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»

Слушатель Шумкина Мария Игоревна

Москва, 2024

Оглавление

[Введение 3](#_Toc185446344)

[1.Аналитическая часть 1](#_Toc185446345)

[1.1 Постановка задачи. 1](#_Toc185446346)

[1.2 Описание используемых методов. 8](#_Toc185446347)

[1.3 Разведочный анализ данных 16](#_Toc185446348)

[2. Практическая часть 19](#_Toc185446349)

[2.1 Предобработка данных 19](#_Toc185446350)

[2.2 Разработка и обучение модели. 22](#_Toc185446351)

[2.3 Тестирование модели. 22](#_Toc185446352)

[2.3.1 Прогнозирование модуля упругости при растяжении 23](#_Toc185446353)

[2.3.2 Прогнозирование прочности при растяжении 26](#_Toc185446354)

[2.4 Создание нейронной сети для рекомендации соотношения матрицы. 29](#_Toc185446355)

[2.5 Разработка приложения 36](#_Toc185446356)

[2.6 Создание удаленного репозитория 38](#_Toc185446357)

[Заключение 39](#_Toc185446358)

[Список использованной литературы 41](#_Toc185446359)

# Введение

Композиционные материалы в настоящее время занимают важное место в современном производстве благодаря их уникальным характеристикам, таким как высокая прочность, малая плотность и устойчивость к воздействию внешних факторов. Они широко применяются в авиационно-космической, автомобильной и строительной промышленности, где высокие требования к материалам диктуют необходимость точного понимания их свойств и поведения. Однако сложность структуры композитов и многообразие их характеристик создают значительные трудности при прогнозировании механических свойств.

Актуальность данной работы обусловлена потребностью в разработке инструментов, позволяющих с высокой точностью предсказывать свойства материалов на основе их параметров. Применение современных методов машинного обучения открывает новые возможности для решения этой задачи, позволяя сократить время и ресурсы, необходимые для проведения дорогостоящих экспериментов и испытаний.

Целью работы является создание модели машинного обучения для прогнозирования модуля упругости и прочности композиционных материалов при растяжении на основе их характеристик, а также рекомендация соотношения матрица-наполнитель. Для достижения поставленной цели были сформулированы следующие задачи:

* анализ и предобработка исходных данных, включая масштабирование и устранение выбросов;
* выбор и обучение моделей машинного обучения для прогнозирования модуля упругости, прочности и соотношения матрица-наполнитель;
* оценка качества построенных моделей с использованием метрик точности;
* разработка веб-приложения для интеграции выбранной модели и предоставления удобного интерфейса для пользователей.

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

Задача, рассматриваемая в данной работе, заключается в прогнозировании конечных свойств композиционных материалов с использованием методов машинного обучения. Композиты — это искусственно созданные материалы, которые состоят из нескольких компонентов, взаимодействующих друг с другом, чтобы сформировать материал с уникальными характеристиками. Например, железобетон, являющийся классическим примером композиционного материала, сочетает в себе прочность бетона при сжатии и высокую растяжимость стальной арматуры.

Важнейшими характеристиками для таких материалов являются их прочность и упругость, которые хотелось бы предсказать на основе данных о характеристиках входящих в состав компонентов. Эти характеристики включают такие параметры, как плотность, соотношение матрица-наполнитель, количество отвердителя, температура вспышки и другие. Прогнозирование этих свойств позволяет не только оптимизировать процесс производства, но и создавать материалы с заданными характеристиками, которые могут быть использованы в различных отраслях, таких как автомобилестроение, авиастроение и строительная индустрия.

В рамках данной работы используются набор данных, содержащий информацию о свойствах различных композитных материалов. Датасет сформирован из двух файлов с расширением .xlsx. Первый файл содержит 10 признаков и индекс, а также 1023 строки. Второй файл имеет 1040 примеров, 3 дополнительных признака и индекс. Файлы были объединены по индексу методом INNER, в результате чего был получен датасет размерностью 1023 строки и 13 признаков.

