МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**по курсу**

«Data Science»

|  |  |
| --- | --- |
| Слушатель | Олейник Владимир Александрович |

Москва, 2023

Содержание

[Введение 3](#_Toc133098760)

[1 Аналитическая часть 4](#_Toc133098761)

[1.1 Постановка задачи 4](#_Toc133098762)

[1.2 Описание используемых методов 5](#_Toc133098763)

[1.2.1. Линейная регрессия методом наименьших квадратов 6](#_Toc133098764)

[1.2.2. Дерево принятия решений 7](#_Toc133098765)

[1.2.3. Метод случайного леса 8](#_Toc133098766)

[1.2.4. Метод опорных векторов 9](#_Toc133098767)

[1.2.5. Метод k-ближайших соседей 10](#_Toc133098768)

[1.2.6. Нейронная сеть 10](#_Toc133098769)

[1.3 Разведочный анализ данных 12](#_Toc133098770)

[2 Практическая часть 19](#_Toc133098771)

[2.1 Предобработка данных 19](#_Toc133098772)

[2.2 Выбор стратегии 23](#_Toc133098773)

[2.3 Разработка и обучение моделей 25](#_Toc133098774)

[2.4 Оценка моделей 25](#_Toc133098775)

[2.5 Разработка и обучение нейронной сети 27](#_Toc133098776)

[2.6 Разработка приложения 30](#_Toc133098777)

[Заключение 32](#_Toc133098778)

[Библиографический список 33](#_Toc133098779)

Введение

Data Science (наука о данных) – раздел [информатики](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0), изучающий проблемы [анализа](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), [обработки](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85&action=edit&redlink=1) и представления [данных](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5) в цифровой форме. Объединяет методы по обработке данных в условиях больших объёмов и высокого уровня параллелизма, [статистические методы](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0), методы [интеллектуального анализа данных](https://ru.wikipedia.org/wiki/Data_mining) и приложения [искусственного интеллекта](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82) для работы с данными, а также методы проектирования и разработки [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

Основная практическая цель профессиональной деятельности в науке о данных - обнаружение закономерностей в данных, извлечение знаний из данных в обобщённой форме для дальнейшего использования в различных целях.

Результаты, полученные с помощью науки о данных, используются для решения задач в разных отраслях. Одной из таких задач, рассмотренной в данной работе, является прогнозирование конечных свойств новых материалов (композиционных материалов).

Композиционный материал - многокомпонентный материал, изготовленный (человеком или природой) из двух или более компонентов с существенно различными свойствами, которые, в сочетании, приводят к появлению нового материала с характеристиками, отличными от характеристик отдельных компонентов и не являющимися простой их [суперпозицией](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%86%D0%B8%D0%BF_%D1%81%D1%83%D0%BF%D0%B5%D1%80%D0%BF%D0%BE%D0%B7%D0%B8%D1%86%D0%B8%D0%B8). В составе композита принято выделять матрицу и наполнитель, который выполняет функцию армирования. Варьируя состав матрицы и наполнителя, их соотношение, ориентацию наполнителя, получают широкий спектр материалов с требуемым набором свойств.

При изготовлении композиционных материалов имеется недостаток - при наличии, сведений об исходных характеристиках компонентов, достаточно проблематично определить конечные характеристики нового материала, состоящего из этих компонентов. Для решения этой проблемы есть два пути: физические испытания образцов материалов, или прогнозирование характеристик на основе данных о характеристиках входящих компонентов.

# Аналитическая часть

## Постановка задачи

Для решения задачи прогнозирования конечных свойств новых материалов предоставлены два файла Microsoft Excel, содержащие данные о некоторых исходных характеристиках компонентов и конечных характеристиках композиционного материала.

Краткое описание файлов:

1. X\_bp, состоящий из 11 столбцов и 1024 строк, включая столбец индекса и строки наименования столбцов;
2. X\_nup, состоящий из 4 столбцов и 1041 строк, включая столбец индекса и строки наименования столбцов.

Требуется:

1. обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении композиционных материалов;
2. написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель.

Созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

В результате объединения файлов по индексу с использованием INNER JOIN (внутреннее соединение - объединяются только строки, имеющие одинаковый индекс) получен датасет размерностью 1023 строки и 13 колонок (признаков).

Пропуски в данных отсутствуют. Все признаки имеют числовой тип данных. Исходя из задачи, признаки разделены на десять входных и три выходных переменных. Сводная информация о датасете указана в таблице 1.

Таблица 1 – сводная информация о датасете

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование колонки (признака) | Тип переменной | Количество непустых  значений | Тип данных | Количество уникальных значений |
| Соотношение матрица-наполнитель | выходной | 1023 | float64 | 1014 |
| Плотность, кг/м3 | входной | 1023 | float64 | 1013 |
| Модуль упругости, ГПа | входной | 1023 | float64 | 1020 |
| Количество отвердителя, м.% | входной | 1023 | float64 | 1005 |
| Содержание эпоксидных групп,%\_2 | входной | 1023 | float64 | 1004 |
| Температура вспышки, С\_2 | входной | 1023 | float64 | 1003 |
| Поверхностная плотность, г/м2 | входной | 1023 | float64 | 1004 |
| Модуль упругости при растяжении, ГПа | выходной | 1023 | float64 | 1004 |
| Прочность при растяжении, МПа | выходной | 1023 | float64 | 1004 |
| Потребление смолы, г/м2 | входной | 1023 | float64 | 1003 |
| Угол нашивки, град | входной | 1023 | int64 | 2 |
| Шаг нашивки | входной | 1023 | float64 | 989 |
| Плотность нашивки | входной | 1023 | float64 | 988 |

## Описание используемых методов

Предсказание численной характеристики объекта предметной области по определенному набору признаков – это задача регрессии, которая относится к категории задач обучения с учителем.

Регрессионная модель – это уравнение, в котором объясняемая переменная (целевой признак) представляется в виде функций от объясняющих переменных признаков.

Для решения поставленной задачи будут рассмотрены следующие методы построения регрессионной модели:

линейная регрессия методом наименьших квадратов;

дерево принятия решений;

метод случайного леса;

метод опорных векторов;

метод k-ближайших соседей;

нейронная сеть.

### Линейная регрессия методом наименьших квадратов

Линейная регрессия - это регрессионная модель зависимости одной (целевой) переменной от другой или нескольких других переменных (признаков) с линейной функцией зависимости, которая, согласно определённым математическим критериям, наиболее соответствует данным.

При наличии одной входной переменной линейная регрессия называется простой, когда входных переменных несколько – множественной.

Из множества методов обучения линейных регрессионных моделей наиболее распространенным является метод наименьших квадратов. С помощью данного метода вычисляется прямая (гиперплоскость), сумма квадратов между которой и данными минимальна.

Линейная регрессия является линейной моделью, которая предполагает линейную связь между входными переменными и выходной переменной. Эта связь измеряется путем выявления изменений выходной переменной при изменении входных переменных.

Достоинства метода:

простота реализации;

эффективность при линейных зависимостях;

простота интерпретации.

Недостатки метода:

ограниченная эффективность при нелинейных зависимостях;

чувствительность к выбросам.

Линейная регрессия наименьших квадратов реализована в библиотеке scikit-learn - sklearn.linear\_model. LinearRegression.

### Дерево принятия решений

Дерево принятия решений - метод автоматического анализа больших массивов данных. Дерево решений представляет собой иерархическую древовидную структуру. Структура дерева представляет собой листья и ветки. На рёбрах (ветках) дерева решения записаны признаки, от которых зависит целевая переменная, в листьях записаны значения целевой переменной, а в остальных узлах - признаки, по которым различаются случаи. Каждый лист представляет собой значение целевой переменной, изменённой в ходе движения от корня по рёбрам дерева до листа. Каждый внутренний узел сопоставляется с одним из входных признаков.

Основная задача при построении дерева решений - последовательно и рекурсивно разбить обучающее множество на подмножества с применением решающих правил в узлах.

В основе построения лежат «жадные» алгоритмы, допускающие локально-оптимальные решения на каждом шаге разбиения в узлах, которые приводят к оптимальному итоговому решению. При выборе одного признака и произведении разбиения по нему на подмножества, алгоритм не может вернуться назад и выбрать другой признак.

Разбиение должно осуществляться по определенному правилу, для которого выбирают признак. Критериев выбора признака существует много, но наибольшей популярностью пользуется информационная энтропия.

Энтропия рассматривается, как мера неоднородности подмножества. Если выбранный признак разбиения обеспечивает максимальное снижение энтропии результирующего подмножества относительно родительского, его можно считать наилучшим. В таком случае лучшим признаком будет тот, который обеспечит максимальный прирост информации результирующего узла относительно исходного

Достоинства метода:

простота интерпретации;

не требует специальной подготовки данных;

способен работать с разными типами входных признаков.

Недостатки метода:

проблема получения оптимального дерева решений;

возможно переобучение.

Дерево решений для задач регрессии реализовано в библиотеке scikit-learn - sklearn.tree.DecisionTreeRegressor.

### Метод случайного леса

Метод случайного леса - алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга и метод случайных подпространств

Все деревья строятся независимо по следующей схеме:

1. выбирается подвыборка обучающей выборки - по ней строится дерево (для каждого дерева - своя подвыборка);
2. выбираются наилучшие признаки для построения расщепление в дереве, Признаки выбираются из некоторого случайного подмножества признаков (для каждого нового расщепления — свои случайные признаки).

В задаче регрессии результат выводится путём усреднения прогнозов, полученных от каждого дерева. Чем больше деревьев, тем лучше качество, но время настройки и работы также пропорционально увеличиваются.

Достоинства метода:

способность эффективно обрабатывать данные с большим числом [признаков](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%B7%D0%BD%D0%B0%D0%BA_(%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5));

нечувствительность к масштабированию значений признаков.

Недостатки метода:

большой размер получающихся моделей;

сложность интерпретации;

требует больше вычислительных ресурсов.

Случайный лес для задач регрессии реализован в библиотеки scikit-learn - sklearn.ensemble.RandomForestRegressor.

### Метод опорных векторов

Метод опорных векторов - это набор контролируемых методов обучения, используемых для классификации, регрессии и обнаружения выбросов.

Метод опорных векторов конструирует гиперплоскость или набор гиперплоскостей в пространстве большой или бесконечной размерности, которые можно использовать для классификации, регрессии или других задач.

В основе метода опорных векторов для задач регрессии лежит поиск гиперплоскости, при которой риск в многомерном пространстве будет минимальным. Регрессии опорных векторов оценивает коэффициенты путем минимизации квадратичных потерь. Так, если прогнозное значение попадает в область гиперплоскости, то потери равны нулю. В противном случае разности прогнозного и фактического значений

Достоинства метода:

эффективен в пространствах больших размеров;

эффективен в случаях, когда количество измерений больше, чем количество выборок;

эффективен с точки зрения использования памяти, потому использует подмножество обучающих точек в функции принятия решений.

Недостатки метода:

чувствительность к выбросам;

при большом объеме данных увеличивается время обучения и требует больше вычислительных ресурсов.

Метод опорных векторов для задач регрессии реализован в библиотеке scikit-learn - sklearn.svm.LinearSVR.

### Метод k-ближайших соседей

Метод k-ближайших соседей - метрический алгоритм. В случае использования метода для регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны.

Регрессия на основе данного метода может использоваться в случаях, когда метки данных являются непрерывными, а не дискретными переменными.

Алгоритм может быть применим к выборкам с большим количеством признаков (многомерным). Задача поиска ближайшего соседа заключается в отыскании среди множества объектов, расположенных в многомерном метрическом пространстве, объектов близких к заданному, согласно некоторой функции расстояния (близости). классический вариант такой функции - [евклидова метрика](https://ru.wikipedia.org/wiki/%D0%95%D0%B2%D0%BA%D0%BB%D0%B8%D0%B4%D0%BE%D0%B2%D0%B0_%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D0%BA%D0%B0)

Алгоритм выполнения алгоритма k-ближайших соседей состоит в:

1. определении параметра k - число ближайших соседей
2. расчет расстояния;
3. сортировка расстояний и определение ближайшего соседа.

Преимущества метода:

простота интерпретации.

Недостатки метода:

по мере увеличения объема данных алгоритм замедляется и требует больше вычислительных ресурсов;

чувствительность к выбросам;

сложность определения оптимального параметра k.

