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 Pro»

**Тема:** «**Прогнозирование конечных свойств новых материалов**

**(композиционных материалов)»**

Слушатель Радьков Дмитрий Васильевич

Москва, 2024

1. **Аналитическая часть**
   1. **Постановка задачи**

Для исследования в качестве входных данных были предоставлены файлы: X\_nup.xlsx, содержащий 11 колонок и 1023 строк данных, и X\_nup.xlsx - 3 колонки и 1040 строк данных.

Цель исследования разработать модели для прогноза модуля упругости при растяжении, прочности при растяжении и соотношения «матрица-наполнитель».

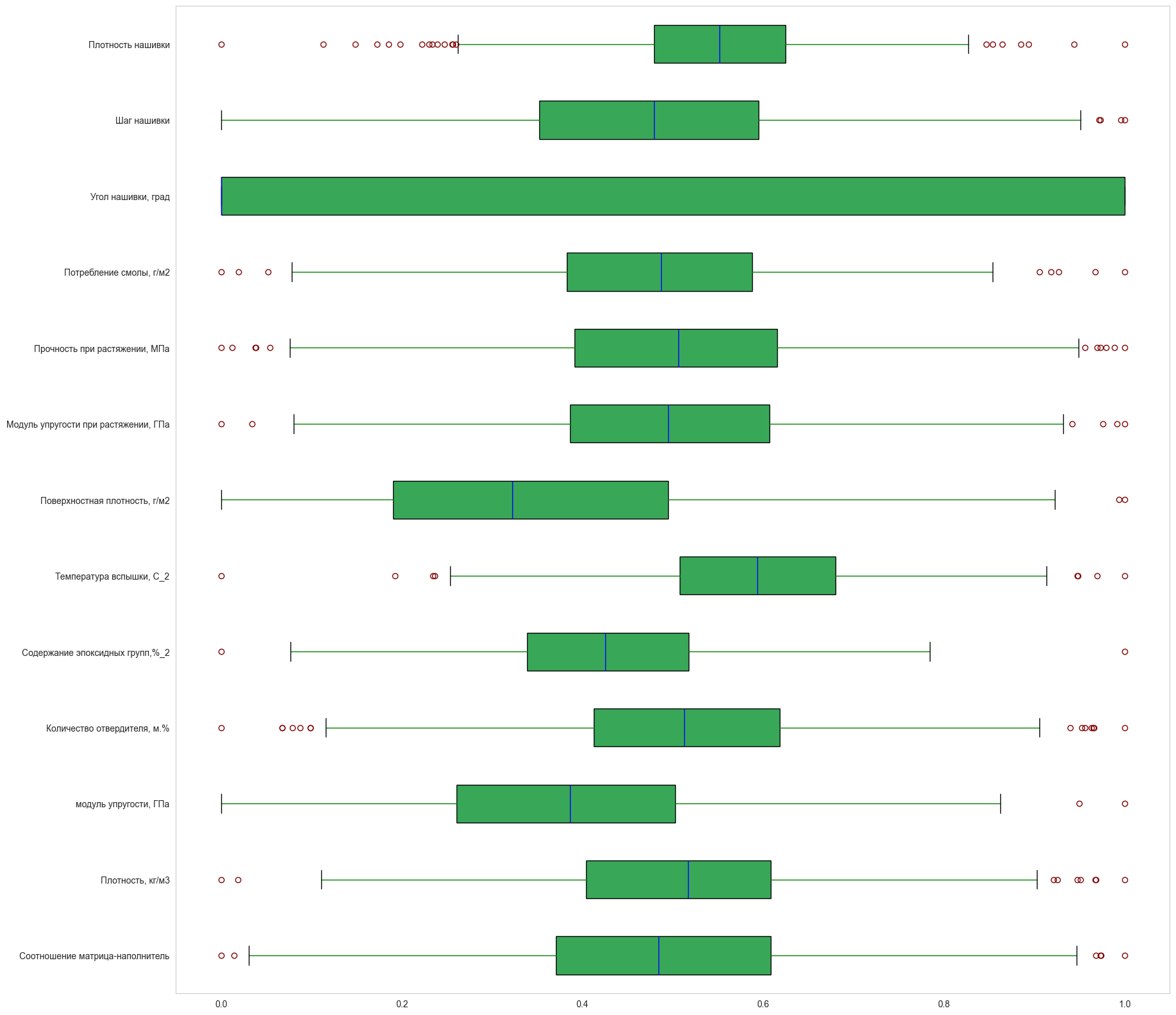
Было произведено объединение двух файлов в итоговый датасет. Объединение производилось по индексу и с использованием внутреннего соединения (INNER) и удалением безымянных столбцов. В результате итоговый датасет имеет размерность (1023, 13).

Далее требуется произвести разведочный анализ данных, отобразить гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек.



Рисунок 1 - пример итогового датасета

Рисунок 2 - диаграмма ящик с усами



Для каждой колонки получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков; пред обработать данные: удалить шумы и выбросы, сделать нормализацию и стандартизацию. Обучить несколько моделей для прогноза модуля упругости при растяжении и прочности при растяжении. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель. Разработать приложение с графическим интерфейсом, которое будет выдавать прогноз соотношения «матрица-наполнитель». Оценить точность модели на тренировочном и тестовом датасете. Создать репозиторий в GitHub и разместить код исследования. Оформить файл README.

* 1. **Описание используемых методов**

Задача исследования - предсказание числового значения, это классическая задача регрессии. С целью определения наилучшей модели осуществим исследование с помощью следующих алгоритмов:

* Лассо регрессии
* Метод опорных векторов
* Метод случайного леса
* Метод градиентного бустинга
* Метод К ближайших соседей
* Методом решающих деревьев

Лассо регрессия (Lasso) — это линейная модель, которая оценивает разреженные коэффициенты.  Это простой метод, позволяющий уменьшить сложность модели и предотвратить переопределение, которое может возникнуть в результате простой линейной регрессии. Данный метод вводит дополнительное слагаемое регуляризации в оптимизацию модели. Это даёт более устойчивое решение. В регрессии лассо добавляется условие смещения в функцию оптимизации для того, чтобы уменьшить коллинеарность и, следовательно, дисперсию модели. Но вместо квадратичного смещения, используется смещение абсолютного значения. Лассо регрессия хорошо прогнозирует модели временных рядов на основе регрессии, таким как авторегрессии.

