

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
по курсу
«Data Science Pro»

Слушатель

Маслов Михаил Дмитриевич

Москва, 2025

Содержание:

Введение.....	3
1. Аналитическая часть.....	5
1.1. Постановка задачи.....	5
1.2. Описание используемых методов.....	7
1.3. Разведочный анализ данных.....	15
2. Практическая часть.....	20
2.1. Предобработка данных.....	21
2.2. Разработка и обучение модели.....	26
2.3. Тестирование модели.....	27
2.4. Нейронная сеть рекомендации соотношения матрицы-наполнителя.....	31
2.5. Разработка приложения.....	34
2.6. Commit на GitHub и размещение на веб-хостинг.....	36
Заключение.....	38
Библиографический список.....	40

Введение

Выпускная квалификационная работа посвящена прогнозированию конечных свойств новых материалов (композиционных материалов). Композиционные материалы — это искусственно созданные материалы, состоящие из нескольких других с четкой границей между ними. Композиты обладают теми свойствами, которые не наблюдаются у компонентов по отдельности. При этом композиты являются монолитным материалом, т. е. компоненты материала неотделимы друг от друга без разрушения конструкции в целом. Яркий пример композита — железобетон. Бетон прекрасно сопротивляется сжатию, но плохо растяжению. Стальная арматура внутри бетона компенсирует его неспособность сопротивляться сжатию, формируя тем самым новые, уникальные свойства. Современные композиты изготавливаются из других материалов: полимеры, керамика, стеклянные и углеродные волокна, но данный принцип сохраняется. У такого подхода есть и недостаток: даже если известны характеристики исходных компонентов, определить характеристики композита, состоящего из этих компонентов, достаточно проблематично. Для решения этой проблемы есть два пути: физические испытания образцов материалов или прогнозирование характеристик. Физическое испытание само по себе требует много ресурсов, денежных и трудовых, а также время. Но для сокращения затрат на научно-исследовательские и опытно-конструкторские работы можно начать с компьютерного моделирования будущего композита с требуемыми свойствами. Суть прогнозирования заключается в симуляции представительного элемента объема композита на основе данных о характеристиках входящих компонентов (связующего и армирующего компонента).

Таким образом, актуальность данной темы подтверждается важностью сокращения расходов и ускорением разработки новых материалов для

достижения эффективности в деятельности предприятий строительной отрасли и освоении новых территорий. К примеру, перспективной территории Арктики, где предполагаются работы по освоению и разработке месторождений полезных ископаемых, включая международные проекты, что само по себе предполагает колоссальный бюджет. Так, созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

Так, для достижения цели ознакомления с полем деятельности специалиста DataScience, в ходе работы были созданы различные проекты прогноза. На входе имеются данные о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.). На выходе необходимо спрогнозировать ряд конечных свойств получаемых композиционных материалов.

Задачи данной работы включают в себя:

- 1) Изучение теоретических основ и методов Data Science;
- 2) Проведение разведочного анализа предложенных данных;
- 3) Проведение предобработки данных;
- 4) Обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении;
- 5) Написания нейронной сети recommending соотношение матрица-наполнитель;
- 6) Ознакомление с распределённой системой управления версиями (Git).

1. Аналитическая часть

1.1. Постановка задачи

Для работы имеются 2 датасета: X_{br} размерности 1023×10 как представлено на Рисунке 1 и X_{nup} размерности 1040×3 как представлено на Рисунке 2, далее требуется объединить оба датасета, предполагая, что нумерация (индекс) является первичным ключом для одного и внешним для второго датасета, тип объединения – Inner, которая в данном случае предполагала присоединение двух матриц с сохранением всех столбцов, но с количеством строк первой матрицы X_{br} . По итогам объединения получилась матрица с количеством строк первого датасета и суммарным количеством столбцов (1023×13), структура которого показана на Рисунке 3.

```
x_br.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1023 entries, 0 to 1022
Data columns (total 10 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Соотношение матрица-наполнитель          1023 non-null   float64
1   Плотность, кг/м3                          1023 non-null   float64
2   модуль упругости, ГПа                     1023 non-null   float64
3   Количество отвердителя, м.%               1023 non-null   float64
4   Содержание эпоксидных групп, %_2         1023 non-null   float64
5   Температура вспышки, C_2                  1023 non-null   float64
6   Поверхностная плотность, г/м2             1023 non-null   float64
7   Модуль упругости при растяжении, ГПа      1023 non-null   float64
8   Прочность при растяжении, МПа             1023 non-null   float64
9   Потребление смолы, г/м2                   1023 non-null   float64
dtypes: float64(10)
memory usage: 87.9 KB
```

Рисунок 1 – Структура датасета X_{br}

```
x_nup.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1040 entries, 0 to 1039
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Угол нашивки, град    1040 non-null   int64
1   Шаг нашивки           1040 non-null   float64
2   Плотность нашивки     1040 non-null   float64
dtypes: float64(2), int64(1)
memory usage: 32.5 KB
```

Рисунок 2 – Структура датасета X_{nup}

Далее была проведена проверка уникальных значений объединенного датасета df.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1023 entries, 0 to 1022
Data columns (total 13 columns):
#   Column                                     Non-Null Count  Dtype  
---  -
0   Соотношение матрица-наполнитель          1023 non-null   float64
1   Плотность, кг/м3                          1023 non-null   float64
2   модуль упругости, ГПа                     1023 non-null   float64
3   Количество отвердителя, м.%               1023 non-null   float64
4   Содержание эпоксидных групп,%_2          1023 non-null   float64
5   Температура вспышки, C_2                  1023 non-null   float64
6   Поверхностная плотность, г/м2             1023 non-null   float64
7   Модуль упругости при растяжении, ГПа      1023 non-null   float64
8   Прочность при растяжении, МПа             1023 non-null   float64
9   Потребление смолы, г/м2                   1023 non-null   float64
10  Угол нашивки, град                         1023 non-null   int64   
11  Шаг нашивки                               1023 non-null   float64
12  Плотность нашивки                         1023 non-null   float64
dtypes: float64(12), int64(1)
memory usage: 111.9 KB
```

Рисунок 3 – Структура датасета df

```
df.nunique()

