**Технически Университет – София**

Факултет приложна математика и информатика

**Курсова работа**

по

Приложен изкуствен интелект

на тема

Методи за попълване на липсващи данни

Shape

Description automatically generated with medium confidence

**Изпълнили:**

Елина Календерова – ИСН – 471 219 051

Даяна Димитрова – ИСН – 471 219 010

**Възложил:**

**Съдържание**

[**I.** **Въведение** 2](#_Toc102135317)

[**II.** **Класификация** 3](#_Toc102135318)

[**Видове липсващи данни** 3](#_Toc102135319)

[**1.** **Missing Completely At Random (MCAR) - Липсва напълно на случаен принцип** 3](#_Toc102135320)

[**2.** **Missing At Random (MAR) - Липсва на случаен принцип** 4](#_Toc102135321)

[**3.** **Missing Not At Random (MNAR) - Липсва не на случаен принцип** 4](#_Toc102135322)

[**III.** **Приложения** 5](#_Toc102135323)

[**Подходи за справяне с липсващите данни** 5](#_Toc102135324)

[**1.** **Deleting rows or columns** 5](#_Toc102135325)

[**2.** **Imputation of data with Mean/Median** 6](#_Toc102135326)

[**3.** **Imputation method for categorical columns** 7](#_Toc102135327)

[**4.** **Improve dataset find data** 7](#_Toc102135328)

[**IV.** **Алгоритъм на програмата** 8](#_Toc102135329)

[**Блок схема** 8](#_Toc102135330)

[**Легенда на блок схемата** 9](#_Toc102135331)

[**Описание на блок схемата** 9](#_Toc102135332)

[**V.** **Имплементация на Python** 11](#_Toc102135333)

[**ScreenShots** 11](#_Toc102135334)

[**Линк към GitHub** 12](#_Toc102135335)

**Част  1 - Въведение, термини, класификация, приложения**

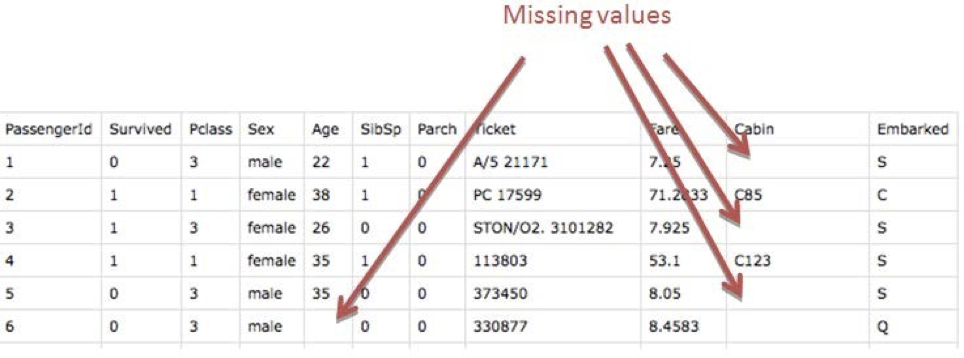
**Част  2 - Алгоритми, теория**

**Част 3 - Имплементация на Python**

**Част 4 - Експеримент(и), резултати, анализ, заключение**

# **Въведение**

Липсващите данни се дефинират като стойности или данни, които не се съхраняват (или не присъстват) за някаква променлива в дадения набор от данни. По-долу е дадена извадка от липсващите данни от набора от данни на Титаник. Можете да видите, че колоните „Възраст“ и „Кабина“ имат някои липсващи стойности.



Може да има множество причини, поради които определени стойности липсват в данните.  Причините за липсващите данни от набора от данни засягат подхода за обработка на липсващи данни. Така че е необходимо да се разбере защо данните могат да липсват. Някои от причините са както следва:

* Минали данни може да се повредят поради неправилна поддръжка.
* Наблюденията не се записват за определени полета поради някои причини. Възможно е да има неуспех при записването на стойностите поради човешка грешка.
* Потребителят не е предоставил стойностите умишлено.