Достоинства метода: легко полностью избавляется от шумов в данных; быстро работает; не очень энергоёмко; способно полностью убрать признак из датасета.

Недостатки метода: выбор модели не помогает и обычно вредит; часто страдает качество прогнозирования; выдаёт ложное срабатывание результата; случайным образом выбирает одну из коллинеарных переменных; не оценивает правильность формы взаимосвязи между независимой и зависимой переменными; не всегда лучше, чем пошаговая регрессия.

Метод опорных векторов (Support Vector Regression) –  Метод опорных векторов пытается найти оптимальную разделяющую гиперплоскость между различными классами данных. Эта гиперплоскость максимизирует расстояние между ближайшими точками каждого класса, которые называются опорными векторами. Таким образом, метод опорных векторов пытается найти наилучшее разделение между классами, даже если данные не являются линейно разделимыми.

Метод опорных векторов также можно использовать для задач регрессии с помощью SVR. SVR пытается найти функцию, которая лежит в пределах заданного расстояния от точек обучающей выборки. Это позволяет модели обобщать на новые данные и делать предсказания.

Достоинства метода: для классификации достаточно небольшого набора данных. При правильной работе модели, построенной на тестовом множестве, вполне возможно применение данного метода на реальных данных. Эффективен при большом количестве гиперпараметров. Существует возможность гибко настраивать разделяющую функцию.  Алгоритм максимизирует разделяющую полосу, которая позволяет уменьшить количество ошибок классификации.

Недостатки метода: неустойчивость к шуму, поэтому была проведена очистка данных от выбросов; для больших наборов данных требуется долгое время обучения; параметры модели сложно интерпретировать, поэтому были рассмотрены и другие методы.

Случайный лес (RandomForest) — это множество решающих деревьев. Универсальный алгоритм машинного обучения с учителем, представитель ансамблевых методов.  Если точность дерева решений оказалось недостаточной, мы можем множество моделей собрать в коллектив.

Достоинства метода: не переобучается; не требует предобработки входных данных; эффективно обрабатывает пропущенные данные, данные с большим числом классов и признаков; имеет высокую точность предсказания и внутреннюю оценку обобщающей способности модели, а также высокую параллелизуемость и масштабируемость.

Недостатки метода: построение занимает много времени; сложно интерпретируемый; не обладает возможностью экстраполяции; может недо обучаться; трудоёмко прогнозируемый; иногда работает хуже, чем линейные методы.

Градиентный бустинг (Gradient Boosting) — это ансамбль деревьев решений, обученный с использованием градиентного бустинга. В основе данного алгоритма лежит итеративное обучение деревьев решений с целью минимизировать функцию потерь. Основная идея градиентного бустинга: строятся последовательно несколько базовых классификаторов, каждый из которых как можно лучше компенсирует недостатки предыдущих. Финальный классификатор является линейной композицией этих базовых классификаторов.

Достоинства метода: новые алгоритмы учатся на ошибках предыдущих; требуется меньше итераций, чтобы приблизиться к фактическим прогнозам; наблюдения выбираются на основе ошибки; прост в настройке темпа обучения и применения; легко интерпретируем.

Недостатки метода: необходимо тщательно выбирать критерии остановки, иначе это может привести к переобучению; наблюдения с наибольшей ошибкой появляются чаще; слабее и менее гибок чем нейронные сети.

Метод ближайших соседей - К-ближайших соседей (kNN - k Nearest Neighbours) ищет ближайшие объекты с известными значения целевой переменной и основывается на хранении данных в памяти для сравнения с новыми элементами. Алгоритм находит расстояния между запросом и всеми примерами в данных, выбирая определенное количество примеров (k), наиболее близких к запросу, затем голосует за наиболее часто встречающуюся метку (в случае задачи классификации) или усредняет метки (в случае задачи регрессии).

Достоинства метода: прост в реализации и понимании полученных результатов; имеет низкую чувствительность к выбросам; не требует построения модели; допускает настройку нескольких параметров; позволяет делать дополнительные допущения; универсален; находит лучшее решение из возможных; решает задачи небольшой размерности.

Недостатки метода: замедляется с ростом объёма данных; не создаёт правил; не обобщает предыдущий опыт; основывается на всем массиве доступных исторических данных; невозможно сказать, на каком основании строятся ответы; сложно выбрать близость метрики; имеет высокую зависимость результатов классификации от выбранной метрики; полностью перебирает всю обучающую выборку при распознавании; имеет вычислительную трудоёмкость.

Дерево принятия решений (DecisionTreeRegressor) – метод автоматического анализа больших массивов данных.  Это инструмент принятия решений, в котором используется древовидная структура, подобная блок-схеме, или модель решений и всех их возможных результатов, включая результаты, затраты и полезность. Дерево принятия решений - эффективный инструмент интеллектуального анализа данных и предсказательной аналитики. Алгоритм дерева решений подпадает под категорию контролируемых алгоритмов обучения. Он работает как для непрерывных, так и для категориальных выходных переменных. Правила генерируются за счёт обобщения множества отдельных наблюдений (обучающих примеров), описывающих предметную область.  Регрессия дерева решений отслеживает особенности объекта и обучает модель в структуре дерева прогнозированию данных в будущем для получения значимого непрерывного вывода. Дерево решений один из вариантов решения регрессионной задачи, в случае если зависимость в данных не имеет очевидной корреляции.