Метод k-ближайших соседей для задач регрессии реализован в библиотеке scikit-learn - sklearn.neighbors.KNeighborsRegressor.

### Нейронная сеть

Нейронная сеть (искусственная нейронная сеть) - математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей - сетей нервных клеток живого организма.

Искусственная нейронная сеть представляет собой систему соединенных и взаимодействующих между собой простых процессоров (искусственных нейронов).

Основными элементами структуры нейронной сети являются:

1. искусственные нейроны, представляющие собой элементарные, связанные между собой единицы;
2. синапс - это соединение, которые используется для отправки - получения информации между нейронами;
3. сигнал - информация, подлежащая передаче.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения - одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами.

Преимущества нейронных сетей:

адаптивность и эффективность;

в процессе обучения способны выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение;

могут аппроксимировать непрерывные функции.

Недостатки нейронных сетей:

возможно переобучение;

результат работы зависит от выбора исходных данных для обучения;

выбор топологии сети, подбор характеристик сети и параметров обучения;

требует больше вычислительных ресурсов;

в процессе обучения могут проявиться проблемы паралича или попадания сети в локальный минимум поверхности ошибок.

## Разведочный анализ данных

Разведочный анализ данных - предварительное исследование датасета с целью определения его основных характеристик, взаимосвязей между признаками.

Описание колонок (признаков) датасета:

* «Соотношение матрица-наполнитель» - характеристика композита;
* «Плотность, кг/м3» - характеристика матрицы;
* «Модуль упругости, ГПа» - характеристика матрицы;
* «Количество отвердителя, м.%» - характеристика матрицы;
* «Содержание эпоксидных групп,%\_2» - характеристика матрицы;
* «Температура вспышки, С\_2» - характеристика матрицы;
* «Поверхностная плотность, г/м2» - характеристика матрицы;
* «Модуль упругости при растяжении, ГПа» - характеристика композита;
* «Прочность при растяжении, МПа» - характеристика композита;
* «Потребление смолы, г/м2» - характеристика наполнителя;
* «Угол нашивки, град» - характеристика наполнителя;
* «Шаг нашивки» - характеристика наполнителя;
* «Плотность нашивки» - характеристика наполнителя.

Дублирующие записи искажают статистические данные датасета и снижают качество обучения модели, потому необходимо удалить такие строки. После удаления дублирующих записей с помощью pandas.drop\_duplicates(), размерность датасета не изменилась, значит дубликаты отсутствовали.

Проверим датасет на пропуски pandas.DataFrame.isnull().any() и уникальные значения pandas.DataFrame.nunique(). Пропуски отсутствуют. Признак «Угол нашивки, град» имеет только два уникальных значения, поэтому в дальнейшем изменим тип данных на категориальный.

Выведем описательные статистические данные, воспользовавшись pandas.DataFrame.describe(), добавив дополнительно информацию о медиане и моде. Полученные данные отображены на рисунке 1.

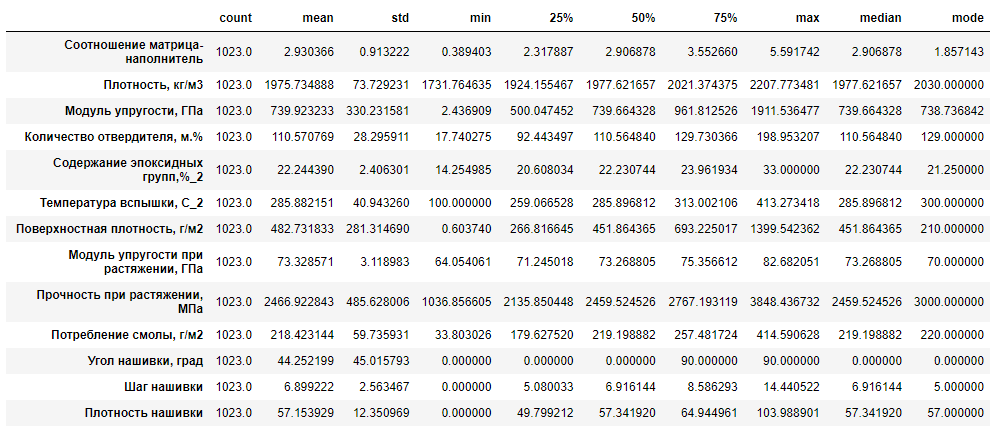


Рисунок 1 - Описательные статистические данные

Построим и визуализируем корреляционную матрицу (рисунок 2).

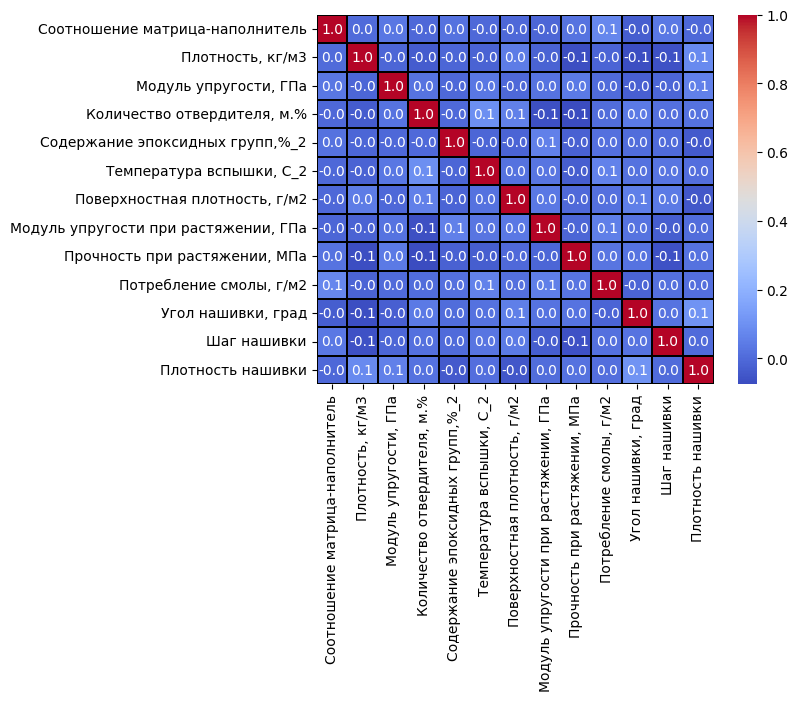


Рисунок 2 - Матрица корреляции

Выведем графики распределения, попарные графики рассеяния точек (рисунок 3).

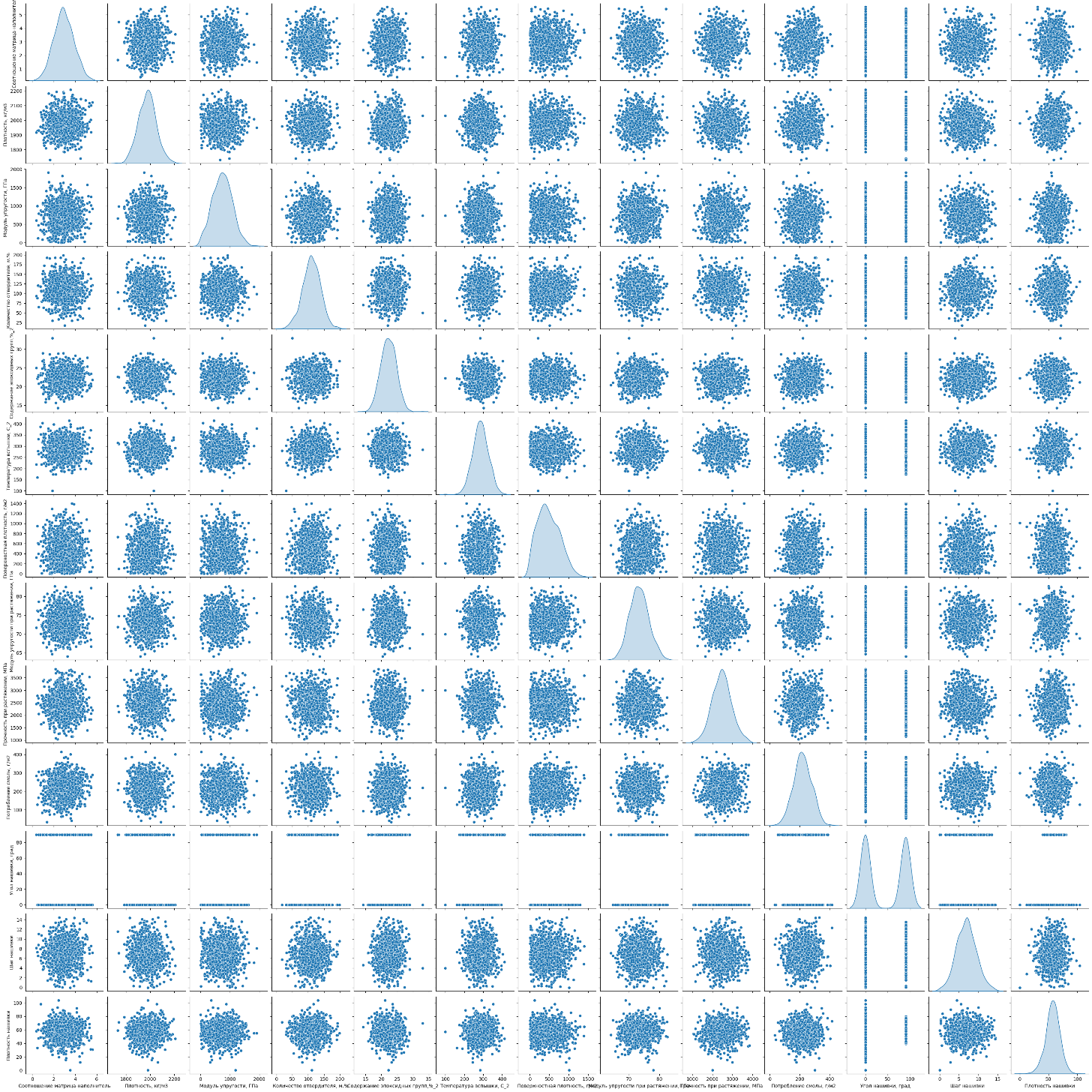


Рисунок 3 - Попарные графики рассеяния точек

С помощью пользовательской функции выведем графики распределения, диаграммы «ящик с усами» (boxplot) и график «квантиль – квантиль» (Q-Q plot) для каждого признака (рисунок 4). На графиках распределения дополнительно отобразим среднее значение, медиану, моду.