# **Класификация**

## **Видове липсващи данни**

Има три  типа липсващи данни, които обикновено са категоризирани. Липсва напълно произволно (MCAR), липсва произволно и липсва не произволно. Всеки тип може да се среща във данните или дори комбинация от множество липсващи типове данни.

### **Missing Completely At Random (MCAR) - Липсва напълно на случаен принцип**

В MCAR вероятността да липсват данни е една и съща за всички наблюдения. В този случай няма връзка между липсващите данни и всякакви други наблюдавани или ненаблюдавани стойности (данните, които не са записани) в рамките на даден набор от данни. Тоест липсващите стойности са напълно независими от други данни. Няма шаблон. В случай на MCAR данните може да липсват поради човешка грешка, някаква повреда на системата/оборудването, загуба на проба или някои незадоволителни технически характеристики при записване на стойностите.

Например, да предположим, че в библиотека има просрочени книги. Липсват някои стойности на просрочени книги в компютърната система. Причината може да е човешка грешка, тъй като библиотекарят е забравил да въведе стойностите. Така че липсващите стойности на просрочени книги не са свързани с друга променлива/данни в системата. Не трябва да се предполага, че това е рядък случай. Предимството на такива данни е, че статистическият анализ остава безпристрастен.

### **Missing At Random (MAR) - Липсва на случаен принцип**

Липсващи на случаен принцип (MAR) означава, че причината за липсващите стойности може да се обясни с променливи, за които имате пълна информация, тъй като има някаква връзка между липсващите данни и други стойности/данни. В този случай данните не липсват за всички наблюдения. Липсва само в подизвадки от данни и има някакъв модел в липсващите стойности.

Например, ако проверите данните от проучването, може да откриете, че всички хора са отговорили на своя „Пол“, но стойностите на „Възраст“ липсват предимно за хората, които са отговорили на своя „Пол“ като „жена“. (Причината е, че повечето жени не искат да разкрият възрастта си.) Така че вероятността данни да липсват зависи само от наблюдаваните данни. В този случай променливите „Пол“ и „Възраст“ са свързани и причината за липсващите стойности на променливата „Възраст“ може да се обясни с променливата „Пол“, но не можете да предвидите самата липсваща стойност.

Да предположим, че се прави анкета за просрочени книги на библиотека. Полът и броят на просрочените книги се задават в анкетата. Да приемем, че повечето жени отговарят на анкетата, а мъжете са по-малко склонни да отговорят. Така че защо липсват данните може да се обясни с друг фактор, който е пол. В този случай статистическият анализ може да доведе до пристрастия. Получаването на безпристрастна оценка на параметрите може да се направи само чрез моделиране на липсващите данни.

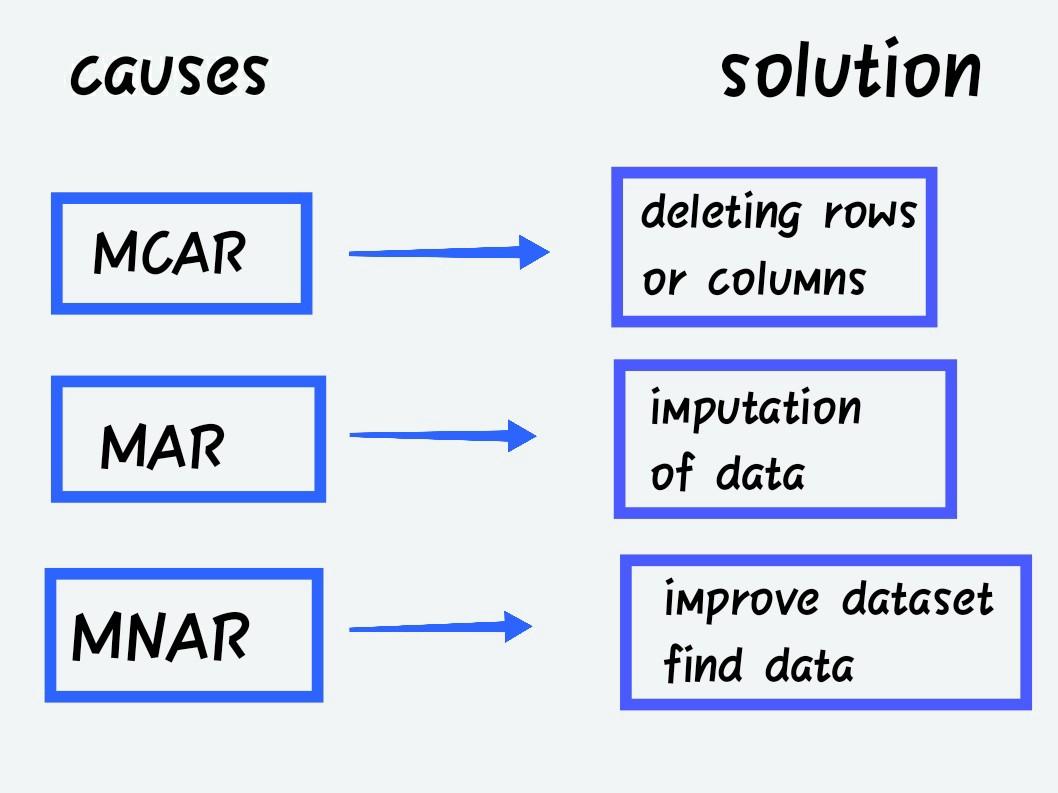
### **Missing Not At Random (MNAR) - Липсва не на случаен принцип**

