МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Информационных систем

ОТЧЕТ

по практической работе
по дисциплине «Машинное обучение»
Тема: «Анализ датасета с данными о шагах»

Студент гр. 2373 Нупрейчик М.

Преподаватель Татчина Я. А.

Санкт-Петербург
2024

В данной работе проводится изучение датасета о шагах. Исходный датасет состоит из 7 колонок: *id*, *timestamp*, *data_version*, *data_type*, *data*, *provider*, *user_id*.

	id	timestamp	data_version	data_type	data	provider	user_id
0	applehealth_activity_steps_2261472_065994e97b5...	1704297546	2	applehealth_activity_steps	[{"dataType":"quantity","entryType":"activity_...	applehealth	2261472
1	applehealth_activity_steps_2261472_d565ff4cfde...	1717525192	2	applehealth_activity_steps	[{"dataType":"quantity","entryType":"activity_...	applehealth	2261472
2	applehealth_activity_steps_2261472_600c2ccda7b...	1730481905	2	applehealth_activity_steps	[{"dataType":"quantity","entryType":"activity_...	applehealth	2261472
3	applehealth_activity_steps_2261472_49c6dcdd76d...	1725171352	2	applehealth_activity_steps	[{"dataType":"quantity","entryType":"activity_...	applehealth	2261472
4	applehealth_activity_steps_2261472_d1a7e9c4d60...	1725653223	2	applehealth_activity_steps	[{"dataType":"quantity","entryType":"activity_...	applehealth	2261472

Рисунок 1. "Шапка" исходного датасета

Вся интересующая нас информация хранится в формате JSON в поле data. Данное поле необходимо преобразовать в отдельную таблицу. Полученный фрейм содержит следующие столбцы: *dataType*, *entryType*, *health_kit_id*, *bundleIdentifier*, *hardwareVersion*, *manufacturer*, *model*, *name*, *operatingSystemVersion*, *productType*, *softwareVersion*, *sourceName*, *version*, *sourceGroup*, *sourceType*, *timeEnd*, *timeStart*, *unit*, *value*, *HKExternalUUID*, *HKMetadataKeySyncIdentifier*, *HKMetadataKeySyncVersion*. Вся дальнейшая работа ведется с данным фреймом.

1. Предобработка данных.

Прежде чем анализировать данные, необходимо разобраться с пропущенными значениями (если таковые имеются), определить неинформативные признаки.

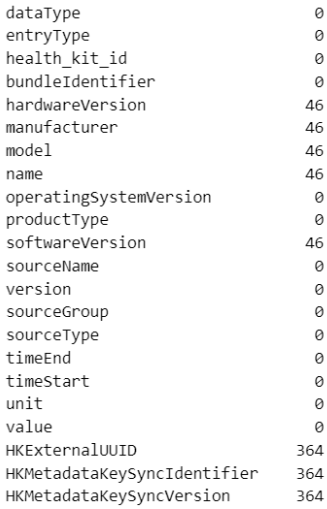

Рисунок 2. Количество пропущенных значений в каждом столбце

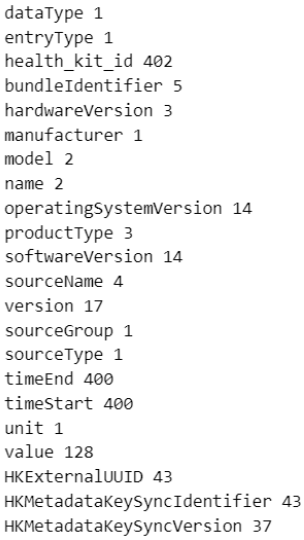

Рисунок 3. Количество уникальных значений в каждом столбце

Есть несколько колонок, в которых очень много пропущенных значений. Это колонки: *HKExternalUUID*, *HKMetadataKeySyncIdentifier*, *HKMetadataKeySyncVersion*. В этих колонках заполненные значения уникальны. Вопрос лишь в важности данных колонок в исследовании - исходя из логики, сложно представить, что технические характеристики каждой записи могут влиять на активность человека. В связи с этим, колонки будут удалены.

