

# ინფორმაციული საზოგადოება, მონაცემთა მენეჯმენტის ახალი ტექნოლოგიები და ექსტრემალური სიტუაციების მართვის სისტემები

გოჩა ჩოგოვაძე, არჩილ ფრანგიშვილი,

გიორგი კვიციანი, გია სურგულაძე, გულბათ ნარეშელაშვილი

საქართველოს ტექნიკური უნივერსიტეტი

## რეზიუმე

ინფორმაციული საზოგადოების ფორმირების მიზნით ნაშრომში განხილულია მონაცემთა მენეჯმენტის ახალი ტექნოლოგიების ინტეგრაციის საკითხები მასობრივი მომსახურების საინფორმაციო სისტემებთან. შემოთავაზებულია ეკოსისტემების და დიდ მონაცემთა ანალიზის სისტემების გამოყენების კონცეფცია, აგრეთვე და მანქანური დასწავლის ალგორითმების შემუშავების და პროგრამული რეალიზაციის ასპექტები ჰიბრიდული დაპროგრამების ტექნოლოგიების საფუძველზე Python ენაზე. ექსპერიმენტული ნაწილი წარმოდგენილია ექსტრემალურ სიტუაციებში გადაწყვეტილების მიღების სისტემის მხარდამჭერი ინფორმაციული და პროგრამული უზრუნველყოფის პაკეტებით. კერძოდ, გადაწყვეტილია სასწრაფო დახმარების ბრიგადების ოპერატიული მართვის ამოცანისათვის ინფორმაციის შეგროვებისა და მონაცემთა საცავის ფორმირების ამოცანა, შემდგომში მათი დამუშავებისათვის მანქანური დასწავლის ალგორითმებით.

**საკვანძო სიტყვები:** ინფორმაციული საზოგადოება. მასობრივი მომსახურების სისტემა. Hadoop. BigData. Machine Learning. გადაუდებელი დახმარება. „112“. პროგრამირება. Python.

## 1. შესავალი

საქართველოში „ინფორმაციული საზოგადოების“ ჩამოყალიბებისა და მისი შემდგომი მდგრადი განვითარების კონცეფციის რეალიზაციის ხელშეწყობა მნიშვნელოვანი ფაქტორია ჩვენი ქვეყნისა და ხალხის კეთილდღეობის დონის ასამაღლებლად [1]. საზოგადოების ინფორმატიზაციის ხარისხის დონის ამაღლება ახალი საინფორმაციო ტექნოლოგიების, ინფორმატიკის დიდაქტიკისა და მედიაინფორმატიკის მეცნიერებათა კომპლექსური გამოყენების ბაზაზე.

ამ მიმართულებით საქართველოს ტექნიკური უნივერსიტეტის „მართვის ავტომატიზებული სისტემების (პროგრამული ინჟინერიის)“ დეპარტამენტში სტუ-ს UNESCO-ს კათედრის ხელმძღვანელობით მიმდინარეობს გარკვეული სამუშაოები სტუდენტთა განათლებისა და მეცნიერული კვლევების სფეროში. შემუშავებულია ახალი სასწავლო პროგრამები, შესრულებულია არაერთი დისერტაცია ამ მიმართულებით, გამოქვეყნებულია მრავალი სახელმძღვანელო და მონოგრაფია „ინფორმაციული საზოგადოების“ განახლებულ ძირითად დისციპლინებში, როგორებიცაა საინფორმაციო და კომუნიკაციური სისტემები, დიდ მონაცემთა ანალიზის- და ეკო-სისტემები და სხვ. [2,3]. ამ საკითხების ერთ კონკრეტულ დეტალურ განხილვას შევხებით ნაშრომის შემდეგ ნაწილში, რომელიც საზოგადოების ინფორმატიზაციის სფეროში, მასობრივი მომსახურების

სისტემების სახით, ახალი ტექნოლოგიების გამოყენებას ეხება ექსტრემალური სიტუაციების, კერძოდ გადაუდებელი სასწრაფო დახმარების ოპერატიული მართვის (დისპეტჩერიზაციის) სისტემის ახალი კონცეფციის ჩამოყალიბებითა და მისი რეალიზაციით, რომელიც „112“-ში დაინერგება.

### 1.1. დისპეტჩერიზაციის კომპიუტერული სისტემა ბრიგადების შესარჩევად

სასწრაფოს ბრიგადის გათავისუფლებისა და ახალ მისამართზე მისვლის სავარაუდო დროის არცოდნის გამო დისპეტჩერებს ხშირად უწევთ არაოპტიმალური გადაწყვეტილებების მიღება.

