1. 1 ГЛАВА

1.1. Постановка задачи

– Познакомиться с понятием «большие данные» и способами их обработки;
– Познакомиться с инструментом Apache Spark и возможностями, которые он предоставляет для обработки больших данных.
– Получить навыки выполнения разведочного анализа данных использованием pyspark.
– Использовать pyspark.DataFrame, который используется для обработки больших данных

1.2. Описание датасета

Название датасета: Smart meters in London (Умные счетчики в Лондоне)

Ссылка на датасет: https://www.kaggle.com/datasets/jeanmidev/smart-meters-in-london

Описание:

Чтобы лучше следить за потреблением энергии, правительство хочет, чтобы поставщики энергии установили умные счетчики в каждом доме в Англии, Уэльсе и Шотландии. Поставщики энергии могут добраться до более чем 26 миллионов домов, и к 2020 году в каждом доме будет установлен интеллектуальный счетчик.

Это развертывание счетчиков возглавляет Европейский союз, который попросил все правительства-члены рассмотреть интеллектуальные счетчики в рамках мер по модернизации нашего энергоснабжения и борьбе с изменением климата. После первоначального исследования британское правительство

решило внедрить интеллектуальные счетчики в рамках своего плана по обновлению нашей стареющей энергетической системы.

В этом наборе данных вы найдете рефакторизованную версию данных из лондонского хранилища данных, которая содержит показания энергопотребления для выборки из 5 567 лондонских домохозяйств, которые приняли участие в проекте Low Carbon London под руководством UK Power Networks. в период с ноября 2011 г. по февраль 2014 г. Данные интеллектуальных счетчиков, похоже, связаны только с потреблением электроэнергии. Датасет содержит файлы блоков с ежедневной информацией, такой как количество измерений, минимум, максимум, среднее значение, медиана, сумма и стандартное значение.

1.3. Разведочный анализ

Датасет представлен в виде таблицы, представленной на рисунке 1.

LCLid	day	energy_median	energy_mean	energy_max	energy_count	energy_std	energy_sum	energy_min
MAC000131	2011-12-15 0...	0.485	0.4320454545...	0.868	22	0.2391457967...	9.505	0.0720000000...
MAC000131	2011-12-16 0...	0.1415	0.2961666687...	1.1160001	48	0.2814713178...	14.216000100...	0.031
MAC000131	2011-12-17 0...	0.1015	0.1898125	0.685	48	0.1884046862...	9.111	0.064
MAC000131	2011-12-18 0...	0.114	0.2189791666...	0.6759999999...	48	0.2029192785...	10.510999999...	0.065
MAC000131	2011-12-19 0...	0.191	0.3259791666...	0.7879999999...	48	0.2592049619...	15.646999999...	0.066
MAC000131	2011-12-20 0...	0.2180000000...	0.3575	1.077	48	0.2875965702...	17.16	0.066
MAC000131	2011-12-21 0...	0.1305	0.2350833333...	0.705	48	0.2220696491...	11.284	0.066
MAC000131	2011-12-22 0...	0.0890000000...	0.2213541666...	1.094	48	0.2672388754...	10.625	0.062
MAC000131	2011-12-23 0...	0.1604999999...	0.291125	0.7490000000...	48	0.2490760479...	13.973999999...	0.065
MAC000131	2011-12-24 0...	0.107	0.1689999999...	0.613	47	0.1506846693...	7.943	0.065
MAC000131	2011-12-25 0...	0.2175	0.3391875000...	0.866	48	0.2631011985...	16.281000000...	0.069
MAC000131	2011-12-26 0...	0.1495000000...	0.2617083333...	0.838	48	0.2447927441...	12.562000000...	0.066
MAC000131	2011-12-27 0...	0.1430000000...	0.2740000000...	0.778	48	0.2521274584...	13.152000000...	0.068
MAC000131	2011-12-28 0...	0.1455000000...	0.3005208333...	1.207	48	0.2986802880...	14.425	0.066
MAC000131	2011-12-29 0...	0.152	0.3070416666...	0.888	48	0.2644546341...	14.738	0.066
MAC000131	2011-12-30 0...	0.135	0.2768541666...	0.782	48	0.2611857568...	13.289000000...	0.064
MAC000131	2011-12-31 0...	0.1515	0.3257291666...	1.252	48	0.3098882941...	15.635000000...	0.066
MAC000131	2012-01-01 0...	0.151	0.2560208333...	0.812	48	0.2252494116...	12.289	0.068
MAC000131	2012-01-02 0...	0.134	0.2520833333...	0.851	48	0.2372129695...	12.1	0.068
MAC000131	2012-01-03 0...	0.1475000000...	0.2355000000...	0.674	48	0.2099953393...	11.304000000...	0.068