Вместе с тем удаляются колонки, которые содержат единственное уникальное значение - *dataType*, *entryType*, *manufacturer*, *sourceGroup*, *sourceType, unit*.

```
health_kit_id 402
bundleIdentifier 5
hardwareVersion 3
model 2
name 2
operatingSystemVersion 14
productType 3
softwareVersion 14
sourceName 4
version 17
timeEnd 400
timeStart 400
value 128
```

Рисунок 4. Оставшиеся после преобразований столбцы

Колонки *model*, *name*, *productType*, *sourceName*, *hardwareVersion* отражают информацию об устройстве, с которого получены данные.

model	name	productType	sourceName	hardwareVersion
iPhone	iPhone	iPhone15,2	iPhone 14 Pro	iPhone15,2
Watch	Apple Watch	Watch5,9	Watch5,9	Watch5,9
		iPhone15,2	Oura	

Рисунок 5. Строки из фрейма

Колонки дублируют информацию. В целях сокращения количества признаков, и, соответственно, уменьшения объема занимаемой памяти, оставляем только колонку *model*.

Остается вопрос пропущенных значений в данном столбце. Все пропуски связаны с тем, что это данные, полученные с устройства **Oura**. По каким-то причинам, в api не предусмотрен перенос данных в этот столбец.

Так как теперь информация хранится в столбце model, столбцы *name*, *productType*, *sourceName*, *hardwareVersion* больше не нужны.

	health_kit_id	bundleIdentifier	model	operatingSystemVersion	softwareVersion	version	timeEnd	timeStart	value
0	49D29566-6524-4FDB-B1E9-98419D373D72	com.apple.health.285A5E5B-B5D6-4182-BC34-9A821...	iPhone	17.1.2	17.1.2	17.1.2	2024-01-02T14:09:58+0100	2024-01-02T14:09:53+0100	13
1	AE54F0C4-BD1B-46FF-BB3C-55BC438E09A0	com.apple.health.441DF187-A35B-42C3-AA35-37173...	Watch	10.4.0	10.4	10.4	2024-06-03T00:27:49+0200	2024-06-03T00:27:47+0200	16
2	C687C0F0-F9BA-45CC-80E5-C758C2AF4196	com.ouraring.oura	Oura	18.0.0	NaN	2409241358	2024-10-31T16:51:00+0100	2024-10-31T16:51:00+0100	38
3	EBE1C9E4-DE18-486C-8966-71E5E84588CA	com.apple.health.285A5E5B-B5D6-4182-BC34-9A821...	iPhone	17.6.1	17.6.1	17.6.1	2024-08-31T20:06:17+0200	2024-08-31T19:59:38+0200	518
4	84851539-5F29-4C60-9595-5D7BDCBFF87D	com.apple.health.285A5E5B-B5D6-4182-BC34-9A821...	iPhone	17.6.1	17.6.1	17.6.1	2024-09-06T20:54:41+0200	2024-09-06T20:49:27+0200	112

Рисунок 6. Первые 5 строк фрейма на данный момент

Признаки *health_kit_id, bundleIdentifier* - системные поля. При анализе не нужны, поэтому удаляются.

	model	operatingSystemVersion	softwareVersion	version	timeEnd	timeStart	value
0	iPhone	17.1.2	17.1.2	17.1.2	2024-01-02T14:09:58+0100	2024-01-02T14:09:53+0100	13
1	Watch	10.4.0	10.4	10.4	2024-06-03T00:27:49+0200	2024-06-03T00:27:47+0200	16
2	Oura	18.0.0	NaN	2409241358	2024-10-31T16:51:00+0100	2024-10-31T16:51:00+0100	38
3	iPhone	17.6.1	17.6.1	17.6.1	2024-08-31T20:06:17+0200	2024-08-31T19:59:38+0200	518
4	iPhone	17.6.1	17.6.1	17.6.1	2024-09-06T20:54:41+0200	2024-09-06T20:49:27+0200	112

Рисунок 7. Первые пять строк фрейма после удаления системных полей

Из неявных признаков на данном этапе остались *operatingSystemVersion*, *softwareVersion*, *version*. Признак *operatingSystemVersion* содержит информацию о версии ПО, которая стоит на устройстве. В этом столбце нет пропусков.

Строки, где *model* = **iPhone** или **Watch** - версии во всех трех столбцах совпадают. Проблема возникает в строках, где *model* = **Oura**. В этих строках значение *softwareVersion* не задано, а значение *version* имеет странный формат. Будем считать, что в столбце *operatingSystemVersion* версия отражается корректно.