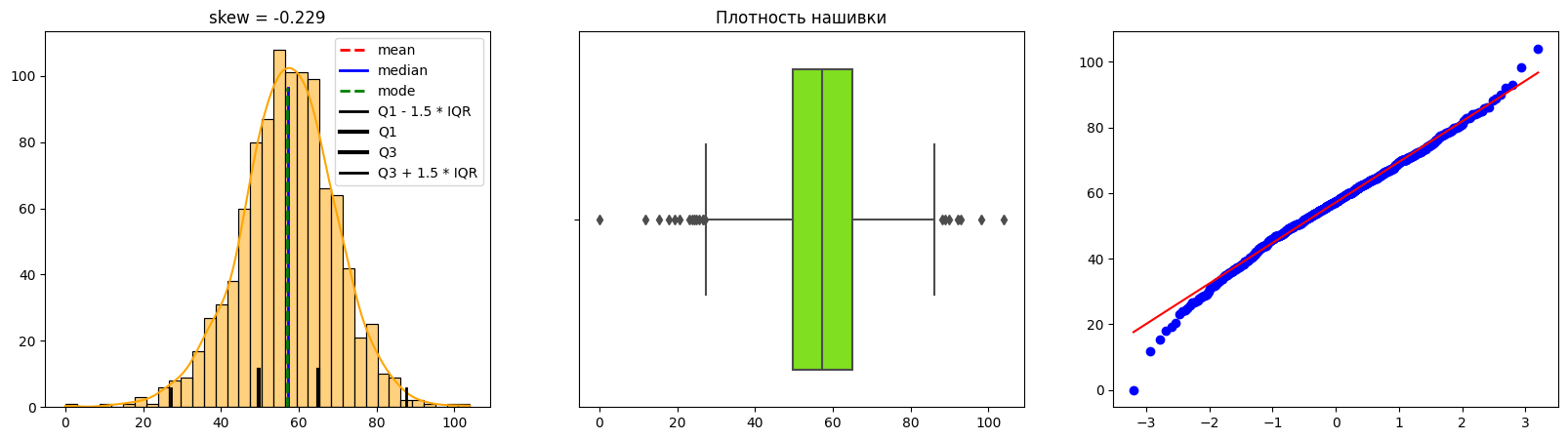
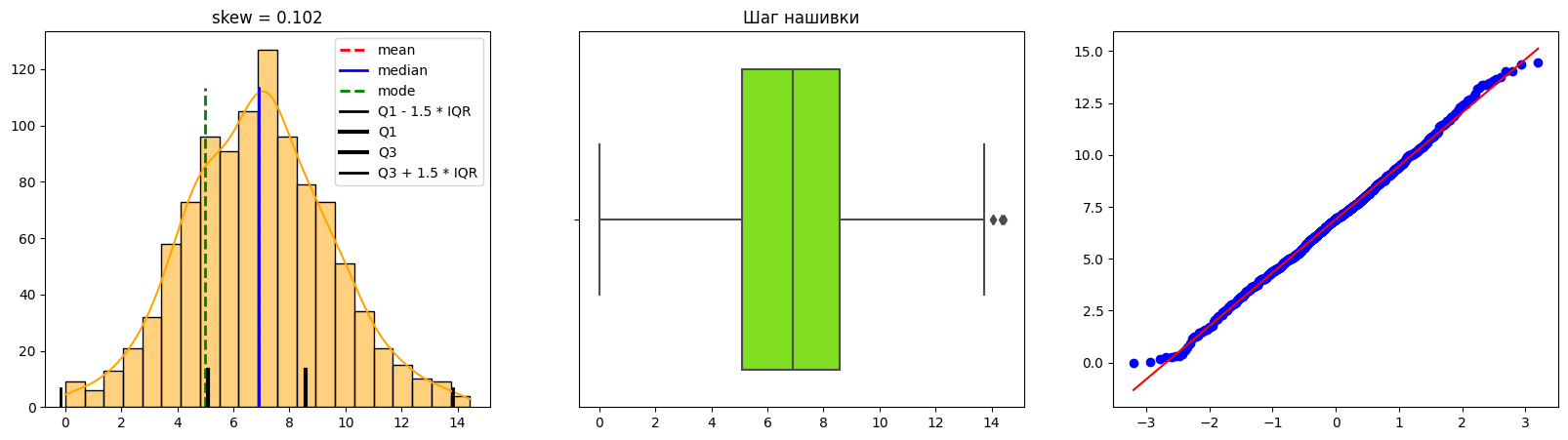
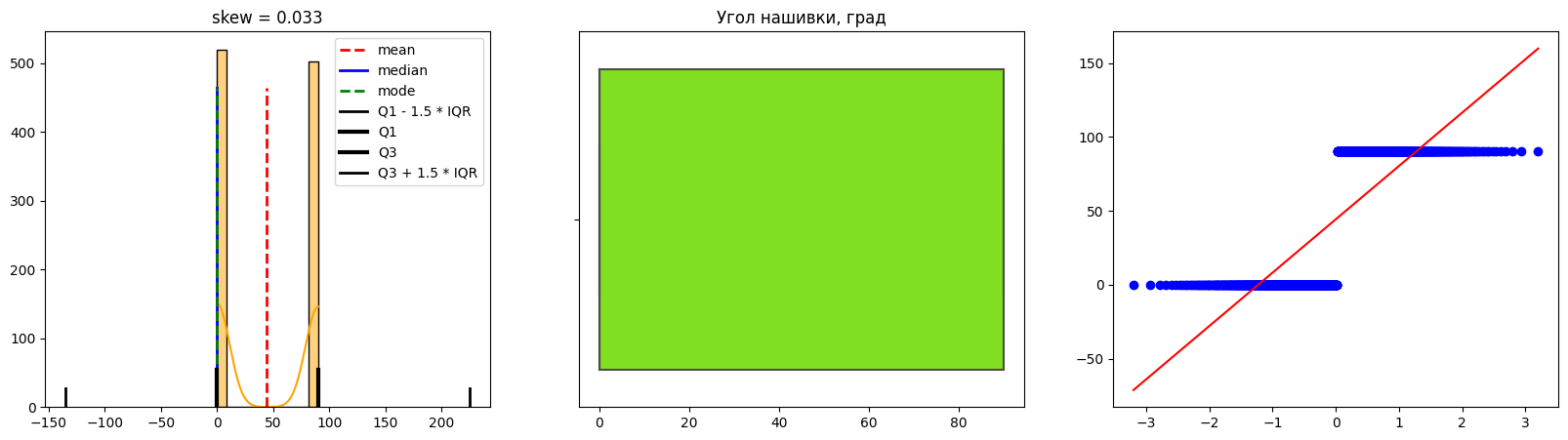
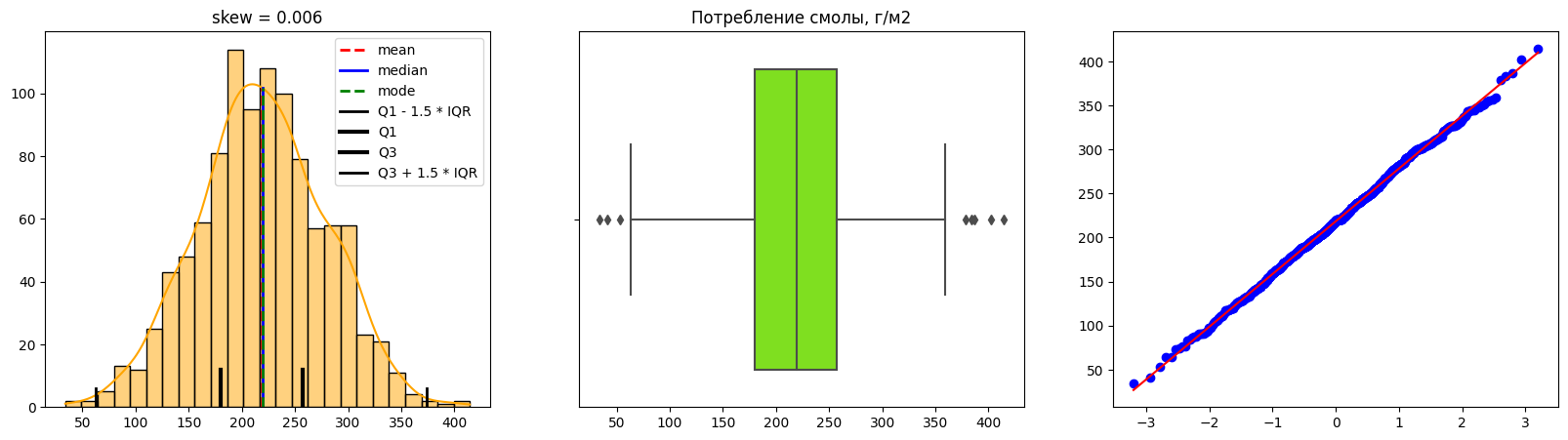
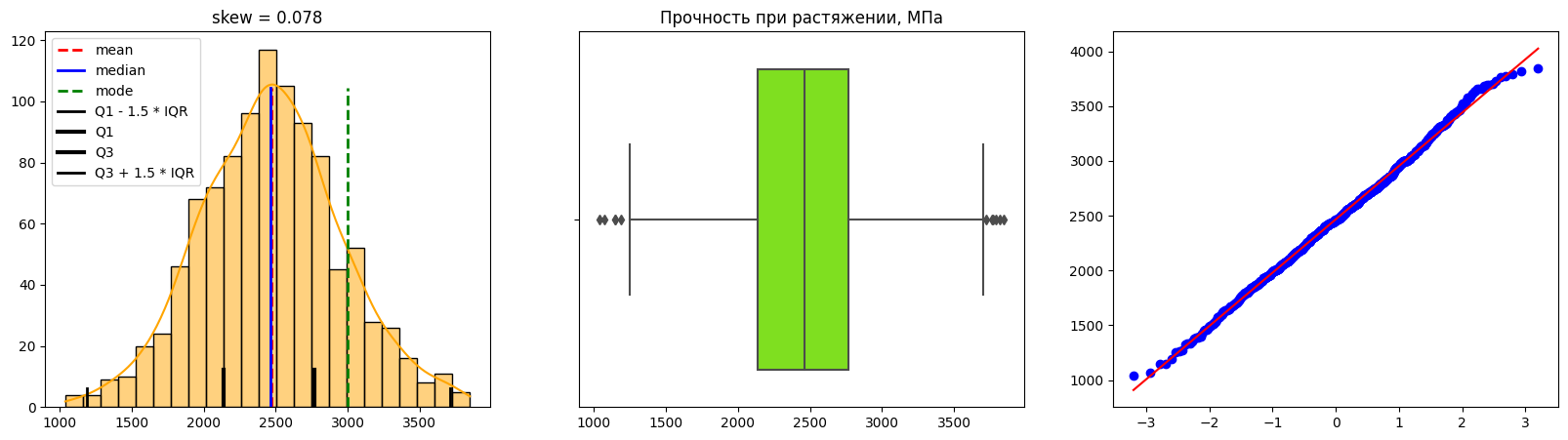
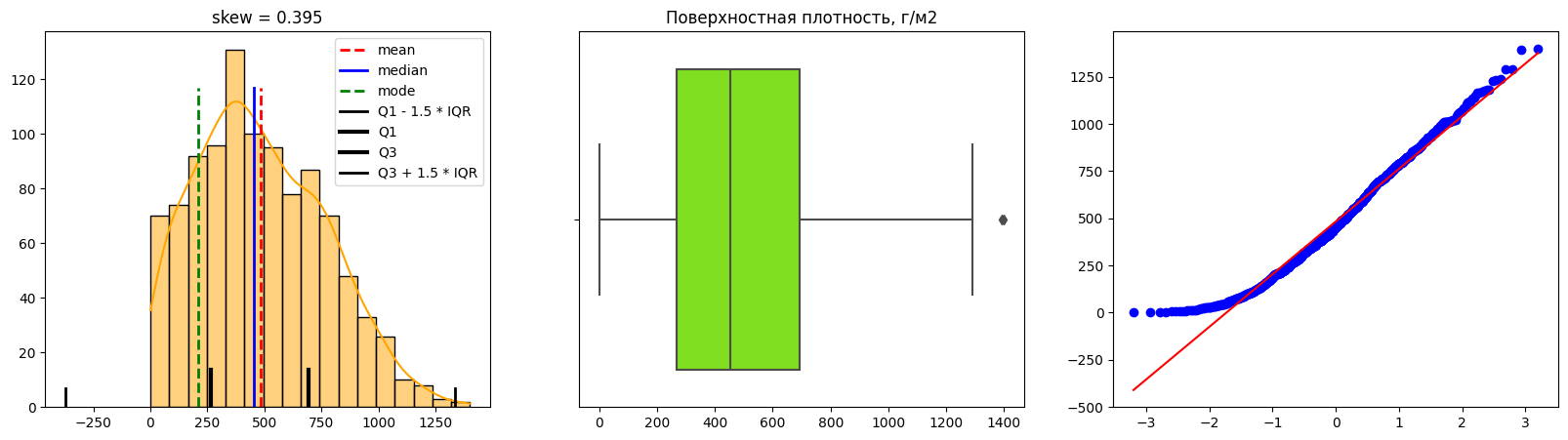
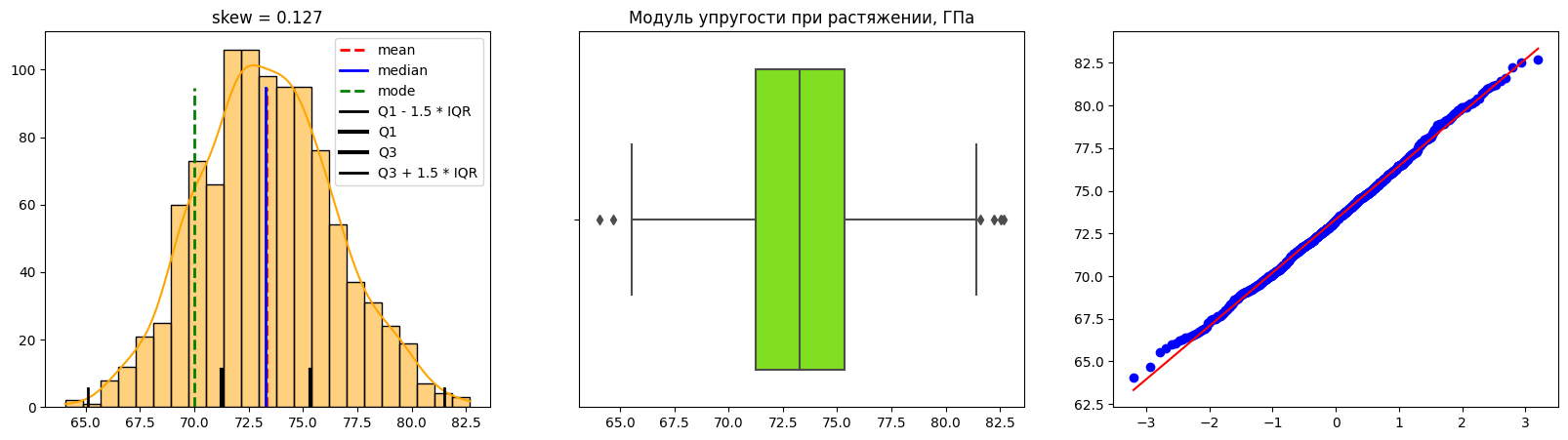
