythonu mogu da se proširuju, skraćuju, spajaju, kao i da se obriše deo liste. Python podržava mogućnost da se liste kreiraju u okviru liste

- Python ima sledeće tipove podataka koji mogu da se nađu u listama

Tabela 1: Tipovi podataka u Pythonu

String	Predstavljaju tekstualne podatke koji se prikazuju pod navodnicima, na primer 'Lista'
Integer	Predstavljaju cele brojeve, na primer -2, 0, 5
Float	Predstavljaju realne brojeve, na primer 4.2, -3.6
Complex	Predstavljaju kompleksne brojeve, npr 3.0 + 2.0j
Boolean	Predstavlja tip koji može imati logičke vrednosti, tačno, ili netačno, odnosno True, ili False

Primer 1: Formiranje liste u Pythonu

Input:

```
#Formiranje liste u Pythonu
listaElemenata = ['a','b','c',['e','m'],'d','f',5,6,10,11.5]
print(listaElemenata)
```

Output:

```
In [1]:
   ...: listaElemenata = ['a','b','c',['e','m'],'d','f',5,6,10,11.5]
   ...: print(listaElemenata)
['a', 'b', 'c', ['e', 'm'], 'd', 'f', 5, 6, 10, 11.5]
```

Primer 2: Pristupanje elementu liste

Input:

```
#Pristupanje elementima liste pomocu indeksa
listaElemenata[3][1]
```

Output:

```
In [2]:
   ...: listaElemenata[3][1]
Out[2]: 'm'
```

Ovo je primer ugnježdene liste. Pošto indeksi u Pythonu kreću od nule, lista u okviru liste sa elementima ['e','m'] se nalazi na trećem indeksu, a elemenat 'm' je u okviru date liste na prvom mestu, tj.indeksu.

Primer 3: Dodavanje elementa na kraj listi

Input:

```
#Dodvanje Elementa na kraju liste
listaElemenata.append('DodatiElement')
print(listaElemenata)
```

Output:

```
In [3]:
   ...: listaElemenata.append('DodatiElement')
   ...: print(listaElemenata)
['a', 'b', 'c', ['e', 'm'], 'd', 'f', 5, 6, 10, 11.5, 'DodatiElement']
```

U prethodnom primeru 3. je prikazana jedna od metoda koja može da se primeni na Python listama. Sledi tabela sa metodama koje se koriste u okviru Python listi.

Tabela 2: Metode u okviru lista u Pythonu

append()	Dodavanje elementa na kraju liste
clear()	Briše sve elemente liste
copy()	Vraća kopiju liste
count()	Prebraja elemente liste
extend()	Dodaje elemente neke druge liste, na kraj postojeće liste
index()	Vraća elemenat na definisanoj poziciji indeksa
insert()	Ubacuje elemenat u listu na određenu poziciju
pop()	Briše elemenat u listi na određenoj poziciji indeksa
remove()	Briše tačno određeni elemenat u listi koji se navede u okviru ove metode, odnosno naredbe
reverse()	Obrće elemente u listi
sort()	Sortira elemente u listi

U prethodnoj tabeli 2. Prikazane su neke od naredbi koje mogu da se primene na Python listama. Pored python naredbi, postoji biblioteka NumPy, koja je pogodna za rad sa listama i matricama. Kreirao ju je Travis Oliphant 2005. godine.

Ova Python biblioteka se nalazi na sledećoj web adresi: https://numpy.org/

Sledi prikaz date biblioteke.

Prednosti biblioteke NumPy

Neke od prednosti korišćenja biblioteke Numpy [2]:

- NumPy kreira niz objekata koji je 50x puta brži od klasičnih Python listi

- Niz objekata u NumPy se naziva ndarray, koji obezbeđuje korišćenje gomile funkcija sa listama

- Sadrži funkcije koje mogu da se koriste u matematičkim oblastima linearne algebre, Furijeove transformacije, kao i matricama, kao i mnogih drugih matematičkih funkcija

- Ima veliku upotrebu u oblasti veštačke inteligencije, odnosno pri korišćenju ML agoritama

- NumPy liste se smeštaju na jednom mestu u memoriji i iz tog razloga, lako im je pristupiti, što mogućava optimizaciju i brži rad sa listama

- Numpy ima mogućnost kreiranja različitih distribucija, kao što su Normalna, Poasonova, Uniformna, Logistička itd…

- Pored svega navedenog NumPy ima sve metode, kao i klasične Python liste, dok je lista tipova podataka proširena

Numpy liste

- Kod Numpy listi važe sve operacije sa listama kao i kod klasičnih Python listi

- Elementima NumPy liste pristupamo takođe pomoću indeksa, kao što je prikazano u primeru 2

Slede primeri kreiranja listi. Da bi se koristila neka biblioteka u Pythonu, ona mora da se pozove pomoću naredbe *import*.