Достоинства метода: помогают визуализировать процесс принятия решения и сделать правильный выбор в ситуациях, когда результаты одного решения влияют на результаты следующих решений; создаются по понятным правилам; просты в применении и интерпретации; заполняют пропуски в данных наиболее вероятным решением; работают с разными переменными; выделяют наиболее важные поля для прогнозирования;

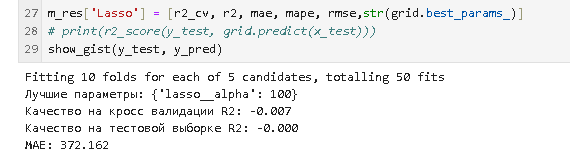
Недостатки метода: ошибаются при классификации с большим количеством классов и небольшой обучающей выборкой; имеют нестабильный процесс (изменение в одном узле может привести к построению совсем другого дерева); имеет затратные вычисления; необходимо обращать внимание на размер; ограниченное число вариантов решения проблемы.

Используемые метрики качества:

R2 - коэффициент детерминации измеряет долю дисперсии, объяснённую моделью, в общей дисперсии целевой переменной.

Если он близок к единице, то модель хорошо объясняет данные, если же он близок к нулю, то качество прогноза идентично средней величине целевой переменной (т.е. очень низкое). Отрицательные значение коэффициента детерминации означают плохую объясняющую способность модели.