Рисунок 1 – Представление датасета

В данном датасете большую часть типов данных представлена типом double/число с плавающей точкой, что продемонстрировано на рисунке 2. Но также, присутствуют такие типы, как: integer, string и timestamp.

```
Обзор данных
root
 |-- LCLid: string (nullable = true)
 |-- day: timestamp (nullable = true)
 |-- energy_median: double (nullable = true)
 |-- energy_mean: double (nullable = true)
 |-- energy_max: double (nullable = true)
 |-- energy_count: integer (nullable = true)
 |-- energy_std: double (nullable = true)
 |-- energy_sum: double (nullable = true)
 |-- energy_min: double (nullable = true)
```

Рисунок 2 – Обзор данных датасета

Как можно заметить, от столбцов LCLid и day можно избавиться, так как в первом хранятся id номера, а во втором временные метки. Что мы и сделали, это показано на рисунке 3.

energy_median	energy_mean	energy_max	energy_count	energy_std	energy_sum	energy_min
0.485	0.43204545454545457	0.868	22	0.23914579678767536	9.505	0.07200000000000001
0.1415	0.29616666875000003	1.1160001	48	0.2814713178628203	14.216000100000002	0.031
0.1015	0.1898125	0.685	48	0.1884046862418033	9.111	0.064
0.114	0.21897916666666666	0.6759999999999999	48	0.20291927853038208	10.510999999999996	0.065
0.191	0.32597916666666665	0.7879999999999999	48	0.2592049619947409	15.646999999999998	0.066
0.21800000000000005	0.3575	1.077	48	0.28759657027517305	17.16	0.066
0.1305	0.23508333333333333	0.705	48	0.2220696491599295	11.284	0.066
0.08900000000000001	0.22135416666666666	1.094	48	0.26723887549908265	10.625	0.062
0.16049999999999998	0.291125	0.7490000000000001	48	0.24907604794434665	13.973999999999998	0.065
0.107	0.16899999999999998	0.613	47	0.15068466931050878	7.943	0.065
0.2175	0.33918750000000003	0.866	48	0.26310119857478675	16.281000000000002	0.069
0.14950000000000002	0.2617083333333333	0.838	48	0.2447927441503373	12.562000000000001	0.066
0.14300000000000002	0.2740000000000001	0.778	48	0.25212745847913703	13.152000000000005	0.068
0.14550000000000002	0.3005208333333333	1.207	48	0.29868028801773083	14.425	0.066
0.152	0.30704166666666667	0.888	48	0.2644546341928976	14.738	0.066
0.135	0.27685416666666673	0.782	48	0.261185756802965	13.289000000000005	0.064
0.1515	0.32572916666666674	1.252	48	0.3098882941898363	15.635000000000005	0.066
0.151	0.25602083333333336	0.812	48	0.2252494116065079	12.289	0.068
0.134	0.2520833333333333	0.851	48	0.23721296951853504	12.1	0.068
0.14750000000000002	0.23550000000000004	0.674	48	0.2099953393606427	11.304000000000002	0.068

Рисунок 3 – Датасет без столбцов LCLid и day

Далее мы приступили к нахождению пропущенных значений. Как правило, пропущенные значения заносятся в таблицу, как None, NaN или NULL. При поиске было обнаружено более 11 тыс. пропусков, как можно заметить на рисунке 4. Так как, существует несколько методов по устранению пропущенных значений, например, заполнение пропущенных значений медианным или средним значением, заполнение при помощи линейной регрессии или заполнение на основе соседних клеток. Но самым простым является удаление строк с пропущенным значениями. Так как в нашем

датасете представлено около 3,5 млн строчек, то удаление 11 тыс., т.е. менее 1%, не повлияет на достоверность будущих предсказаний и не увеличит погрешность. На рисунке 5 продемонстрировано, что пропущенные значения были устранены.