Primer 4: Kreiranje različitih Numpy listi

Input:

```
#Kreiranje listi pomocu Numpy biblioteke
import numpy as np
L1 = np.array([1, 1])# jednodimenzionalna lista
print(L1)

L2 = np.array([[3, 2, 1], [7, 8, 9]]) # Dvodimenzionalna lista sa integerima
print(L2)

L3 = np.array([[3.5, 2, 1.0], [7.5, 8, 9.1]]) # Dvodimenzionalna lista sa realnim brojevima
print(L3)

L4 = np.array([3+7j, 2+8j, 1+9j], dtype = complex) # Jednodimenzionalna lista sa kompleksnim brojevima
print(L4)
```

Output:

```
In [1]:
   ...: import numpy as np
   ...: L1 = np.array([1, 1])# jednodimenzionalna lista
   ...: print(L1)
[1 1]

In [2]: L2 = np.array([[3, 2, 1], [7, 8, 9]]) # Dvodimenzionalna lista sa
integerima
   ...: print(L2)
[[3 2 1]
 [7 8 9]]

In [3]: L3 = np.array([[3.5, 2, 1.0], [7.5, 8, 9.1]]) # Dvodimenzionalna
lista sa realnim brojevima
   ...: print(L3)
[[3.5 2.  1. ]
 [7.5 8.  9.1]]

In [4]: L4 = np.array([3+7j, 2+8j, 1+9j], dtype = complex) #
Jednodimenzionalna lista sa kompleksnim brojevima
   ...: print(L4)
[3.+7.j 2.+8.j 1.+9.j]
```

Postoji i drugi načini da kreiraju brojčane liste u zavisnosti od potrebe, a što će biti prikazano u narednom primeru.

Primer 5: Nekoliko drugih načina kreiranja Numpy listi

Input:

```
#Drugi način kreiranja listi
L5 = np.arange(1,6) # kreiranje jednodimenzionalne liste integera od 1 do 5
#Ovde lista mora da ide do 6, jer indeksi u Python idu do nekog broja, ali ne obuhvataju i sam broj
print(L5)

L6 = np.ones(5) #kreiranje liste gde su svi elementi jedinice
print(L6)

L7 = np.zeros(5) #kreiranje liste gde su svi elementi nule
print(L7)

L8 = np.linspace(0,5,6) # Kreiranje liste brojeva sa jednakim intervalima
print(L8)

L9 = np.random.randn(5) # kreiranje liste brojeva sa normalnom raspodelom
print(L9)

L10 = np.random.randint(1,10,5) # kreiranje liste brojeva sa slučajnim brojevima
print(L10)
```

Output:

```
In [5]: L5 = np.arange(1,6) # kreiranje jednodimenzionalne liste integera od 1 do 5
   ...: #Ovde lista mora da ide do 6, jer indeksi u Python idu do nekog broja, ali ne obuhvataju i sam broj
   ...: print(L5)
[1 2 3 4 5]

In [6]: L6 = np.ones(5) #kreiranje liste gde su svi elementi jedinice
   ...: print(L6)
[1. 1. 1. 1. 1.]

In [7]: L7 = np.zeros(5) #kreiranje liste gde su svi elementi nule
   ...: print(L7)
[0. 0. 0. 0. 0.]

In [8]: L8 = np.linspace(0,5,6) # Kreiranje liste brojeva sa jednakim intervalima
   ...: print(L8)
[0. 1. 2. 3. 4. 5.]

In [9]: L9 = np.random.randn(5) # kreiranje liste brojeva sa normalnom raspodelom
   ...: print(L9)
[ 0.77733734  0.8105555  -0.95392299 -0.38262497  0.97440534]

In [10]: L10 = np.random.randint(1,10,5) # kreiranje liste brojeva sa slučajnim brojevima
    ...: print(L10)
[4 5 6 7 2]
```

MATRICE

Neka je zadat sistem od *m* linearnih jednačina sa *n* nepoznatih $x_1, x_2, \cdots, x_n$.

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{12}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{gathered}$$

Brojevi $a_{ij}, b_i, i = 1,2, \cdots, m, j = 1,2, \cdots, n$ nazivaju se koeficijenti sistema.