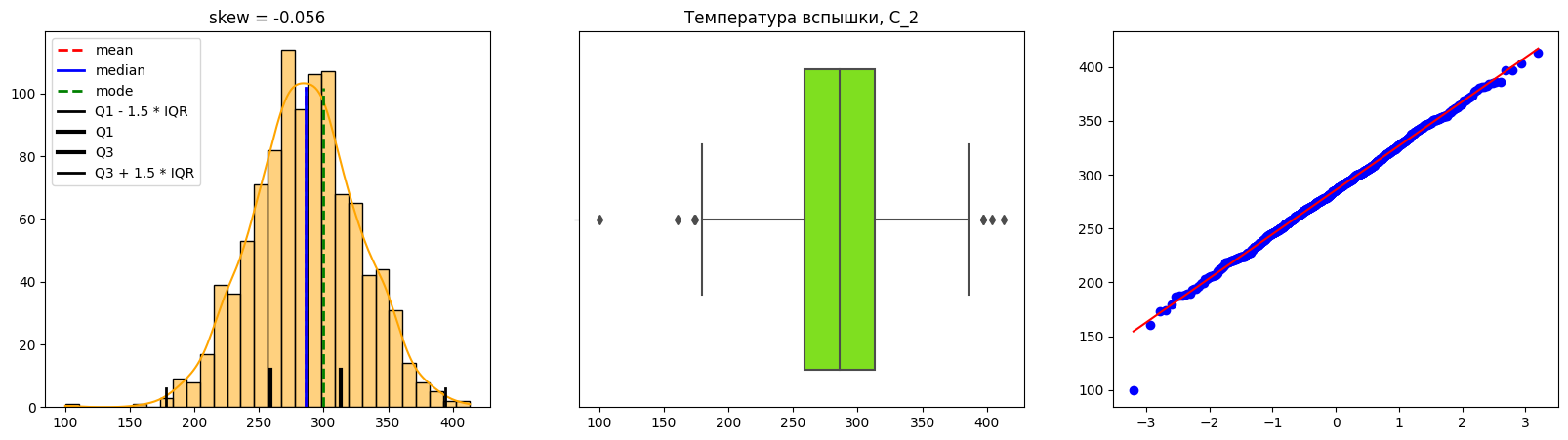
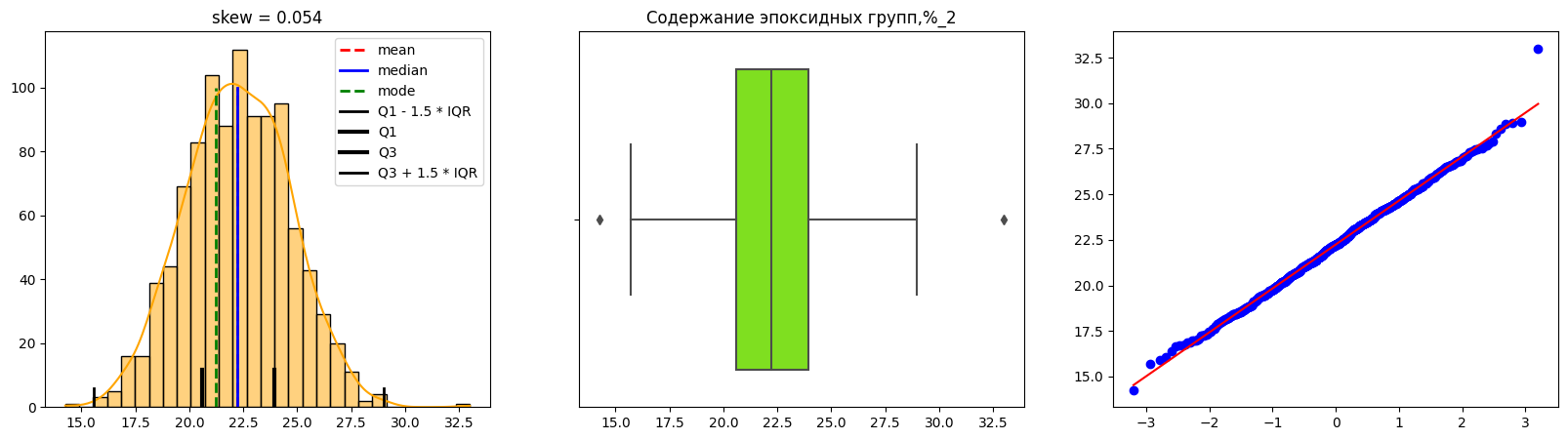
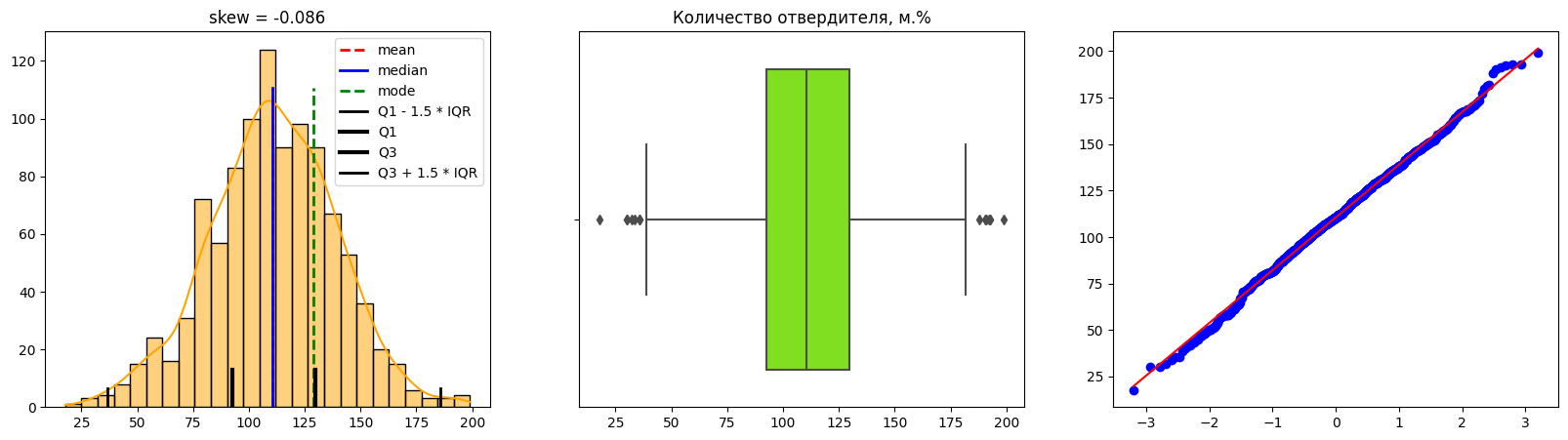
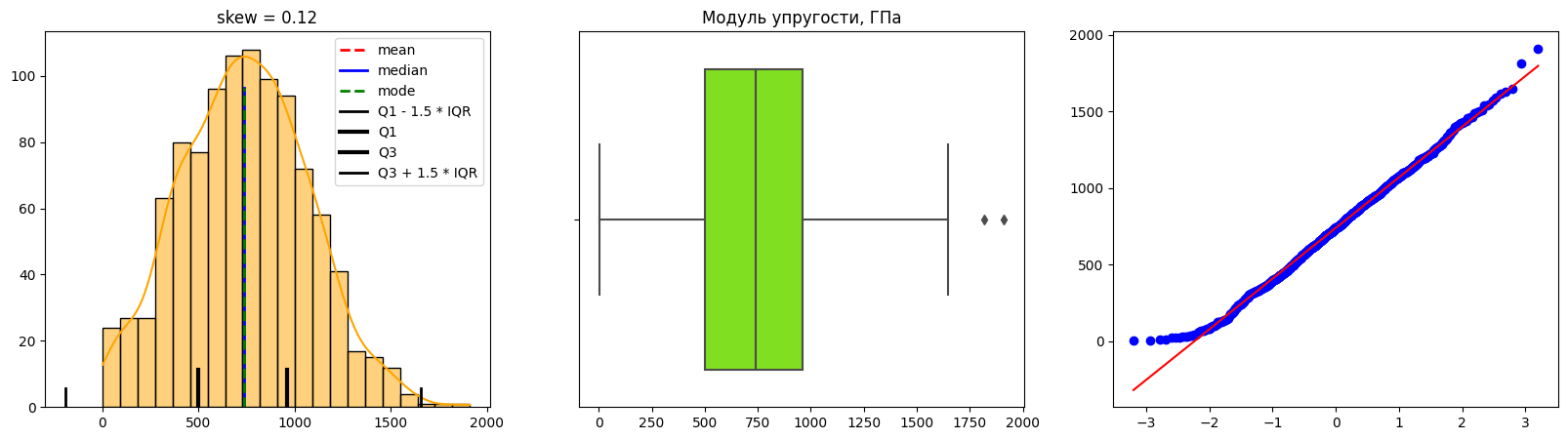
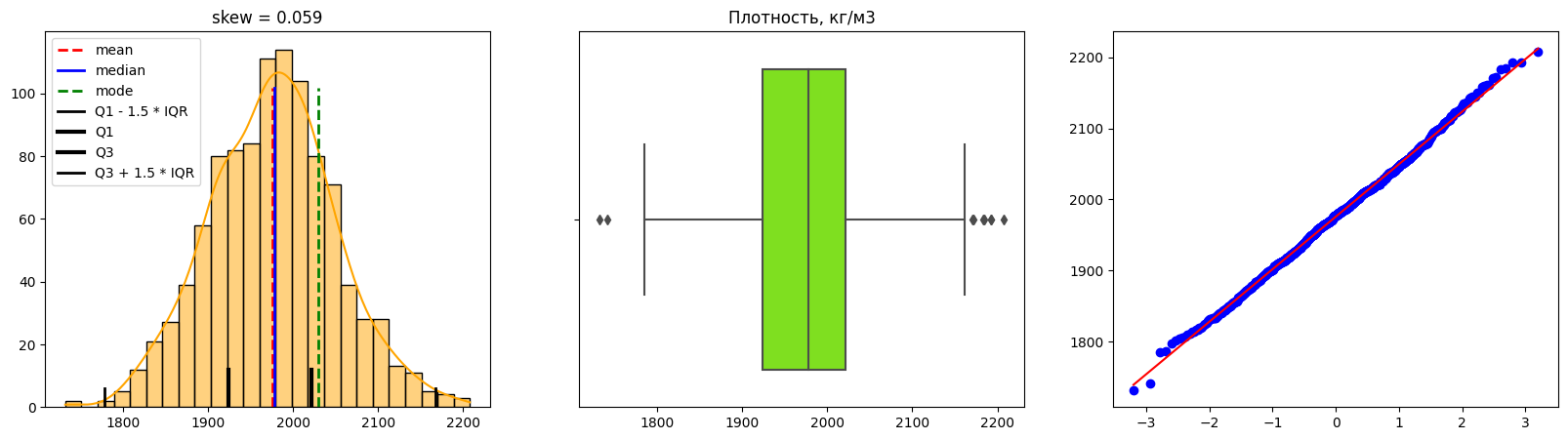
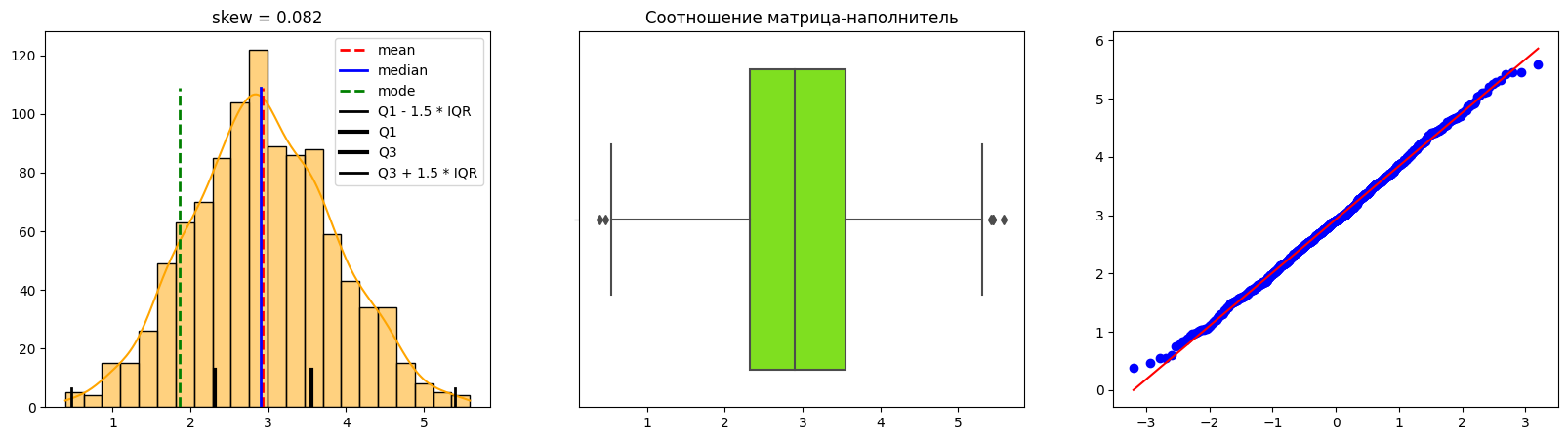