Кол-во пропущенных значений в датафрейме

energy_median	energy_mean	energy_max	energy_count	energy_std	energy_sum	energy_min
30	30	30	0	11331	30	30

Рисунок 4 – Кол-во пропущенных значений по столбцам

Кол-во пропущенных значений в датафрейме, после удаления

energy_median	energy_mean	energy_max	energy_count	energy_std	energy_sum	energy_min
0	0	0	0	0	0	0

Рисунок 5 – Демонстрация устраненных пропущенных значений

Далее было решено начать поиск выбросов. Выброс – это наблюдение, удаленное от других в выборке. Другими словами, это Наблюдение, которое расходится с общей закономерностью Выборки. Выбросы могут появляться из-за некорректно собранных данных, новых процессов или различных методов сборки данных. Также, стоит различать выбросы с несбалансированным датасетом. Хоть в определениях и есть некоторые сходства, однако несбалансированный набор данных с точки зрения Машинного обучения – это меньший размер выборки одного класса в сравнении с другим. Мы решили начать с количественного отображения выбросов, что показано на рисунке 6. Также, выбросы продемонстрированы на “Ящиках с усами” (BoxPlot) на рисунке 7. Все что выше максимума или ниже минимума является выбросами. Так, как чаще всего выбросы устраняются удалением, то мы прибегли именно к этому методу. Проверку того, что выбросы успешно удалены, можно заметить на рисунке 8. А также, на рисунке 9, где продемонстрированы графики с данными без выбросов.

Кол-во выбросов по столбцам:

energy_median_out	energy_mean_out	energy_max_out	energy_count_out	energy_std_out	energy_sum_out	energy_min_out
224973	201317	121012	29750	158187	201339	221460