Соотношение матрица-наполнитель          1014
Плотность, кг/м3                          1013
модуль упругости, ГПа                     1020
Количество отвердителя, м.%               1005
Содержание эпоксидных групп,%_2          1004
Температура вспышки, C_2                  1003
Поверхностная плотность, г/м2             1004
Модуль упругости при растяжении, ГПа      1004
Прочность при растяжении, МПа             1004
Потребление смолы, г/м2                   1003
Угол нашивки, град                         2
Шаг нашивки                               989
Плотность нашивки                         988
dtype: int64
```

Рисунок 4 – количество уникальных значений датасета df

По итогу первоначального осмотра не выявлено пропусков в данных, объединенный датасет стал иметь 13 столбцов и 1023 строки, что можно рассматривать как 13 переменных и 1023 их значения. Всю выборку предстоит на этапе разведочного анализа проверить на выбросы, корреляцию, задвоенные значения, а потом очистить и масштабировать, чтобы подготовить к дальнейшим процедурам машинного обучения. Также стоит отметить, что данные числовые и непрерывные, поэтому можно воспользоваться методами регрессионного анализа для построения моделей прогноза.

1.2. Описание используемых методов

В ходе работы на разных стадиях требуются различные методы. Так для разведочного анализа данных предстоит провести корреляционный анализ. Так, корреляционный анализ – статистический метод изучения взаимосвязи между двумя и более случайными величинами. Корреляционный анализ применяется для оценки степени линейной зависимости между парами факторов, производится с целью отбора и предобработки входных полей для использования в обучаемых на данных моделях. При построении моделей (например, в машинном обучении) отсутствие корреляционного анализа может привести к использованию нерелевантных факторов, что снизит точность модели.

Корреляционный анализ методами Пирсона, Спирмена и Кендалла – это три разных метода измерения взаимосвязи между переменными, но они отличаются по типу измеряемой связи и типу данных, для которых подходят. Корреляция Пирсона измеряет линейную взаимосвязь, в то время как Спирмена и Кендалла – монотонную (как меняется один показатель относительно другого, не обязательно линейно). Пирсона работает с количественными данными, а Спирмена и Кендалла могут работать с ранговыми данными или данными, которые не имеют нормального распределения.

Таблица 1 – Сравнение методов корреляции

Критерий	Пирсон	Спирмен	Кендалл
1	2	3	4
Тип связи	Только линейная	Монотонная	Монотонная
Требования к данным	Нормальное распределение	Любое распределение	Любое распределение

Продолжение таблицы 1

1	2	3	4
Устойчивость к выбросам	Низкая	Средняя	Высокая
Шкала данных	Интервальная/отношенческая	Порядковая и выше	Порядковая и выше
Интерпретация	Сила линейной зависимости	Сила монотонной зависимости	Вероятность согласованности
Сильные стороны	Точен для линейных зависимостей	Лучше для нелинейных данных	Устойчив к выбросам и связям
Слабые стороны	Чувствителен к нарушениям предпосылок	Теряет информацию при ранжировании	Медленный для больших данных

Таким образом, на основании сравнения методов корреляции, представленного в Таблице 1, можно сделать вывод, что для первоначального анализа больше подходят методы Спирмена и Кендалла.

После определения наличия зависимых переменных последует выявление выбросов. Для их обнаружения и очистки датасетов полезно применять метод IQR (Межквартильных расстояний) и как альтернативный метод будет использован IsolationForest.

Выявление выбросов — одна из старейших проблем анализа данных. Причиной появления выбросов могут быть ошибки измерений, ошибки отбора выборки, преднамеренное искажение или некорректная фиксация результатов анализа выборки, ошибочные предположения о распределении данных или модели, малое количество наблюдений и так далее. В свою очередь такие ошибки могут повлиять на точность и предсказательную способность прогнозных моделей. Для определения выбросов применяется метод межквартильных расстояний (IQR). Принято начинать с его

графического отображения в виде боксплотов, также известный как диаграмма размаха или ящик с усами, это графический метод представления распределения числовых данных. Он показывает медиану, квартили, минимальные и максимальные значения, а также выбросы, что позволяет быстро оценить и сравнить распределения различных наборов данных. Боксплот состоит из "ящика" (который представляет межквартильный размах, т.е. разницу между 25-м и 75-м перцентилями) и "усов", которые указывают на минимальное и максимальное значения, не являющиеся выбросами. Медиана, или 50-й перцентиль, обычно отмечается внутри ящика. Выбросы, то есть значения, выходящие за пределы усов (обычно за $1.5 \cdot$ межквартильный размах от границы ящика), отображаются отдельными точками. Сам метод IQR (межквартильный размах) — это статистический метод для определения выбросов в данных и измерения статистической дисперсии. Основан на квартилях распределения: Q1 (25-й перцентиль): значение, ниже которого расположено 25% данных, Q3 (75-й перцентиль): значение, ниже которого расположено 75% данных.

Также одним из эффективных способов обнаружения выбросов в высокоразмерных наборах данных является использование случайных лесов. Ансамблевый метод IsolationForest “изолирует” наблюдения путем случайного выбора признака, а затем случайного выбора значения разбиения между максимальным и минимальным значениями выбранного признака. Поскольку рекурсивное разбиение можно представить в виде древовидной структуры, количество разбиений, необходимых для изоляции выборки, равно длине пути от корневого узла до конечного узла. Эта длина пути, усредненная по лесу таких случайных деревьев, является мерой нормальности и нашей функцией принятия решения. Случайное разбиение дает заметно более короткие пути для аномалий. Следовательно, если лес случайных деревьев в совокупности дает более короткие длины путей для определенных образцов, они с высокой вероятностью являются аномалиями.

Таблица 2 – Сравнение методов борьбы с выбросами

Критерий	IQR	Isolation Forest
Принцип	Статистический (квартили)	Алгоритм ML (ансамбль деревьев)
Измерения	Только одномерные данные	Многомерные данные
Чувствительность	Только к крайним значениям	К любым атипичным паттернам
Взаимосвязи признаков	Не учитывает	Автоматически учитывает
Производительность	Очень быстрый (миллисекунды)	Средняя (зависит от размера данных)
Интерпретируемость	Высокая	Низкая ("черный ящик")
Распределение данных	Не требует предположений	Не требует предположений
Типичное применение	EDA, очистка данных	Мониторинг систем, обнаружение мошенничества
Плюсы	<ul style="list-style-type: none"> - Простота реализации - Наглядность - Нет параметров 	<ul style="list-style-type: none"> - Работает с многомерными данными - Обнаружение сложных паттернов - Хорошая масштабируемость