Рисунок 4 – Графики распределения, диаграммы «ящик с усами», графики «квантиль-квантиль»

По графикам можно сделать выводы:

* наличие выбросов;
* явная корреляция между признаками отсутствует;
* распределение близкое к нормальному по всем признакам, кроме «Поверхностная плотность, г/м2», признак имеет положительную асимметрию.

Все статистические данные и другую полезную информацию можно получить также с помощью метода ProfileReport библиотеки ydata\_profiling.

Выброс - это элемент маломощного подмножества выборки, существенно отличающийся от остальных элементов выборки. Причинами появления выбросов могут быть ошибки в измерениях или технические ошибки форматирования данных.

Выбросы могут негативно повлиять на построение модели, поэтому лучше выполнить обнаружение и устранение (удаление или замена) выбросов.

Существует несколько подходов для обнаружения выбросов:

1. использование трех сигм, которые следуют правилу 68-95-99,7, при этом 68% данных находятся в пределах одного стандартного отклонения от среднего, 95% данных находятся в пределах двух стандартных отклонений от среднего и 99,7% данных находятся в пределах трех стандартных отклонений от среднего;
2. использование 5-95 квантилей - значения ниже 5% квантиля и выше 95% квантиля можно считать выбросами;
3. использование межквартильного размаха (interquartile range, IQR), который является мерой статистической дисперсии. Для вычисления IQR набор данных делится на квартили, или четыре упорядоченные по рангу четные части, с помощью линейной интерполяции. Эти квартили обозначаются Q1 (нижний квартиль, соответствует 25-му процентилю), Q2 (медиана) и Q3 (верхний квартиль, соответствует 75-му процентилю). IQR определяется как разность третьего квартиля и первого квартиля.

Использую подходы, указанные выше получили следующее:

* правила трех сигм - удалено 23 строки, при повторных итерациях еще 4;
* межквартильный размах - удалено 87 строк, при повторных итерациях еще 14;
* 5- 95 квантилей - удалено 727 строк (большая потеря данных).

# Практическая часть

## Предобработка данных

Предварительная обработка данных в машинном обучении – это важный шаг, который помогает повысить качество данных. Предобработка данных в машинном обучении относится к технике подготовки необработанных данных с целью сделать их пригодными для построения и обучения моделей машинного обучения. Иными словами, это метод интеллектуального анализа данных, который преобразует необработанные данные в понятный и читаемый формат.

Для предварительной обработки данных воспользуемся следующими трансформаторными классами пакета sklearn.preprocessing библиотеки scikit-learn:

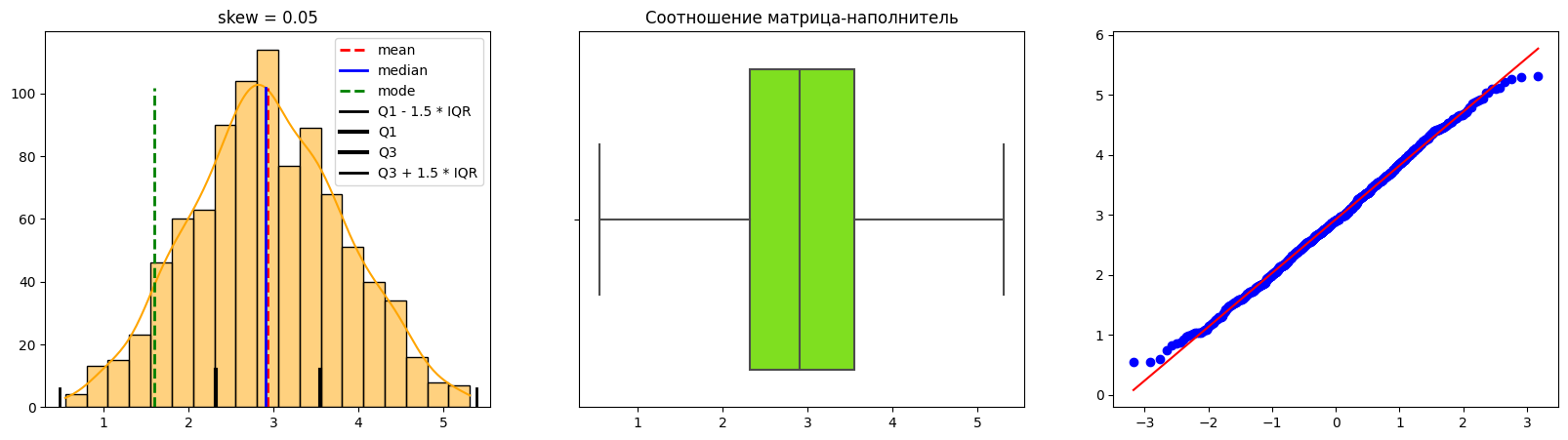
1. StandardScaler – стандартизация путем удаления среднего значения и масштабирования до единичной дисперсии (центрирование и масштабирование происходят независимо для каждого объекта путем вычисления соответствующей статистики по выборкам в обучающем наборе);
2. MinMaxScaler - преобразование объектов, масштабируя каждый объект до заданного диапазона (масштабируется и преобразуется каждый признак по отдельности таким образом, чтобы он находился в заданном диапазоне в обучающем наборе, чаще всего, между нулем и единицей);
3. PowerTransformer - функциональное преобразование мощности, чтобы сделать данные более похожими на гауссовы (семейство параметрических монотонных преобразований, которые применяются для придания данным более гауссовского типа; поддерживает преобразование Бокса-Кокса и преобразование Йео-Джонсона);
4. QuantileTransformer - преобразование объектов, используя информацию о квантилях (применяется к каждому элементу независимо; этот метод преобразования имеет тенденцию распространять наиболее часто встречающиеся значения, что уменьшает влияние незначительных выбросов);
5. RobustScaler - масштабирование объектов, используя статистику, устойчивую к выбросам (удаляет медиану и масштабирует данные в соответствии с диапазоном квантилей, центрирование и масштабирование происходят независимо для каждого объекта путем вычисления соответствующей статистики по выборкам в обучающем наборе).

Предобработку данных сделаем для всех полученных датасетов после удаления выбросов с помощью StandardScaler (для признаков с нормальным распределением) совместно с PowerTransformer (для признаков с явной асимметрией, преобразованием Йео-Джонсона) и MinMaxScaler. Дополнительно выполним предобработку датасета без удаления выбросов с помощью RobustScaler и QuantileTransformer.

Целевые переменные преобразовывать не будем.

После предобработки данных получили восемь датасетов:

* 3sig\_SC (выбросы – правило трех сигм, преобразование - StandardScaler + PowerTransformer);
* 3sig\_MMS (выбросы – правило трех сигм, преобразование - MinMaxScaler);
* IQR\_SC (выбросы – межквартильный размах, преобразование - StandardScaler + PowerTransformer) - графики распределения, диаграммы «ящик с усами», графики «квантиль-квантиль» каждого признака отображены на рисунке 5;
* IQR\_MMS (выбросы – межквартильный размах, преобразование - MinMaxScaler);
* 5Q95\_SC (выбросы – 5-95 квантилей, преобразование - StandardScaler + PowerTransformer);
* 5Q95\_MMS (выбросы – 5-95 квантилей, преобразование - MinMaxScaler);
* all\_RS (выбросы – не устранены, преобразование - RobustScaler);
* all\_QT (выбросы – не устранены, преобразование - QuantileTransformer).