Matricom nazivamo pravougaonu šemu oblika mxn sastavljenu od elemenata a_{ij}, raspoređenih u *m* vrsta i *n* kolona.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{matrix} \rightarrow 1.VRSTA \\ \rightarrow 2.VRSTA \\ \vdots \\ \rightarrow m.VRSTA \end{matrix}$$

$$\begin{matrix} \downarrow & \downarrow & & \downarrow \\ 1. & 2. & & n. \\ K & K & & K \\ O & O & & O \\ L & L & & L \\ O & O & & O \\ N & N & & N \\ A & A & & A \end{matrix}$$

Matrica se može označiti na sledeći način $[a_{ij}]_{mxn}$. Ova matrica ima koja ima *m* vrsta i *n* kolona ima dimenziju mxn.

Matrica $\begin{bmatrix} 5 & 1 & 2 \\ 6 & 6 & 7 \end{bmatrix}$ je reda $2x3$, jer ima 2 vrste i 3 kolone

NAPOMENA: Determinanta je realan broj, koji je zapisan kao šema brojeva, za razliku od matrice koja je šema proizvoljnih elemenata.

Elementi $a_{11}, a_{22}, \cdots, a_{mn}$ su na GLAVNOJ DIJAGONALI (označeno plavom bojom), a elementi $a_{1n}, \cdots, a_{m1}$ su na SPOREDNOJ DIJAGONALI (označeno crvenom bojom).

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

OPERACIJE SA MATRICAMA

Sabiranje matrica

Zbir matrica $A = [a_{ij}]_{mxn}$ i $B = [b_{ij}]_{mxn}$ koje su **istih** dimenzija je matrica $C = [c_{ij}]_{mxn}, i = 1,2,\cdots,m,\ j = 1,2,\cdots,n,$ gde je $[a_{ij}]_{mxn} + [b_{ij}]_{mxn} = [c_{ij}]_{mxn}$. Mogu da se sabiraju samo matrice **istih** dimenzija. Sabira se svaki elemenat prve matrice sa odgovarajućim elementom druge matrice.

$$C = A + B = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

$$C = A + B = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}$$

Važe i sledeće osobine:

- Komutativnost $A + B = B + A$
- Asocijativnost $(A + B) + C = A + (B + C)$

Primer 6: Sabrati matrice

$$A = \begin{bmatrix} 1 & 3 & 1 \\ 1 & 5 & 4 \\ 1 & 2 & 4 \end{bmatrix}_{3x3} \text{ i } B = \begin{bmatrix} 1 & 2 & 1 \\ -1 & 4 & 8 \\ -2 & -1 & 4 \end{bmatrix}_{3x3}$$

$$C = A + B = \begin{bmatrix} 1 & 3 & 1 \\ 1 & 5 & 4 \\ 1 & 2 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 1 \\ -1 & 4 & 8 \\ -2 & -1 & 4 \end{bmatrix} = \begin{bmatrix} 1+1 & 3+2 & 1+1 \\ 1-1 & 5+4 & 4+8 \\ 1-2 & 2-1 & 4+4 \end{bmatrix}$$

$$C = A + B = \begin{bmatrix} 2 & 5 & 2 \\ 0 & 9 & 12 \\ -1 & 1 & 8 \end{bmatrix}$$

Proizvod matrice sa skalarom

Proizvod matrice sa skalarom je matrica koja se dobije kada scalar $\gamma \in R$ pomnožimo sa svakim elementom matrice $A = [a_{ij}]_{mxn}$, odnosno $\gamma \cdot A = \gamma \cdot [a_{ij}]_{mxn} = [\gamma \cdot a_{ij}]_{mxn}$

$$\gamma \cdot A = \gamma \cdot \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} \gamma \cdot a_{11} & \gamma \cdot a_{12} & \cdots & \gamma \cdot a_{1n} \\ \gamma \cdot a_{21} & \gamma \cdot a_{22} & \cdots & \gamma \cdot a_{2n} \\ \vdots & \vdots & & \vdots \\ \gamma \cdot a_{m1} & \gamma \cdot a_{m2} & \cdots & \gamma \cdot a_{mn} \end{bmatrix}$$

Važe i sledeće osobine:

- Komutativnost $\gamma \cdot A = A \cdot \gamma$
- Asocijativnost $(\alpha \cdot \beta) \cdot A = \alpha \cdot (\beta \cdot A)$
- Distributivnost u odnosu na zbir skalara $(\alpha + \beta) \cdot A = \alpha \cdot A + \beta \cdot A$
- Distributivnost u odnosu na zbir matrica $\alpha \cdot (A + B) = \alpha \cdot A + \alpha \cdot B$

Primer 7: Pomnožiti matricu $A = \begin{bmatrix} 1 & 3 & 1 \\ 1 & 5 & 4 \\ 1 & 2 & 4 \end{bmatrix}_{3x3}$ sa skalarom $\gamma = 2$.