Минусы	<ul style="list-style-type: none"> - Игнорирует корреляции - Пропускает кластерные выбросы - Не работает для $n > 1$ 	<ul style="list-style-type: none"> - Случайность влияет на результат - Требуется подбора параметров - Ресурсоемкость для big data

Поскольку представленный датасет не является большим для обработки и можно рисовать боксплоты, то подойдет метод IQR, однако, IsolationForest тоже стоит применить ради сравнения.

Для масштабирования данных предстоит применить метод стандартизации StandardScaler и метод нормализации MinMaxScaler. StandardScaler преобразует данные так, чтобы распределение имело среднее

$= 0$ и стандартное отклонение $= 1$. Его ключевыми особенностями является то, что он центрирует данные вокруг 0, сохраняет форму исходного распределения, не ограничивает диапазон значений, чувствителен к выбросам. Лучше всего подходит для линейных моделей (регрессия, SVM); методов, основанных на расстояниях (KNN, K-means); PCA и других методов снижения размерности; когда данные, близкие к нормальному распределению.

В свою очередь MinMaxScaler сжимает данные в фиксированный диапазон обычно от 0 до 1. Его ключевыми особенностями является то, что он гарантирует фиксированные границы значений; искажает распределение при наличии выбросов; сохраняет исходное соотношение расстояний, по умолчанию диапазон от 0 до 1, но можно задать иной, например, от -1 до 1. Наиболее уместными областями его применения являются: нейронные сети (особенно входные слои); алгоритмы, требующие единого масштаба (градиентный спуск); изображения (пиксельные значения); древовидные алгоритмы (Decision Trees, Random Forest)

Таблица 3 – Сравнение методов масштабирования данных

Критерий	Standard Scaler	MinMax Scaler
1	2	3
Тип преобразования	Стандартизация (z-score)	Нормализация (масштабирование диапазона)
Центрирование	Среднее $= 0$	Не центрирует
Диапазон	Неограничен	Фиксированный
Чувствительность к выбросам	Высокая	Очень высокая
Сохранение распределения	Да	Нет (меняет форму)
Интерпретируемость	Стандартные отклонения от среднего	Относительные позиции в диапазоне

Продолжение Таблицы 3

1	2	3
Лучшие алгоритмы	SVM, Линейная регрессия, K-means	Нейронные сети, KNN
Плюсы	Сохраняет выбросы (информативно)	-Гарантированный диапазон значений -Быстрые вычисления
Минусы	Не гарантирует границы значений	-Искажает данные при выбросах - Плохо для ненормальных распределений

Основываясь на сравнении методов, более уместным является применение стандартизации, так как прогнозирование будет основываться на регрессионных моделях в виду имеющихся данных. Но в качестве эксперимента стоит опробовать оба метода.

Как методы машинного обучения моделей прогноза модуля упругости при растяжении и прочности при растяжении были выбраны несколько методов: линейной регрессии, «случайного леса», градиентный бустинг и метод опорных векторов для регрессии. Как завершающий этап будет написана нейронная сеть многослойного перцептрона (MLP)

Линейная регрессия моделирует зависимость между независимыми переменными (признаками) и зависимой переменной (целевой) как линейную комбинацию параметров. Модель пытается найти веса (коэффициенты) для каждой независимой переменной, чтобы минимизировать сумму квадратов ошибок (остатков) между предсказанными и фактическими значениями. Её особенностями является интерпретируемость (каждый коэффициент показывает влияние признака на цель); предполагает линейную связь; аддитивность и независимость признаков; чувствительна к мультиколлинеарности и выбросам.

Случайный лес (Random Forest Regression) – это ансамбль решающих деревьев регрессии. Каждое дерево обучается на бутстрапированной подвыборке данных, и при построении дерева рассматривается случайное подмножество признаков для разделения. Предсказание — среднее предсказаний всех деревьев. Этот метод обладает своими особенностями, к которым можно отнести устойчивость к переобучению (благодаря бэггингу и случайности признаков); может обрабатывать нелинейные зависимости; не требует масштабирования признаков; менее интерпретируема, чем линейная модель.

Градиентный бустинг (Gradient Boosting Regression) – это метод, который последовательно строит ансамбль слабых предсказателей (обычно деревьев решений). Каждая следующая модель обучается на ошибках (остатках) предыдущих моделей с целью их минимизации. Используется градиентный спуск в пространстве функций. Алгоритм его работы:

1. Инициализация начальным значением (например, среднее значение целевой переменной).
2. Для каждой итерации:
 - Вычисление остатков (градиента) для текущей модели.
 - Обучение нового базового предсказателя (дерева) на остатках.
 - Обновление модели с учетом предсказаний нового дерева (с шагом обучения).

Особенности данного метода машинного обучения заключаются в том, что часто дает высокую точность; может переобучаться при недостаточном контроле; чувствителен к шуму и выбросам.

Метод опорных векторов для регрессии (Support Vector Regression, SVR) – это метод, который ищет гиперплоскость, которая минимизирует ошибку, при этом допуская отклонения в пределах заданной ширины трубки (ϵ). Использует ядра (kernel trick) для отображения данных в пространство

более высокой размерности, где может существовать линейная гиперплоскость.

Отличительными чертами данного метода является то, что он эффективен в пространствах высокой размерности; требует тщательного подбора гиперпараметров (C , ϵ , ядро); чувствителен к масштабированию данных.

Многослойный перцептрон (MLP Regression) – это искусственная нейронная сеть прямого распространения. Состоит из входного слоя, одного или нескольких скрытых слоев и выходного слоя. Каждый нейрон применяет нелинейную функцию активации к взвешенной сумме входов.

Данный метод имеет также свои особенности, которыми является то, что он может аппроксимировать сложные нелинейные функции; требует много данных и вычислительных ресурсов; чувствителен к масштабированию данных и начальным весам; склонен к переобучению.

Таблица 4 – Сравнение методов машинного обучения

Критерий	Линейная регрессия	Random Forest	Gradient Boosting	SVR	MLP
1	2	3	4	5	6
Точность	Низкая (линейные задачи)	Высокая	Очень высокая	Средняя-высокая	Очень высокая
Интерпретируемость	Отличная	Средняя (важность признаков)	Средняя	Низкая	Очень низкая
Время обучения	Мгновенное	Среднее	Длительное	Длительное (большие данные)	Очень длительное
Параметры настройки	2-3 (регуляризация)	5-7 (деревья, глубина)	10+ (скорость, глубина, др.)	5+ (C , ϵ , ядро, γ)	15+ (слои, нейроны, LR, др.)

Продолжение Таблицы 4

1	2	3	4	5	6
Требования к данным	Жесткие (линейность, нормальность)	Минимальные	Минимальные	Масштабирование	Масштабирование
Устойчивость к шуму	Низкая	Высокая	Средняя	Высокая (ε-трубка)	Средняя
Переобучение	Редко	Возможно (без ограничений)	Контролируется шагом	Контролируется C	Высокий риск
Плюсы	Простота, скорость, интерпретация	Робастность, параллелизм	Лучшая точность, гибкость	Теория ВПР, эффективность	Универсальность, сложные зависимости