მაგალითად, ზარის ინიციატორთან ტერიტორიულად უახლოესი თავისუფალი ბრიგადა საცობის გათვალისწინებით 1 საათის სავალზეა, სხვა საქმეზე გასული ბრიგადა კი შეიძლება ინიციატორის მეზობლად იყოს და 5 წუთში გათავისუფლდეს. არაოპტიმალური მენეჯმენტით დროც ბევრად მეტი იხარჯება და ფინანსური თუ ადამიანური რესურსიც.

პროექტის მიზანია ბრიგადების განაწილების გაუმჯობესება. ანუ, გამომახების საპასუხოდ ისეთი ბრიგადის არჩევა, რომელსაც შეუძლია, ყველაზე სწრაფად მივიდეს შემთხვევის ადგილზე.

ბრიგადის შერჩევის პროცესი იწყება პაციენტის შესაძლო ჰოსპიტალიზაციის ალბათობის გამოთვლით. თუ ჰოსპიტალიზაციის ალბათობა მაღალია ( $P \geq 0.5$ ) მოცემულ შემთხვევაზე გაგზავნილი სასწრაფო დახმარების მანქანა აღარ განიხილება როგორც თავისუფალი დროის უახლოეს ინტერვალში.

თუ შემთხვევის ჰოსპიტალიზაციის ალბათობა დაბალია ( $P < 0.5$ ) ალგორითმი შეაფასებს სასწრაფო დახმარების მანქანის შესაძლო დაკავებულობის დროს და მიაწვდის ამ ინფორმაციას დისპეტჩერს. ამ დროის შუალედის გავლის შემდეგ, სავარაუდოა, რომ ბრიგადა განთავისუფლდება.

უნდა აღინიშნოს, რომ ალგორითმი იძლევა მხოლოდ ალბათობის სიდიდეებს და არა დროის ზუსტ შუალედებს. მიღებული რეკომენდაცია დისპეტჩერმა უნდა გადაამოწმოს უშუალოდ ბრიგადის ექიმთან.

პაციენტის ჰოსპიტალიზაციის ალბათობა, ისევე, როგორც მიახლოებითი დრო, რასაც ბრიგადა ხარჯავს პაციენტზე, გამოითვლება ისეთი მკაცრად განსაზღვრული მონაცემების საფუძველზე, როგორცაა მაგალითად, ინციდენტის ტიპი, პაციენტის ასაკი და სქესი, ბრიგადის ინფორმირების, რეაგირების, ადგილზე მისვლისა და გათავისუფლების დროები.

პაციენტთან დახარჯული ალბათური დროის გამოთვლის გარდა, მნიშვნელოვანი იქნება ბრიგადის გადაადგილების სავარაუდო დროც, რასაც მარშრუტიზაციის სერვისებით მივიღებთ. არსებული სერვისები მოგვცემს საშუალებას, გავარკვიოთ, გზებზე არსებული „საცობების“ გათვალისწინებით, რა დრო იქნება საჭირო ერთი პუნქტიდან მეორემდე გადასაადგილებლად (მაგალითად Google traffic) [4,5].

## 1.2. მონაცემთა სიმრავლე (Dataset)

Dataset-ის შემუშავებისას გამოვყავით ორი ეტაპი. პირველ რიგში, SQL-ის საშუალებით „112“-ის მთავარი მონაცემთა ბაზიდან ამოვიღეთ 5 წლის მონაცემები და სასურველი ფორმატით შევინახეთ მონაცემთა საცავში. მეორე ეტაპზე Python-ის საშუალებით დავამუშავეთ მონაცემები, ამოვშალეთ ანომალიები და გავამზადეთ მანქანური დასწავლის ალგორითმებთან სამუშაოდ.

## 2, ძირითადი ნაწილი

### 2.1. მონაცემების შეგროვება (Data-gathering)

რადგან მიმდინარე პროექტი მჭიდრო კავშირშია გადაუდებელი დახმარების ერთიანი ცენტრის კონფიდენციალურ ინფორმაციასთან, თავს შევიკავებ მონაცემთა ბაზასთან მუშაობის დეტალების აღწერისგან. ქვემოთ მოყვანილია პირველი ეტაპის შედეგი:

- CASE\_ID – საქმის უნიკალური კოდი (საჭიროა მხოლოდ მონაცემთა შეგროვების ეტაპზე);
- CALL\_ID – ზარის უნიკალური კოდი (საჭიროა მხოლოდ მონაცემების შეგროვების ეტაპზე);
- NOM – ინიციატორის ნომერი (საჭიროა მხოლოდ მონაცემების შეგროვების ეტაპზე);
- DATE\_CREATED – ზარის შემოსვლის თარიღი;
- CASE\_TEMPLATE\_ID – ინციდენტის ტიპი (გული, კრუნჩხვა, სუნთქვის გაძნელება...)
- INJURED\_PERSON\_AGE\_YEARS,  
INJURED\_PERSON\_AGE\_MONTHS,  
INJURED\_PERSON\_AGE\_DAYS,  
პაციენტის ასაკი ივსება წლების, თვეების ან დღეების სახით (ნახ.1).