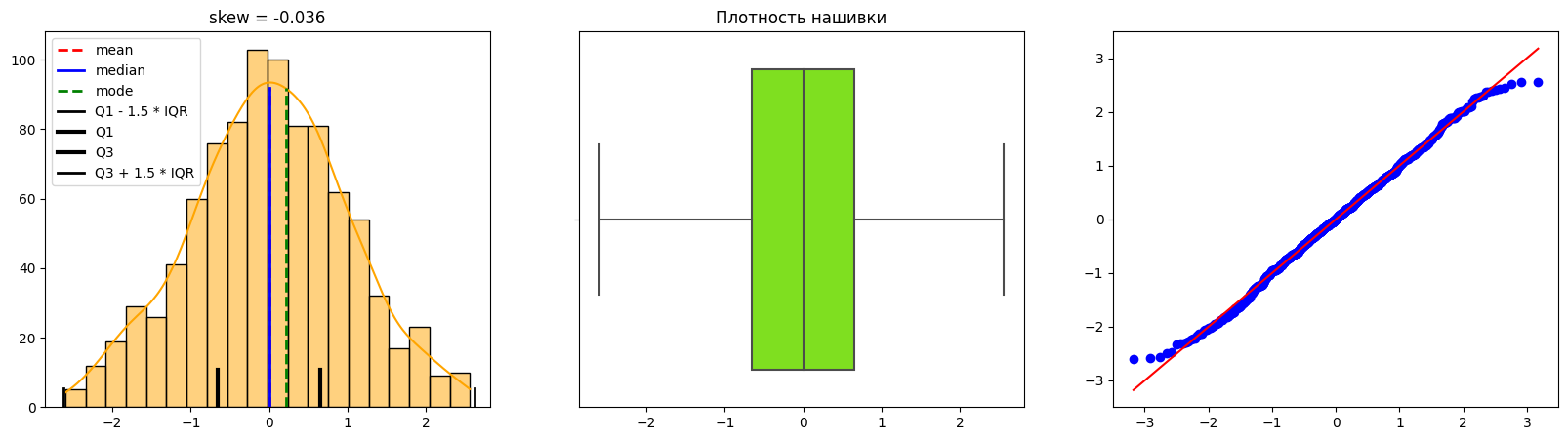
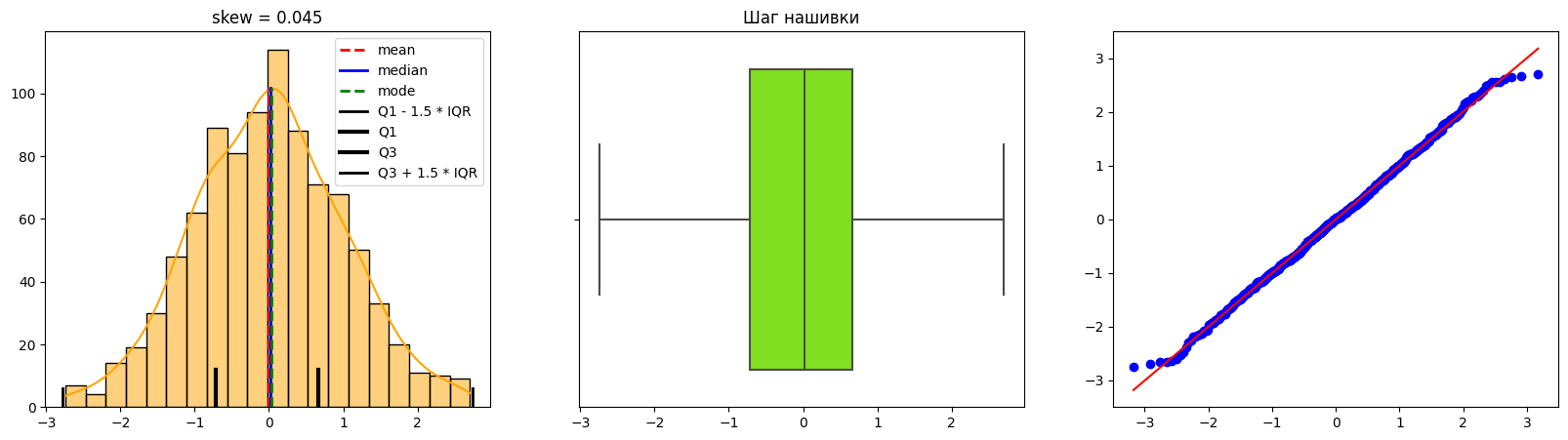
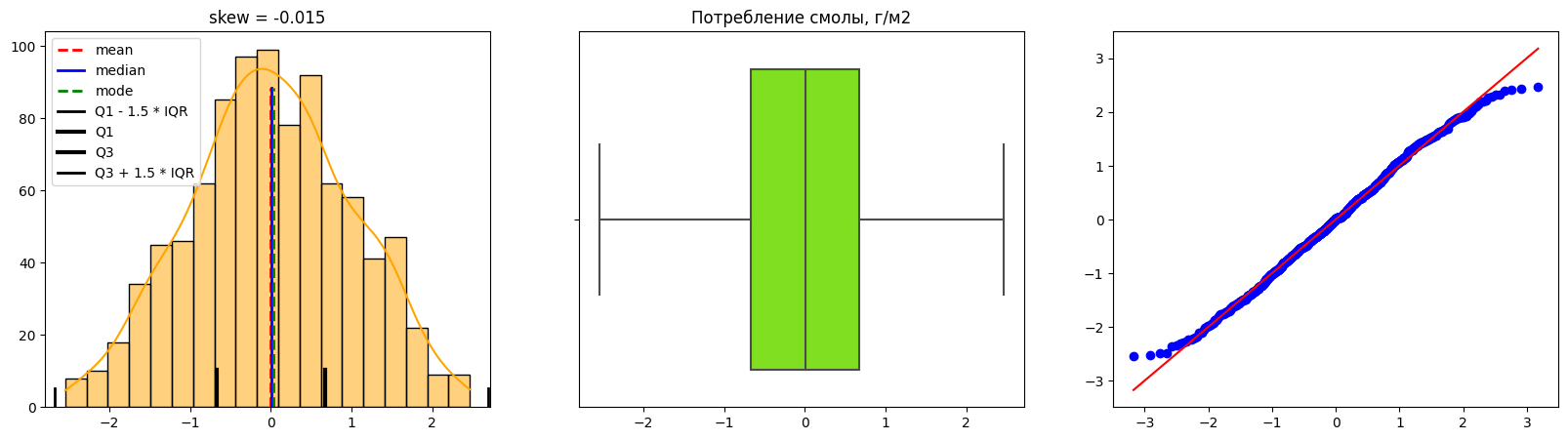
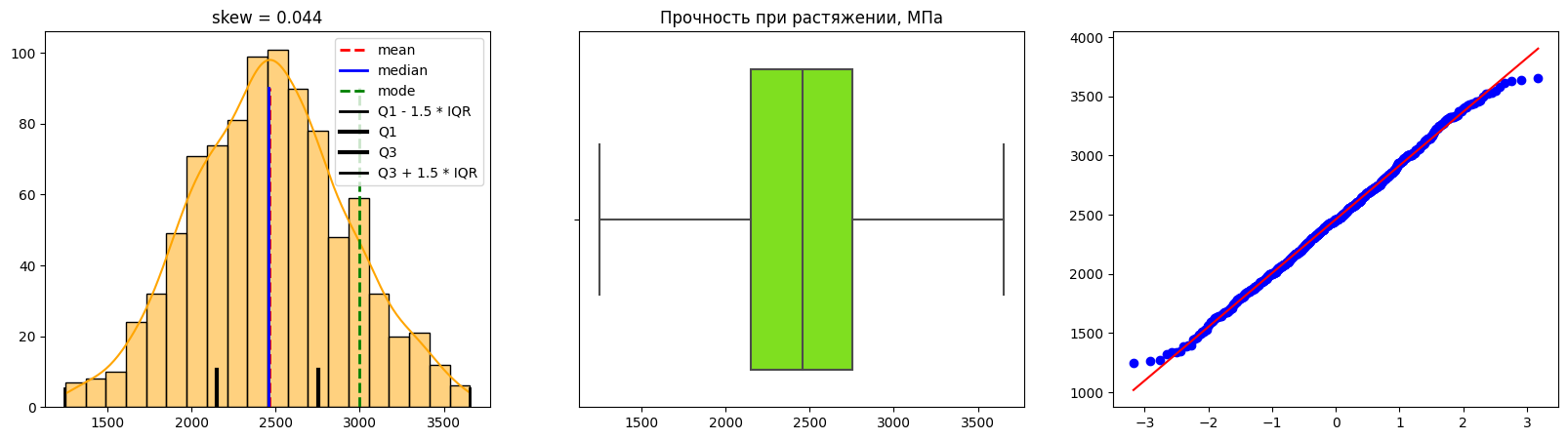
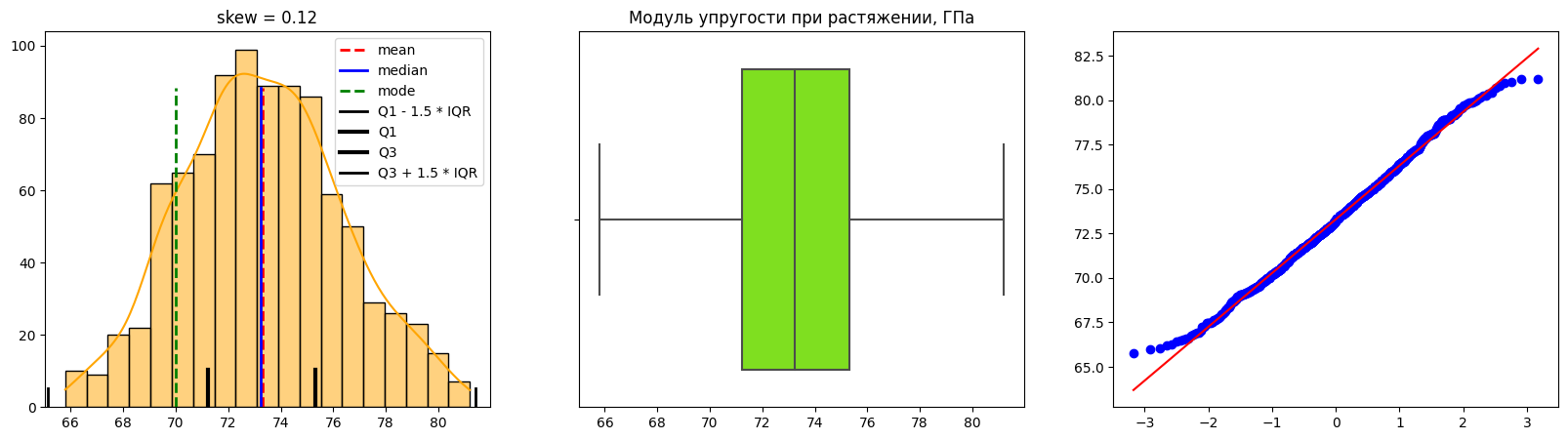
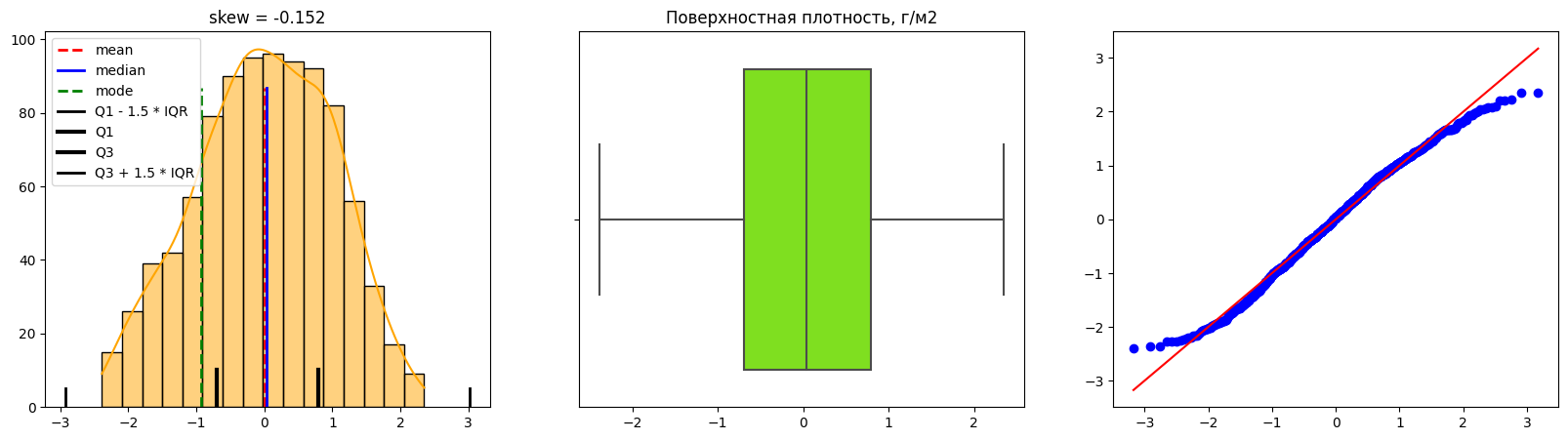
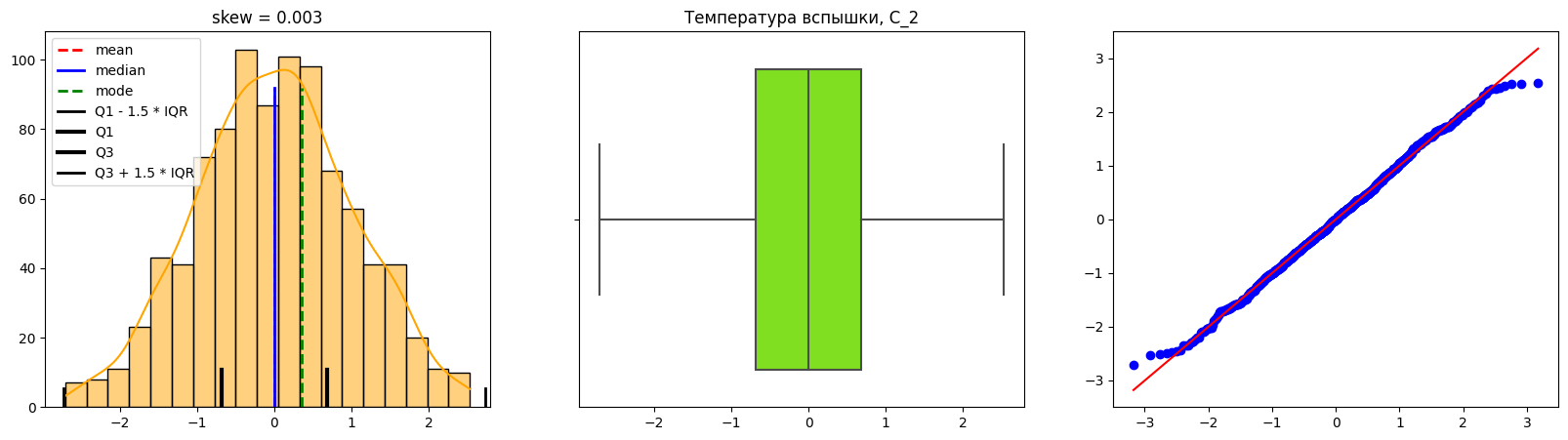
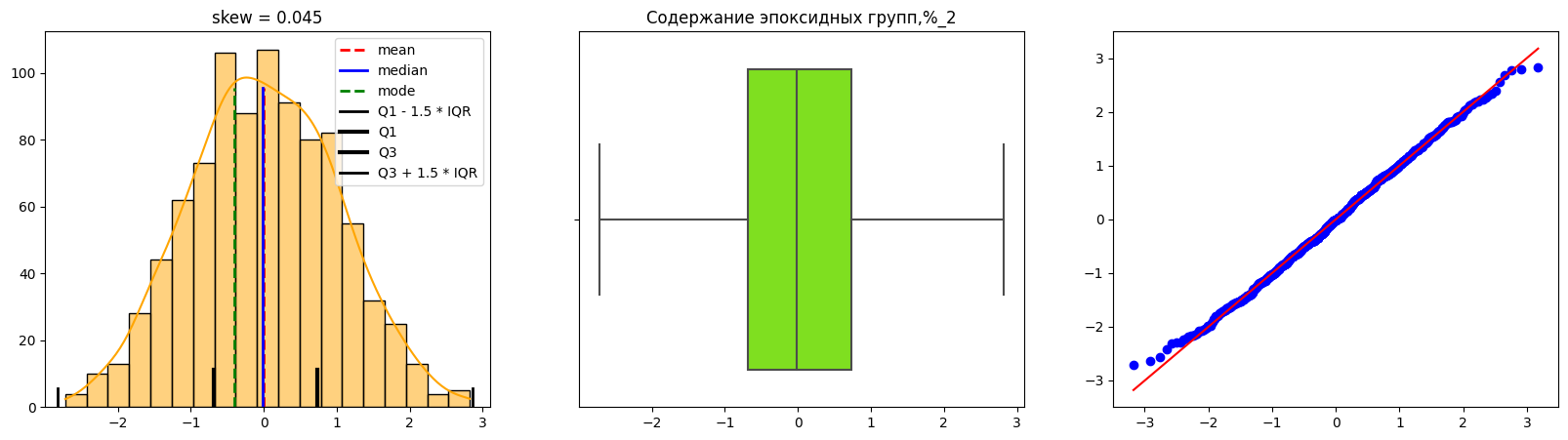
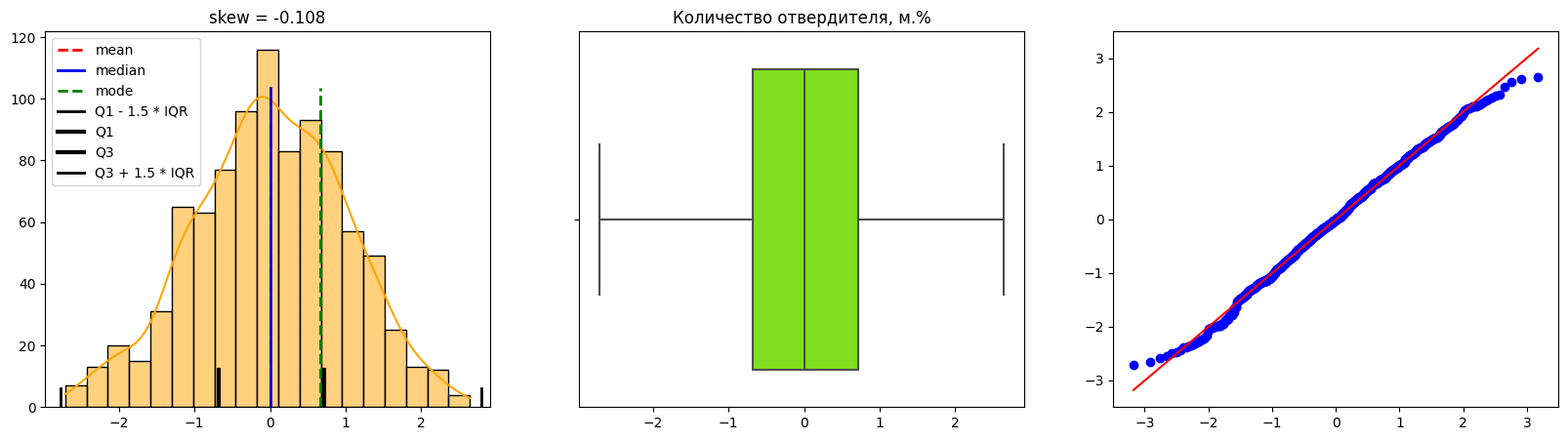
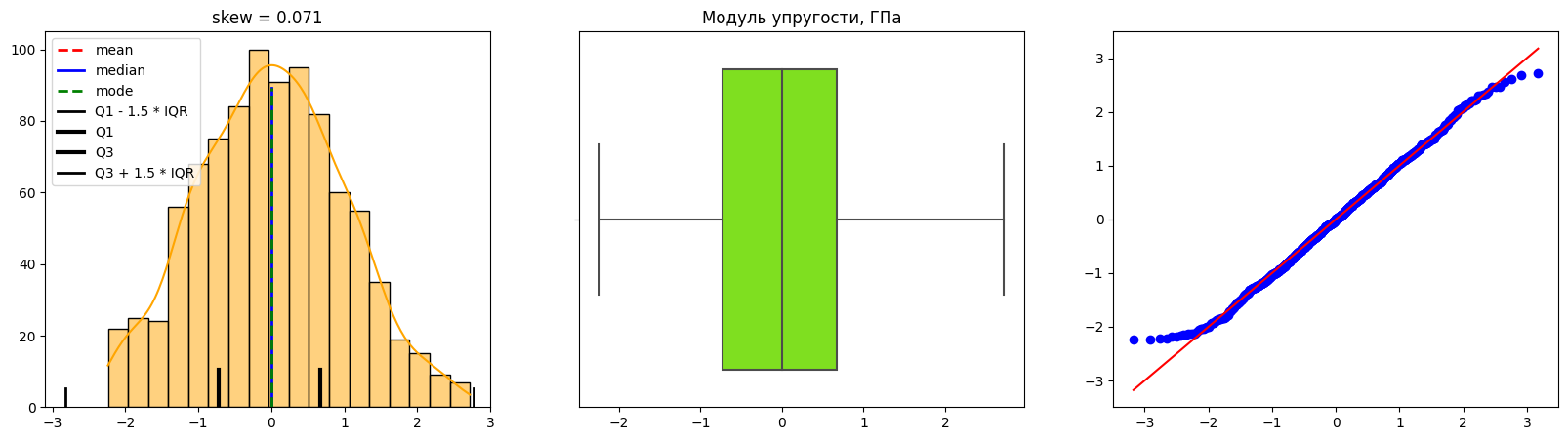
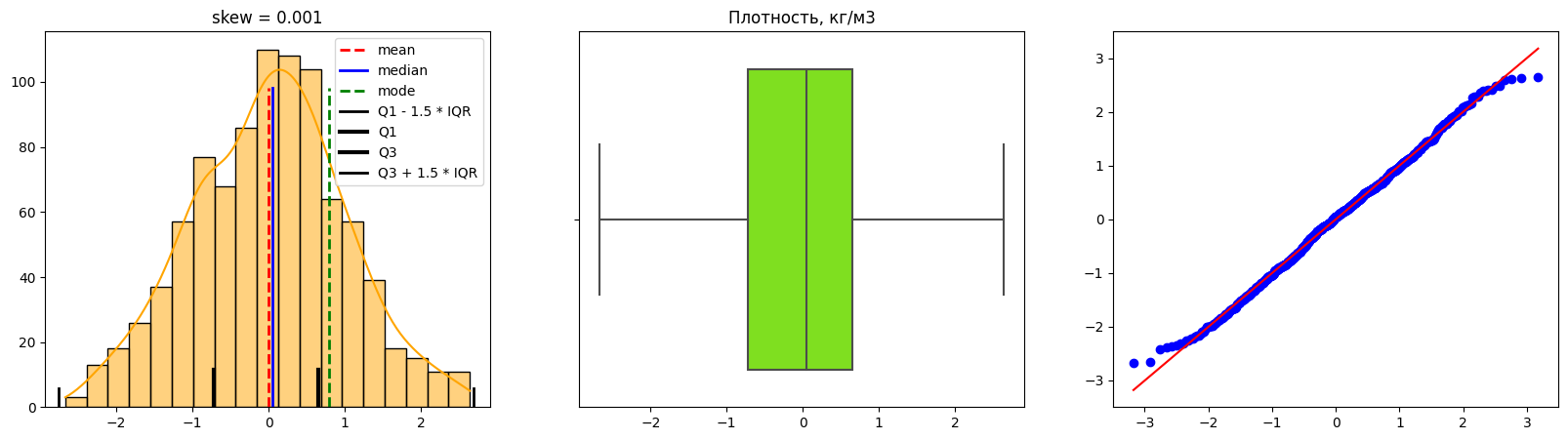