Рисунок 6 – Кол-во выбросов по столбцам

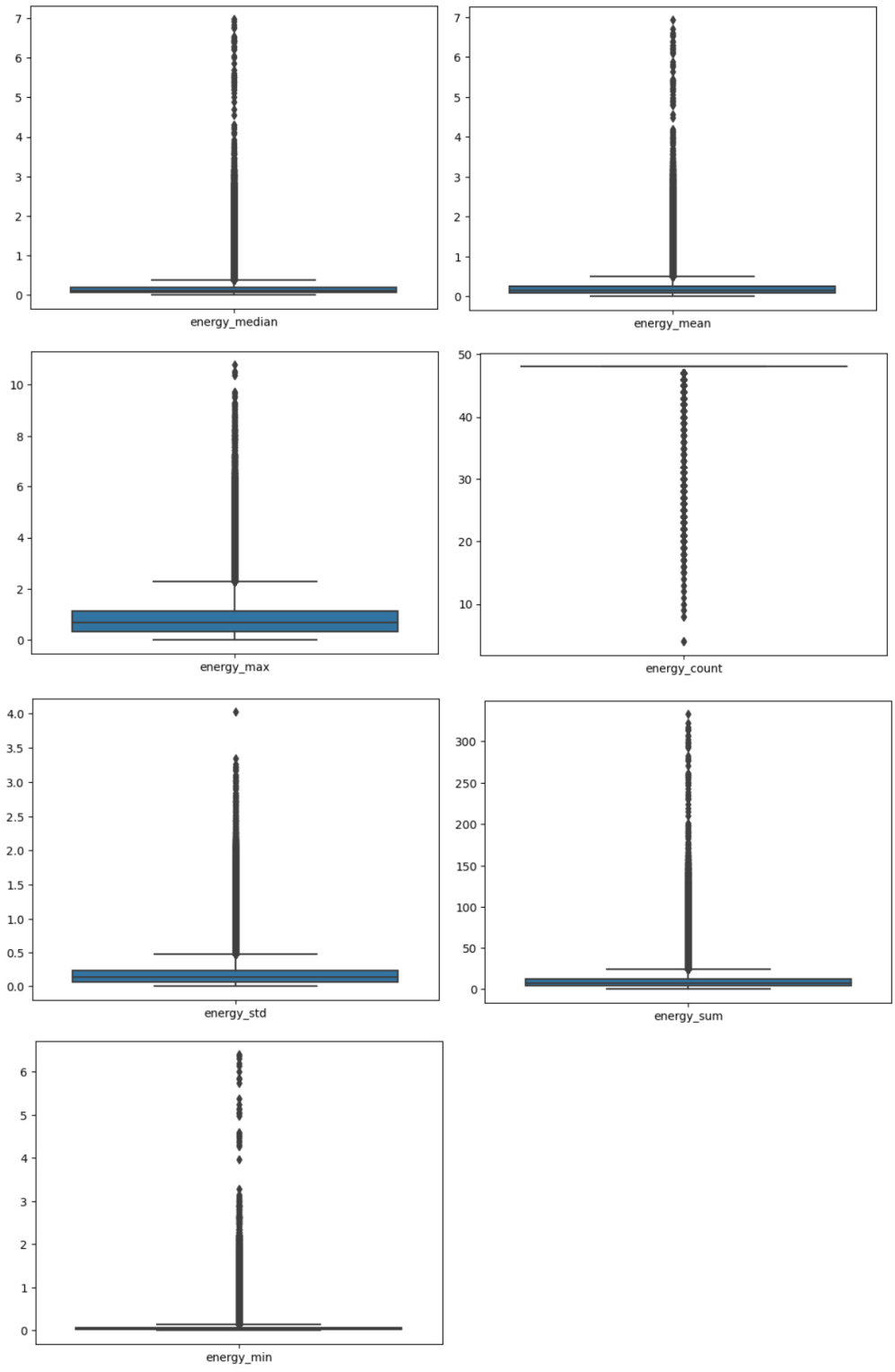

Рисунок 7 – Графики для визуального наблюдения выбросов

Кол-во выбросов по столбцам, после удаления выбросов:

energy_median_out	energy_mean_out	energy_max_out	energy_count_out	energy_std_out	energy_sum_out	energy_min_out
0	0	0	0	0	0	0