Минусы	Только линейные зависимости	Случайность, память	Долгое обучение, настройка	Медленный инференс, ядра	Черный ящик, ресурсоемкость
Лучшие сценарии	Быстрый прототип, линейные данные	Табличные данные, баланс скорости/качества	Соревнования, точность	Малые/средние наборы данных	Изображения, текст, сложные сигналы

Все данные методы являются методами работы с числовыми непрерывными данными, что делает их отличным выбором для регрессионного анализа данных, представленных в датасете. По факту применения данных методов машинного обучения будут построены прогнозные модели необходимых параметров.

1.3. Разведочный анализ данных

Зная, что тип данных в датасете уже пригоден для дальнейших вычислений (float64, int64), а угол нашивки имеет только 2 значения, то этот столбец был приведен к бинарному виду для удобства масштабирования, а

датасет скопирован и переименован в ds, чтобы оригинальный остался неизменным.

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп, %_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки	Шаг нашивки	Плотность нашивки
count	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000
mean	2.930366	1975.734888	739.923233	110.570769	22.244390	285.882151	482.731833	73.328571	2466.922843	218.423144	0.491691	6.899222	57.153929
std	0.913222	73.729231	330.231581	28.295911	2.406301	40.943260	281.314690	3.118983	485.628006	59.735931	0.500175	2.563467	12.350969
min	0.389403	1731.764635	2.436909	17.740275	14.254985	100.000000	0.603740	64.054061	1036.856605	33.803026	0.000000	0.000000	0.000000
25%	2.317887	1924.155467	500.047452	92.443497	20.608034	259.066528	266.816645	71.245018	2135.850448	179.627520	0.000000	5.080033	49.799212
50%	2.906878	1977.621657	739.664328	110.564840	22.230744	285.896812	451.864365	73.268805	2459.524526	219.198882	0.000000	6.916144	57.341920
75%	3.552660	2021.374375	961.812526	129.730366	23.961934	313.002106	693.225017	75.356612	2767.193119	257.481724	1.000000	8.586293	64.944961
max	5.591742	2207.773481	1911.536477	198.953207	33.000000	413.273418	1399.542362	82.682051	3848.436732	414.590628	1.000000	14.440522	103.988091

Рисунок 5 – Статистическое описание датасета ds

Далее было выяснено, что в датасете нет пропусков и задвоенных значений, что могло бы нарушить работу моделей и требовало бы заполнения пропусков и очистку датасета. Также были рассчитаны медианные значения параметров, стандартные отклонения, а так же квартили распределения, представленные на рисунке 5.

Затем был произведен корреляционный анализ датасета с целью выявления взаимосвязи показателей, с графическим представлением в виде тепловой карты, как представлено на Рисунке 6.

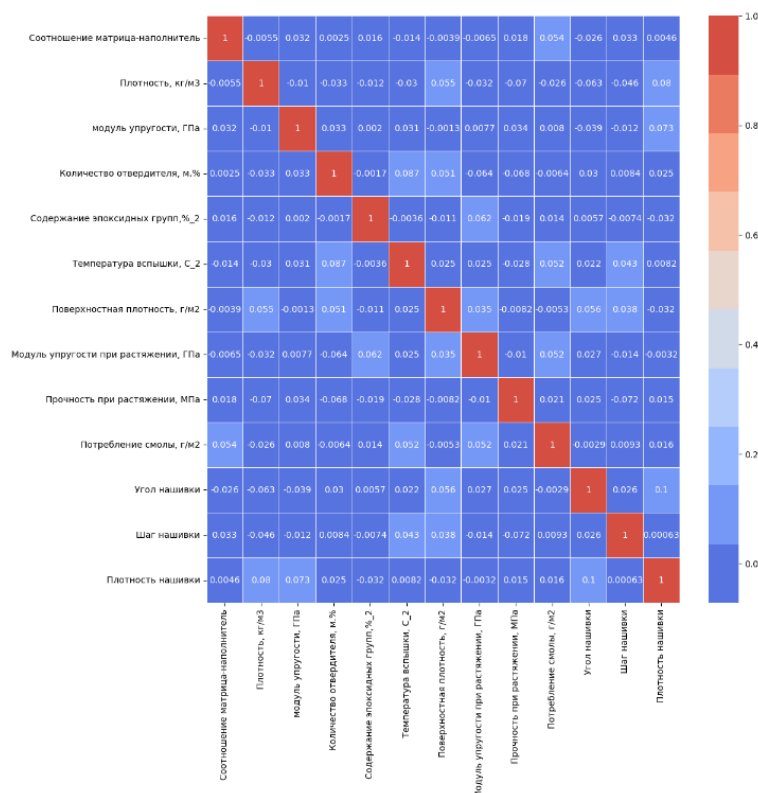


Рисунок 6 – Тепловая карта корреляций датасета ds

Так как был проведен анализ еще сырых данных, то уместнее применить методы ранговой корреляции Спирмена и Кендалла, менее чувствительные к распределению и выбросам, чем корреляция Пирсона. На Рисунке 6 представлена тепловая карта корреляций по Спирмену, но результат был схожий при каждом способе, включая корреляцию методом Пирсона, который также не показал линейной взаимосвязи переменных.

Для быстрой оценки зависимостей и распределения переменных, стоит представить данные в виде попарных графиков рассеяния точек, который представлен на Рисунке 7.

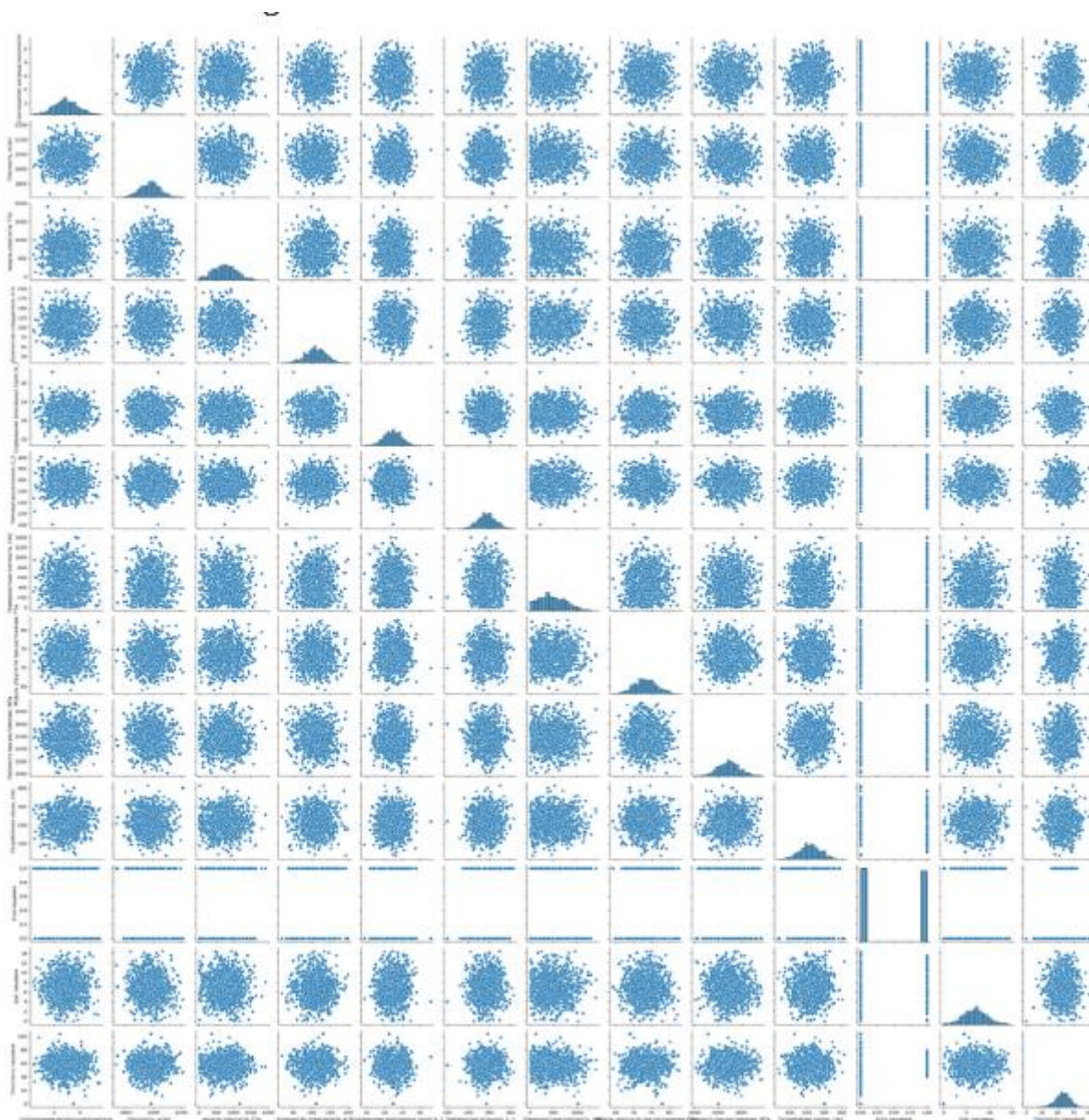


Рисунок 7 – Попарный график рассеяния датасета ds

На первый взгляд, данные не имеют явной линейной связи, а коэффициенты корреляции, рассчитанные по методам Спирмена и Кендалла ниже 0,1 по модулю, что говорит об отсутствии значимой связи между данными, так же заметны отдельные точки, которые могут быть выбросами.

Графики попарного рассеяния данных информативны, но плотность распределения в них лучше проанализировать на гистограммах, с наложением кривой плотности распределения, представленном на Рисунке 8.

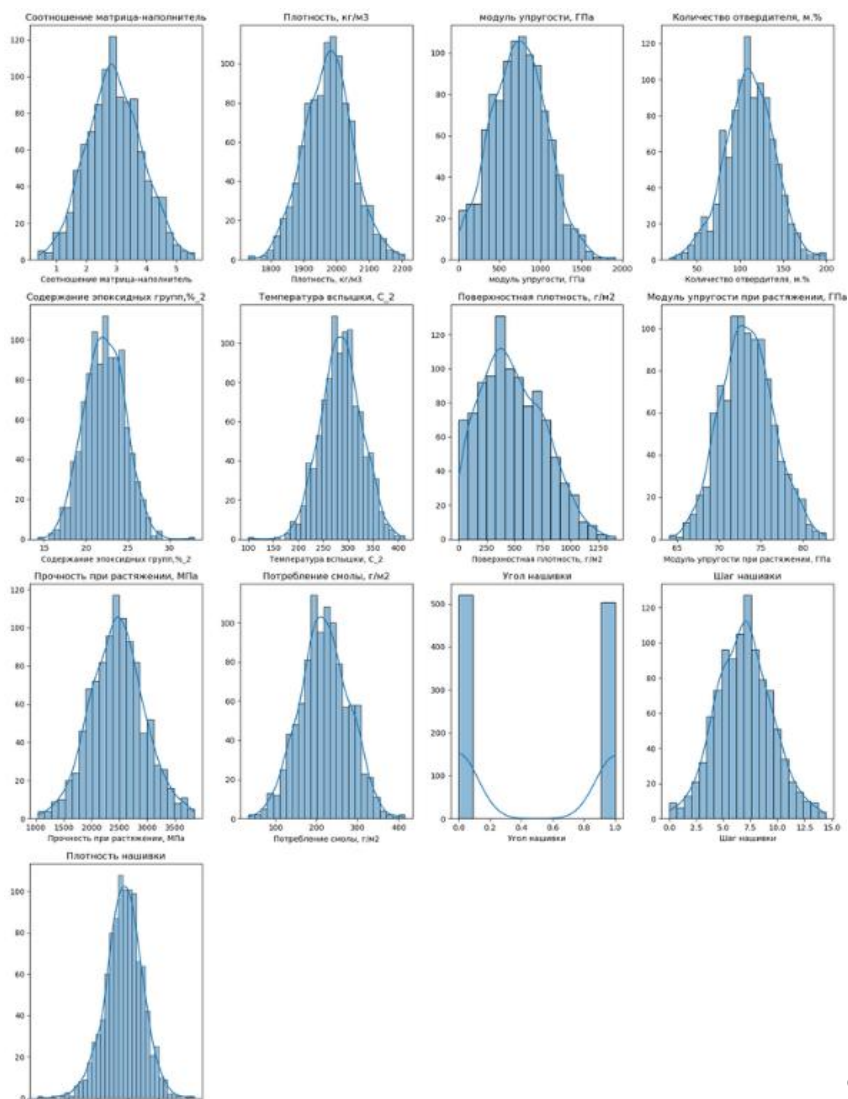


Рисунок 8 – Гистограммы параметров датасета ds с наложением кривой плотности распределения

Исходя из визуального анализа гистограмм на Рисунке 8, можно сделать вывод, что данные в датасете близки к нормальному распределению, но, возможно, имеют выбросы.

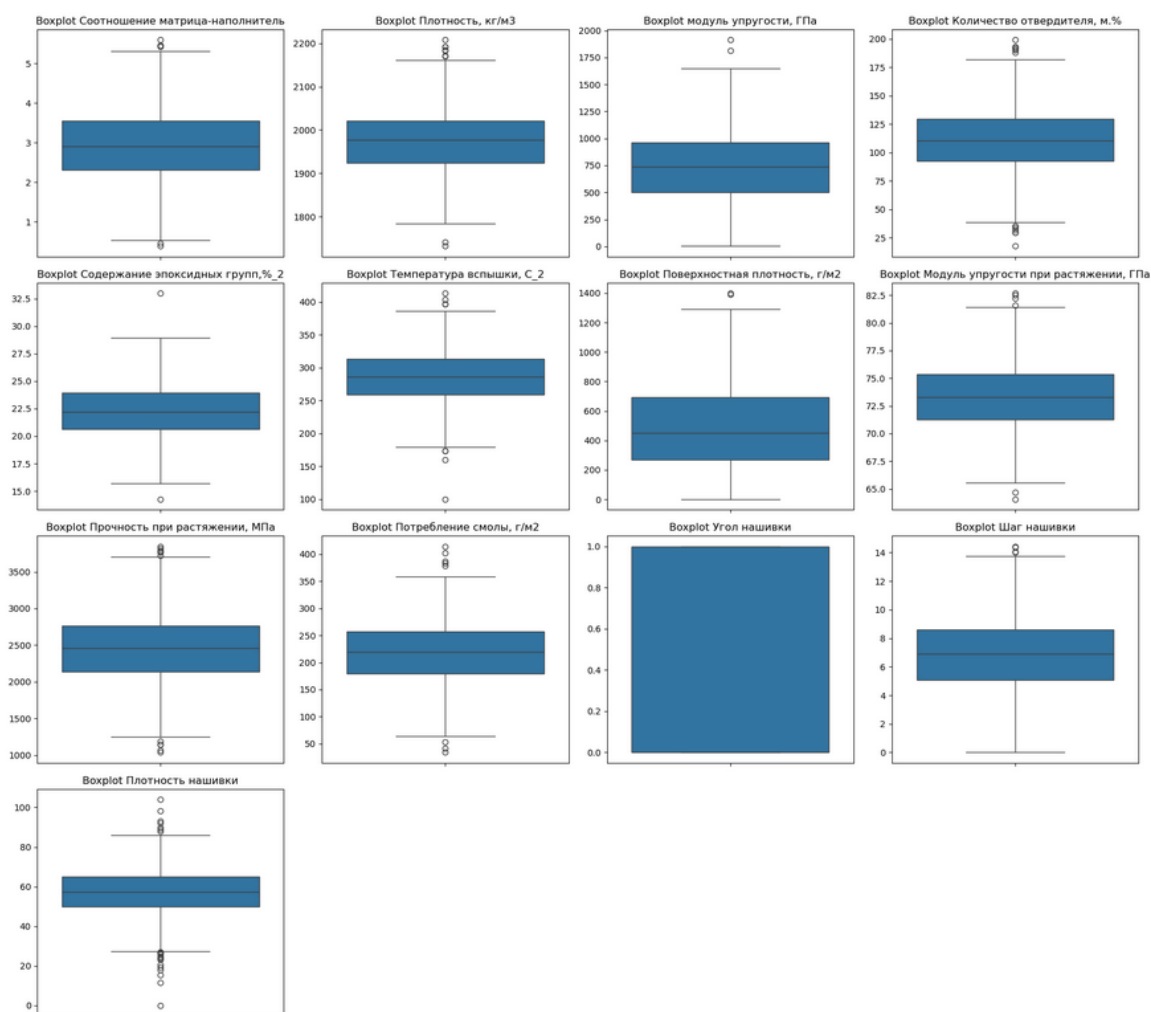


Рисунок 9 – Диаграммы боксплот для датасета ds

Для анализа выбросов были сгенерированы боксплоты – диаграммы размаха, отражающие представление распределения данных, которое показывает медиану, квартили (первый и третий) и значения выбросов. Как видно на Рисунке 9, за пределами «усов» находятся точки, которые и являются выбросами.

Таким образом, выявлено, что датасет не имеет пропусков, тип данных в столбцах верный и пригодный для анализа, нет коррелирующих переменных. Но также присутствуют следующие проблемы датасета:

- Разный масштаб данных;
- Присутствие выбросов.

Данные проблемы необходимо устранить для дальнейшей работы с датасетом в моделях машинного обучения и нейронных сетях.

2. Практическая часть

2.1. Предобработка данных

В теории предобработка данных включает этапы: обработку пропусков, выбросов, ошибок в значениях. При визуальном осмотре данных, замечено, что они имеют схожий формат, одинаковый тип данных в каждом столбце. На этапе разведочного анализа, выявлено, что пропусков в данных не имеется, но присутствуют выбросы, и требуется нормализация и стандартизация для дальнейшего применения моделей для прогноза модуля упругости при растяжении и прочности при растяжении и рекомендательной нейронной сети, которая будет рекомендовать соотношение матрица-наполнитель.

Предобработка началась с применения метода IQR, который предполагает удаление точек, лежащих вне интервала от $Q1 - 1.5 * IQR$ до $Q3 + 1.5 * IQR$, где $Q1$ и $Q3$ - первый и третий квартили соответственно. На Рисунке 9 аутлайеры, они же выбросы – находятся за концами «усов».

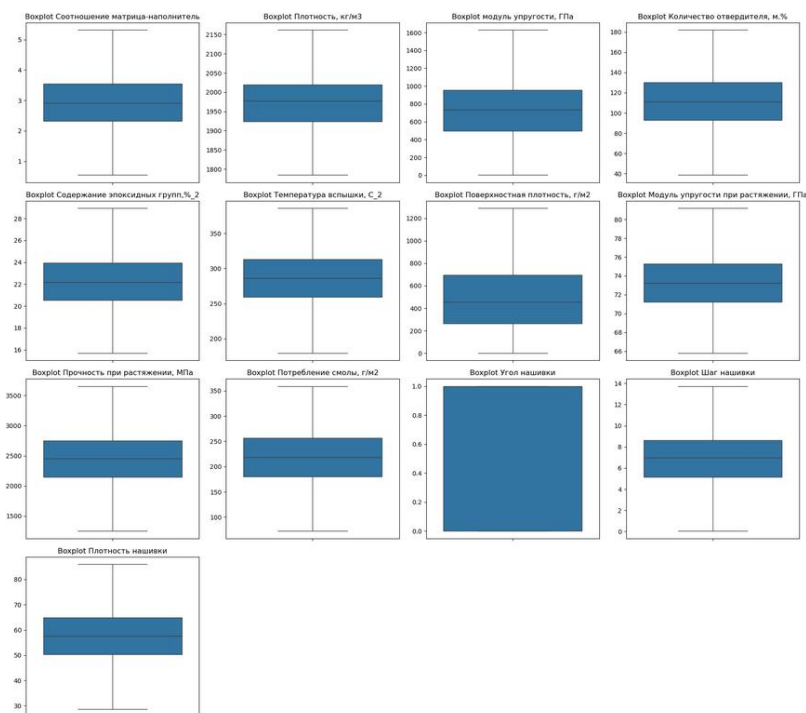


Рисунок 10 – Диаграммы боксплот для очищенного датасета

Процесс удаления выбросов потребовал проведения трех итераций применения метода к данным, пока не был получен результат, представленный на Рисунке 10. Как видно по диаграммам, выбросов не наблюдается, что можно представить и на графике попарного рассеяния параметров, представленном на Рисунке 11.

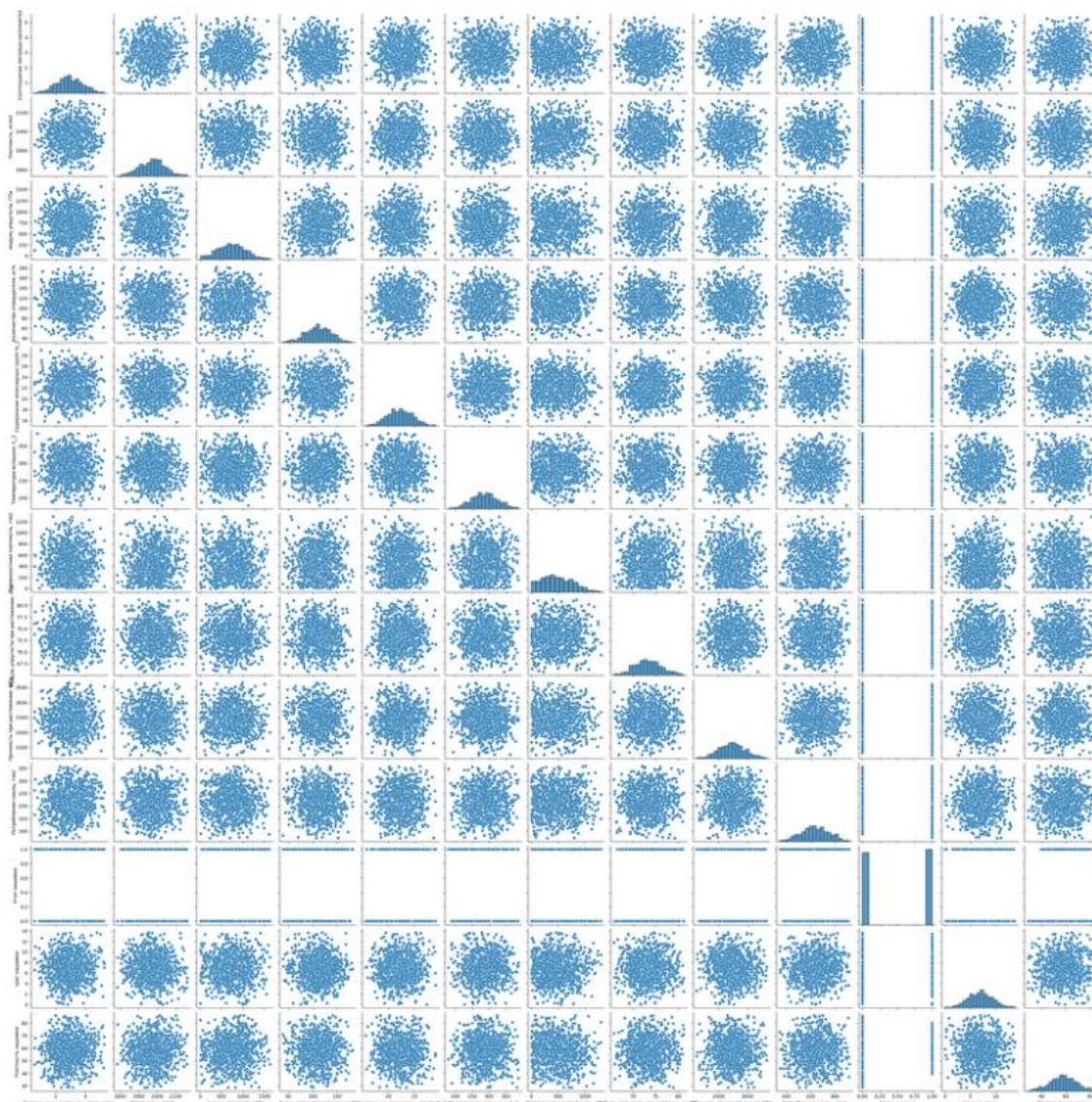


Рисунок 11 – График попарного рассеяния величин очищенного датасета

На данном графике видно, что величины рассеяны случайным образом, с уплотнением к центру и, предположительно, в большинстве графиков равноудалены друг от друга.

Также на данном этапе стоит отметить, что метод IQR для очистки от выбросов работает лучше. После первой итерации в столбце 'Плотность