При анализе столбцов выяснилось, что значения в строке между столбцами совпадают. Поэтому столбца *operatingSystemVersion* будет более чем достаточно.

	model	operatingSystemVersion	timeEnd	timeStart	value
0	iPhone	17.1.2	2024-01-02T14:09:58+0100	2024-01-02T14:09:53+0100	13
1	Watch	10.4.0	2024-06-03T00:27:49+0200	2024-06-03T00:27:47+0200	16
2	Oura	18.0.0	2024-10-31T16:51:00+0100	2024-10-31T16:51:00+0100	38
3	iPhone	17.6.1	2024-08-31T20:06:17+0200	2024-08-31T19:59:38+0200	518
4	iPhone	17.6.1	2024-09-06T20:54:41+0200	2024-09-06T20:49:27+0200	112

Рисунок 8. Оставшиеся поля фрейма

Для дальнейшего анализа были созданы следующие признаки: *datetimeStart* и *datetimeEnd* – это признаки *timeStart* и *timeEnd* соответственно, приведенные к типу datetime; *timeStart* и *timeEnd* – время, выделенные из признаков *datetimeStart* и *datetimeEnd*; *dateStart* и *dateEnd* – даты, выделенные из признаков *datetimeStart* и *datetimeEnd*.

	model	operatingSystemVersion	timeEnd	timeStart	value	datetimeEnd	datetimeStart	dateEnd	dateStart
0	iPhone	17.1.2	14:09:58	14:09:53	13	2024-01-02 14:09:58+01:00	2024-01-02 14:09:53+01:00	2024-01-02	2024-01-02
1	Watch	10.4.0	00:27:49	00:27:47	16	2024-06-03 00:27:49+02:00	2024-06-03 00:27:47+02:00	2024-06-03	2024-06-03
2	Oura	18.0.0	16:51:00	16:51:00	38	2024-10-31 16:51:00+01:00	2024-10-31 16:51:00+01:00	2024-10-31	2024-10-31
3	iPhone	17.6.1	20:06:17	19:59:38	518	2024-08-31 20:06:17+02:00	2024-08-31 19:59:38+02:00	2024-08-31	2024-08-31
4	iPhone	17.6.1	20:54:41	20:49:27	112	2024-09-06 20:54:41+02:00	2024-09-06 20:49:27+02:00	2024-09-06	2024-09-06

Рисунок 9. Фрейм после добавления дополнительных признаков

2. Анализ источников данных.

Есть 3 источника данных в датасете: смартфон **IPhone**, умные часы **Watch**, умное кольцо **Oura**. На данный момент не ясно, работали ли устройства, например, одновременно - не произошел ли двойной учет шагов? Какой источник предоставляет большее количество информации?

Взглянем на гистограмму признака *model*.

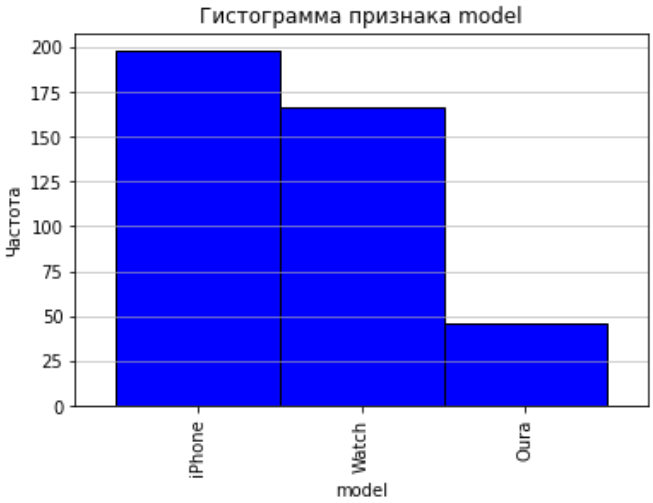

Рисунок 10. График количественного соотношения источников

Судя по гистограмме, наименьшее количество информации поступает с устройства **Oura**.

Чтобы выбрать наиболее информативный источник, необходимо выяснить, с какого из них поступало больше всего информации. Для этого построим график для всех трех устройств, в какие дни с них поступала информация.

Рисунок 11. Поступление информации с устройств по датам

Информации с устройства **Oura** значительно меньше, также срок его использования меньше чем у смартфона и часов. Остается выбрать одно из устройств - **iPhone** или **Watch**.