Рисунок 5 - Графики распределения, диаграммы «ящик с усами», графики «квантиль-квантиль» после предобработки данных

## Выбор стратегии

Для выбора стратегии устранения выбросов и предобработки данных было принято решений прогнать все полученные датасеты через обучение моделей с параметрами по умолчанию, и выбрать наилучший вариант на основе метрик полученных моделей.

С помощью цикла все датасеты были разделены на входные и целевые переменные, потом на обучающую и тестовую выборки в соотношении 70% обучающая и 30% тестовая, далее произведено обучение моделей и снятие метрик.

Полученные метрики моделей показывают, что в большинстве случаев наилучшие результаты выдает модель обычной линейной регрессии.

Далее выполнено сравнение датасетов между собой по модели линейной регрессии, чтоб определится с каким датасетом будем работать дальше. Данные для сравнения изображены на рисунке 6.

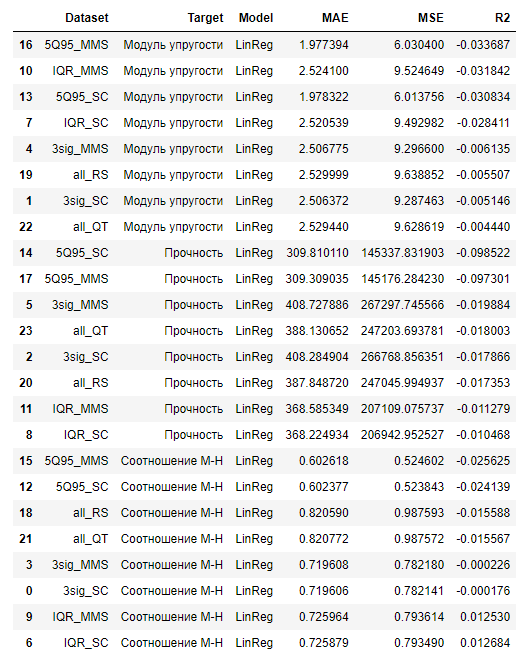


Рисунок 6 – Метрики модели линейной регрессии на разных датасетах

Дальнейшее обучение моделей будет на датасете с устранением выбросов, используя межквартильный размах, и с преобразованием StandardScaler, совместно с PowerTransformer.

## Разработка и обучение моделей

Для прогнозирования модуля упругости при растяжении и прочности при растяжении будут обучены модели с применением методов, указанных в подразделе 1.2.

Сначала разделим датасет на входные и целевую переменные, потом с помощью sklearn.model\_selection.train\_test\_split библиотеки scikit-learn разделим датасет на тестовую (70%) и обучающую (30%) выборки.

Далее, используя sklearn.compose.ColumnTransformer, сделаем препроцессинг данных, который позволит автоматизировать предобработку данных перед обучением модели.

Препроцессинг будет состоять из следующих преобразований:

* StandardScaler для числовых признаков без асимметрии;
* PowerTransformer для числовых признаков с явной асимметрией;
* OneHotEncoder для категориальных признаков.

Воспользуемся функцией sklearn.model\_selection.GridSearchCV библиотеки scikit-learn для автоматического подбирания параметров моделей машинного обучения. GridSearchCV находит наилучшие параметры, путем обычного перебора, создавая модель для каждой возможной комбинации параметров.

После получения наилучших параметров, инициализируем модели для обучения. Используя sklearn.pipeline.Pipeline библиотеки scikit-learn, объединим препроцессинг данных и модель в пайплайн, произведем обучение, снимем метрики и сохраним полученные модели в формате pkl для дальнейшего использования.