Рисунок 3 - фрагмент кода для метрики R2 модели Лассо



MSE (Mean Squared Error) или средняя квадратичная ошибка принимает значениях в тех же единицах, что и целевая переменная. Чем ближе к нулю MSE, тем лучше работают предсказательные качества модели.

* 1. **Разведочный анализ данных**

Для работы моделей машинного обучения, необходимо обработать и очистить входные данные, т.к. они могут содержать искажения и пропущенные значения – это ненадёжно, поскольку способно привести к крайне неверным результатам по итогам моделирования.

Цель разведочного анализа - получение первоначальных представлений о характерах распределений переменных исходного набора данных, формирование оценки качества исходных данных (наличие пропусков, выбросов), выявление характера взаимосвязи между переменными с целью последующего выдвижения гипотез о наиболее подходящих для решения задачи моделях машинного обучения.

Рисунок 5 - тест на уникальность строк

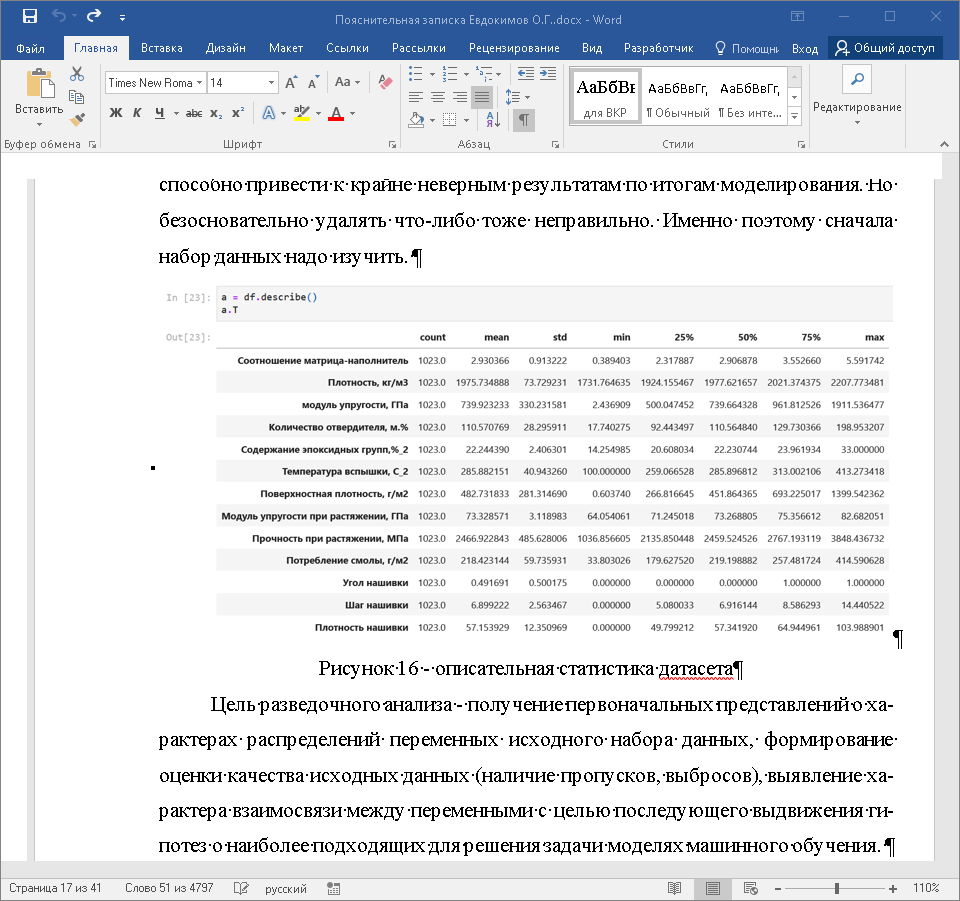
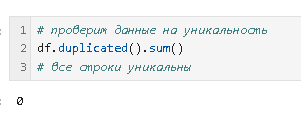
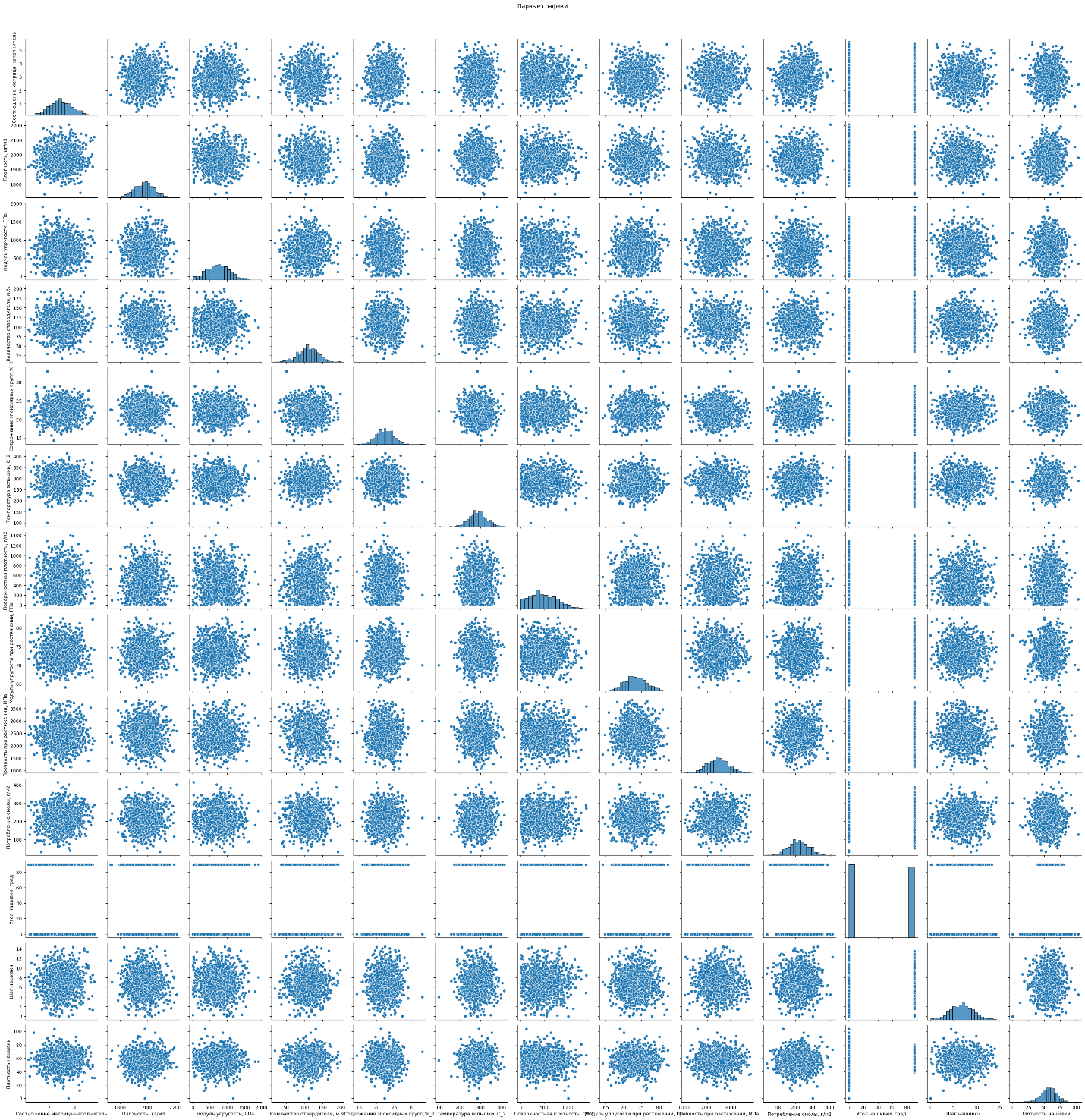