Судя по графику значений, есть дни, когда использовалось одно из устройств, и есть дни, когда использовались оба. Дней, когда в качестве устройства выступал **iPhone**, больше.

Интересно, что пропуски в датах устройства **iPhone** можно заполнить с помощью **Watch**. Таким образом, получится сохранить больше данных для дальнейшего изучения.

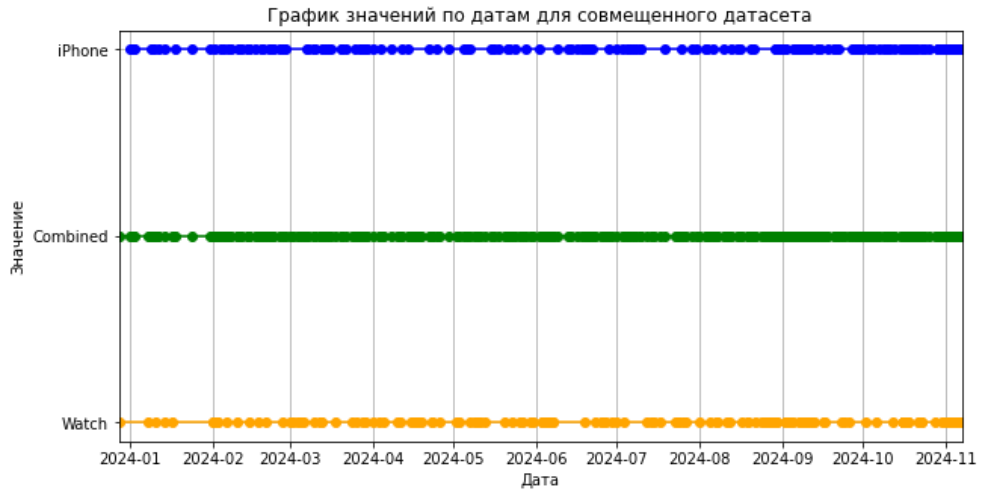

Рисунок 12. Поступление информации с устройств по датам после объединения источников

Пропусков стало значительно меньше.

3. Объединение данных и сбор статистики.

На данном этапе гистограмма признака *value* (количество шагов) выглядит следующим образом:

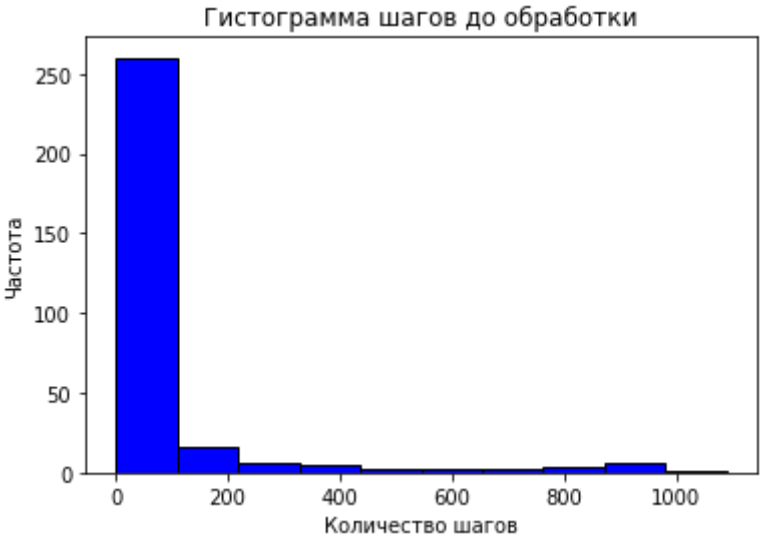

Рисунок 13. Гистограмма признака *value* до обработки

На данный момент она отражает шаги с учетом всех остановок носителя. Нетрудно заметить, что устройства гораздо чаще фиксировали короткие участки активности. Было бы интересно посмотреть, какова длительность (в секундах) данных участков. Для этого был создан новый признак – *moveTime*, который рассчитывался как разность между признаками *datetimeEnd* и *datetimeStart*.