$$2 \cdot A = 2 \cdot \begin{bmatrix} 1 & 3 & 1 \\ 1 & 5 & 4 \\ 1 & 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 \cdot 1 & 2 \cdot 3 & 2 \cdot 1 \\ 2 \cdot 1 & 2 \cdot 5 & 2 \cdot 4 \\ 2 \cdot 1 & 2 \cdot 2 & 2 \cdot 4 \end{bmatrix} = \begin{bmatrix} 2 & 6 & 2 \\ 2 & 10 & 8 \\ 2 & 4 & 8 \end{bmatrix}$$

Proizvod matrica

Proizvod matrica je matrica $C = [c_{ij}]_{mxn}$ koja se dobije kada se pomnože matrice $A = [a_{ij}]_{mxp}$ i $B = [b_{ij}]_{pxn}$. Elementi matrice C se dobijaju na sledeći način:

$$c_{ij} = \sum_{K=1}^{p} a_{ik} \cdot b_{kj}$$

$$C = A \cdot B = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mp} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pn} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} \cdot b_{11} + a_{12} \cdot b_{21} + \cdots + a_{1p} \cdot b_{p1} & a_{11} \cdot b_{12} + a_{12} \cdot b_{22} + \cdots + a_{1p} \cdot b_{p2} & \cdots & a_{11} \cdot b_{1n} + a_{12} \cdot b_{2n} + \cdots + a_{1p} \cdot b_{pn} \\ a_{21} \cdot b_{11} + a_{22} \cdot b_{21} + \cdots + a_{2p} \cdot b_{p1} & a_{21} \cdot b_{12} + a_{22} \cdot b_{22} + \cdots + a_{2p} \cdot b_{p2} & \cdots & a_{21} \cdot b_{1n} + a_{22} \cdot b_{2n} + \cdots + a_{2p} \cdot b_{pn} \\ \vdots & \vdots & & \vdots \\ a_{m1} \cdot b_{11} + a_{m2} \cdot b_{21} + \cdots + a_{mp} \cdot b_{p1} & a_{m1} \cdot b_{12} + a_{m2} \cdot b_{22} + \cdots + a_{mp} \cdot b_{p2} & \cdots & a_{m1} \cdot b_{1n} + a_{m2} \cdot b_{2n} + \cdots + a_{mp} \cdot b_{pn} \end{bmatrix}$$

Napomena: Proizvod dve matrice je definisan *samo* ako je *broj kolona prve matrice jednak sa brojem vrsta druge matrice!*

Važe i sledeće osobina:

- Asocijativnost $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

U opštem slučaju **ne važi** komutativnost $A \cdot B \neq B \cdot A$.

Primer 8: Pomnožiti matrice $A = \begin{bmatrix} 1 & 3 & 1 \\ 1 & 5 & 4 \\ 1 & 2 & 4 \end{bmatrix}_{3x3}$ i $B = \begin{bmatrix} 1 & 2 \\ -1 & 4 \\ -2 & -1 \end{bmatrix}_{3x2}$

$$C = A \cdot B = \begin{bmatrix} 1 & 3 & 1 \\ 1 & 5 & 4 \\ 1 & 2 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ -1 & 4 \\ -2 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \cdot 1 + 3 \cdot (-1) + 1 \cdot (-2) & 1 \cdot 2 + 3 \cdot 4 + 1 \cdot (-1) \\ 1 \cdot 1 + 5 \cdot (-1) + 4 \cdot (-2) & 1 \cdot 2 + 5 \cdot 4 + 4 \cdot (-1) \\ 1 \cdot 1 + 2 \cdot (-1) + 4 \cdot (-2) & 1 \cdot 2 + 2 \cdot 4 + 4 \cdot (-1) \end{bmatrix}$$

$$C = A \cdot B = \begin{bmatrix} 1 - 3 - 2 & 2 + 12 - 1 \\ 1 - 5 - 8 & 2 + 20 - 4 \\ 1 - 2 - 8 & 2 + 8 - 4 \end{bmatrix} = \begin{bmatrix} -4 & 13 \\ -12 & 18 \\ -9 & 6 \end{bmatrix}$$

TRANSPONOVANA MATRICA

Transponovana matrica je matrica u kojoj su vrste(kolone) zamenile mesta sa odgovarajućim kolonama(vrstama), odnosno za matricu $A = [a_{ij}]_{mxn}$ njena transponovana matrica je $A^T = [a_{ji}]_{nxm}$.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \Longrightarrow A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$$

Primer 9: Neka je data matrica $A = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$. Naći njenu transponovanu maticu.

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \Longrightarrow A^T = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Primer 10: Neka je data matrica $A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$. Naći njenu transponovanu maticu.

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \Longrightarrow A^T = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

INVERZNA MATRICA

Inverzna matrica je *kvadratna* matrica A^{-1}, koja zadovoljava sledeću osobinu $A^{-1} \cdot A = A \cdot A^{-1} = I$, gde je *I jedinična* matrica.

Kvadratna matrica A je **regularna** ako je $detA \neq 0$, dok je **singularna** ako je $detA = 0$.
Minor M_{ij} je determinanta koje sa dobija kada iz determinante *D* odbacimo *i*-tu vrstu i *j*-kolonu.

Minor M_{33} se nalazi na sledeći način.

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \qquad M_{33} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}$$

Kofaktor elementa a_{ij} je broj $A_{ij} = (-1)^{i+j} \cdot M_{ij}$.

Primer 11: Naći kofaktor A_{33} determinante $\begin{vmatrix} 2 & 5 & 3 \\ 1 & 4 & 3 \\ 2 & 6 & 4 \end{vmatrix}$.

$$A_{33} = (-1)^{3+3} \cdot \begin{vmatrix} 2 & 5 & 3 \\ 1 & 4 & 3 \\ 2 & 6 & 4 \end{vmatrix} = (-1)^6 \cdot \begin{vmatrix} 2 & 5 \\ 1 & 4 \end{vmatrix} = 1 \cdot (2 \cdot 4 - 5 \cdot 1)$$
$$= 8 - 5 = 3$$

Adjugovana matrica matrice *A* u oznaci $adjA$ je *transponovana kvadratna* matrica koja se dobija od kofaktora matrice *A*.

$$A = \begin{bmatrix} A_{11} & A_{21} & \cdots & A_{m1} \\ A_{12} & A_{22} & \cdots & A_{m2} \\ \vdots & \vdots & & \vdots \\ A_{1n} & A_{2n} & \cdots & A_{mn} \end{bmatrix}$$

Inverzna matrica A^{-1} koja se dobija od *kvadratne regularne* matrice *A* jednaka je:

$$A^{-1} = \frac{1}{detA} \cdot adjA$$

Primer 12: Naći inverznu matricu A^{-1} matrice $\begin{bmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{bmatrix}$.

$$detA = \begin{vmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{vmatrix}\begin{matrix}1 & 1\\ 1 & 3\\ 4 & 1\end{matrix}$$

Determinantu rešavamo primenom Sarusovog pravila, koje važi isključivo za determinante reda 3x3.

$$detA = 1\cdot 3\cdot 1 + 1\cdot 1\cdot 4 + 2\cdot 1\cdot 1 - 2\cdot 3\cdot 4 - 1\cdot 1\cdot 1 - 1\cdot 1\cdot 1 = -17 \neq 0$$

$$A = \begin{vmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{vmatrix} \qquad \text{šema znakova } \begin{vmatrix}+ & - & +\\ - & + & -\\ + & - & +\end{vmatrix}$$

$$\text{Determinanta } adjA = \begin{vmatrix}A_{11} & A_{12} & A_{13}\\ A_{21} & A_{22} & A_{23}\\ A_{31} & A_{32} & A_{33}\end{vmatrix}^T = \begin{vmatrix}A_{11} & A_{21} & A_{31}\\ A_{12} & A_{22} & A_{32}\\ A_{13} & A_{23} & A_{33}\end{vmatrix}$$

$A_{11} = (-1)^{1+1}\cdot\begin{vmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{vmatrix} = \begin{vmatrix}3 & 1\\ 1 & 1\end{vmatrix} = 2$ $A_{21} = (-1)^{2+1}\cdot\begin{vmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{vmatrix} = -\begin{vmatrix}1 & 2\\ 1 & 1\end{vmatrix} = 1$ $A_{31} = (-1)^{3+1}\cdot\begin{vmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{vmatrix} = \begin{vmatrix}1 & 2\\ 3 & 1\end{vmatrix} = -5$

$A_{12} = (-1)^{1+2}\cdot\begin{vmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{vmatrix} = -\begin{vmatrix}1 & 1\\ 4 & 1\end{vmatrix} = 3$ $A_{22} = (-1)^{2+2}\cdot\begin{vmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{vmatrix} = \begin{vmatrix}1 & 2\\ 4 & 1\end{vmatrix} = -7$ $A_{32} = (-1)^{3+2}\cdot\begin{vmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{vmatrix} = -\begin{vmatrix}1 & 2\\ 1 & 1\end{vmatrix} = 1$

$A_{13} = (-1)^{1+3}\cdot\begin{vmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{vmatrix} = \begin{vmatrix}1 & 3\\ 4 & 1\end{vmatrix} = -11$ $A_{23} = (-1)^{2+3}\cdot\begin{vmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{vmatrix} = -\begin{vmatrix}1 & 1\\ 4 & 1\end{vmatrix} = 3$ $A_{33} = (-1)^{3+3}\cdot\begin{vmatrix}1 & 1 & 2\\ 1 & 3 & 1\\ 4 & 1 & 1\end{vmatrix} = \begin{vmatrix}1 & 1\\ 1 & 3\end{vmatrix} = 2$

$$adjA = \begin{bmatrix}2 & 1 & -5\\ 3 & -7 & 1\\ -11 & 3 & 2\end{bmatrix} \qquad A^{-1} = \frac{1}{-17}\cdot\begin{bmatrix}2 & 1 & -5\\ 3 & -7 & 1\\ -11 & 3 & 2\end{bmatrix}$$

Kreiranje NumPy matrica

Matrice dimenzija m*x*n mogu da se kreiraju pomoću biblioteke Numpy na više načina, a što će biti prikazano u narednim primerima. Najčešće se to radi uz pomoć naredbe *array*. Za više detalja pogledati [3]. Ovo je već prikazano kroz *primer 2.* kada su prikazane dvodimenzinalne liste *L2* i *L3*, a sledi još jedan primer. Treba obratiti pažnju da se matrice definišu pomoću duplih uglastih zagrada, odnosno [[,],[,], … , [,]], a pri tom se mora voditi računa da su iste dužine liste, odnosno o dimenziji matrice. Slede primeri u kojima će biti prikazani načini kreiranja matrica.

Primer 13: Kreiranje matrice korišćenjem naredbe array

Input:

```
#Kreiranje matrica pomocu Numpy biblioteke
import numpy as np
M1 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) # Kreiranje matrice koriscenjem naredbe array
print(M1)
```

Output:

```
In [1]:
   ...: import numpy as np
   ...: M1 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) # Kreiranje matrice
koriscenjem naredbe array
   ...: print(M1)
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

Takođe matrica može da se napravi od jednodimenzionalnog niza elemenata, pomoću naredbe *reshape*.

Kada se koristi ova naredba, mora se obratiti pažnja na dimenzije matrice.

U narednom primeru biće kreiran jednodimenzionalni niz brojeva od 1 do 9, ali uz pomoć naredbe numpy naredbe arange, koja je prikazana u *primeru 5*. Korišćenjem date naredbe dobiće se matrica dimenzije 3x3.

Primer 14: Kreiranje matrice korišćenjem naredbe reshape

Input:

```
M2 = np.arange(1,10).reshape(3,3) # Kreiranje matrice korscenjem naredbe reshape
print(M2)
```

Output:

```
In [2]: M2 = np.arange(1,10).reshape(3,3) # Kreiranje matrice korscenjem naredbe
reshape
   ...: print(M2)
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

Matrica može da nastane više listi, ili jedne iste, ali moraju biti iste dužine, a što će biti prikazano u naredna dva primera, korišćenjem prethodno definisanih lista iz *primera 4*.

Primer 15: Kreiranje matrice spajanjem dve iste liste *L1* iz *primera 4.*

Input:

```
M3 = np.array([L1,L1]) #Kreiranje matrice, spajanjem dve iste NumPy liste
print(M3)
```

Output:

```
In [2]: M3 = np.array([L1,L1]) #Kreiranje matrice, spajanjem dve iste
NumPy liste
   ...: print(M3)
[[1 1]
 [1 1]]
```

Primer 16: Kreiranje matrice spajanjem dve različite Python liste

Input:

```
P1 = [1, 2.5, 3]
P2 = [4, 5, 6]
M4 = np.array([P1,P2]) #Kreiranje matrice, spajanjem dve različite Python liste
print(M4)
```

Output:

```
In [3]: P1 = [1, 2.5, 3]
   ...: P2 = [4, 5, 6]
   ...: M4 = np.array([P1,P2]) #Kreiranje matrice, spajanjem dve različite Python liste
   ...: print(M4)
[[1.  2.5 3. ]
 [4.  5.  6. ]]
```

Postoji mogućnost da se kreira matrica i pomoću naredbe *matrix*. Za više detalja pogledati [4].

Primer 17: Kreiranje matrice korišćenjem naredbe matrix

Input:

```
M5 = np.matrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) # Kreiranje matrice koriscenjem naredbe matrix
print(M5)
```

Output:

```
In [5]: M5 = np.matrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) # Kreiranje matrice koriscenjem naredbe matrix
   ...: print(M5)
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

Ako spojimo dve liste *L1* iz *primera 4.* na sledeći način, dobićemo listu, koja liči na matricu, ali nije, nego je dupla ista lista.

Primer 18: Spajanje dve liste u jednu listu

Input:

```
#Formiramo novu listu,ne matricu, pomocu dve liste L1
import numpy as np
L1 = np.array([1,1])# Dobija se dupla ista lista u okviru liste
DuplaLista=[L1,L1]
print(DuplaLista)
```

Output:

```
In [2]:
   ...: import numpy as np
   ...: L1 = np.array([1,1])# Dobija se dupla ista lista u okviru liste
   ...: DuplaLista=[L1,L1]
   ...: print(DuplaLista)
[array([1, 1]), array([1, 1])]
```

Ako prvom elementu promenimo vrednost na nula, vrednost nula ćemo imati u obe liste, jer je ista lista, a što se vidi iz narednog primera.

Primer 19: Dodeljivanje prvom elementu liste vrednost nula

Input:

```
#Ako prvom elementu, promenimo vrednost na nulu, promenice se u obe liste
DuplaLista[0][0]=0
print(DuplaLista)
```

Output:

```
In [3]:
   ...: DuplaLista[0][0]=0
   ...: print(DuplaLista)
[array([0, 1]), array([0, 1])]
```

Ako istu dodelu vrednosti nula primenimo na matricu M3, iz primera 15. Koja se isto sastoji iz duple liste L1, dobija se vrednost nula na prvom elementu matrice, jer je to matrica, a ne lista. Upravo na ovaj primer treba obratiti pažnju. Sledi primer koji prikazuje promenu vrednosti matrice.

Primer 20: Promena vrednosti prvog elementa matrice na nulu

Input:

```
#Neka je data matrica sastavljena od dve spojene liste
#Ako se zada ista promena vrednosti na nulu, tada se samo prvi elemenat matrice manja na nulu
M3[0][0]=0
print(M3)
```

Output:

```
In [4]:
   ...: M3[0][0]=0
   ...: print(M3)
[[0 1]
 [1 1]]
```

Operacije sa matricama korišćenjem NumPy biblioteke

Za prikaz ovih operacije biće korišćeni primeri koji su prethodno objašnjeni u okviru poglavlja Operacije sa matricama. Za početak sledi primer sabiranja matrica, korišćenjem matrica iz *primera 6*.

Primer 21: Sabiranje matrica

Input:

```
#Sabiranje matrica
import numpy as np
A = np.array([[1, 3, 1], [1, 5, 4], [1, 2, 4]])
B = np.array([[1, 2, 1], [-1, 4, 8], [-2, -1, 4]])
C = A + B      # pri sabiranju matrica se koristi plus
print(C)
```

Output:

```
In [1]:
   ...: import numpy as np
   ...:
   ...: A = np.array([[1, 3, 1], [1, 5, 4], [1, 2, 4]])
   ...: B = np.array([[1, 2, 1], [-1, 4, 8], [-2, -1, 4]])
   ...: C = A + B      # pri sabiranju matrica se koristi plus
   ...: print(C)
[[ 2  5  2]
 [ 0  9 12]
 [-1  1  8]]
```

Iz prethodnog primera može se uočiti da se rezultati podudaraju, sa primerom 6.

Sledi primer množenja matrice sa skalarom. Za ovo će biti korišćen primer 7. iz istog poglavlja. Kada se matrica množi sa skalarom, koristi se klasična operacija množenja(*).

Primer 22: Množenje matrice skalarom

Input:

```
# Mnozenje matrice skalarom
D = 2*A
print(D)
```

Output:

```
In [3]:
   ...: D = 2*A
   ...: print(D)
[[ 2  6  2]
 [ 2 10  8]
 [ 2  4  8]]
```

Kada se množe dve matrice, tada *ne može* da se koristi klasično množenje, nego se koristi Numpy funkcija *dot*. Sledi primer množenja matrica.

Primer 23: Množenje dve matrice

Input:

```
#Množenje matrica
import numpy as np
A = np.array([[1, 3, 1], [1, 5, 4], [1, 2, 4]])
B = np.array([[1, 2], [-1, 4], [-2, -1]])
C = A.dot(B) # mnozenje matrica sa naredbom dot
print(C)
```

Output:

```
In [4]:
   ...: import numpy as np
   ...: A = np.array([[1, 3, 1], [1, 5, 4], [1, 2, 4]])
   ...: B = np.array([[1, 2], [-1, 4], [-2, -1]])
   ...: C = A.dot(B) # mnozenje matrica sa naredbom dot
   ...: print(C)
[[ -4  13]
 [-12  18]
 [ -9   6]]
```

Rešavanje sistema linearnih jednačina uz korišćenje ChatGPT

U okviru ovog poglavlja biće prikazan Python kod za rešavanje sistema lineranih jednačina *matričnom* metodom, ali uz upotrebu *NumPy* bibliotke. Kod je izgenerisan primenom veštačke inteligencije ***ChatGPT***. Za više detalja pogledati [6]. Biće prikazan kompletan matematički postupak rešavanja narednog primera, radi boljeg razumevanja zadatka. Dobijeni rezultati su ***identični***.

Primer 24: Rešiti sistem jednačina matričnom metodom.

$$x + 2y + 2z = 2$$

$$x - 2y - 3z = 0$$

$$x - y - 2z = 1$$

Rešenje:

$X = A^{-1} \cdot B$

$A^{-1} = \frac{1}{detA} \cdot adjA \qquad B = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \qquad X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

$$detA = \begin{vmatrix} 1 & 2 & 2 \\ 1 & -2 & -3 \\ 1 & -1 & -2 \end{vmatrix} \begin{matrix} 1 & 2 \\ 1 & -2 \\ 1 & -1 \end{matrix}$$

$detA = (1 \cdot (-2) \cdot (-2) + 2 \cdot (-3) \cdot 1 + 2 \cdot 1 \cdot (-1)) - (1 \cdot (-2) \cdot 2 + (-1) \cdot (-3) \cdot 1 + (-2) \cdot 1 \cdot 2)$

$detA = (4 - 6 - 2) - \big((-4) + 3 + (-4)\big) = -4 + 5 = 1$

$detA = 1$

$$A = \begin{vmatrix} 1 & 2 & 2 \\ 1 & -2 & -3 \\ 1 & -1 & -2 \end{vmatrix} \qquad \text{šema znakova} \begin{vmatrix} + & - & + \\ - & + & - \\ + & - & + \end{vmatrix}$$

Determinantu $adjA = \begin{vmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{vmatrix}^T = \begin{vmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{vmatrix}$

$A_{11} = \begin{vmatrix} -2 & -3 \\ -1 & -2 \end{vmatrix} = 1$ $\quad A_{21} = -\begin{vmatrix} 2 & 2 \\ -1 & -2 \end{vmatrix} = 2$ $\quad A_{31} = \begin{vmatrix} 2 & 2 \\ -2 & -3 \end{vmatrix} = -2$

$A_{12} = -\begin{vmatrix} 1 & -3 \\ 1 & -2 \end{vmatrix} = -1$ $\quad A_{22} = \begin{vmatrix} 1 & 2 \\ 1 & -2 \end{vmatrix} = -4$ $\quad A_{32} = -\begin{vmatrix} 1 & 2 \\ 1 & -3 \end{vmatrix} = 5$

$A_{13} = \begin{vmatrix} 1 & -2 \\ 1 & -1 \end{vmatrix} = 1$ $\quad A_{23} = -\begin{vmatrix} 1 & 2 \\ 1 & -1 \end{vmatrix} = 3$ $\quad A_{33} = \begin{vmatrix} 1 & 2 \\ 1 & -2 \end{vmatrix} = -4$

$$adjA = \begin{bmatrix} 1 & 2 & -2 \\ -1 & -4 & 5 \\ 1 & 3 & -4 \end{bmatrix}, \; X = A^{-1} \cdot B = \frac{1}{detA} \cdot adjA \cdot B$$

$$X = 1 \cdot \begin{bmatrix} 1 & 2 & -2 \\ -1 & -4 & 5 \\ 1 & 3 & -4 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} = 1 \cdot \begin{bmatrix} 2+0-2 \\ -2+0+5 \\ 2+0-4 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ -2 \end{bmatrix}$$

Rešenje je $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = (\mathbf{0}, \mathbf{3}, \mathbf{-2})$**.**

U narednom primeru prikazan je kod koji je izgenerisao ChatGpt.

Primer 25: Python kod izgenerisan od strane ChatGPT

Input:

```
#Kod generisan od strane ChaGpt
import numpy as np

# Define the coefficient matrix and the constant matrix
A = np.array([[1, 2, 2],
              [1, -2, -3],
              [1, -1, -2]])
b = np.array([2, 0, 1])

# Find the inverse of A
A_inv = np.linalg.inv(A)

# Find the solution vector
x = np.dot(A_inv, b)

# Print the solution vector
print(f"x = {x[0]}")
print(f"y = {x[1]}")
print(f"z = {x[2]}")
```

Output:

```
x = 0.0
y = 3.0
z = -2.0
```

Reference

[1] https://www.programiz.com/python-programming/matrix

[2] https://www.w3schools.com/python/numpy/numpy_intro.asp

[3] https://numpy.org/doc/stable/reference/generated/numpy.array.html#numpy.array

[4] https://numpy.org/doc/stable/reference/generated/numpy.matrix.html

[5] https://www.digitalocean.com/community/tutorials/concatenate-lists-python

[6] https://openai.com/product/gpt-4