Рисунок 8 – Численное отображение удаления выбросов

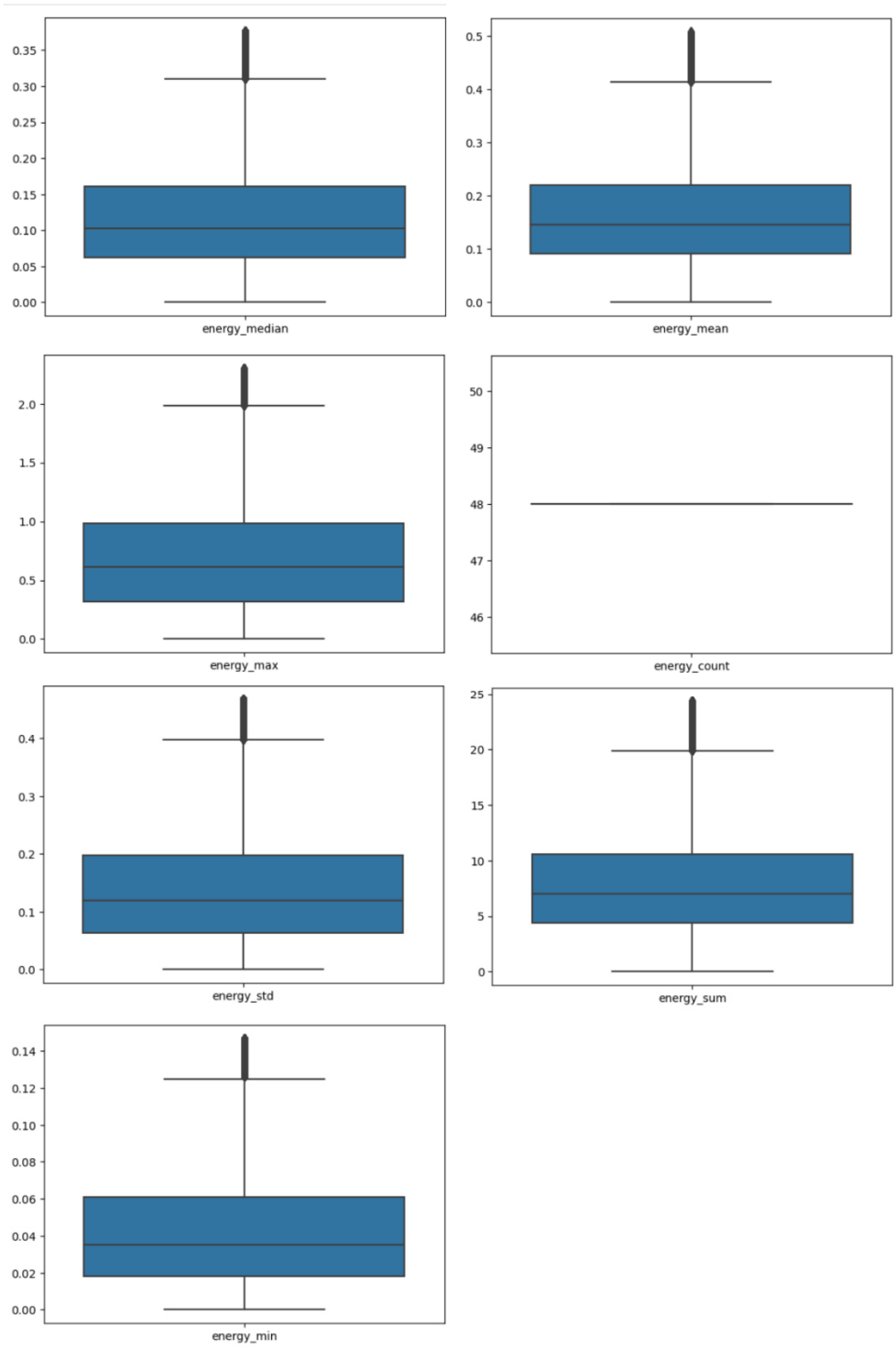

Рисунок 9 – Графики для визуального наблюдения отсутствия выбросов

После всех преобразований в датасете, было решено визуализировать статистические данные в них входят: среднее, минимальное, максимальное, стандартное отклонение и квантили, это показано на рисунке 10. Можно заметить по столбцу count, что после обработки данных, осталось еще чуть больше 3 млн. строк, что означает, что было удалено около 14% процента данных, в основном эти проценты составляют выбросы.

	count	mean	std	min	25%	50%	75%	max
energy_median	3053252.0	0.118768	0.074528	0.0	0.062000	0.103000	0.161000	0.377000
energy_mean	3053252.0	0.163161	0.094318	0.0	0.090979	0.146500	0.220083	0.508542
energy_max	3053252.0	0.695165	0.465291	0.0	0.316000	0.612000	0.983000	2.303000
energy_std	3053252.0	0.140305	0.097866	0.0	0.063570	0.118744	0.197008	0.469602
energy_sum	3053252.0	7.831735	4.527253	0.0	4.367000	7.032000	10.564000	24.410000
energy_min	3053252.0	0.042886	0.032184	0.0	0.018000	0.035000	0.061000	0.147000

Рисунок 10 – Визуализация статистических данных

Далее была построена матрица корреляция на основе датасета и показана на рисунке 11. В ней можно заметить, что признаки столбца energy_count не коррелируют ни с одним другим признаком других столбцов, то есть столбец содержит в себе одинаковые признаки и никак не меняется, что и доказывает то, почему его признаки не коррелируют с другими признаками. Т.е. данный столбец может быть удален. Также, есть признаки, которые коррелируют с процентом выше 80%, то есть являются сильно коррелирующими.

energy_median	energy_mean	energy_max	energy_count	energy_std	energy_sum	energy_min
1.0	0.8862012883618461	0.4941876529868468	NaN	0.5149833234916236	0.8862012883618424	0.6829627699218158
0.8862012883618461	1.0	0.7476251232287053	NaN	0.8145762056794023	1.0000000000001228	0.6214967136610641
0.4941876529868468	0.7476251232287053	1.0	NaN	0.9454504347867345	0.747625123228614	0.28539156332222265
NaN	NaN	NaN	1.0	NaN	NaN	NaN
0.5149833234916236	0.8145762056794023	0.9454504347867345	NaN	1.0	0.8145762056793269	0.2438036109341937
0.8862012883618424	1.0000000000001228	0.747625123228614	NaN	0.8145762056793269	1.0	0.6214967136610219
0.6829627699218158	0.6214967136610641	0.28539156332222265	NaN	0.2438036109341937	0.6214967136610219	1.0

Рисунок 11 – Матрица корреляций

И последним действием мы решили визуализировать данные на графиках распределения. Они показаны на рисунке 12 в виде гистограмм. Можно заметить, что все графики имеют вид нормального распределения.