Рисунок 4 - описательная статистика датасета

В качестве инструментов разведочного анализа используется: оценка статистических характеристик датасета; гистограммы распределения каждой из переменной; диаграммы ящика с усами; попарные графики рассеяния точек; график «квантиль-квантиль»; тепловая карта; описательная статистика для каждой переменной; анализ и полное исключение выбросов; проверка наличия пропусков и дубликатов; ранговая корреляция Пирсона.

Рисунок 6 - попарные графики рассеяния



Гистограммы используются для изучения распределений частот значений переменных. Мы видим очень слабую корреляцию между переменными.

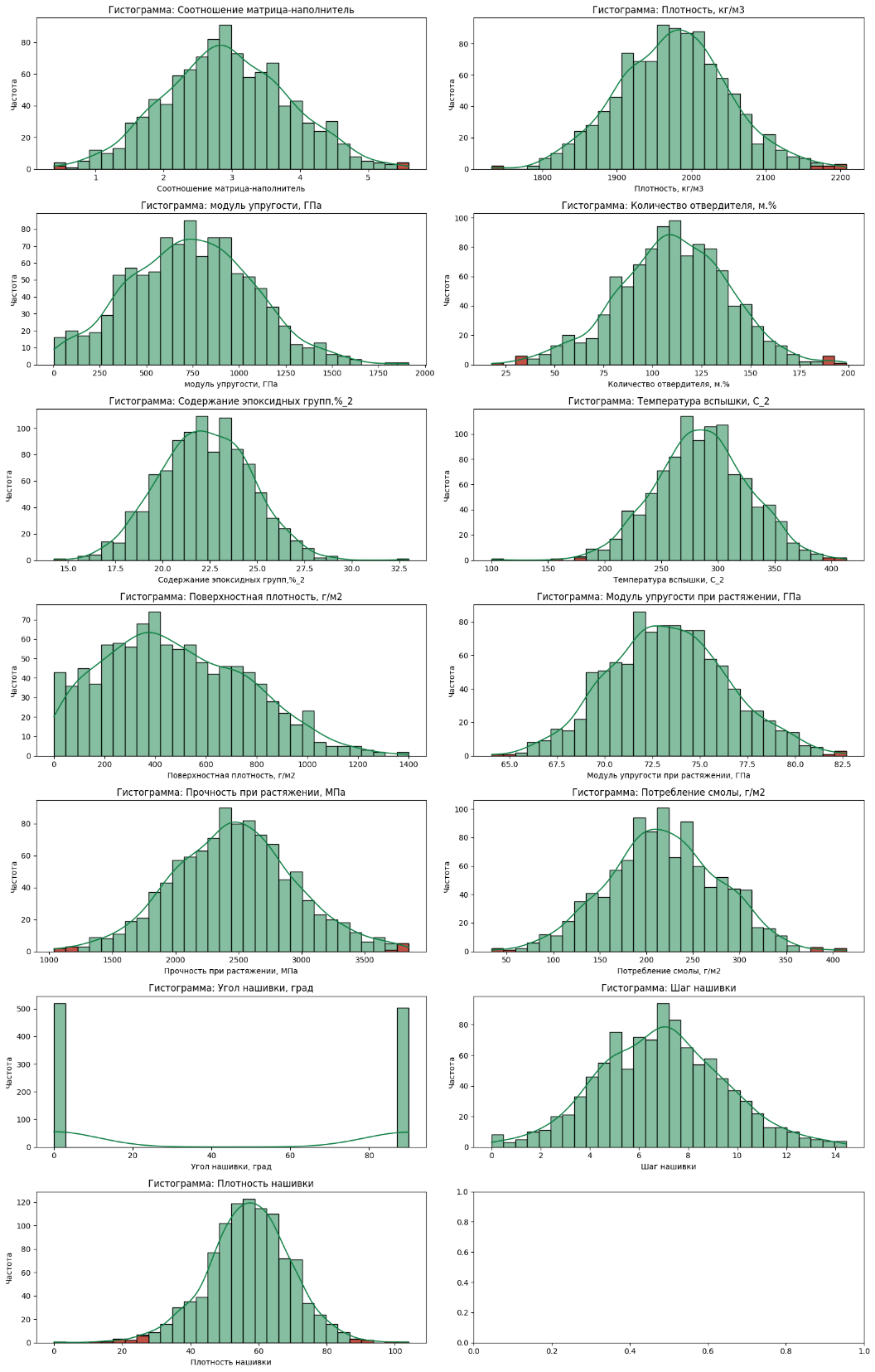


Рисунок 7 - гистограммы распределения

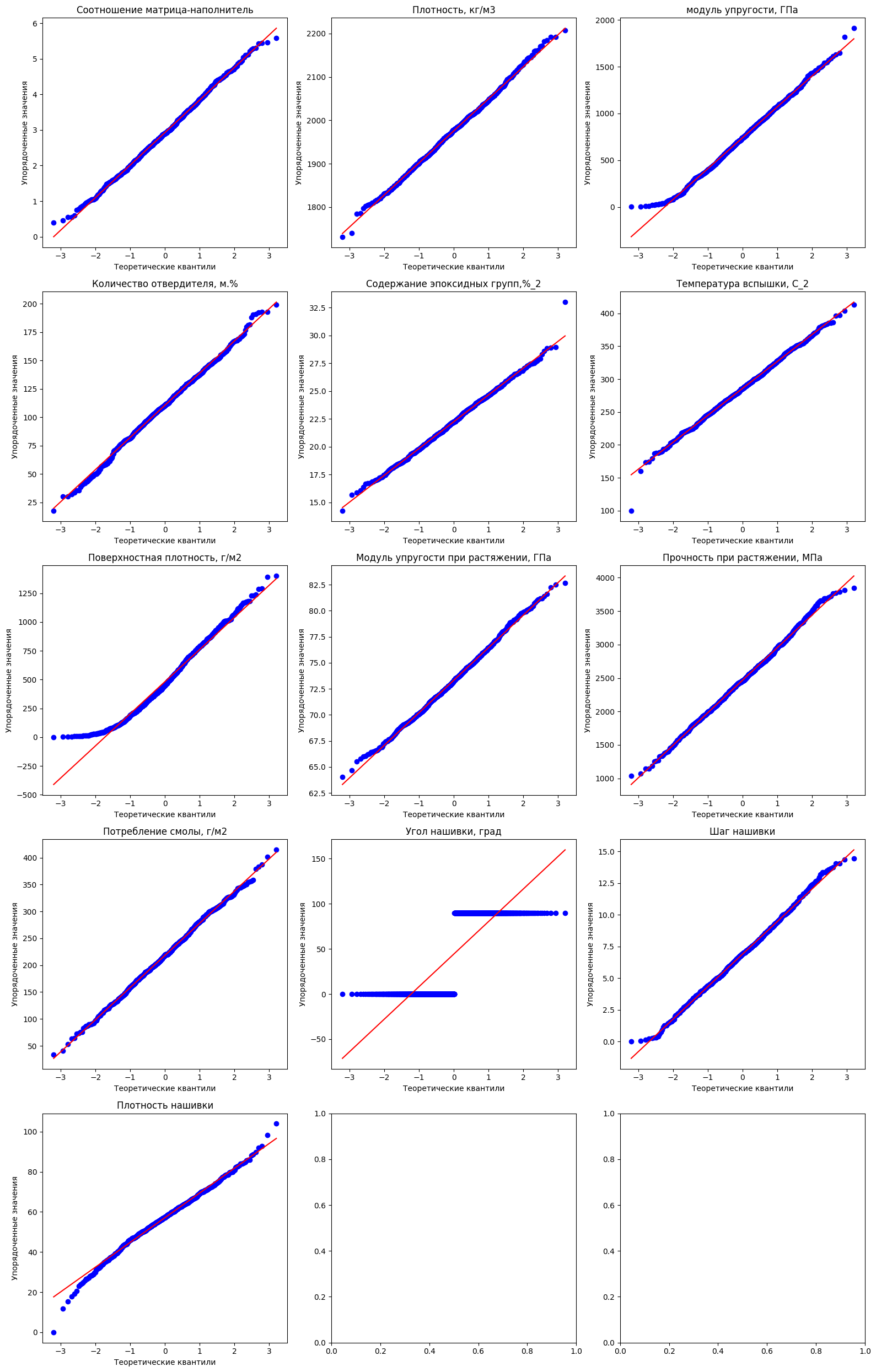


Рисунок 8 - графики «квантиль-квантиль»