Липсващите стойности зависят от ненаблюдаваните данни. Ако има някаква структура/модел в липсващите данни и други наблюдавани данни не могат да го обяснят, тогава това е липсващо не на случаен принцип (MNAR). Ако липсващите данни не попадат в MCAR или MAR, тогава те могат да бъдат категоризирани като MNAR. Това може да се случи поради нежеланието на хората да предоставят необходимата информация. Определена група хора може да не отговори на някои въпроси в анкета.

Например, да предположим, че името и броят на просрочените книги са попитани в анкетата за библиотека. Така че повечето хора, които нямат просрочени книги, вероятно ще отговорят на анкетата. Хората с повече просрочени книги е по-малко вероятно да отговорят на анкетата. Така че в този случай липсващата стойност на броя на просрочените книги зависи от хората, които имат повече просрочени книги. Друг пример, хората с по-малко доходи може да откажат да споделят тази информация в анкета. В случай на MNAR също статистическият анализ може да доведе до пристрастия.

# **Приложения**

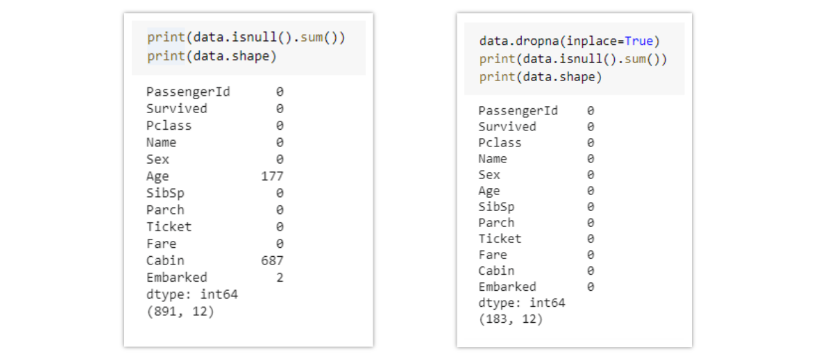
## **Подходи за справяне с липсващите данни**

Все още не е ясно как точно да се разграничат видовете липсващи данни и как да се обработва всеки тип данни. Освен това, дори данните, за които сме сигурни, че са от конкретен тип, може да имат елементи от различни липсващи типове данни, което усложнява проблема. Най-общо казано са приети следните подходи за отделните типове:

***Python*** е подходящ език за извършване на анализ на данни, главно поради наличието от пакети на ***Python***, ориентирани към данни. ***Pandas*** е един от тези пакети и прави импортирането и анализирането на данни много по-лесно. Той разпознава липсващите данни и ги заменя с ***NaN*** (това означава ***Not a Number***), като по този начин последващата обработка се опростява. Съществуват множество подходи за справяне с тези данни в зависимост от техния вид и предназначение. 