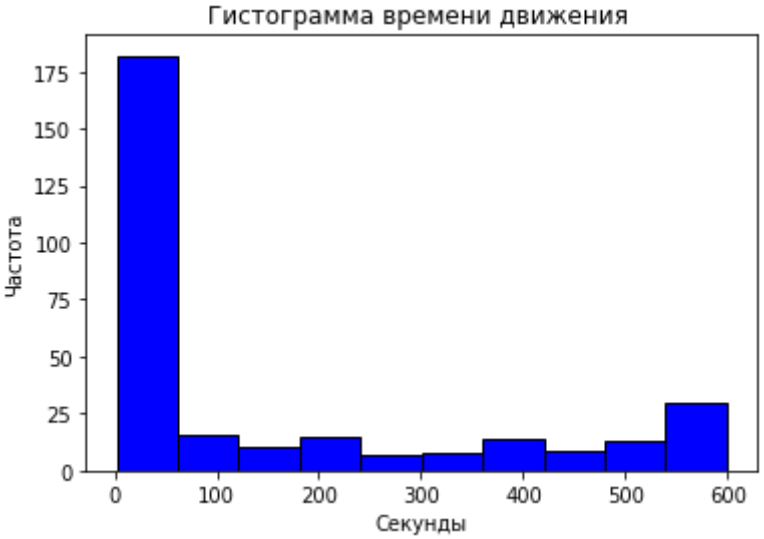

Рисунок 14. Гистограмма признака *moveTime* до обработки

Очень много строк, где время движения меньше минуты.

Будем считать, что остановки до минуты - паузы при ходьбе (ожидание зеленого света на светофоре, к примеру). Объединим те временные промежутки, между которыми разница в секундах меньше 60.

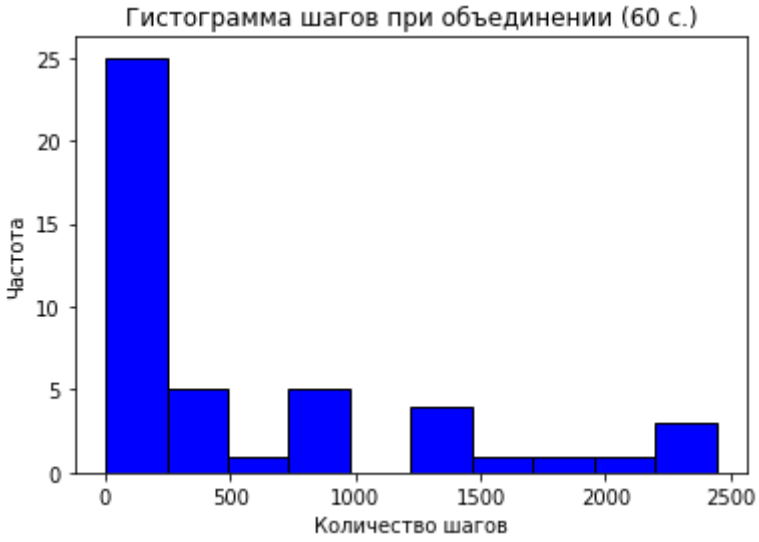

Рисунок 15. Гистограмма признака *value* (60 с.)

Ради интереса посмотрим на аналогичный график, но при разнице между промежутками в 120 секунд.

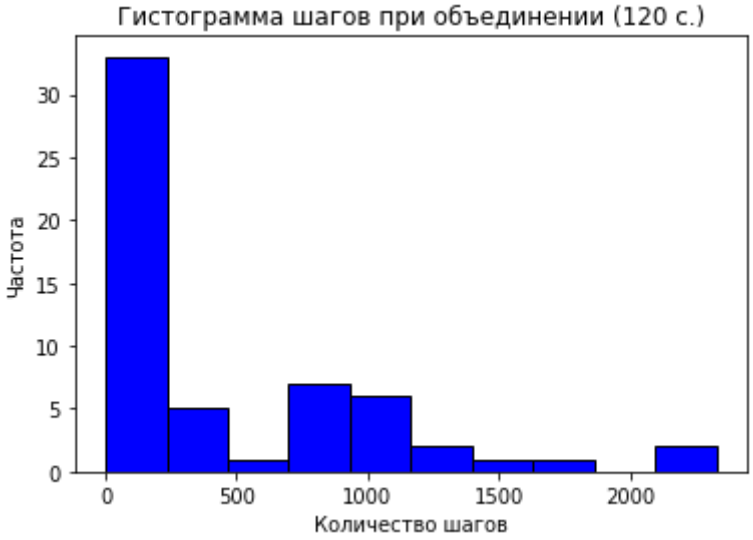

Рисунок 16. Гистограмма признака *value* (120 с.)