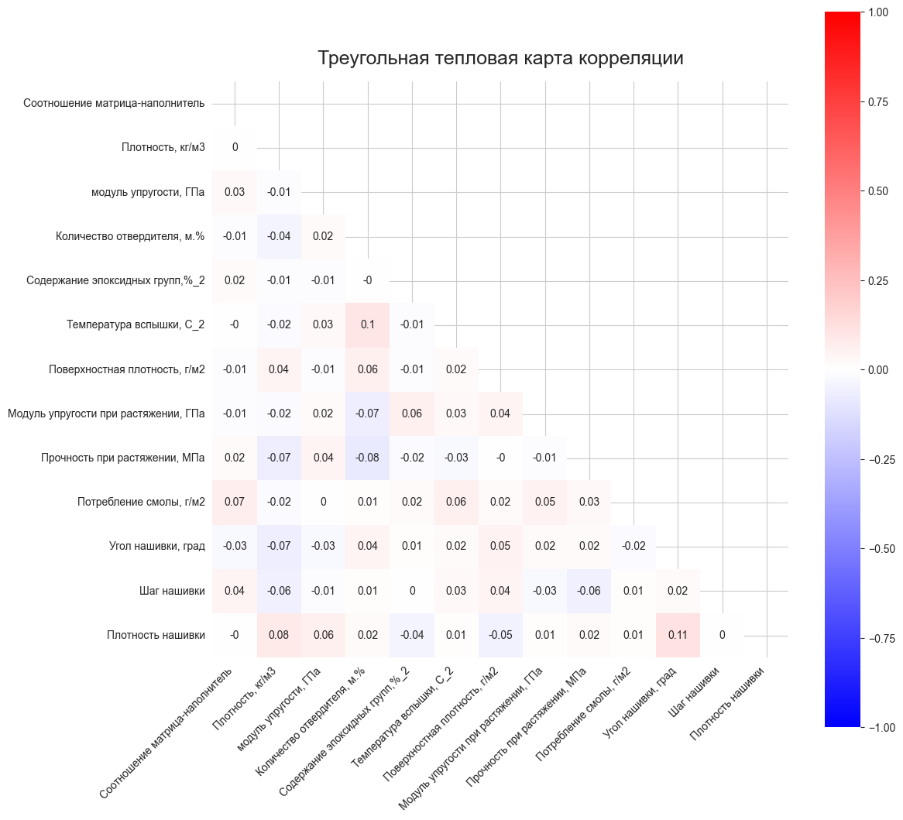
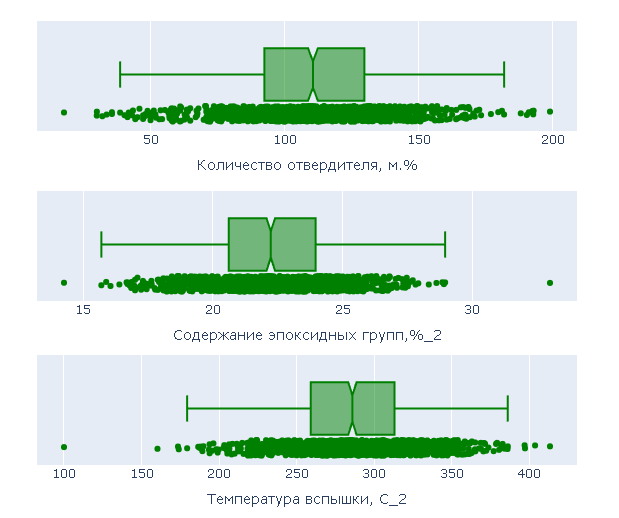


Рисунок 10 - тепловая карта с корреляцией данных

Рисунок 9- графики ящик с усами для признаков с явными выбросами



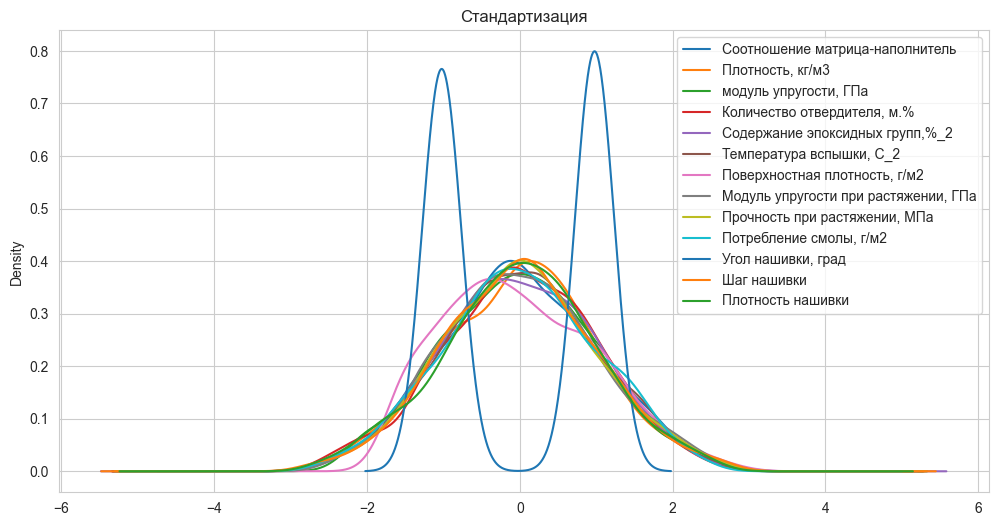
Для определения выбросов данных мы будем использовать метод межквартильного расстояния.

Данные объединённого датасета не имеют чётко выраженной зависимости, что подтверждает тепловая карта с матрицей корреляции и матрицы диаграмм рассеяния.

1. **Практическая часть**
   1. **Предобработка данных**

Выполним стандартизацию данных с помощью StandardScaler().