ნახ.1. პაციენტის ასაკი

ახალდაბადებული ან ჩვილი ბავშვის შემთხვევაში ოპერატორი ავსებს დღეების ველს, წლამდე ასაკის ბავშვებისთვის – თვეების ველს, დანარჩენი პაციენტების ასაკი კი აღირიცხება წლების სახით [6].

- INJURED\_PERSON\_GENDER - სქესი;
- CATEGORIES – ინფორმაცია ყველა იმ გადაუდებელი დახმარების სერვისზე, რაც ჩართული იყო ინციდენტის მართვაში (სასწრაფო, სახანძრო, სამაშველო, პოლიცია...);
- SPENT\_MINS – სასწრაფოს ბრიგადის მიერ პაციენტთან, ბინაზე გატარებული დრო (ჰოსპიტალიზაციის შემთხვევაში ემატება პაციენტის ჰოსპიტალიზაციაზე დახარჯული დროც);
- PREV\_HOSP\_COUNT – ამავე ნომერზე იგივე ინციდენტის ტიპის მიხედვით ჰოსპიტალიზაციების რაოდენობა ბოლო 1 წლის განმავლობაში;
- LASTHOSPDD – ბოლო ჰოსპიტალიზაციიდან გასული დღეების რაოდენობა (ბოლო 1 წელი);

• CALL\_COUNT – იმავე ნომერზე ბოლო 1 თვის განმავლობაში შექმნილი საქმეების რაოდენობა.

## 2.2. მონაცემების წინასწარი დამუშავება (Data preprocessing)

მონაცემების წინასწარი დამუშავება და ანალიზისთვის მომზადება ხშირად ყველაზე მნიშვნელოვანი პროცესია მანქანური დასწავლის დროს [7]. მონაცემების პირველადი შეგროვების ეტაპზე ხშირად, მინიმალურად გვაქვს კონტროლის საშუალება. შესაბამისად, ხშირია ისეთი ანომალიური მონაცემები, როგორცაა:

- სპექტრის გარე მნიშვნელობები – მაგალითად, დროის ან ანაზღაურების უარყოფითი მნიშვნელობები;
- მონაცემთა შეუძლებელი კომბინაციები;
- შეუვსებელი მონაცემები;
- დასწავლისთვის გამოუსადეგარი მონაცემები – მაგალითად, უსიმბარათო ტელეფონებიდან განხორციელებული ზარები.

ამ და სხვა მრავალი ანომალიური შემთხვევის გაფილტვრის გარეშე მონაცემების გაანალიზებამ შეიძლება მცდარ შედეგებამდე მიგვიყვანოს.

ასე რომ, მონაცემების წინასწარი დამუშავება მანქანური დასწავლის პროექტის უპირველესი და ხშირად, უმთავრესი ეტაპია [8].

მონაცემების დამუშავებისა და ანალიზისთვის ვიყენებთ ჰიბრიდული, მრავალ-პარადიგმული დაპროგრამების Python ენას [9,10].

თავდაპირველად, ვშლით "ტრიაჟის" ტიპის საქმეებს (საქმეებს, სადაც ერთდროულად სასწრაფო დახმარების რამდენიმე ბრიგადა გავიდა):

```
# keep=False პარამეტრი უზრუნველყოფს დუბლირებული ჩანაწერების სრულად წაშლას. df_orig = df_orig.drop_duplicates(subset=['CALL_ID'], keep=False)
```

მნიშვნელოვანია, რომ შემდეგ სვეტებში არ გვქონდეს შეუვსებელი მონაცემები:

```
# dropna მეთოდი how='any' პარამეტრის მითითებით შლის ყველა იმ სტრიქონს, სადაც აღნიშნული სვეტებიდან ერთ-ერთი მაინც არის მონაცემის გარეშე. df_orig.dropna(subset=['CASE_ID', 'CALL_ID', 'DATE_CREATED', 'CASE_TEMPLATE_ID', 'NOM', 'CATEGORIES', 'SPENT_MINS'], how='any', inplace=True)
```