## Оценка моделей

Оценку моделей будем производить с помощью коэффициента детерминации, дополнительно посмотрим результаты среднеквадратичной ошибки и средней абсолютной ошибки.

Коэффициент детерминации (R-квадрат) - это доля дисперсии зависимой переменной, объясняемая рассматриваемой моделью зависимости, то есть объясняющими переменными.

Среднеквадратичная ошибка (MSE) рассчитывается путем получения среднего квадрата разницы между исходным и прогнозируемым значениями данных.

Средняя абсолютная ошибка (MAE) представляет собой среднюю абсолютную разницу между исходным и прогнозируемым значениями данных.

На рисунках 7, 8 отображены полученные метрики работы моделей.

На рисунке 7

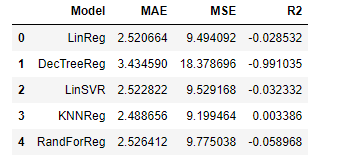


Рисунок 7 – Метрики моделей для прогноза модуля упругости при растяжении

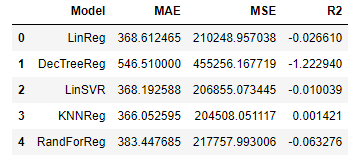


Рисунок 8 – Метрики моделей для прогноза прочности при растяжении

Модели, реализованные с помощью метода k-ближайших соседей, показывают наилучшие значения коэффициента детерминации для прогнозирования целевых переменных.

## Разработка и обучение нейронной сети

Для прогнозирования соотношения матрицы-наполнитель разработаем и обучим нейронную сеть с помощью библиотеки tensorflow.

Так как мы уже обучили модели для прогнозирования двух других целевых переменных, то будем считать их входными переменными.

Сначала сделаем два препроцессинга данных: один аналогичен препроцессингу созданному ранее, но с добавлением двух новых входных переменных, второй будет использовать MinMaxScaler для преобразования числовых признаков и OneHotEncoder для категориальных признаков. Для дальнейшего использования сохраним полученные модели препроцессинга в формате pkl.

Инициализируем модель нейронной сети и попробуем вручную подбирать параметры сети для датасета с разной предобработкой данных, будем изменять количество слоев и нейронов, использовать разные функции активации и оптимизаторы.

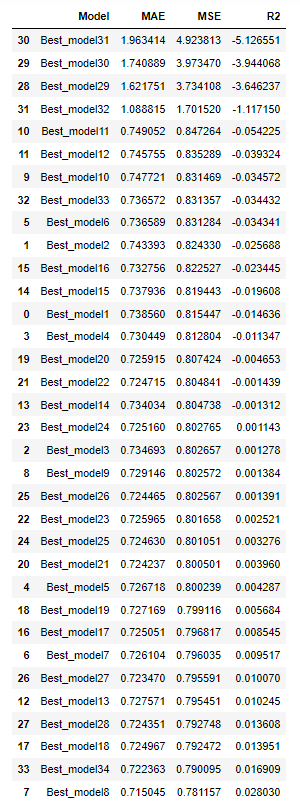
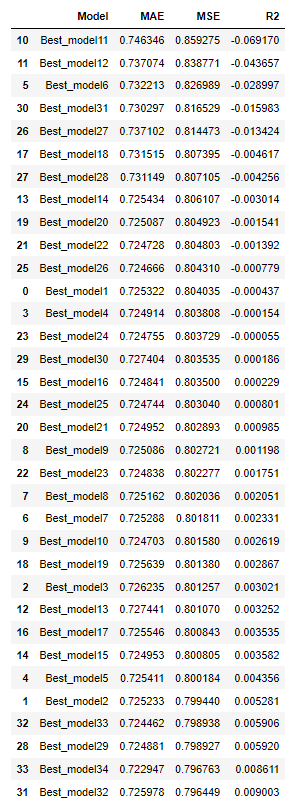
При обучении нейронной сети воспользуемся ModelCheckpoint для сохранения лучшей модели в формате hdf5 и EarlyStopping для ранней остановки обучения, при отсутствии улучшения точности модели после 20 эпох обучения

На рисунке 9 изображены метрики моделей нейронной сети, полученных в результате ручного перебора параметров.

Наилучший показатель детерминации показала модель со следующими параметрами сети:

* входной слой (количество нейронов - 8, инициализация матрицы весов – 'normal', функция активации – 'relu');
* выходной слой (количество нейронов - 1, инициализация матрицы весов – 'normal', функция активации – 'linear');
* функция ошибки 'mean\_squared\_error';
* оптимизатор - Adam (скорость обучения - 0.0001);
* метрика - 'mean\_squared\_error'.

Рисунок 9 – Метрики моделей нейронной сети



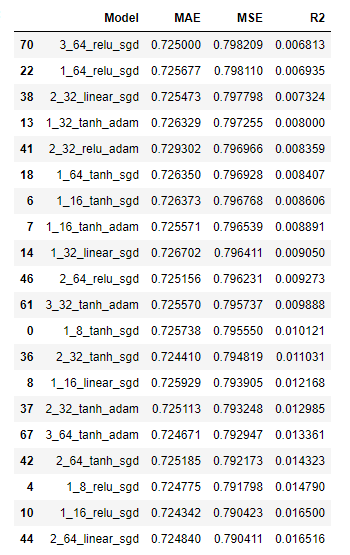
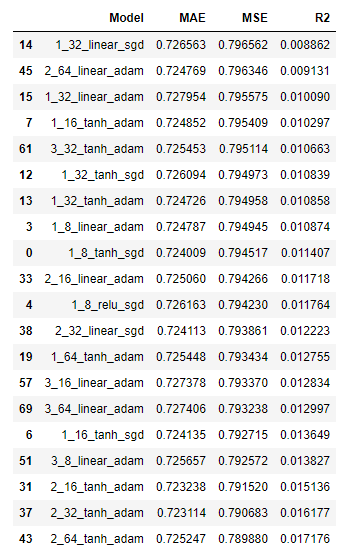
(слева – датасет после StandardScaler, справа – датасет после MinMaxScaler)

Дополнительно было решено выполнить попытку параметров сети перебором через цикл. Параметры цикла:

* количество слоев – [1, 2, 3];
* количество нейронов - [8, 16, 32, 64];
* функции активации - ['tanh', 'linear', 'relu'];
* оптимизаторы - ['sgd', 'adam'].

Полученные модели не дали коэффициент детерминации выше, чем модель, указанная выше. Метрики моделей изображены на рисунке 10.

Рисунок 10 – Метрики моделей нейронной сети, полученные циклом



(слева – датасет после StandardScaler, справа – датасет после MinMaxScaler)

Для дальнейшего использования наилучшая модель и ее веса сохранены в формате h5.

## Разработка приложения

Для разработки приложения будем использовать библиотеку Streamlit.

Streamlit - это фреймворк с открытым исходным кодом, специально используемый для задач, связанных с машинным обучением и наукой о данных. Он может создавать веб-приложения с гораздо меньшим количеством кода. Он широко используется, потому что поддерживает большинство научных библиотек данных.

Разработанное приложение позволяет пользователю:

1. внести характеристики матрицы и наполнителя (рисунок 11) - по умолчанию, заполнены значениями из тестовой выборки;

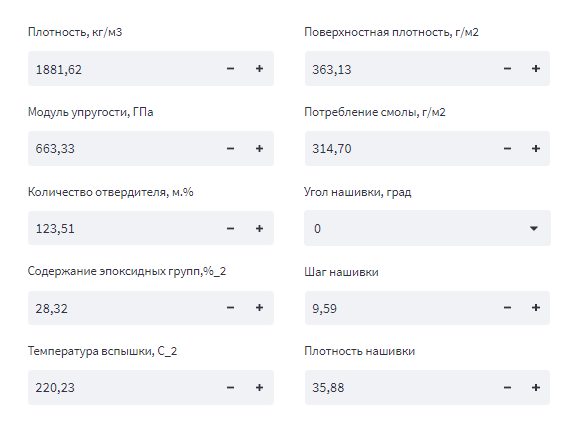


Рисунок 11 – Интерфейс ввода значений

1. выбрать модель машинного обучения (рисунок 12) – по умолчанию выбрана модель линейной регрессии;

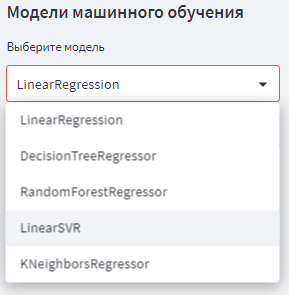


Рисунок 12 – Интерфейс выбора модели

1. получить прогнозы модуля упругости при растяжении, прочность при растяжении, соотношение матрица-наполнитель после нажатия кнопки «Получить прогнозы».

Принцип работы приложения после нажатия кнопки «Получить прогнозы» следующий:

1. формируется датасет из внесенных значений;
2. в зависимости от выбора пользователя загружаются модели для прогнозирования модуля упругости при растяжении и прочности при растяжении, и выполняется прогнозирование этих значений;
3. формируется датасет из внесенных и полученных значений из предыдущего пункта;
4. загружается модель препроцессинга данных и выполняется преобразования датасета из предыдущего пункта;
5. загружается модель нейронной сети и выполняется прогнозирование соотношения матрица-наполнитель;
6. вывод результатов работы (рисунок 13).



Рисунок 13 – Общий вид интерфейса приложения после вывода результатов

<https://olvovchik-data-science-learn-applicationapp-tfgy2e.streamlit.app/> - ссылка на приложение в сети Интернет.

Заключение

Полученный коэффициент детерминации обученных моделей и нейронной сети практически нулевой – это означает, что связь между переменными регрессионной модели отсутствует и получаемые прогнозы ничем не отличаются от прогноза средним значением.

Итоговое решение поставленной задачи не достигнуто, требуется более детальный анализ данных (желательно с привлечением специалистов предметной области). Также можно попробовать использовать другие методы и модели прогнозирования, которые не были рассмотрены в данной работе.

Библиографический список

1. Библиотека Matplotlib документация [Электронный ресурс] : – Режим доступа: [https://matplotlib.org/](https://matplotlib.org/%20) (дата обращения 28.03.2023).
2. Библиотека Pandas документация [Электронный ресурс] : – Режим доступа: <https://pandas.pydata.org/> (дата обращения 27.03.2023).
3. Библиотека Seaborn документация [Электронный ресурс] : – Режим доступа: <https://seaborn.pydata.org/index.html> (дата обращения 28.03.2023).
4. Библиотека Scikit-learn документация [Электронный ресурс] : – Режим доступа: <https://scikit-learn.org/stable/index.html> (дата обращения 01.04.2023).
5. Библиотека Scipy документация [Электронный ресурс] : – Режим доступа: <https://scipy.org/> (дата обращения 31.03.2023).
6. Библиотека TensorFlow документация [Электронный ресурс] : – Режим доступа: <https://www.tensorflow.org/> (дата обращения 07.04.2023).
7. Библиотека Streamlit документация [Электронный ресурс] : – Режим доступа: <https://docs.streamlit.io/> (дата обращения 10.04.2023).
8. Википедия [Электронный ресурс] : – Режим доступа: <https://ru.wikipedia.org/wiki/> (дата обращения 31.03.2023).
9. Гапанюк Ю. Е. Репозиторий курсов по машинному обучению [Электронный ресурс] : – Режим доступа: <https://github.com/ugapanyuk/ml_course_2022> (дата обращения 28.03.2023).
10. Коротеев М. Машинное обучение обучению [Электронный ресурс] : – Режим доступа: <https://koroteev.site/ml/> (дата обращения 10.04.2023).
11. Капаца Е. Машинное обучение доступным языком [Электронный ресурс] : – Режим доступа: <https://www.helenkapatsa.ru/> (дата обращения 30.03.2023).
12. Метрики качества линейных регрессионных моделей [Электронный ресурс] : – Режим доступа: <https://loginom.ru/blog/quality-metrics> (дата обращения 02.04.2023).
13. Оптимизация гиперпараметров с помощью поиска по сетке и случайного поиска в Python [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/companies/otus/articles/698370/> (дата обращения 05.04.2023).
14. Умная нормализация данных [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/articles/527334/> (дата обращения 01.04.2023).
15. Streamlit. Поиск кратчайшего пути [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/articles/568836/> (дата обращения 10.04.2023).
16. Joseph Misiti Awesome Machine Learning [Электронный ресурс] : – Режим доступа: <https://github.com/josephmisiti/awesome-machine-learning> (дата обращения 07.04.2023).