

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО



Факультет Безопасных Информационных Технологий

Управление мобильными устройствами

Лабораторная работа №1

Выполнил:

студент группы N3348

Колесников Н. Д.

Проверил:

Федоров И. Р.

Санкт-Петербург

2020

Цель работы

В данной работе необходимо реализовать простейшее правило тарификации для услуг типа «Телефония» по длительности разговора и «СМС» по общему количеству.

Средство реализации

Для реализации мною был выбран язык программирования Python 3.x, т. к. я считаю что для выполнения данной задачи, он является наиболее подходящим вариантом (Так как присутствует большое количество модулей, используемых для работы с данными и их визуализации).

Для работы была выбрана среда Jupyter Notebook (также известный как IPython Notebook), которая крайне удобна в случае, если ты используешь Python (Так как весь код разделён на секции, что 1. упрощает процесс отладки программы 2. позволяет при необходимости выводить содержимое данных в любом месте программы).

Исходный код

https://github.com/LordKzoth/MOBILE_MANAGEMENT/blob/master/LW_1/LR_1_CDR.py

Ход работы (Вариант б)

Первым делом был импортирован модуль pandas, который используется для обработки и анализа данных (Данный модуль строиться поверх модуля numpy, который используется для упрощения работы с матрицами и векторами данных).

```
In [1]: import pandas as pd
```

Далее посредством метода pandas.read_csv('file name') файл data.csv был преобразован к удобному виду и записан в переменную df. После этого для получения информации по данным, с которыми предстоит работать я вывел общее количество записей в таблице, а также отобразил первые 5 записей.

```
In [2]: df = pd.read_csv('data.csv')
print('Количество строк в таблице: {}'.format(len(df)))
df[:5]
```

Количество строк в таблице: 9

Out[2]:

	timestamp	msisdn_origin	msisdn_dest	call_duration	sms_number
0	2020-01-01 00:00:00	915783624	911926375	36.23	15
1	2020-01-01 00:05:00	911926375	968247916	9.20	5
2	2020-01-01 00:10:00	936415793	915642913	7.52	24
3	2020-01-01 00:15:00	914976835	914976835	96.70	97
4	2020-01-01 00:20:00	962365794	933156729	110.44	15

Следующим шагом я создал множество всех абонентов, которые были представлены в столбцах `msisdn_origin` и `msisdn_dest` (Это было необходимо для того чтобы исключить повторения и получить представление о том, сколько действительно абонентов представлено в датасете).

```
In [4]: # Перечень абонентов
abon = set()

for v in df['msisdn_origin']:
    abon.add(v)
for v in df['msisdn_dest']:
    abon.add(v)

abon = sorted(list(abon))
print('Все абоненты:')
print(abon)
```

Все абоненты:
[911926375, 914976835, 915642913, 915783624, 933156729, 936415793, 962365794, 966714385, 968247916]

Далее я создал переменную `abon_number`, которой присвоил значение согласно моему варианту.

```
abon_number = 968247916
```

Следующим шагом, я отобразил все данные по заданному абоненту (Это было сделано для того, чтобы далее на основании полученных результатов в дальнейшем реализовать функцию рассчитывающую стоимость предоставленных абоненту услуг).

Исходящие

```
In [6]: outgoing = df[df['msisdn_origin'] == abon_number].copy()
outgoing
```

```
Out[6]:
```

	timestamp	msisdn_origin	msisdn_dest	call_duration	sms_number
6	2020-01-01 00:30:00	968247916	962365794	91.48	57

```
In [7]: out_calls = outgoing['call_duration'].values
out_calls
```

```
Out[7]: array([91.48])
```

```
In [8]: sms = outgoing['sms_number'].values
sms
```

```
Out[8]: array([57], dtype=int64)
```

Входящие

```
In [9]: incoming = df[df['msisdn_dest'] == abon_number].copy()
incoming
```

```
Out[9]:
```

	timestamp	msisdn_origin	msisdn_dest	call_duration	sms_number
1	2020-01-01 00:05:00	911926375	968247916	9.2	5

```
In [10]: in_calls = out_calls = incoming['call_duration'].values
in_calls
```

```
Out[10]: array([9.2])
```

И наконец, я реализовал функцию рассчитывающую стоимость предоставленных абоненту услуг.

Тарификация

```
In [11]: def tariff(outgoing, incoming):
# Get necessary values from data
out_calls = outgoing['call_duration'].values
sms = outgoing['sms_number'].values
in_calls = incoming['call_duration'].values

# Calculate amount
out_calls_sum = sum([v * 4 for v in out_calls])
in_calls_sum = sum([5*0 + (v-5)*1 if v>=5 else 0 for v in in_calls])
sms_sum = sum([5*0 + (v-5)*1 if v>=5 else 0 for v in sms])
amount = out_calls_sum + in_calls_sum + sms_sum

return amount
```

Описание функции:

1. Я формирую списки out_calls (Исходящие звонки), sms (Отправленные СМС), in_calls (Входящие звонки) (Был выбран тип List для хранения данных из расчёта, что по одному абоненту может быть более чем одна запись, в прочем как видно из пунктов «Исходящие» и «Входящие» для данного абонента представлено по одной записи в каждом из них).
2. Далее я высчитываю стоимость услуги (с учетом коэффициентов и условий, указанных в моём варианте).

Данную программу можно представить в сокращённом формате (с использованием расширения .py):

```
1 import pandas as pd
2
3 df = pd.read_csv('data.csv')
4
5 abon_number = 968247916
6
7 outgoing = df[df['msisdn_origin'] == abon_number].copy() # Исходящие
8 incoming = df[df['msisdn_dest'] == abon_number].copy()   # Входящие
9
10 def tariff(outgoing, incoming):
11     # Get necessary values from data
12     out_calls = outgoing['call_duration'].values
13     sms = outgoing['sms_number'].values
14     in_calls = incoming['call_duration'].values
15
16     # Calculate amount
17     out_calls_sum = sum([v * 4 for v in out_calls])
18     in_calls_sum = sum([5*0 + (v-5)*1 if v>=5 else 0 for v in in_calls])
19     sms_sum = sum([5*0 + (v-5)*1 if v>=5 else 0 for v in sms])
20     amount = out_calls_sum + in_calls_sum + sms_sum
21
22     return amount
23
24
25 print('Итоговая сумма: {:.2f}'.format(tariff(outgoing, incoming)))
```

```
C:\Users\Lord K'zoth\Desktop>py LR_1_CDR.py
Итоговая сумма: 422.12
```

Итоговый результат

```
In [12]: print('Итоговая сумма: {:.2f}'.format(tariff(outgoing, incoming)))
Итоговая сумма: 422.12
```

Видим результат, округленный до сотых: **422.12**

Вывод: в ходе выполнения данной лабораторной работы мною обследован предоставленный файл data.csv, после чего из него были извлечены необходимые данные. После чего было реализовано простейшее правило тарификации для услуг типа «Телефония» по длительности разговора и «СМС» по общему количеству.