### **Deleting rows or columns**

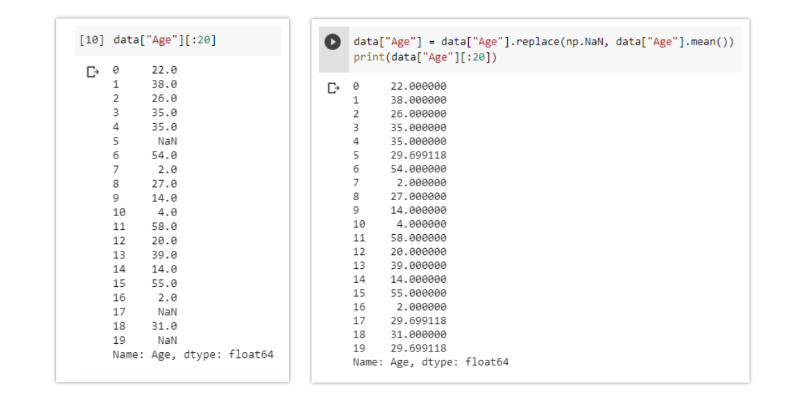
Липсващите стойности могат да бъдат обработени чрез изтриване на редовете или колоните с нулеви стойности. Ако колоните имат повече от половината редове като нула, тогава цялата колона може да бъде отхвърлена. Редовете, които имат стойности на една или повече колони като нула, също могат да бъдат премахнати. Този подход има както своите предимства, така и недостатъци. Едно от предимствата е, че с премахване на всички липсващи стойности, се създава един стабилен модел, а като недостатък може да бъде разгледано това, че се губи много информаци. Работата също не му е добра, ако процентът на липсващите стойности е прекомерен в сравнение с пълния набор от данни.

Примерна таблица:

### **Imputation of data with Mean/Median**

Колоните в набора от данни, които имат числови непрекъснати стойности, могат да бъдат заменени със средна стойност, медиана или режим на оставащите стойности в колоната. Този метод може да предотврати загубата на данни в сравнение с предишния метод. Замяната на горните две приближения (средно, медиана) е статистически подход за обработка на липсващите стойности. В примера по-долу е показано как липсващите стойности се заменят със средната стойност:

Примерна таблица:

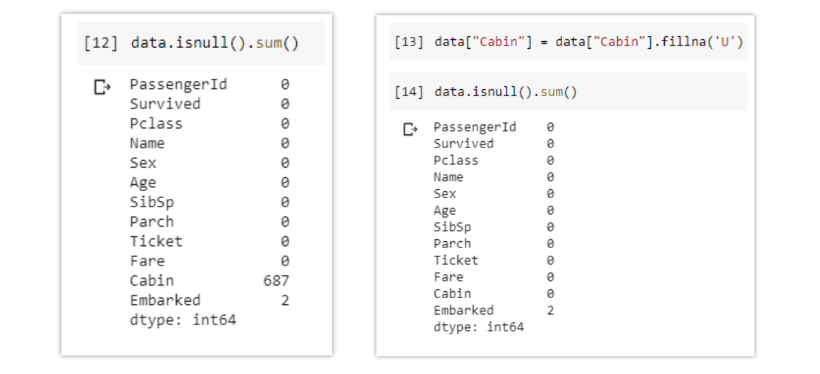


Този метод също има както своите предимства така и недостатъци. Две от предимствата, които се наблюдават са: предотвратяването на загуба на данни, което води от изтриване на редове или колони, работи добре с малък набор от данни и е лесен за изпълнение. Недостатъците, които могат да бъдат предизвикани са: работи само с числени непрекъснати променливи и това, че може да причини изтичане на данни 

### **Imputation method for categorical columns**

Когато липсващите стойности са от категорийни колони (низови или числови), тогава липсващите стойности могат да бъдат заменени с най-честата категория. Ако броят на липсващите стойности е много голям, тогава той може да бъде заменен с нова категория.

Примерна таблица:



Тук също се наблюдават както позитиви, така и негативи. Няколко от предимствата са: предотвратяване от загуба на данни, която води от изтриване на редове или колони, работи добре с малък набор от данни, лесен за изпълнение и отрича загубата на данни чрез добавяне на уникална категория. Някои от негативите са: работи само с категорични променливи и добавяне на нови функции към модела по време на кодиране, може да доведе до лошо представяне

### **Improve dataset find data**

 Данните за MNAR нямат ясна причина за отсъствието им. Знанията в областта са основните познания, които притежаваме за решаването на проблема. Учен по данни в областта на образованието например би имал основни познания за това как функционират училищата и би могъл да разбере по-лесно наборите от данни от тази област. Знанието в областта, заедно с всеки ресурс, който може да ни помогне да определите причината, ще бъде най-добрата стратегия за потенциално завършване на нашия пъзел с данни за MNAR.

# **Алгоритъм на програмата**

## **Блок схема**

## **Легенда на блок схемата**

## **Описание на блок схемата**

Програмата започва със зареждането на данните, които ще се обработват, от csv файл. Dataset-а се състои от 7 колони и 66 реда като един от редовете е с имената на колоните, а останалите 65 с данните. За всяка колона се прилага различен подход в зависимост от особеността на информацията в нея. Най-общо казано, програмата започва с първата колона, през втората, третата и така до последната.

Видовете колони с техните типове данни са както следва:

|  |  |
| --- | --- |
| Country | *String* |
| Total Library Count | *Integer* |
| TV Shows Count | *Integer* |
| Movies Count | *Integer* |
| Basic Cost | *Double* |
| Standard Cost | *Double* |
| Premium Cost | *Double* |

***Важно уточнение е, че когато се обработва съответна колона, не се върви по абсолютно всеки ред от нея, а само по тези, които имат липсващи данни.***

Първата колона е ***Country***. Тъй като липсващите страни са едва около 6% от всички данни, не би било проблем, редовете с липсващи държави да се изтрият. Нещо повече, дори да бъдат обработени по различен начин, то няма как в конкретния случай да са от полза, тъй като всички последващи колони се отнасят за държавата, с която си реферират.

Следващите колони ***Total Library Count***, ***TV Shows Count*** и ***Movies Count*** имат по най-много една липсваща стойност в ред, което дава възможност за обработка на данните спрямо стойностите в другите две колони, тъй като втората колона, ***Total Library Count***, всъщност е сбор от другите две колони след нея, а именно ***TV Shows Count*** и ***Movies Count***. С помощта на тази зависимост и факта, че има само по една липсваща стойност в ред, в една от трите колони, попълването на тези стойности става много по-рационално. Ако колоната ***Total Library Count*** е ***NaN***, то нейната стойност е сборът от колоните ***TV Shows Count*** и ***Movies Count***. Ако колоната ***TV Shows Count*** е ***NaN***, стойността й е ***Total Library Count*** минус ***Movies Count***. И ако ***Movies Count*** е ***NaN***, нейната стойност ще бъде ***Total Library Count*** минус ***TV Shows Count***.

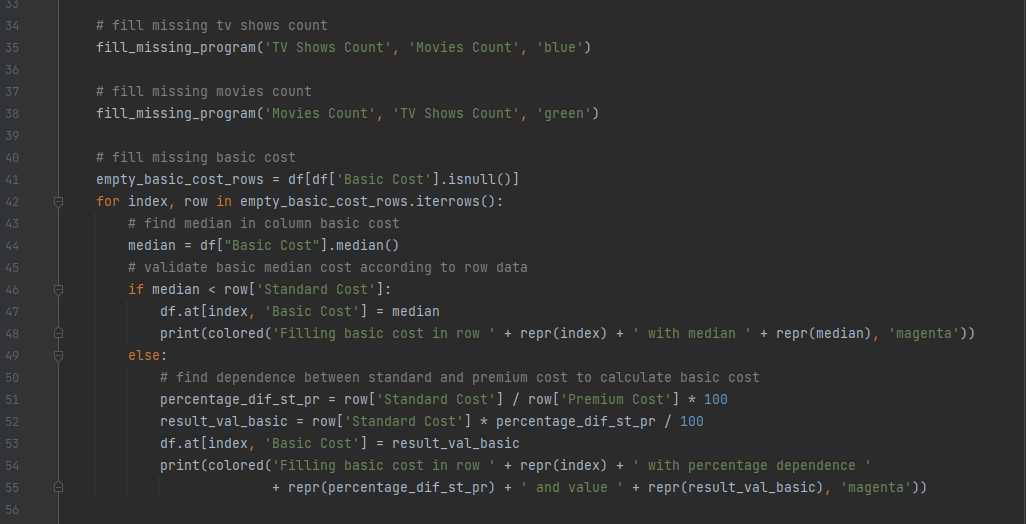
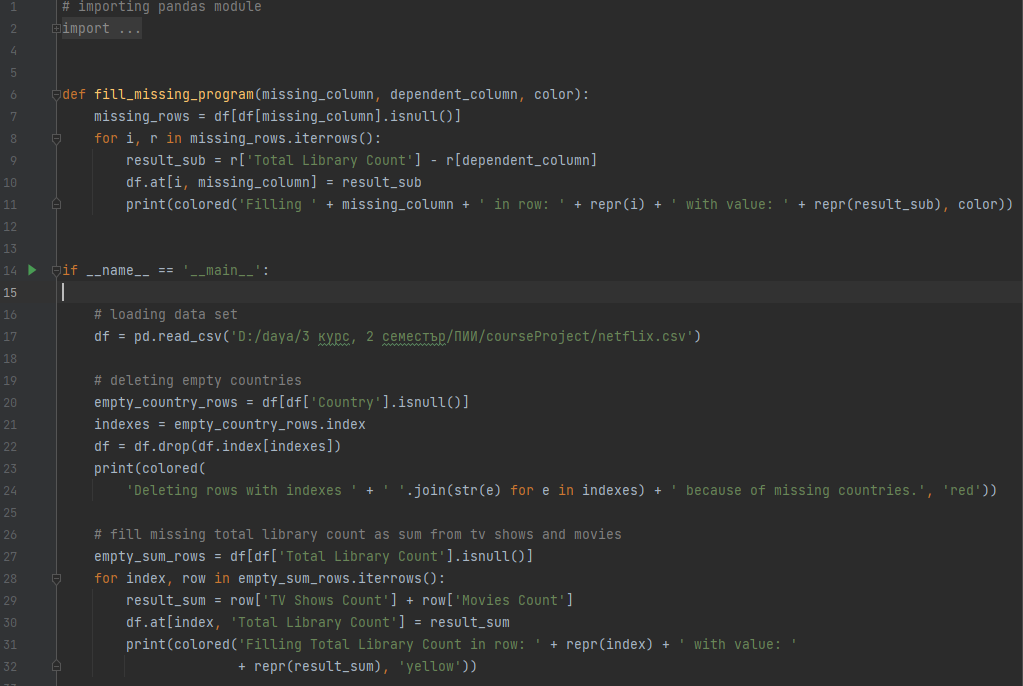
Ако има липсваща стойност в колона ***Basic Cost***, се взима медианата на колоната. Ако медианата е по-малка от съответната стойност в реда за колона ***Standard Cost***, то липсващата стойност се заменя с изчислената медиана. В противен случай обаче, се взима отношението между ***Standard Cost*** и ***Premium Cost*** в процент, а именно с колко процента ***Premium Cost*** е повече от ***Standard Cost***. След това този процент се сваля от ***Standard Cost*** и се присвоява на липсващата стойност в ***Basic Cost*** за съответния ред.

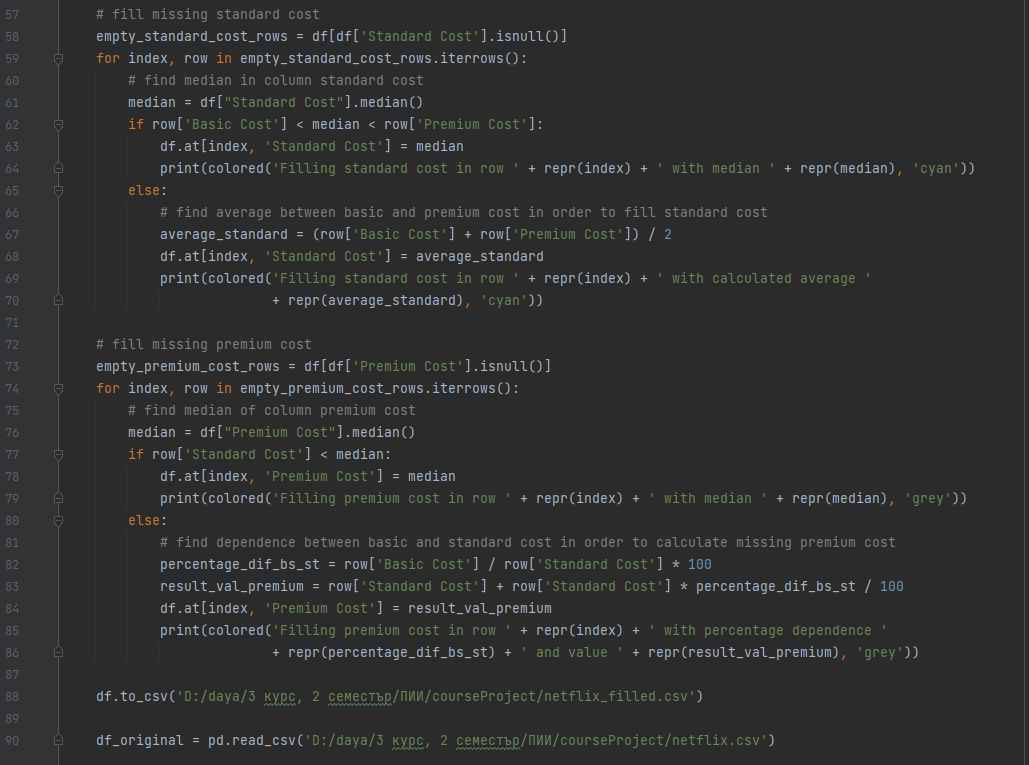
Ако липсващата стойност е в колона ***Standard Cost***, отново се взима медианата на колоната. Ако стойността на медианата е между ***Basic Cost*** и ***Premium Cost***, медианата се присвоява на липсващата стойност. В противен случай се изчислява средно аритметично между стойностите за ***Basic Cost*** и ***Premium Cost*** на съответния ред с ***NaN*** и полученият резултат се записва на мястото на празната клетка.

Последната колона е ***Premium Cost***. В този случай отново първо се взима медианата на колоната. След това се проверява дали тази медиана е по-голяма от ***Standard Cost*** в съответния ред, ако това е така медианата директно се присвоява на липсващата стойност. В противен случай обаче се взима процентът, с който ***Standard Cost*** е по-голяма от ***Basic Cost***. Този процент се прибавя към ***Standard Cost*** и се записва в ***NaN*** клетката на съответния ред в колоната ***Premium Cost***.

Разбира се, изчисленията в последните три колони са възможни именно поради факта, че има по най-много една липсваща стойност из между трите колони в един ред.

# **Имплементация на Python**

******ScreenShots**



**Линк към GitHub** -> <https://github.com/daianadimitrova258/FillingMissingValues>

# importing pandas module

from termcolor import colored

import pandas as pd

def fill\_missing\_program(missing\_column, dependent\_column, color):

   missing\_rows = df[df[missing\_column].isnull()]

   for i, r in missing\_rows.iterrows():

       result\_sub = r['Total Library Count'] - r[dependent\_column]

       df.at[i, missing\_column] = result\_sub

       print(colored('Filling ' + missing\_column + ' in row: ' + repr(i) + ' with value: ' + repr(result\_sub), color))

if \_\_name\_\_ == '\_\_main\_\_':

   # loading data set

   df = pd.read\_csv('D:/daya/3 курс, 2 семестър/ПИИ/courseProject/netflix.csv')

   # deleting empty countries

   empty\_country\_rows = df[df['Country'].isnull()]

   indexes = empty\_country\_rows.index

   df = df.drop(df.index[indexes])

   print(colored(

       'Deleting rows with indexes ' + ' '.join(str(e) for e in indexes) + ' because of missing countries.', 'red'))

   # fill missing total library count as sum from tv shows and movies

   empty\_sum\_rows = df[df['Total Library Count'].isnull()]

   for index, row in empty\_sum\_rows.iterrows():

       result\_sum = row['TV Shows Count'] + row['Movies Count']

       df.at[index, 'Total Library Count'] = result\_sum

       print(colored('Filling Total Library Count in row: ' + repr(index) + ' with value: '

                     + repr(result\_sum), 'yellow'))

   # fill missing tv shows count

   fill\_missing\_program('TV Shows Count', 'Movies Count', 'blue')

   # fill missing movies count

   fill\_missing\_program('Movies Count', 'TV Shows Count', 'green')

   # fill missing basic cost

   empty\_basic\_cost\_rows = df[df['Basic Cost'].isnull()]

   for index, row in empty\_basic\_cost\_rows.iterrows():

       # find median in column basic cost

       median = df["Basic Cost"].median()

       # validate basic median cost according to row data

       if median < row['Standard Cost']:

           df.at[index, 'Basic Cost'] = median

           print(colored('Filling basic cost in row ' + repr(index) + ' with median ' + repr(median), 'magenta'))

       else:

           # find dependence between standard and premium cost to calculate basic cost

           percentage\_dif\_st\_pr = row['Standard Cost'] / row['Premium Cost'] \* 100

           result\_val\_basic = row['Standard Cost'] \* percentage\_dif\_st\_pr / 100

           df.at[index, 'Basic Cost'] = result\_val\_basic

           print(colored('Filling basic cost in row ' + repr(index) + ' with percentage dependence '

                         + repr(percentage\_dif\_st\_pr) + ' and value ' + repr(result\_val\_basic), 'magenta'))

   # fill missing standard cost

   empty\_standard\_cost\_rows = df[df['Standard Cost'].isnull()]

   for index, row in empty\_standard\_cost\_rows.iterrows():

       # find median in column standard cost

       median = df["Standard Cost"].median()

       if row['Basic Cost'] < median < row['Premium Cost']:

           df.at[index, 'Standard Cost'] = median

           print(colored('Filling standard cost in row ' + repr(index) + ' with median ' + repr(median), 'cyan'))

       else:

           # find average between basic and premium cost in order to fill standard cost

           average\_standard = (row['Basic Cost'] + row['Premium Cost']) / 2

           df.at[index, 'Standard Cost'] = average\_standard

           print(colored('Filling standard cost in row ' + repr(index) + ' with calculated average '

                         + repr(average\_standard), 'cyan'))

   # fill missing premium cost

   empty\_premium\_cost\_rows = df[df['Premium Cost'].isnull()]

   for index, row in empty\_premium\_cost\_rows.iterrows():

       # find median of column premium cost

       median = df["Premium Cost"].median()

       if row['Standard Cost'] < median:

           df.at[index, 'Premium Cost'] = median

           print(colored('Filling premium cost in row ' + repr(index) + ' with median ' + repr(median), 'grey'))

       else:

           # find dependence between basic and standard cost in order to calculate missing premium cost

           percentage\_dif\_bs\_st = row['Basic Cost'] / row['Standard Cost'] \* 100

           result\_val\_premium = row['Standard Cost'] + row['Standard Cost'] \* percentage\_dif\_bs\_st / 100

           df.at[index, 'Premium Cost'] = result\_val\_premium

           print(colored('Filling premium cost in row ' + repr(index) + ' with percentage dependence '

                         + repr(percentage\_dif\_bs\_st) + ' and value ' + repr(result\_val\_premium), 'grey'))

   df.to\_csv('D:/daya/3 курс, 2 семестър/ПИИ/courseProject/netflix\_filled.csv')

   df\_original = pd.read\_csv('D:/daya/3 курс, 2 семестър/ПИИ/courseProject/netflix.csv')