Рисунок 11 - распределение признаков после стандартизации



* 1. **Разработка и обучение модели**

Разработка и обучение моделей машинного обучения осуществлялась для двух выходных параметров: «Прочность при растяжении» и «Модуль упругости при растяжении» отдельно. Поиск гиперпараметров моделей производился с помощью GridSearchCV() с предварительной стандартизацией входных данных. Для модели RandomForestRegressor дополнительно применялось RandomizedSearchCV() для охвата большего диапазона параметров. Входные данные разделялись на тренировочную и тестовую части 70/ 30.

* 1. **Тестирование модели**

Тестирование моделей производилось на тестовой выборке. Результаты приведены в Таблицах 1, 2.

## Таблица 1 - Результаты тестирования для целевой переменной Прочность при растяжении, МПа.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **r2\_cv** | **r2** | **MAE** | **MAPE** | **RMSE** |
| **Lasso** | -0,00716 | -0,00036 | 372,1623 | 0,16188 | 461,4009 |
| **SVR** | -0,007 | 0,00105 | 372,8447 | 0,161404 | 461,075 |
| **RFR** | -0,03331 | 0,022693 | 371,9613 | 0,161362 | 456,0529 |
| **gbr** | -0,14254 | -0,02831 | 381,8662 | 0,164447 | 467,8013 |
| **KNN** | -0,00863 | 0,003327 | 372,7264 | 0,162499 | 460,5492 |
| **DTR\_cv** | 0,001155 | -0,02937 | 378,1114 | 0,164635 | 468,0422 |
| **DTR\_rcv** | 0,004835 | 0,00332 | 372,8105 | 0,161538 | 460,5508 |

## Таблица 2 - Результаты тестирования для целевой переменной Модуль упругости при растяжении, ГПа

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **r2\_cv** | **r2** | **MAE** | **MAPE** | **RMSE** |
| **Lasso** | -0,02764 | -0,00440 | 2,38528 | 0,03266 | 2,92255 |
| **SVR** | -0,02435 | -0,01194 | 2,39512 | 0,03285 | 2,93350 |
| **RFR** | -0,07777 | -0,01245 | 2,41823 | 0,03312 | 2,93424 |
| **gbr** | -0,14217 | -0,13818 | 2,52988 | 0,03466 | 3,11111 |
| **KNN** | -0,02839 | -0,00953 | 2,40081 | 0,03289 | 2,93000 |
| **DTR\_cv** | -0,01601 | -0,00572 | 2,40591 | 0,03296 | 2,92447 |
| **DTR\_rcv** | -0,01266 | 0,00211 | 2,38597 | 0,03267 | 2,91307 |

## Все результаты не удовлетворительны.

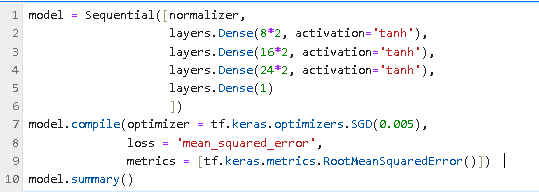
* 1. **Написать нейронную сеть, которая будет рекомендовать соотношение «матрица – наполнитель».**

Нейронную сеть строилась с помощью класса keras.Sequential. На входе сети будем использовать обученный слой layers.Normalization для стандартизации входных данных.

Разработка сети производилась в три этапа:

На этапе 1 создана сеть со структурой, представленной на Рисунке 12:

Рисунок 12 - структура нейронной сети на этапе 1



Далее сеть обучалась на протяжении 500 эпох с целью понимания процесса обучения.

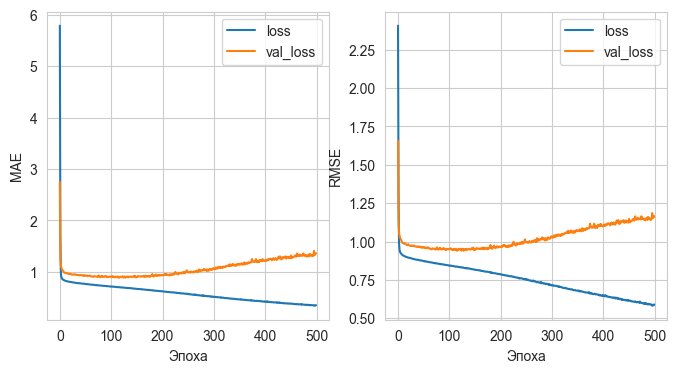


Рисунок 13 - Кривые обучения сети на этапе 1

Кривые показывают переобучение сети.

На этапе 2 сеть была дополнена тремя слоями Dropout и имеет структуру согласно Рисунку 14

Качество на валидации стало лучше, ситуация с переобучением стала лучше

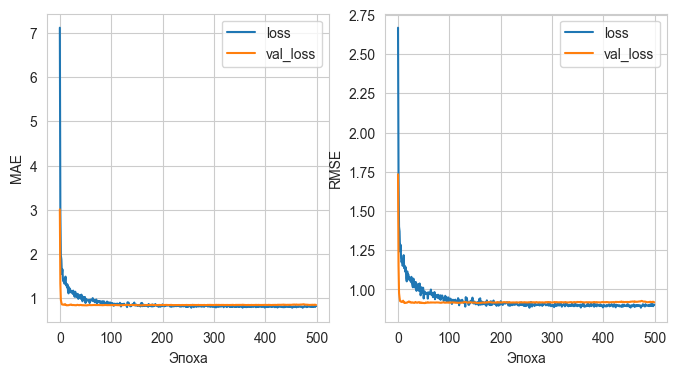


Рисунок 15 - Кривые обучения сети на этапе 2

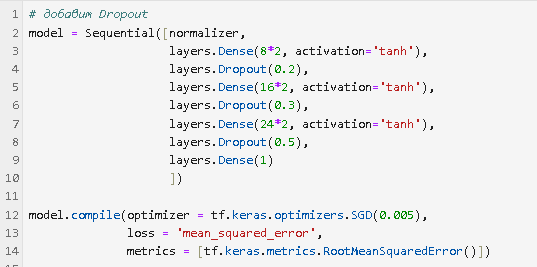
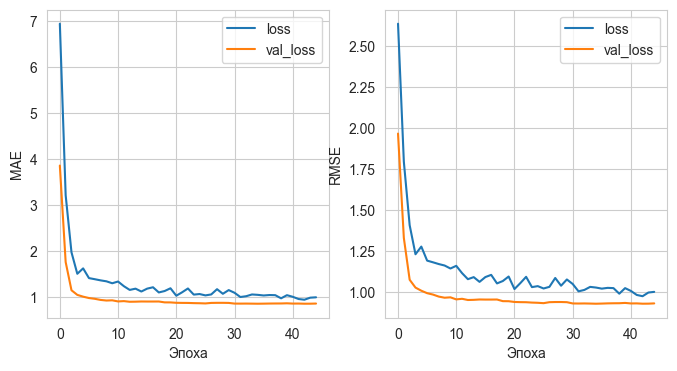