нашивки', где выбросов изначально было больше всего (21) стало 4, а последующие выбросы почти все ушли, в то время как после применения IsolationForest, после первой итерации выбросов осталось больше (21 было – стало 17). Основываясь на коде, можно сказать, что для небольшого количества фич лучше использовать метод IQR, но если столбцов будет кратно больше, а измерениями можно пожертвовать, то быстрее будет IsolationForest. Отсюда также можно предположить, что потерь от оригинального датасета будет больше при применении ансамблевого метода, нежели чем межквартильного размаха. В ходе преобразований датасет сократился на 9,87%, до 922 строк после трех итераций и полного удаления выбросов, в то время как после первой итерации IsolationForest, при наличии выбросов количество строк датасета стало 923, значит, при итеративном подходе, дальше матрица сократится еще больше.

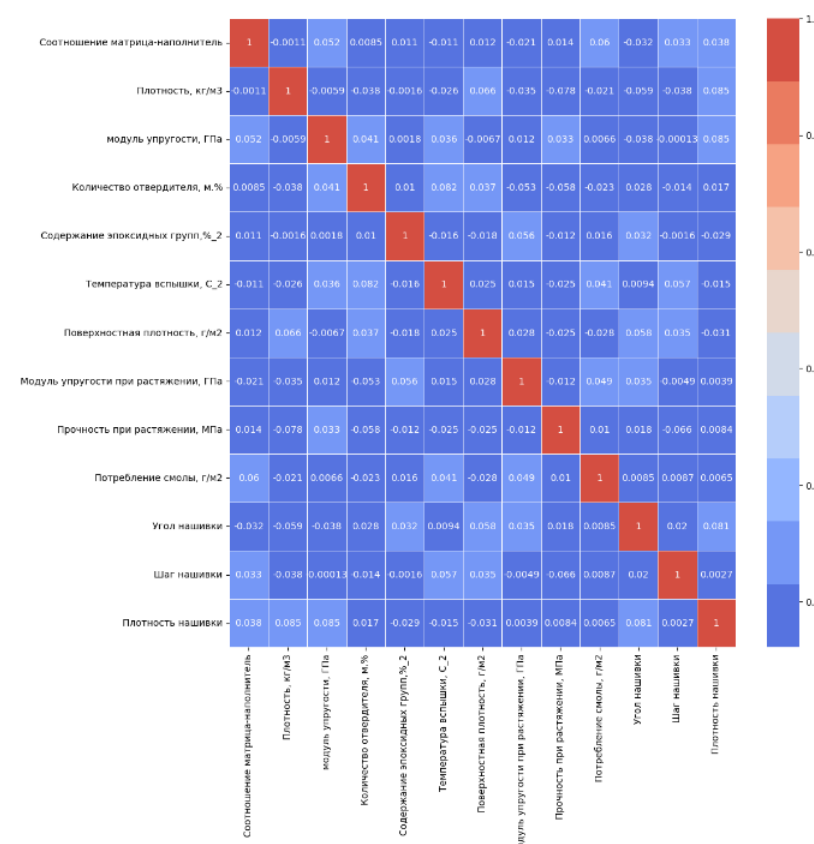


Рисунок 12 – Тепловая карта корреляций очищенного датасета

Далее необходимо было проверить, насколько изменилась связь между показателями, в данном случае методом корреляции Спирмена и вынести результаты в тепловую карту, представленную на Рисунке 12.

Таким образом, после чистки попарное рассеяние показателей стало более скученным, более опасываемым кругом, что говорит о низкой взаимосвязи показателей. Это же подтверждает и тепловая карта корреляции методом Спирмана, статистически значимых связей нет. Кривая плотности распределения показателей все еще не повторяет кривую нормального распределения.

Следующий этап предполагал изменение масштаба данных. Для него было применено два метода StandartScaler и MinMaxScaler. Обычно применяется только один масштабатор, так как вместе они могут нарушить форму распределения данных. Но так как это исследовательская работа и необходимо рассмотреть различные методы, то можно применить оба.

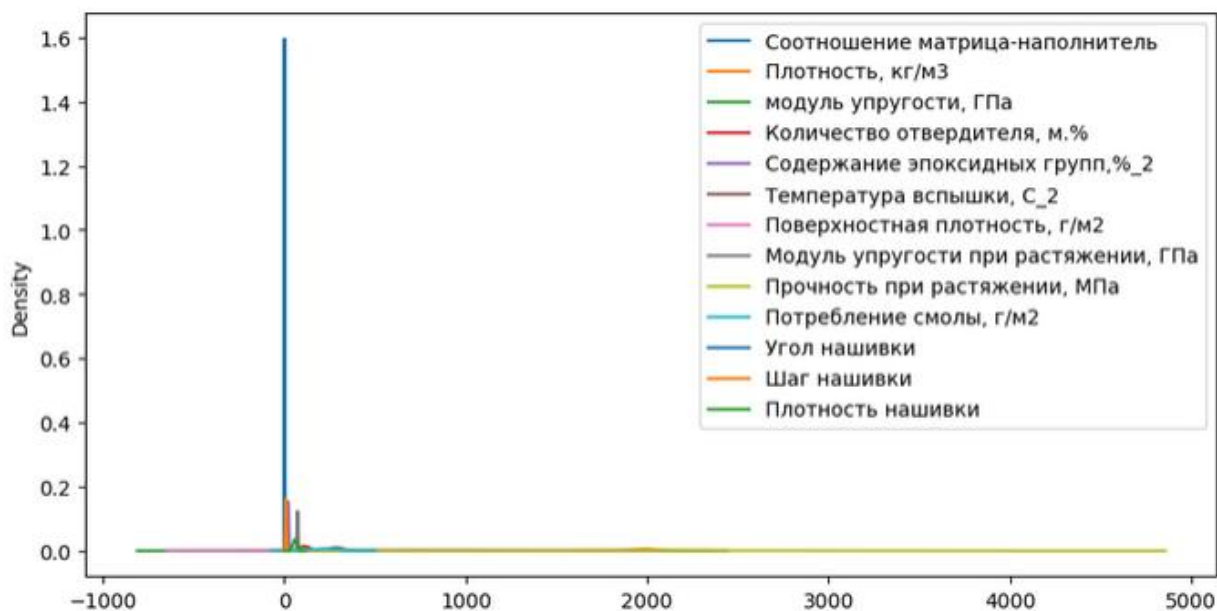


Рисунок 13 – График оценки ядерной плотности величин выборки

На рисунке 13 можно наблюдать, что разный масштаб данных помешает работе методов машинного обучения и все данные должны быть приведены к одному масштабу.

Также наблюдается смещение данных, что подтверждается методом .skew и показано на Рисунке 14.

Соотношение матрица-наполнитель	0.049923
Плотность, кг/м3	0.001483
модуль упругости, ГПа	0.070899
Количество отвердителя, м.%	-0.108432
Содержание эпоксидных групп,%_2	0.045304
Температура вспышки, C_2	0.002646
Поверхностная плотность, г/м2	0.358970
Модуль упругости при растяжении, ГПа	0.119982
Прочность при растяжении, МПа	0.043946
Потребление смолы, г/м2	-0.014610
Угол нашивки	-0.043465
Шаг нашивки	0.044809
Плотность нашивки	-0.035707
dtype: float64	

Рисунок 14 – Таблица смещения величин выборки

Как видно из графика и таблицы выше, данные необходимо не только привести к сопоставимому виду, но и центрировать, поэтому первым этапом предстоит стандартизировать данные, а затем их можно нормализовать.

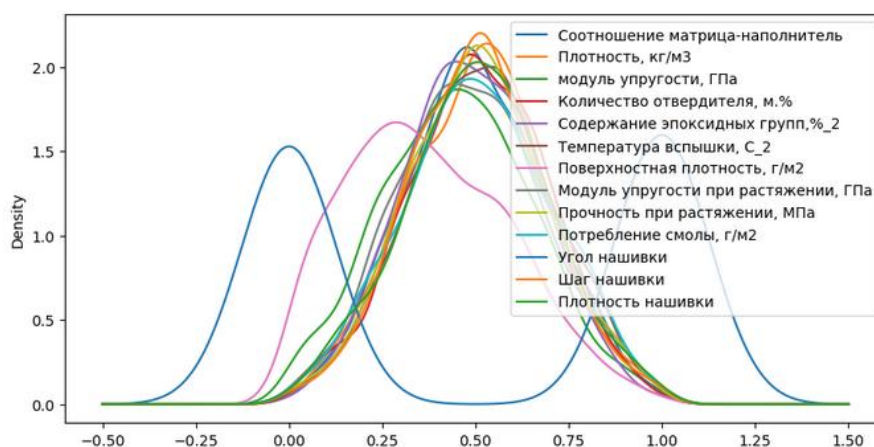


Рисунок 15 – График оценки ядерной плотности величин масштабированной выборки

Как видно из графика оценки ядерной плотности на Рисунке 15, после стандартизации данных, каждая из фич отцентрована и находится в сопоставимом формате, в то время как после только нормализации в ходе эксперимента, объективно, данные были смещены, что подтверждает теоретический аспект нарушения формы выборки, но коэффициенты корреляции методом Спирмене в обоих случаях сохранены и представлены на Рисунке 16.

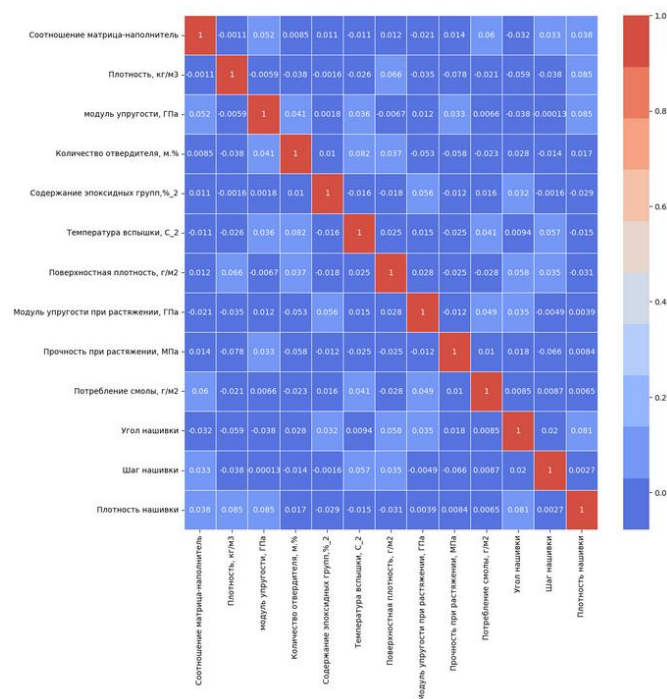


Рисунок 16 – Тепловая карта корреляций масштабированного датасета

Для дальнейших расчетов стоит использовать стандартизованный и нормализованный датасет, как показывают результаты валидации, у него наименьшая, сопоставимая с просто стандартизованным датасетом, накопленная ошибка по каждому столбцу, в то время как нормализация одна портит качество приведенного датасета.

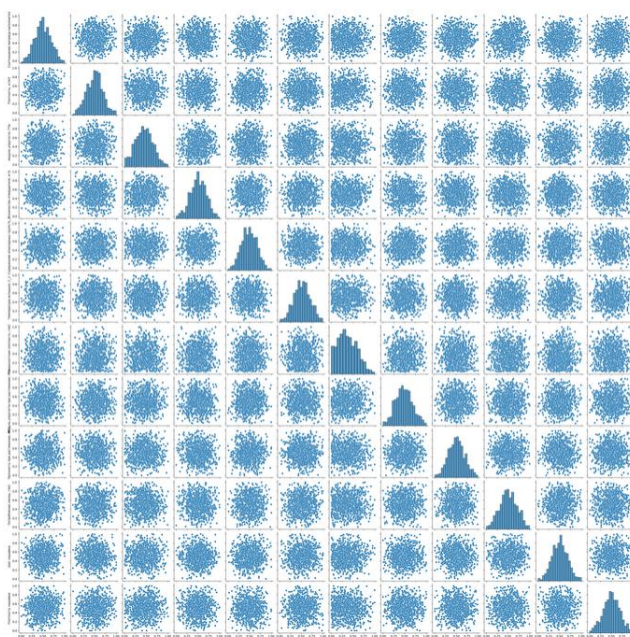


Рисунок 17 – График попарного рассеяния величин масштабированного датасета

	Соотношение матрица- наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки	Шаг нашивки	Плотность нашивки
count	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000
mean	0.499412	0.502904	0.451341	0.506200	0.490578	0.516739	0.373295	0.487343	0.503776	0.507876	0.510846	0.503426	0.503938
std	0.187858	0.188395	0.201534	0.186876	0.180548	0.190721	0.217269	0.196366	0.188668	0.199418	0.500154	0.183587	0.193933
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.371909	0.368184	0.305188	0.378514	0.366571	0.386228	0.204335	0.353512	0.373447	0.374647	0.000000	0.372844	0.376869
50%	0.495189	0.511396	0.451377	0.506382	0.488852	0.516931	0.354161	0.483718	0.501481	0.510143	1.000000	0.506414	0.504310
75%	0.629774	0.624719	0.587193	0.638735	0.623046	0.646553	0.538397	0.617568	0.624299	0.642511	1.000000	0.626112	0.630842
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Рисунок 18 – Описательная таблица подготовленного датасета

Как видно по графику попарного рассеяния, представленного на Рисунке 17, очищенные и приведенные данные все также не имеют какой либо явной связи. Сам датасет очищен, масштабирован и готов к использованию его в методах машинного обучения, а все его значения лежат в граница от нуля до единицы, как видно по Рисунку 18.

2.2. Разработка и обучение модели

Для начала подготовленный датасет надо разбить на тестовую и тренировочную выборки, при этом необходимо 30% данных оставить на тестирование модели, на остальных происходит обучение моделей. Сначала имеет смысл оценить модели для одного показателя 'Модуль упругости при растяжении, ГПа', а затем можно применить наработки для второго 'Прочность при растяжении, МПа'. Данные числовые, непрерывные, значит, можно проводить регрессионный анализ. Стоит рассмотреть линейную регрессию, Случайный лес для регрессии, Градиентный бустинг, Метод опорных векторов. В ходе использования отобранных методов был проведен поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10. ¶

Так, для построения моделей регрессионного анализа как модуля упругости, так и прочности при растяжении оптимальными параметрами оказались следующие, представленные в Таблице 5.

Таблица 5 – Таблица оптимальных гиперпараметров моделей

Модель	Наилучшие параметры
Random Forest	{'max_depth': 10, 'n_estimators': 100}
Gradient Boosting	{'learning_rate': 0.05, 'max_depth': 3, 'n_estimators': 50}
SVR	{'C': 0.1, 'kernel': 'linear'}

В целях проверки гипотезы о нарушении взаимосвязей, было решено рассмотреть также применение модели регрессии выбранными методами, но для датасета, с исключением столбца угла нашивки из-за формы его данных, но, результат не изменился, оценка моделей не изменилась.

2.3. Тестирование модели

Для метрики качества моделей будут использованы следующие: коэффициент детерминации, средняя абсолютная ошибка, средняя квадратическая ошибка.

Коэффициент детерминации (R^2) — это статистический показатель, который измеряет, насколько хорошо модель регрессии описывает фактические данные. Он показывает долю вариации зависимой переменной, которая объясняется независимыми переменными, включенными в модель, также насколько хорошо модель предсказывает данные по сравнению с простым предсказанием средним значением. Чем ближе R к единице, тем лучше модель описывает имеющиеся данные.