При анализе типов полей было выявлено, что все признаки, кроме "Угол нашивки, град" имеют тип float64, что подтверждает таблица 1. Здесь же мы видим, что отсутствующих значений нет, т.к. все параметры отмечены "non-null".

Таблица 1 - Описание признаков исходного датасета

|  |  |  |  |
| --- | --- | --- | --- |
| № | Признак | Количество заполненных значений | Тип |
| 1 | Соотношение матрица-наполнитель | 1023 | float64 |
| 2 | Плотность, кг/м3 | 1023 | float64 |
| 3 | модуль упругости, ГПа | 1023 | float64 |
| 4 | Количество отвердителя, м.% | 1023 | float64 |
| 5 | Содержание эпоксидных групп,% | 1023 | float64 |
| 6 | Температура вспышки, С\_2 | 1023 | float64 |
| 7 | Поверхностная плотность, г/м2 | 1023 | float64 |
| 8 | Модуль упругости при растяжении, ГПа | 1023 | float64 |
| 9 | Прочность при растяжении, МПа | 1023 | float64 |
| 10 | Потребление смолы, г/м2 | 1023 | float64 |
| 11 | Угол нашивки, град | 1023 | int64 |
| 12 | Шаг нашивки | 1023 | float64 |
| 13 | Плотность нашивки | 1023 | float64 |

Чтобы не пренебрегать золотым правилом в машинном обучении датасет сразу был разделен на обучающую (train) и тестовую (test) выборки до начала анализа и моделирования. Это правило помогает избежать утечек данных и переобучения модели. Разделение данных в отношении 30/70 привело к формированию обучающего датасета размерностью 716 экземпляров. Объем данных достаточен для обучения модели и выявления значимых зависимостей. Дальнейшее исследование признаков проводится на обучающей выборке.

В начале анализа было изучено количество уникальных значений каждого признака (таблица 2).

Таблица 2 – Описание признаков обучающего датасета

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Признак | Количество уникальных значений | Минимальное значение | Максимальное значение | Среднее арифметическое |
| 1 | Соотношение матрица-наполнитель | 711 | 0.39 | 5.46 | 2.94 |
| 2 | Плотность, кг/м3 | 712 | 1731.76 | 2192.74 | 1976.2 |
| 3 | модуль упругости, ГПа | 715 | 2.4 | 1911.54 | 740.54 |
| 4 | Количество отвердителя, м.% | 704 | 17.74 | 198.95 | 109.81 |
| 5 | Содержание эпоксидных групп,% | 704 | 14.25 | 33 | 22.28 |
| 6 | Температура вспышки, С\_2 | 702 | 160.26 | 413.28 | 287.76 |
| 7 | Поверхностная плотность, г/м2 | 704 | 0.60 | 1399.54 | 47.73 |
| 8 | Модуль упругости при растяжении, ГПа | 704 | 64.05 | 82.68 | 73.32 |
| 9 | Прочность при растяжении, МПа | 704 | 1036.86 | 3848.44 | 2473.91 |
| 10 | Потребление смолы, г/м2 | 703 | 41.05 | 414.59 | 215.97 |
| 11 | Угол нашивки, град | 2 | 0 | 90 | 44.5 |
| 12 | Шаг нашивки | 693 | 0.15 | 14.05 | 6.93 |
| 13 | Плотность нашивки | 692 | 11.74 | 103.99 | 57.57 |

Отличающийся тип параметра "Угол нашивки, град" спровоцировал интерес к детальному анализу, который показал, что признак принимает только 2 значения: 0 и 90. Это характеризует признак как категориальный, что позволяет отнести его к бинарным переменным и использовать метод One-Hot Encoding. Это обеспечит корректное представление категориальных данных для дальнейшего обучения модели.

One-Hot Encoding (OHE) — это метод преобразования категориальных переменных в числовую форму, который используется для представления категориальных данных в виде бинарных признаков. