
Datu kārtošana

Referāts

Eva Mārtiņšone

24.02.2021.



Contents

Darba mērķis	2
Izmantotie moduļi	2
CSV faila imports	3
Datu indeksēšana	3
Izmantotās funkcijas:	3
Datu indeksēšana pēc tabulas kolonas ‘Organizācija’	3
Datu grupēšana no indeksētās kolonas ‘Organizācija’	3
Atlasīto datu tālākā indeksēšana pēc kolonas ‘Darba grupai’	3
Datu grupēšana pēc indeksētajiem datiem, kas nepieciešams e-pastu saturam	4
Epastsu izsūtīšana	4
SMTP savienojuma izveidi	4
E-pasta pamatelementi	4
Pieslēguma izveide	4
Izsūtītāja e-pasta rekvizīti	4
Pieslēgums e-pasta serverim	5
E-pasta saturs	5
E-pasta izsūtīšanu	5

Darba mērķis

Veikt datu kārtošanu pēc vairākiem atlases kritēriem. Iegūtās datu kopas nosūtītas atbildīgajiem uz e-pastiem

Izmantotie moduļi

pandas.DataFrame modulis izmantots ar mērķi veikt darbības ar datiem tabulā gan pēc rindu, gan pēc kolonnu etiķetēm smtpplib modulis izmatots ar mērķi izsūtīt e-pastus

CSV faila imports

`pandas.read_csv` - nodrošina csv faila nolasīšanu, kurā ir atdalītas vērtības `r` - nodrošina nepabeigtu vērtību atdalīšanu (fails) - importētā faila atrašanās vieta

Izpildītais kods:

```
1 saraksts = pd.read_csv (r'C:\Users\Lietotajs\Desktop\ievades_dati.csv')
2 print(saraksts)
```

Datu indeksēšana

Izmantotās funkcijas:

`.index` - nodrošina katras tabulas kolonas vai rindas indeksēšanu kā atevišķu vienumu `.loc` - nodrošina indeksēto tabulas kolonu vai rindu vienumu atlasi

Datu indeksēšana pēc tabulas kolonas ‘Organizācija’

Nodrošina visu organizāciju pieteikumu indeksēšanu Izpildītais kods:

```
1 pak_kolonas = saraksts.set_index('Organizacija')
2 print(pak_kolonas)
```

Datu grupēšana no indeksētās kolonas ‘Organizācija’

Nodrošina konkrētu organizāciju pieteikumu atlasi Izpildītais kods:

```
1 iestade = pak_kolonas.loc[['Labklājības departaments', 'Rīgas Sociālais
    dienests', 'Rīgas pasvaldības policija', 'Sociālais dienests']]
2 print(iestade)
3 print(pak_kolonas)
```

Atlasīto datu tālākā indeksēšana pēc kolonas ‘Darba grupai’

Nodrošina konkrētu orgaizāciju pieteikumu indeksēšanu Izpildītais kods:

```
1 pak_vaditaji = iestade.set_index('Darba grupai')
2 print(pak_vaditaj)
```

Datu grupēšana pēc indeksētajiem datiem, kas nepieciešams e-pastu saturam

Nodrošina konkrētu orgaizāciju pieteikumu grupēšanu pēc to atbildīgajiem Izpildītais kods:

```
1 Normunds = pak_vaditaji.loc[['Datu bazu un sistemu nodala']]
2 print('DBSN pieteikumi', Normunds)
3
4 Mihails = pak_vaditaji.loc[['Tehnisko resursu uzturesanas sektors']]
5 print('TRU sektota pieteikumi', Mihails)
6
7 Jorens = pak_vaditaji.loc[['Fiziskas drošības un integroto sistemu
   sektors']]
8 print('FDIS sektota pieteikumi', Jorens)
```

Epasts izsūtīšana

Par katru datu kopu, kas iegūta iepriekš aprakstīto datu indeksēšanā un grupēšanā pēc organizāciju pieteikumiem un atbildīgajiem, tiek sagatavots e-pasts pēc turpmāk esošā parauga

SMTP savienojuma izveidi

```
1 import smtplib
```

E-pasta pamatelementi

```
1 To = "mmmartinsone@gmail.com"
2 Subject = "neizpildītie uzdevumi"
3 Text = 'tetsa e-pasts'
```

Pieslēguma izveide

```
1 smtp_server = "smtp.gmail.com"
2 port = 587
```

Izsūtītāja e-pasta rekvizīti

```
1 e_pasta_sender = "mmmartinsone@gmail.com"
2 e_pasta_parole = "parole"
```

Pieslegums e-pasta serverim

```
1 server = smtplib.SMTP('smtp.gmail.com', 587)
2 server.ehlo()
3 server.starttls()
4 server.ehlo()
5 server.login(e_pasta_sender, e_pasta_parole)
```

E-pasta saturs

```
1 Text = '\r\n'.join([
2     'To: %s' % To,
3     'From: %s' % e_pasta_sender,
4     'Subject: %s' % Subject,
5     '',
6     Normunds
7 ])
```

E-pasta izsūtīšanu

```
1 try:
2     server.sendmail(e_pasta_sender, [To], saturs)
3     print('epasts nosutits')
4 except:
5     print('e_pasts nav nosutits')
6 server.quit()
```