Будем считать, что наиболее подходящим интервалом объединения промежутков в один является расстояние между ними не более 60 секунд.

Далее переходим к статистике.

Определим часы активности пользователя.

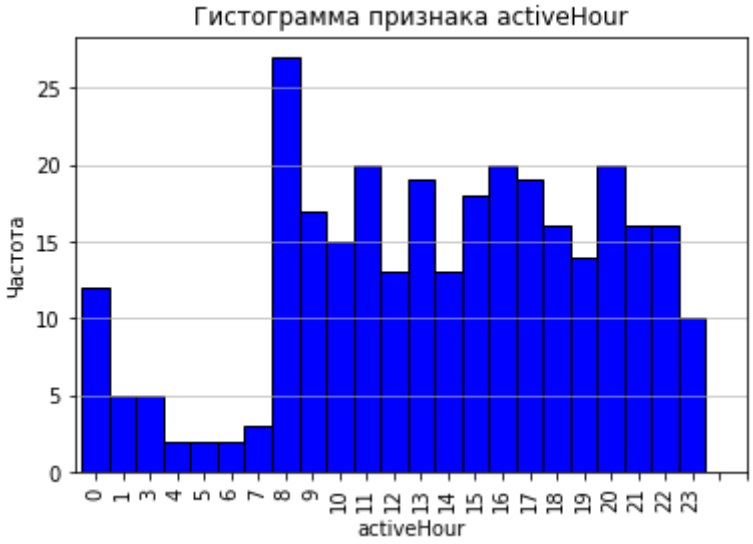

Рисунок 17. Гистограмма признака *activeHour*

Чаще всего пользователь двигается (встает и совершает некоторое количество шагов) в 8 часов утра.

Изучим аналогичный график, но не для количества движений, а для количества шагов во временные промежутки.

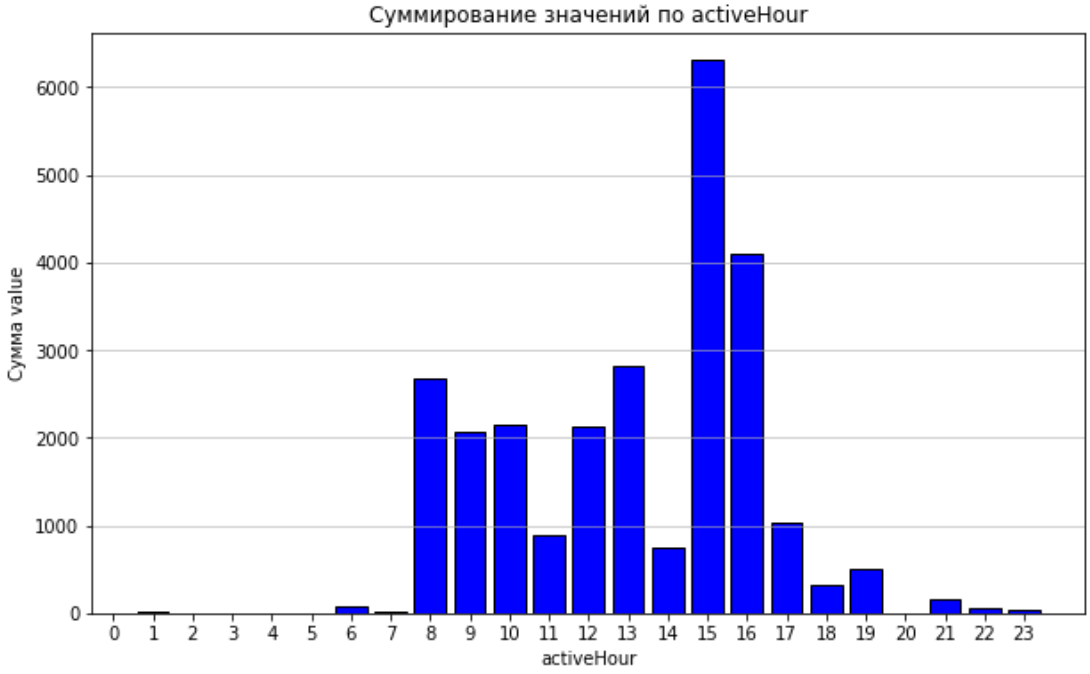

Рисунок 18. Гистограмма движений по признаку *activeHour*

Хотя и движений было значительно больше в ранние часы, гистограмма по шагам показала, что больше всего шагов делается в послеобеденное время.