შემდგომ დამუშავებამდე, ვშლით ისეთ მონაცემებს, რომლებსაც სხვადასხვა მიზეზების გამო, ვერ გამოვიყენებთ დასწავლის პროცესისთვის:

```
# ვშლით არასტანდარტული სიგრძის ნომრებს
df = df[df['NOM'].map(len) <= 9]
# ვფილტრავთ მობილურისა და ქალაქის ნომრებს (მხოლოდ თბილისი)
pattern='5\d\d\d\d\d\d\d\d\d\d|2\d\d\d\d\d\d\d|322\d\d\d\d\d\d|79[01]\d\d\d\d\d\d'
# ვფილტრავთ მობილურისა და ქალაქის ნომრებს წინასწარ გამოცხადებული პატერნის მიხედვით
df = df[df['NOM'].str.match(pattern)]
# ვშლით სხვადასხვა მიზეზის გამო არსებულ ანომალიებს
df = df[df['SPENT_MINS'] <= 120]
df = df[df['SPENT_MINS'] >= 0]
```

მონაცემების წინასწარი დამუშავების შედეგად ვიღებთ შემდგომი დასწავლისთვის გამზადებულ 2.7 მილიონი ჩანაწერისგან შემდგარ მონაცემთა ერთობლიობას 219 სვეტის სახით:

- DATE\_CREATED – ზარის შემოსვლის თარიღს ჩავუტარეთ კატეგორიზაცია სსიპ „112“-სთვის სპეციფიური დროითი შუალედების მიხედვით. კატეგორიზაციისთვის გამოვიყენეთ One Hot Encoding მეთოდი, რაც ნიშნავს, რომ ყოველი სტრიქონისათვის შევსებულია მხოლოდ ერთერთი [6]: [ TIME\_02\_08, TIME\_08\_12, TIME\_12\_17, TIME\_17\_23, TIME\_23\_02 ].

ამ ეტაპზე მხოლოდ საათებს ვაქცევთ ყურადღებას, თუმცა, სამომავლოდ, შესაძლებელი იქნება სეზონების გამოყოფაც.

```
# დროითი დაჯგუფება "112"-ის სპეციფიკის მიხედვით [HH_HH)
```

```
df["TIME_02_08"] = 0
```

```
df["TIME_08_12"] = 0
```

```
df["TIME_12_17"] = 0
```

```
df["TIME_17_23"] = 0
```

```
df["TIME_23_02"] = 0
```

```
for i, row in df.iterrows():
```

```
    if 2 <= int(row["DATE_CREATED"][11:13]) < 8 :
```

```
        df.loc[i, "TIME_02_08"] = 1
```

```
    elif 8 <= int(row["DATE_CREATED"][11:13]) < 12 :
```

```
        df.loc[i, "TIME_08_12"] = 1
```

```
    elif 12 <= int(row["DATE_CREATED"][11:13]) < 17 :
```

```
        df.loc[i, "TIME_12_17"] = 1
```

```
    elif 17 <= int(row["DATE_CREATED"][11:13]) < 23 :
```

```
        df.loc[i, "TIME_17_23"] = 1
```

```
    elif int(row["DATE_CREATED"][11:13]) == 23 :
```

```
        df.loc[i, "TIME_23_02"] = 1
```

```
    elif 0 <= int(row["DATE_CREATED"][11:13]) < 2 :
```

```
        df.loc[i, "TIME_23_02"] = 1
```

- CASE\_TEMPLATE\_ID – ინციდენტის ტიპები გადმოტანილია One Hot Encoding მეთოდით. მივიღეთ 194 უნიკალური სვეტი, სადაც ყველა შემთხვევისთვის გვაქვს 193 ცალი '0' და მხოლოდ ერთი '1'.

```
#CASE_TEMPLATE_ID სვეტის კატეგორიზაცია
```

```
df_with_CTID=pd.get_dummies(df_orig, columns=["CASE_TEMPLATE_ID"], prefix='CTID')
```

- PATIENT\_AGE – ასაკის დასამუშავებლად მიზანშეწონილია ორი მეთოდის გამოყენება, რადგან წინასწარ არ ჩანს, თუ რომელი მეთოდი მოგვცემს უკეთეს შედეგს:

- წავშალეთ ანომალიები, ჩავატარეთ ასაკის კატეგორიზაცია One Hot Encoding მეთოდით, კატეგორიებად კი ავარჩიეთ შემდეგი გარდამტეხი ასაკები: 0 - 40 დღე, 1, 6, 15, 60 წლები.

- მივდივართ მეორე მიმართულებითაც,- არ ვატარებთ ასაკის კატეგორიზაციას, ვინახავთ ასაკს დღეების სახით და ვუკეთებთ ნორმალიზებას შემდეგი მეთოდით:  $X_n = (X - X_{min}) / (X_{max} - X_{min})$ , (1) სადაც  $X_n$  პაციენტის ასაკის ნორმალიზებულ მნიშვნელობას აღნიშნავს.