Средняя абсолютная ошибка (MAE) — это степень несоответствия между фактическими и прогнозируемыми значениями. Абсолютная ошибка представляет собой разность между спрогнозированным и фактическим значениями. MAE — это среднее от таких ошибок, что помогает понять эффективность модели, чем она ближе к нулю, тем качественнее модель. Но MAE возвращается в том же масштабе значений, что и исходные данные,

поэтому небольшая ошибка на низком масштабе данных может сбить с толку и для наглядности используется средняя абсолютная ошибка в процентах.

Среднеквадратическая ошибка (MSE) — это метрика, используемая для оценки качества модели машинного обучения или статистической модели. Она показывает среднее значение квадратов разностей между предсказанными и фактическими значениями. Чем меньше MSE, тем лучше модель предсказывает данные.

Таблица 6 – Сравнительная таблица MAE, MSE, R²

Название	Средняя Абсолютная Ошибка	Средняя Квадратическая Ошибка / Корень из MSE	Коэффициент Детерминации
1	2	3	4
Что измеряет?	Среднюю абсолютную величину ошибок	Средний квадрат ошибок	Долю дисперсии зависимого фактора, объясненную моделью
Диапазон	[0, +∞)	[0, +∞)	(-∞, 1] (обычно [0, 1])
Идеальное значение	0	0	1
Единицы измерения	Такие же как у зависимого фактора	Квадрат единиц зависимого фактора	Безразмерная (доля или %)
Чувствительность к выбросам	Низкая (ошибка линейна)	Высокая (ошибка квадратична)	Высокая (зависит от квадратов остатков)
Основные Преимущества	1. Простая и понятная интерпретация. 2. Устойчивость к выбросам. 3. Единицы измерения зависимого фактора.	1. Чувствительность к выбросам (один выброс может сильно исказить MSE). 2. Плохая интерпретируемость единиц (MSE - квадраты, RMSE - не средняя ошибка). 3. Не дает представления об относительном качестве модели.	1. Безразмерность (легко сравнивать модели на разных данных). 2. Интуитивная интерпретация (% объясненной дисперсии). 3. Стандартная метрика для сравнения моделей.

Продолжение Таблицы 6

Основные Недостатки	<p>1. Не выделяет крупные ошибки (может маскировать редкие, но катастрофические промахи).</p> <p>2. Менее "гладкая" функция для оптимизации (из-за модуля).</p>	<p>Когда крупные ошибки недопустимы и должны строго штрафоваться.</p> <p>Когда модель оптимизируется градиентными методами (дифференцируемость). (RMSE: то же и когда нужны интерпретируемые единицы с квадратичным штрафом).</p>	<p>1. Не показывает величину ошибки в единицах зависимого фактора.</p> <p>2. Растет с добавлением любых переменных (даже бесполезных).</p> <p>3. Не подходит для сравнения моделей с разным u (только для одного набора данных).</p> <p>4. Может вводить в заблуждение при неверной спецификации модели (особенно нелинейности).</p>
---------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Исходя из анализа метрик качества моделей можно сделать вывод, что лучше применять их совместно, чтобы учитывать разные их вариации для полноты выводов, так как ни одна метрика не дает полной картины.

Например, высокий R^2 при высоком MAE/MSE говорит о том, что модель уловила тренд, но делает грубые ошибки в абсолютных величинах, в свою очередь низкий R^2 при низком MAE/MSE возможен, если предсказываемая величина имеет очень маленькую дисперсию (все значения близки к среднему).

Таблица 7 – Сравнительная таблица метрик качества моделей

Параметр	Model	R^2 train	MAE train	MSE train	RMSE train	R^2 test	MAE test	MSE test	RMSE test
1	2	3	4	5	6	7	8	9	10
Модуль упругости при растяжении	Linear Regression	0.017761	0.154297	0.036366	0.190700	-0.026243	0.169520	0.042833	0.206960
Модуль упругости при растяжении	Random Forest	0.627365	0.096330	0.013796	0.117458	-0.040132	0.169388	0.043412	0.208356

Продолжение Таблицы 7

1	2	3	4	5	6	7	8	9	10
Модуль упругости при растяжении	Gradient Boosting	0.188023	0.139788	0.030063	0.173386	-0.029195	0.169313	0.042956	0.207258
Модуль упругости при растяжении	SVR	0.010552	0.154069	0.036633	0.191398	-0.026280	0.168836	0.042834	0.206964
Прочность при растяжении	Linear Regression	0.017607	0.147792	0.034306	0.185219	-0.005456	0.155191	0.037238	0.192972
Прочность при растяжении	Random Forest	0.703969	0.082464	0.010338	0.101675	-0.028212	0.157050	0.038081	0.195143
Прочность при растяжении	Gradient Boosting	0.218386	0.132207	0.027295	0.165211	-0.001203	0.155306	0.037081	0.192563
Прочность при растяжении	SVR	0.013447	0.147819	0.034451	0.185611	-0.022893	0.156749	0.037884	0.194638
Прочность при растяжении-УН	Linear Regression	0.017527	0.147845	0.034309	0.185227	-0.004113	0.155074	0.037188	0.192843
Прочность при растяжении-УН	Random Forest	0.705071	0.082302	0.010299	0.101485	-0.029210	0.157208	0.038118	0.195238
Прочность при растяжении-УН	Gradient Boosting	0.218386	0.132207	0.027295	0.165211	0.001839	0.155044	0.036968	0.192270
Прочность при растяжении-УН	SVR	0.013754	0.147917	0.034441	0.185582	-0.014068	0.156340	0.037557	0.193796

Имеющиеся данные не взаимосвязаны, алгоритмы машинного обучения не показывают какую либо значимую возможность прогнозирования. Коэффициент детерминации на тренировочных данных в модели Случайного леса показывает самую высокую эффективность метода, однако, R^2 на тестовой выборке во всех моделях даже отрицательный, что говорит о том, что ни одна модель совершенно не объясняет вариации

данных, будь то Модуль упругости при растяжении или Прочность при растяжении. Также можно утверждать, что простое среднее дает более точный результат для прогнозирования. Для эксперимента был убран показатель угла нашивки, предполагая, что бинарное значение может нарушить описание зависимого параметра от переменной. Метрики ошибок в сводной таблице указывают на то, что выдвинутая гипотеза ложна о влиянии бинарного показателя на выборку, значит, повысить описательную способность зависимого параметра переменными можно только дополнив датасет данными, которые могут повысить взаимосвязь в регрессии. В данном случае даже ансамблевый метод Градиентного бустинга не показал ощутимо значимой эффективности. ¶

2.4. Нейронная сеть рекомендации соотношения матрицы-наполнителя

Для построения модели будет использована модель многослойного перцептрон (MLP) для регрессии с функцией EarlyStopping по средней абсолютной ошибке (так как имеется регрессия), чтобы избежать переобучения. Архитектура сети:

- Входной слой: 128 нейронов с активацией ReLU
- BatchNormalization для стабилизации обучения
- Dropout (30%) для регуляризации
- Скрытый слой: 64 нейрона с активацией ReLU
- BatchNormalization и Dropout (20%)
- Скрытый слой: 32 нейрона с активацией ReLU
- Выходной слой: 1 нейрон (регрессия)
- Оптимизатор adam
- Функция потерь MSE

Датасет требуется разделить на тренировочную и тестовую выборки в соотношении 80/20. Также, для дальнейшего вынесения нейронной

модели рекомендации соотношения матрица-наполнитель, был взят очищенный датасет, чтобы внутри модели сразу производился препроцессинг данных, когда предстоит вынести модель в отдельное веб-приложение для интерпретации результата, с учетом того, что модель видит данные в том масштабе, в котором она обучена. Код данного шага представлен на Рисунке 19

```

# Разделение на признаки и целевую переменную
B = diy.drop('Соотношение матрица-наполнитель', axis=1)
a = diy['Соотношение матрица-наполнитель']

# Разделение на тренировочную и тестовую выборки
B_train, B_test, a_train, a_test = train_test_split(
    B, a,
    test_size=0.2,
    random_state=42
)

# Масштабирование данных
scaler = StandardScaler()
B_train_scaled = scaler.fit_transform(B_train)
B_test_scaled = scaler.transform(B_test)

print(f"Тренировочные данные: {B_train.shape[0]} строк")
print(f"Тестовые данные: {B_test.shape[0]} строк")

```

Тренировочные данные: 737 строк
 Тестовые данные: 185 строк

Рисунок 19 – Разделение данных на выборки и применение масштабатора

В ходе обучения нейронная сеть показала наилучшую описательную способность тестовой выборки на 20 эпохе из 500 и остановкой на 25 эпохе, при ожидании снижения метрики MSE. Обученная модель имеет веса, представленные на Рисунке 20.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	1,664
batch_normalization (BatchNormalization)	(None, 128)	512
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8,256
batch_normalization_1 (BatchNormalization)	(None, 64)	256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2,080
dense_3 (Dense)	(None, 1)	33

Total params: 37,637 (147.02 KB)
 Trainable params: 12,417 (48.50 KB)
 Non-trainable params: 384 (1.50 KB)
 Optimizer params: 24,836 (97.02 KB)

Рисунок 20 – Веса обученной модели многослойного перцептрона

Если повысить число ожидания эпох, то модель уходит в переобучение и все сильнее растет расхождение между функцией потерь, которая минимизирует среднюю квадратическую ошибку на валидационном наборе данных. Оптимальной оказалась 20 эпоха, далее функция валидационных потерь выходит на плато, а кривая потерь на тренировочных данных снижается дальше, лучший результат представлен на Рисунке 21.

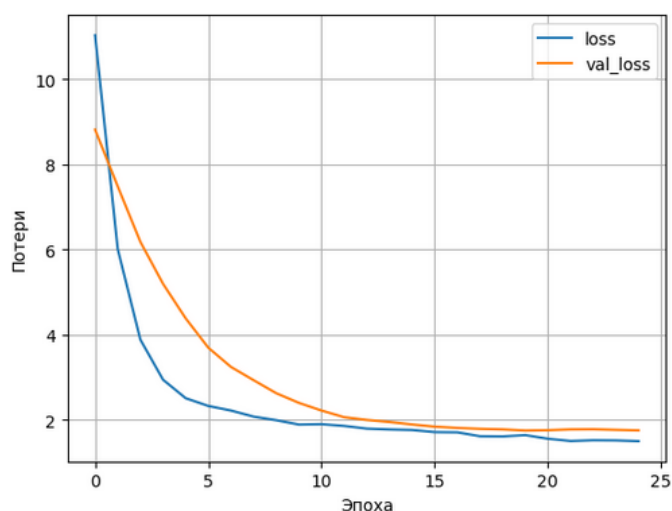


Рисунок 21 – графики потерь на тренировочном и валидационном наборах данных

Но, при всем при этом, метрики ошибок на тестовых данных получились следующими:

- MAE – 0.786625
- MSE – 0.951334
- R^2 – -0.224622

Если проанализировать метрики качества, то видно, что модель плохо описывает зависимый параметр, прогноз на его основании строить нельзя, особенно, когда средняя абсолютная ошибка кажется низкой, но если вспомнить, что медианное значение параметра соотношения матрица наполнитель в первоначальном очищенном датасете составляет примерно 2,93, что примерно 3 к 1. Уместно посчитать среднюю абсолютную ошибку в

процентах, которая говорит о том, что при валидации результатов предсказанное значение в среднем отличается от фактического на 30,38%.

По итогу проделанной работы можно отметить, что данные для построения моделей были независимыми для регрессионного анализа. Ни один из методов машинного обучения не показал значимого результата, как и нейросеть на третьей зависимой от переменных. Нейросеть при заданных 500 эпохах и ожиданием 5 итераций снижения средней абсолютной ошибки показала хороший результат, лучше, чем при аналогичных параметрах и более высокой толерантности (от 10 до 50), с которой модель уходит в переобучение. В свою очередь коэффициент детерминации получился так же низким, как и в аналогичных испытаниях датасета. Показатель среднего абсолютного отклонения выглядит низким, но если посмотреть на среднюю абсолютную процентную ошибку, то видно, что средняя ошибка составляет 30.38% от реальных значений, что много и для практического применения модели нейронной сети необходима серьезная доработка подхода и расширение данных для тренировки и теста модели, чтобы между данными появилась какая-либо связь.

2.5. Разработка веб-приложения

Для разработки веб-приложения на библиотеке Flask была сохранена нейронная модель с весами в файл `neural_model.keras`, а для масштабирования данных сохранен оригинальный скейлер с помощью библиотеки `joblib`, код сохранения модели представлен на Рисунке 22.

```
# Сохранение всей модели
model.save('neural_model.keras')
print("Модель сохранена как 'neural_model.keras'")

# Дополнительно: сохранение scaler для дальнейшей работы с нейросетью
joblib.dump(scaler, 'scaler.pkl')
print("Скалер сохранен как 'scaler.pkl'")

Модель сохранена как 'neural_model.keras'
Скалер сохранен как 'scaler.pkl'
```

Рисунок 22 – Сохранение модели и масштабатора

Далее для функционирования приложения был написан файл приложения `app.py`, содержащий модель обработки и вывода данных, а также `index.html` – файл графической оболочки для функционирования приложения в браузере. Архитектура папки приложения представлена на Рисунке 23.

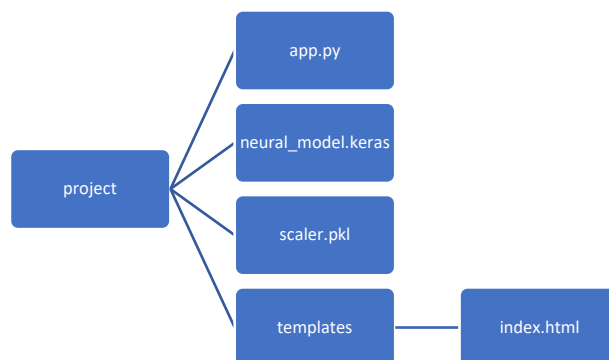


Рисунок 23 – Архитектура папки веб-приложения

Приложение требует ввода чисел предполагаемых характеристик, а выводит соотношение матрицы к наполнителю в формате десятичной дроби, с разделителем точкой, в свою очередь дробные значения тоже требуется вводить с разделителем точкой. Особенности работы приложения, которые были предусмотрены:

- Валидация данных:
 - Проверка заполнения всех полей
 - Контроль числовых значений
 - Специальная проверка для "Угла нашивки" (0 или 90)
- Обработка ошибок:
 - Четкие сообщения о незаполненных полях
 - Указание некорректных значений
 - Обработка исключений при прогнозировании
- Интерфейс:
 - Белый фон страницы
 - Светло-серая панель формы
 - Выделение ошибок красным цветом
 - Наглядное отображение результата

- Безопасность:
 - Обработка некорректного ввода
 - Защита от исключений при прогнозировании

Прогнозирование соотношения матрица-наполнитель

Плотность, кг/м3:
Введите число

модуль упругости, ГПа:
Введите число

Количество отвердителя, м. %:
Введите число

Содержание эпоксидных групп, %_2:
Введите число

Примечание: дробная часть должна быть после точки

Температура вспышки, С_2:
Введите число

Поверхностная плотность, г/м2:
Введите число

Модуль упругости при растяжении, ГПа:
Введите число

Прочность при растяжении, МПа:
Введите число

Потребление смолы, г/м2:
Введите число

Угол нашивки:
Введите число

Примечание: допустимые значения - 0 или 90

Шаг нашивки:
Введите число

Плотность нашивки:
Введите число

Рассчитать

Прогнозирование соотношения матрица-наполнитель

Плотность, кг/м3:
2160

модуль упругости, ГПа:
933

Количество отвердителя, м. %:
129

Содержание эпоксидных групп, %_2:
21.25

Примечание: дробная часть должна быть после точки

Температура вспышки, С_2:
300

Поверхностная плотность, г/м2:
1010

Модуль упругости при растяжении, ГПа:
78

Прочность при растяжении, МПа:
2000

Потребление смолы, г/м2:
300

Угол нашивки:
0

Примечание: допустимые значения - 0 или 90

Шаг нашивки:
7

Плотность нашивки:
70

Рассчитать

Прогнозируемое соотношение: **2.8742**

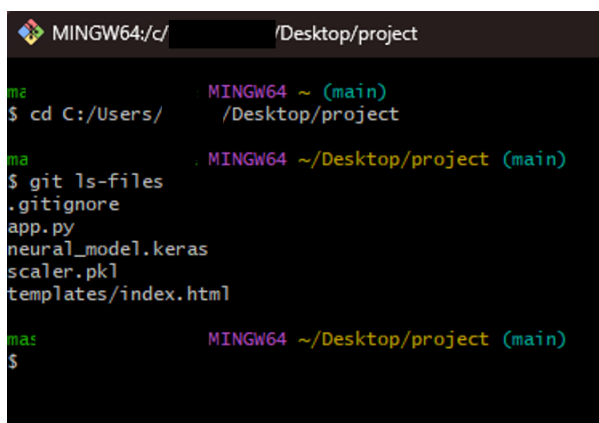
Рисунок 24 – Веб-интерфейс приложения Flask

Веб-отображение интерфейса представлено на Рисунке 24. Локально на компьютере приложение открывается по адресам: <http://192.168.0.10:5000/> и <http://127.0.0.1:5000/>

2.6. Commit на GitHub и размещение на веб-хостинг

Для контроля за версиями и выгрузкой файлов на портал GitHub была использована распределенная система управления версиями (Git), создан аккаунт на портале GitHub и репозиторий с проектом веб-приложения.

В папке project с файлами веб-приложения настроено отслеживание версий файлов, представленных на Рисунке 25.



```

MINGW64/c/ /Desktop/project

ma MINGW64 ~ (main)
$ cd C:/Users/ /Desktop/project

ma MINGW64 ~/Desktop/project (main)
$ git ls-files
.gitignore
app.py
neural_model.keras
scaler.pkl
templates/index.html

ma MINGW64 ~/Desktop/project (main)
$
  
```

Рисунок 25 – Командная строка среды Git с расположением проекта

Также из интерфейса командной строки был совершен commit с выгрузкой отслеживаемых данных в репозиторий на сайт GitHub по адресу: https://github.com/maslovmd/BKP_DataScience. Commit из командной строки Git Bash был назван «Создание_веб-приложения», репозиторий представлен на Рисунке 26.

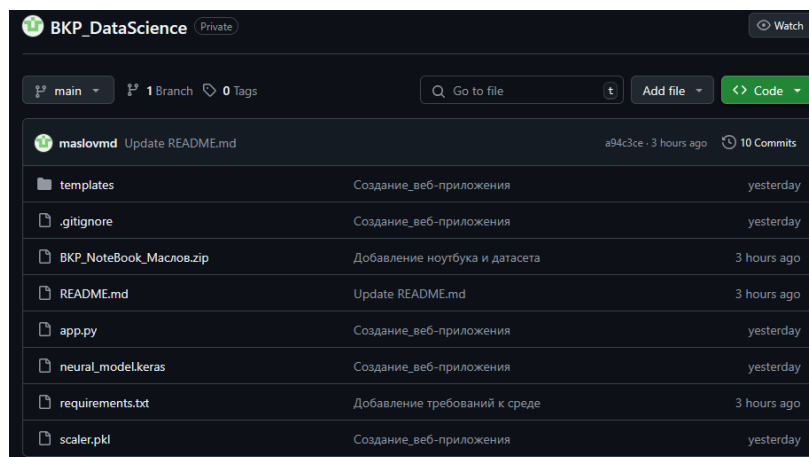


Рисунок 26 – Результат операции Commit с сайта GitHub

Так же был опробован метод прямой загрузки на сайт через веб-интерфейс. Данный Commit указан как «Добавление ноутбука и датасета» и «Добавление требований к среде», был написан и доработан файл README.

Веб-приложением также можно воспользоваться до 10 октября 2025 года в сети интернет по адресу <https://cladkybubbalech.pythonanywhere.com/>. Для сайта приложение было оптимизировано, чтобы полный функционал был доступен в ограниченной дисковым пространством среде.

Заключение

В ходе рабочего процесса было замечено, что хотя первоначальные данные довольно чистые, как показал разведочный анализ, данные полные без большого количества выбросов. В данном случае в объединенном датасете выбросы составили чуть менее 10%. Распределение близко к нормальному, а связь между самими переменными отсутствует, что говорит о том, что каждый параметр уникален и не является иным представлением другого. Это не гарантирует результата достаточного для построения прогноза для обоснования управленческих решений.

По итогу проделанной работы можно отметить, что данные для построения моделей были независимыми для регрессионного анализа. Ни один из методов машинного обучения не показал значимого результата, как и нейросеть на третьей зависимой от переменных. Нейросеть при заданных 500 эпохах и ожиданием 5 итераций снижения средней абсолютной ошибки показала хороший результат, лучше, чем при аналогичных параметрах и более высокой толерантности (от 10 до 50), с которой модель уходит в переобучение. В свою очередь коэффициент детерминации получился так же низким, как и в аналогичных испытаниях датасета, статистически неотличимым от нуля, что можно интерпретировать как низкую описательную возможность модели, а значит, непригодную для практического применения. Показатель среднего абсолютного отклонения выглядит низким, но если посмотреть на среднюю абсолютную процентную ошибку, то видно, что она составляет 30.38% от реальных значений, что много и для практического применения модели нейронной сети необходима серьезная доработка и расширение данных для тренировки и теста модели, чтобы между данными появилась какая-либо связь. Однако, это позволило применить некоторые из методов анализа данных на практике и увидеть их

результат, а также ознакомиться с возможностью визуального представления возможностей модели.

Данное исследование показывает специфику работы специалиста по Data Science. Можно предположить, что данный случай не является чем-то редким в процессе изучения данных, чем он и полезен для начинающего программиста. Работа специалиста данной отрасли информационных наук незаменима, но и бывает муторной, не приносящей ожидаемого результата, а сколько предстоит построить моделей, доработать их, пересмотреть исходные данные, собрать новые, чтобы выявить закономерность и взаимосвязь между переменными для дальнейшего практического применения знаний, полученных в ходе моделирования процессов и прогнозирования конечного результата – только практика покажет. Но несмотря на это, подтверждается факт, что этот подход экономит не только финансовые ресурсы предприятия, но и не распыляет трудовые на бесполезные эксперименты по НИОКР. Также не тратится много времени в сравнении с практическими опытами по составлению композитных материалов в данном случае, хотя данный подход можно применить абсолютно к любой отрасли, поэтому специалисты-аналитики так ценны в обществе прогресса.

Но следует учитывать, что в социальной сфере предсказательная способность моделей будет в практическом смысле все равно ниже, чем при прочих равных в прогнозировании свойств композитных материалов, разработке финансовых стратегий, так как человеческое поведение во многом сложно моделировать из-за его иррациональности, особенно индивидуальное поведение.

Библиографический список

1. ГОСТ Р ИСО 16269-4 — 2017 Статистические методы. Статистическое представление данных [Текст] ; введ. 2017–08–10. – Москва : Федеральное агентство по техническому регулированию и метрологии ; М. : Изд-во стандартов, 2002. – 3 с.
2. Андерсон К. Аналитическая культура. От сбора данных до бизнес-результатов / Карл Андерсон; пер. с англ. Юлии Константиновой; [науч. ред. РусланСалахиев]. — М.: Манн, Иванов и Фербер, 2017.
3. Брюс, П. Практическая статистика для специалистов Data Science: Пер. с англ. /П. Брюс, Э. Брюс. — СПб.: БХВ-Петербург, 2018. — 304 с.: ил.
4. Грас Д. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2021. - 416 с.: ил.
5. Демидова, Л. А. Разведочный анализ данных. Python : учебно-методическое пособие / Л. А. Демидова. — Москва : РТУ МИРЭА, 2022 — Часть 1 — 2022. — 107 с.
6. Кибзун, А. И. Теория вероятностей и математическая статистика. Базовый курс с примерами и задачами : справочник / А. И. Кибзун, Е. Р. Горяинова, А. В. Наумов. — 3-е изд. — Москва : ФИЗМАТЛИТ, 2007. — 232 с.
7. Лагутин, М. Б. Наглядная математическая статистика : учебное пособие / М. Б. Лагутин. — 7-е изд. — Москва : Лаборатория знаний, 2019. — 475 с.
8. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.:Символ-Плюс, 2011. – 992 с., ил.
9. Лутц М. Изучаем Python, том 2, 5-е изд. : Пер. с англ. — СПб. : ООО “Диалектика”, 2020. — 720 с. : ил.
- 10.Любанович Б. Простой Python. Современный стиль программирования. — СПб.: Питер, 2016. — 480 с.: ил.

- 11.Маккинни У. Python и анализ данных: Первичная обработка данных с применением pandas, NumPy и Jupiter / пер. с англ. А. А. Слинкина. 3-е изд. – М.:МК Пресс, 2023. – 536 с.: ил.
- 12.Неделько В.М.. Основы статистических методов машинного обучения. Учебное пособие. – Новосибирск, 2011. — 79 с..
- 13.Плас Дж. Вандер, Python для сложных задач: наука о данных и машинное обучение. Санкт-Петербург: Питер, 2018, 576 с.
- 14.Полякова, В. В. Основы теории статистики : [учеб. пособие] / В. В. Полякова, Н. В. Шаброва ; М-во образования и науки Рос. Федерации, Урал. федер. ун-т. – 2-е изд., испр. и доп. – Екатеринбург : Изд-во Урал. ун-та, 2015. – 148 с.
- 15.Реутов Ю.А.: Прогнозирование свойств полимерных композиционных материалов и оценка надёжности изделий из них, Диссертация на соискание учёной степени кандидата физико-математических наук, Томск 2016.
- 16.Роббинс, Дж. HTML5: карманный справочник, 5-е издание.: Пер. с англ. - М.: ООО «И.Д. Вильямс»: 2015. - 192 с.: ил.
- 17.Aurélien Géron. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems—СА.: O'Reilly Media, Inc., 2019. — 484 с.
- 18.Документация по библиотеке keras: – Режим доступа: <https://keras.io/api/>. (дата обращения: 08.07.2025).
- 19.Документация по библиотеке matplotlib: – Режим доступа: <https://matplotlib.org/stable/users/index.html>. (дата обращения: 08.07.2025)
- 20.Документация по библиотеке numpy: – Режим доступа: <https://numpy.org/doc/1.22/user/index.html#user>. (дата обращения: 08.07.2025).

21. Документация по библиотеке pandas: – Режим доступа: https://pandas.pydata.org/docs/user_guide/index.html#user-guide. (дата обращения: 08.07.2025).
22. Документация по библиотеке scikit-learn: – Режим доступа: https://scikit-learn.org/stable/user_guide.html. (дата обращения: 08.07.2025).
23. Документация по библиотеке seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>. (дата обращения: 08.07.2025).
24. Документация по библиотеке Tensorflow: – Режим доступа: <https://www.tensorflow.org/overview> (дата обращения: 08.07.2025).
25. Документация по языку программирования python: – Режим доступа: <https://docs.python.org/3.8/index.html>. (дата обращения: 08.07.2025).
26. «Газпромнефть-Заполярье» представило арктические проекты компаниям тюменского нефтегазового кластера : – Режим доступа: <https://polar.gazprom-neft.ru/press-center/news/gazpromneft-zapolyare-predstavilo-arkticheskie-proekty-kompaniyam-tyumenskogo-neftegazovogo-klastera>. (дата обращения: 08.07.2025).
27. Путин: Россия готова к сотрудничеству по освоению Арктики с западными странами : – Режим доступа: <https://www.1tv.ru/news/2025-03-27/505330-putin-rossiya-gotova-k-sotrudnichestvu-po-osvoeniyu-arktiki-s-zapadnymi-stranami>. (дата обращения: 08.07.2025).
28. Справочник Scikit-Learn, 2.7. Обнаружение новизны и выбросов: – Режим доступа: https://scikit-learn.ru/stable/modules/outlier_detection.html. (дата обращения: 08.07.2025).
29. Справочник Scikit-Learn, 13. Choosing the right estimator: – Режим доступа: https://scikit-learn.org/stable/machine_learning_map.html. (дата обращения: 08.07.2025).

- 30.Руководство по быстрому старту в flask: – Режим доступа: <https://flask-russian-docs.readthedocs.io/ru/latest/quickstart.html>. (дата обращения: 08.07.2025).
- 31.Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning [Электронный ресурс] : – Режим доступа: [http://alvarestech.com/temp/deep/Deep%20Learning%20by%20Ian%20Goodfellow,%20Yoshua%20Bengio,%20Aaron%20Courville%20\(z-lib.org\).pdf](http://alvarestech.com/temp/deep/Deep%20Learning%20by%20Ian%20Goodfellow,%20Yoshua%20Bengio,%20Aaron%20Courville%20(z-lib.org).pdf). (дата обращения: 08.07.2025).
- 32.Trevor Hastie, Robert Tibshirani, Jerome Friedman. The Elements of Statistical Learning Data Mining, Inference, and Prediction [Электронный ресурс] : – Режим доступа: <https://www.sas.upenn.edu/~fdiebold/NoHesitations/BookAdvanced.pdf>. (дата обращения: 08.07.2025).