Теперь посмотрим, в какие дни недели пользователь совершает наибольшее количество подъемов и шагов. Для этого был добавлен признак *weekday* – день недели (где 0 – понедельник, 6 – воскресенье), рассчитанный на основе признака *datetimeStart*.

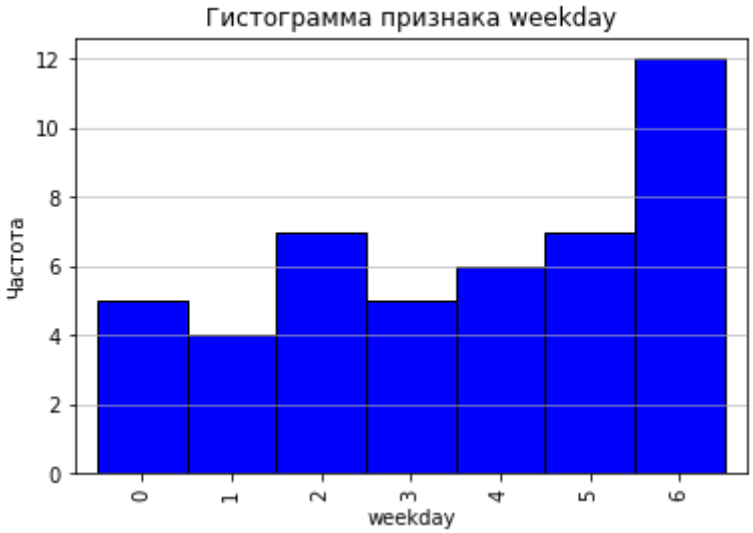

Рисунок 19. Гистограмма признака *weekday*

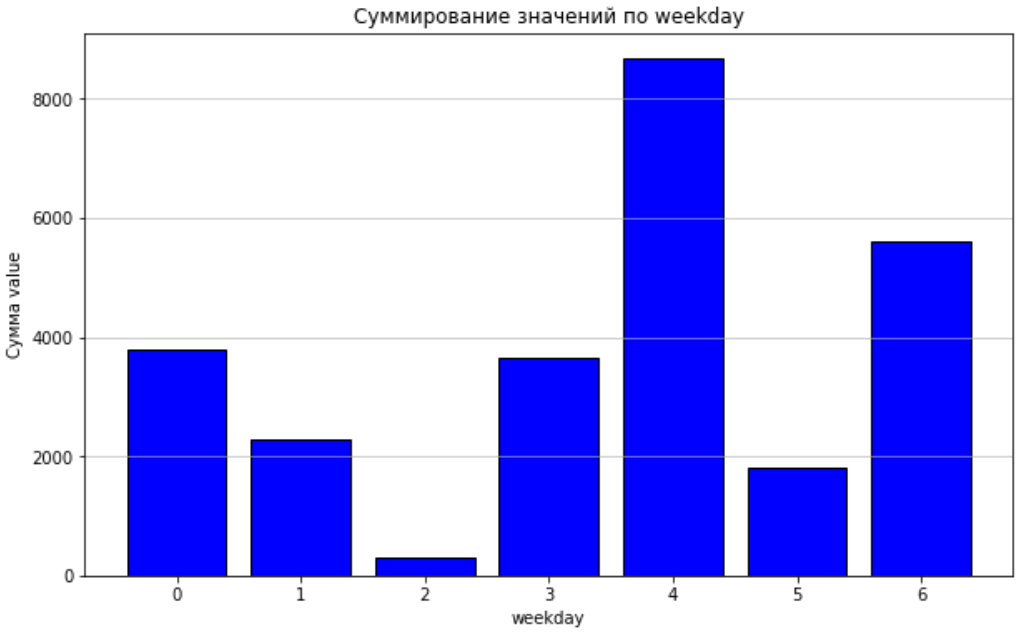

Рисунок 20. Гистограмма шагов по признаку *weekday*

Исходя из графиков, больше всего пользователь совершает шагов по пятницам и воскресеньям. Больше всего подъемов пользователю совершает в воскресенье.