• INJURED\_PERSON\_GENDER – შეცვალეთ NaN (არ არსებული) მნიშვნელობები საშუალოთი.

#გაურკვეველი სქესის შემთხვევაში ვიყენებთ საშუალოს

```
df['INJURED_PERSON_GENDER'].fillna(df['INJURED_PERSON_GENDER'].mean(), inplace=True)
```

• CATEGORIES – განვაცალკევებთ ერთად შენახული კატეგორიები და ჩავატარებთ კატეგორიზაციას One Hot Encoding მეთოდით. 8 განსხვავებული კატეგორიიდან დადებითადაა შევსებული ყველა ის სვეტი, სადაც საქმესთან დაკავშირებულია შესაბამისი კატეგორია (სასწრაფო, სახანძრო, პოლიცია, კრიმინალური...)

#CATEGORIES სვეტის დაჭრა და შემდეგ კატეგორიზაცია

```
df_CAT_TMP = df_orig.set_index('CASE_ID').CATEGORIES.str.split(r',', expand=True).stack().reset_index(level=1, drop=True).to_frame('CATEGORIES');
df_only_CAT=pd.get_dummies(df_CAT_TMP, prefix='CAT', columns=['CATEGORIES']).groupby(level=0).sum()
```

# ორი dataframe-ის გაერთიანება

```
df = pd.merge(df_with_CTID, df_only_CAT, on=df_with_CTID.CASE_ID, how='inner')
```

- SPENT\_MINS – უცვლელია
- PREV\_HOSP\_COUNT – უცვლელია
- LASTHOSPDD – უცვლელია
- CALL\_COUNT – უცვლელია
- HOSPITALISATION – მოხდა თუ არა პაციენტის ჰოსპიტალიზაცია და ამ შემთხვევაში SPENT\_MINS პარამეტრს ემატება ჰოსპიტალიზაციისათვის დახარჯული დროც.

# ჰოსპიტალიზაციის სვეტის გენერაცია

# ბოლო ჰოსპიტალიზაციის დრო თუ მიმდინარე გამოძახების თარიღს ემთხვევა, ვთვლით, რომ გამოძახება ჰოსპიტალიზაციით დასრულდა.

```
df['HOSPITALISATION'] = np.where(df['LASTHOSPDD']==0, '1', '0')
```

# დუბლირებული მონაცემების თავიდან ასაცილებლად, თუ მიმდინარე გამოძახება ჰოსპიტალიზაციით დასრულდა (row['LASTHOSPDD']==0),

#ძველი ჰოსპიტალიზაციების რაოდენობას ვაკლებთ ერთს.

```
for i, row in df.iterrows():
```

```
    if row['LASTHOSPDD']==0:
```

```
        df.loc[i,'PREV_HOSP_COUNT'] -= 1
```

```
        df.loc[i,'LASTHOSPDD'] = np.nan
```

```
    if df.loc[i,'PREV_HOSP_COUNT'] == 0:
```

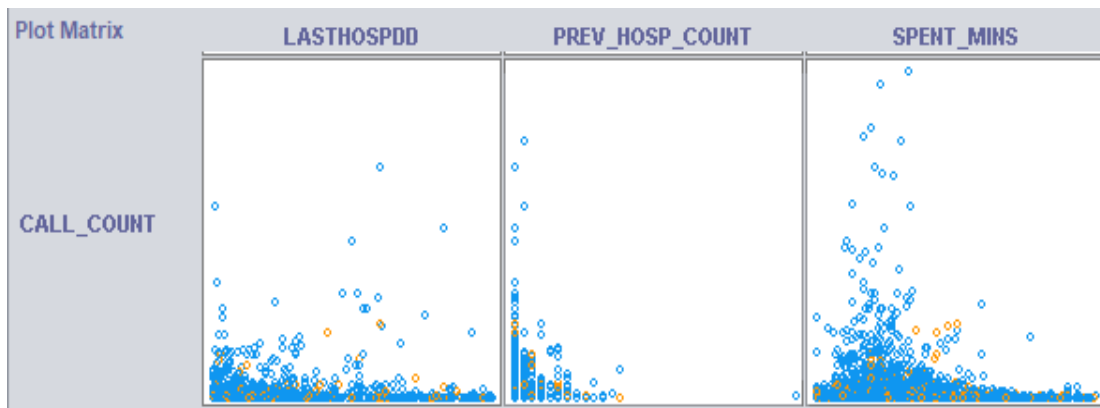
```
        df.loc[i,'PREV_HOSP_COUNT'] = np.nan
```

### 2.3. მონაცემთა ანალიზი და ვიზუალიზაცია

მიღებულ Dataset-ზე ერთდროულად ორი მიმართულებით ვმუშაობთ, Python პროგრამირება და Weka ხელსაწყო.