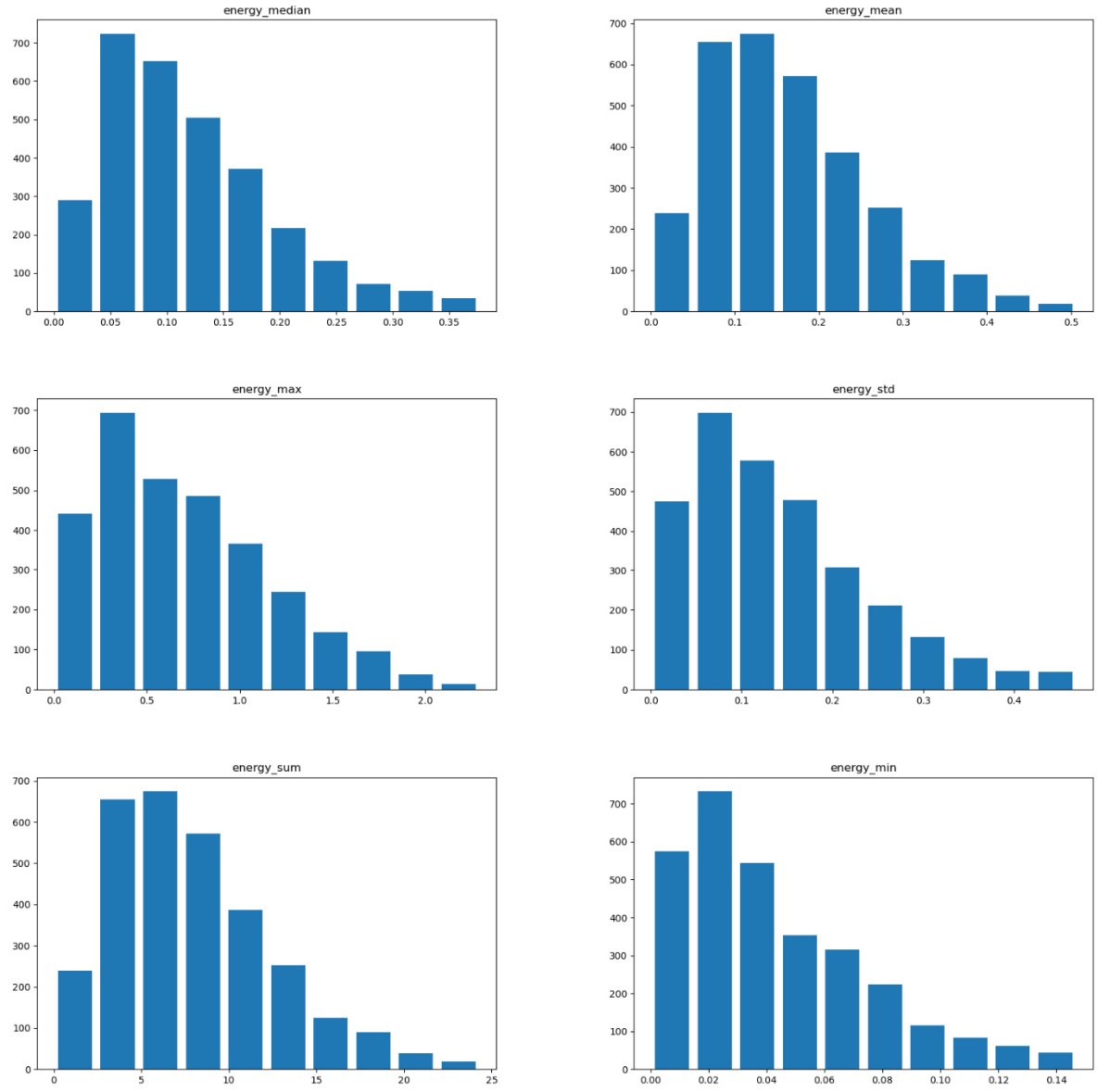

Рисунок 12 – Графики распределения

1.4. Выводы

В данной главе мы работали с инструментом Apache Spark, использовав spark.DataFrame, который предназначен для работы с большим количеством данных. Также, нашли подходящий датасет и научились проводить разведочный анализ данных.

2. 2 ГЛАВА
3. 3 ГЛАВА