Рисунок 14 - структура нейронной сети на этапе 2

На этапе 3 сеть была дополнена критерием останова: качество не улучается на протяжении 10 эпох.

Обучение остановлено в районе 50-ой эпохи.

Рисунок 16 - Кривые обучения сети на этапе 3



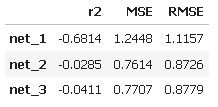
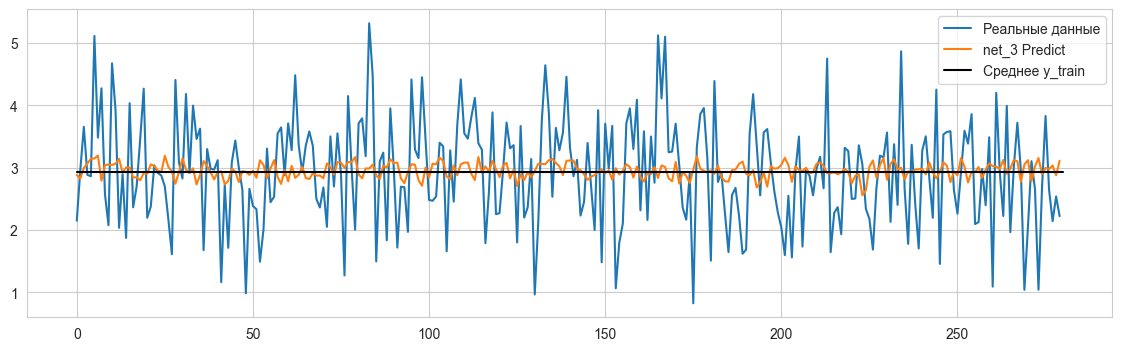


Рисунок 17 - Итоговые результаты обучения сетей

Выведем график прогноза сети net\_3

Рисунок 18 - график прогноза сети net\_3



* 1. **Разработка приложения**

Приложение основано на Framework Flask. Запуск приложения осуществляется через командную строку: python main.py. Далее в браузере нужно перейти по адресу: <http://127.0.0.1:5000/>. Откроется форма приложения с предварительно заполненными значениями одного из примеров входного датасета. Далее нужно нажать кнопку "Получить прогноз".

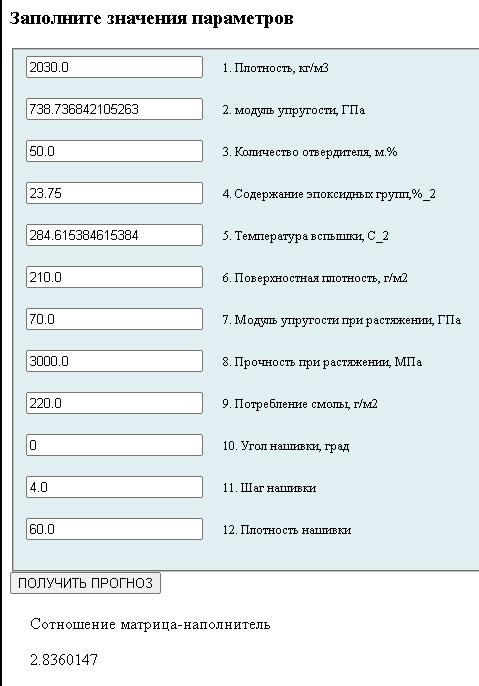


Рисунок 18 - скриншот приложения

* 1. **Создание удалённого репозитория и загрузка**

Репозиторий был создан на github.com по адресу:

<https://github.com/kiber7575/VKR_Radkov_DV>

* 1. **Заключение**

Данная исследовательская работа позволяет сделать некоторые основные выводы по теме. Распределение полученных данных в объединённом датасете близко к нормальному, но коэффициенты корреляции между парами признаков стремятся к нулю. Использованные при разработке моделей подходы не позволили получить сколько-нибудь достоверных прогнозов. Применённые модели регрессии не показали высокой эффективности в прогнозировании свойств композитов. Лучшие метрики для модуля упругости при растяжении, ГПа – метод опорных векторов, для прочности при растяжении, МПа - лассо-регрессия.

Был сделан вывод, что невозможно определить из свойств материалов соотношение «матрица – наполнитель». Данный факт не указывает на то, что прогнозирование характеристик композитных материалов на основании предоставленного набора данных невозможно, но может указывать на недостатки базы данных, подходов, использованных при прогнозе, необходимости пересмотра инструментов для прогнозирования.

* 1. **Список используемой литературы и веб ресурсы.**

1. Сара Бослаф. Статистика для всех. 2015 г. - 586 с.
2. Sebastian Raschka, Yuxi (Hayden) Liu, Vahid Mirjalili. Machine Learning with PyTorch and Scikit-Learn. 2022 г. - 771 с.
3. В.Ш. Берикашвилли, С.П. Оськин - Статистическая обработка данных, планирование эксперимента и случайные процессы. 2022 - 164с.
4. ГрасД. Data Science. Наука о данных с нуля. 2021 - 416 с.
5. Жерон, Орельен - Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. 2018 - 688 с.