Weka მანქანური დასწავლის ალგორითმების ნაკრებია. ასევე, შეიცავს ისეთ საშუალებებს, რომლებიც გამოიყენება მონაცემთა წინასწარი დამუშავების, კლასიფიკაციის, რეგრესიისა და დაჯგუფებისათვის, მონაცემთა ასოციაციის წესების დასადგენად და ვიზუალიზაციისათვის [10].

მონაცემების კორელაციაზე წარმოდგენის შესაქმნელად გამოვიყენეთ Weka-ს ვიზუალიზაციის ხელსაწყოები. ქვემოთ მოყვანილია ვიზუალიზაციის რამდენიმე ისეთი ფრაგმენტი. მე-2 ნახაზზე ასახულია სასწრაფოს ბრიგადის მიერ პაციენტთან გატარებული დროისა და წინა ჰოსპიტალიზაციათა რაოდენობების ურთიერთდამოკიდებულება (პაციენტი იგივე ნომრით და იმავე ინციდენტის ტიპით). თითოეული წერტილი გრაფიკზე ასახავს ცალკეულ შემთხვევას. ფორთოხლისფრად გაფერადებულია ის საქმეები, რომლებიც დასრულდა ჰოსპიტალიზაციით.



ნახ.2. სასწრაფოს ბრიგადის მიერ პაციენტთან გატარებული დროისა, წინა ჰოსპიტალიზაციის შემთხვევების რაოდენობის და ჰოსპიტალიზაციის მოხდენის ურთიერთ დამოკიდებულება.

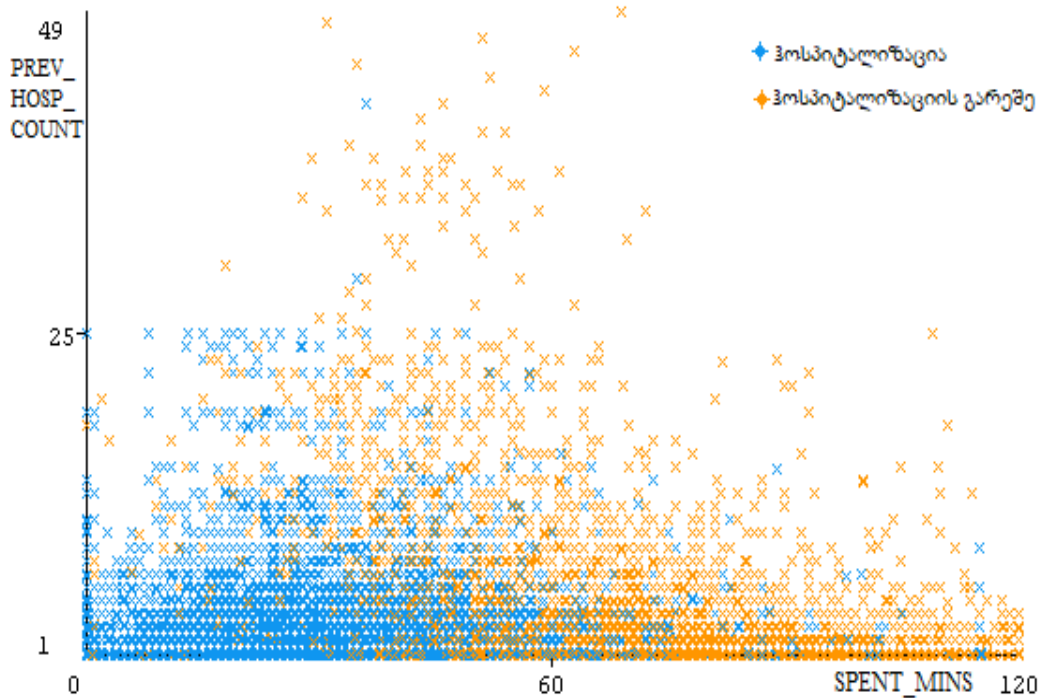
როგორც ვიზუალიზაციიდან ჩანს, მონაცემების დისპერსია (გაბნევა) დიდია, ამიტომ გადაწყდა, მონაცემთა სიმრავლე ბოლო 5 წლის მონაცემებით შემდგარიყო.

საინტერესოა აღინიშნოს, რომ უკუპროპორციული დამოკიდებულება გამოვლინდა გამოძახებების რიცხვსა და ჰოსპიტალიზაციის წინა შემთხვევების რაოდენობას შორის (იხ. შუა გრაფიკი ნახ.3).

რაც მეტია გამოძახებათა რიცხვი, მით ნაკლებია ჰოსპიტალიზაციის წინა შემთხვევების რაოდენობა. ეს შეიძლება აიხსნას ორი ტიპის პაციენტების არსებობით. ერთი მხრივ, არსებობს ბევრი ისეთი პაციენტი, რომელიც ხშირად იძახება სასწრაფო დახმარებას, მაგრამ ჩვეულებრივ, საჭიროებს არა ჰოსპიტალიზაციას, არამედ მხოლოდ ფსიქოლოგიურ მხარდაჭერას (ნუგეშს, დამშვიდებას).

მეორე მხრივ, არსებობენ პაციენტები, რომლებიც უკვე იყვნენ მოხვედრილი გარკვეულ კრიტიკულ სიტუაციაში და საჭირო გახდა მათი ჰოსპიტალიზაცია. ამის შემდეგ კი, მიღებული გამოცდილების გათვალისწინებით, ისინი ახერხებენ თავიანთი ჯანმრთელობის პრობლემების მოგვარებას ისე, რომ ჩვეულებრივ, სასწრაფო დახმარებას

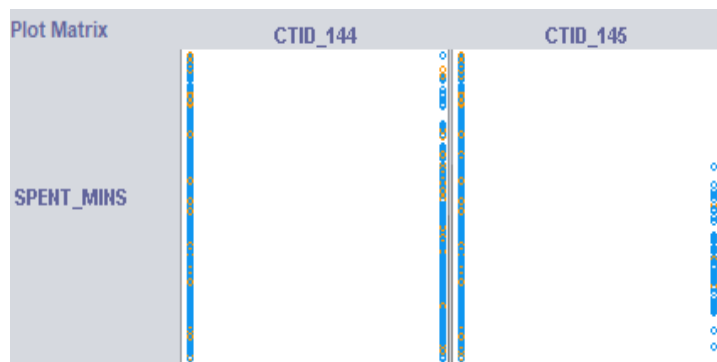
გაცილებით იშვიათად იძახებენ და თანაც, მხოლოდ მაშინ, როცა მათი ჯანმრთელობის მდგომარეობა მართლაც კრიტიკულია. ასე რომ, ასეთი გამოძახებები ძირითადად ჰოსპიტალიზაციით მთავრდება.



ნახ.3. გამოძახებათა რაოდენობის კავშირი სხვადასხვა პარამეტრთან

მარჯვენა გრაფიკიდან ჩანს, რომ წინა გამოძახებათა შემთხვევების დიდი რაოდენობის არსებობისას სასწრაფო დახმარების ბრიგადა შემთხვევაზე 20-30 წუთს ხარჯავს.

კავშირი ინციდენტის ტიპებსა და სასწრაფოს ბრიგადის მიერ შემთხვევაზე დახარჯულ დროს შორის ასახულია მე-4 ნახაზზე.



ნახ.4. კავშირი ინციდენტის ტიპსა და სასწრაფოს ბრიგადის გათავისუფლებას შორის

მარცხენა და მარჯვენა გრაფიკებზე წარმოდგენილია ორი სხვადასხვა ინციდენტის ტიპი (კოდებით 144 და 145). მარცხენა, ნულოვანი სვეტები გრაფიკებში შეესატყვისება



ყველა იმ ინციდენტის ტიპს, რომლებიც განსხვავებულია არჩეული ტიპებისგან. როგორც ვხედავთ, დროის მონაცემები ყველა ინციდენტის ტიპისთვის განაწილებულია დროის მთელ შუალედზე, მარჯვენა სვეტები კი გვიჩვენებს კონკრეტული ინციდენტის ტიპების შემთხვევაში დახარჯული დროის მონაცემების განაწილებას. აქედან ჩანს, რომ სხვადასხვა ინციდენტის ტიპის შესატყვისი მონაცემების განაწილება განსხვავებულია ერთმანეთისგან და განსხვავებულია ზოგადი განაწილებისაგან. ასე რომ, შესაძლებელია, ინციდენტის ყოველი ტიპი დახასიათდეს საშუალო დროის სიდიდით და განაწილების სახით.

### 3. დასკვნა

ნაშრომში განხილულია დისპეტჩერიზაციის კომპიუტერული სისტემის მოდელი და მონაცემთა სიმრავლის შემუშავების დეტალები. სისტემა ობიექტური გამოთვლების საფუძველზე საშუალებას მოგვცემს, ერთის მხრივ, გამოვლინდეს ყველაზე მაღალი კატეგორიის გადაუდებლობის მქონე შემთხვევები (გამომახებები), მეორეს მხრივ კი შეირჩეს სასწრაფო დახმარების ის ბრიგადები, რომლებიც შეძლებენ არჩეული შემთხვევების ყველაზე სწრაფად მომსახურებას.

### ლიტერატურა - References – Литература:

1. ჩოგოვაძე გ., ფრანგიშვილი ა., ჯაგოდნიშვილი თ., სურგულაძე გ. (2017). საინფორმაციო სისტემებიდან ინფორმაციული საზოგადოებისაკენ. სტუ-ს შრ.კრ. „მას“, N1(23). „ტექნიკური უნივერსიტეტი“, თბ., გვ.7-16
2. ჩოგოვაძე გ., ფრანგიშვილი ა., სურგულაძე გ. (2017). მართვის საინფორმაციო სისტემების დაპროგრამების ჰიბრიდული ტექნოლოგიები და მონაცემთა მენეჯმენტი. მონოგრ., ISBN 978-9941-20-790-7. სტუ, „ტექნიკური უნივერსიტეტი“, თბილისი, -1001 გვ.
3. სურგულაძე გ., კვიციანი გ. (2017). შესავალი NoSQL მონაცემთა ბაზებში (MongoDB). სტუ, „IT-კონსალტინგის ცენტრი“. თბ.
4. Fleischman RJ, Lundquist M, Jui J, Newgard CD, Warden C. (2013). Predicting Ambulance Time of Arrival to the Emergency Department Using Global Positioning System and Google Maps. *Prehospital emergency care : official journal of the National Association of EMS Physicians and the National Association of State EMS Directors.* 17(4):458-465
5. Propp DA, Rosenberg CA. (1991). A comparison of prehospital estimated time of arrival and actual time of arrival to an emergency department. *Am J Emerg Med.* 9: 301-3.
6. Kiviladze G. (2018). Semi Automated management of defining the case priority in the flu epidemic season for Emergency Medical Service. *GESJ:Computer Sciences and Telecommunications.* 1(53):101-105.
7. Chicco D. (2017). Ten quick tips for machine learning in computational biology. *BioData Mining.* 10 (35): 1-17. doi:10.1186/s13040-017-0155-3.
8. Luo G. (2015). MLBCD: a machine learning tool for big clinical data. *Health Information Science and Systems.* 3(3). doi:10.1186/s13755-015-0011-0.
9. <https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/>
10. Data Mining Software in Java. <https://www.cs.waikato.ac.nz/ml/weka/>

11. Wes McKinney. (2017). Python for Data Analysis Data Wrangling with Pandas, NumPy, and IPython. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. <http://opencarts.org/sachlaptrinh/pdf/28232.pdf>

**INFORMATION SOCIETY, NEW DATA MANAGEMENT TECHNOLOGIES  
AND EMERGENCY MANAGEMENT SYSTEMS**

Chogovadze Gocha, Prangishvili Archil, Kiviladze Giorgi,  
Surguladze Gia, Nareshelashvili Gulbaat  
Georgian Technichal University

**Summary**

With the purpose of forming an information society, the issues of integration of new data management technologies in information systems of mass service are considered. The concept of the use of ecosystems and large data systems are proposed, as well as aspects of the development and software implementation of machine learning algorithms based on hybrid programming technologies of the Python language. The experimental part of the work is represented by information and software packages supporting the decision-making system in extreme situations. In particular, the tasks of collecting relevant information and forming the data warehouse of the system of emergency assistance ("112") are solved for their subsequent processing by machine learning algorithms.

**ИНФОРМАЦИОННОЕ ОБЩЕСТВО, НОВЫЕ ТЕХНОЛОГИИ МЕНЕДЖМЕНТА  
ДАННЫХ И СИСТЕМЫ УПРАВЛЕНИЯ ЧРЕЗВЫЧАЙНЫМИ СИТУАЦИЯМИ**

Чоговадзе Г., Прангишвили А., Кивиладзе Г.,  
Сургуладзе Г., Нарешелашвили Г.  
Грузинский Технический Университет

**Резюме**

С целью формирования информационного общества в статье рассматриваются вопросы интеграции новых технологий менеджмента данных в информационные системы массового обслуживания. Предлагается концепция использования экосистем и систем анализа больших данных, а также аспекты разработки и программной реализации алгоритмов машинного обучения на основе технологий гибридного программирования языка Python. Экспериментальная часть работы представлена информационными и программными пакетами, поддерживающими систему принятия решений в экстремальных ситуациях. В частности, решаются задачи сбора релевантной информации и формирования хранилища данных системы неотложной помощи („112“) для последующей их обработки алгоритмами машинного обучения.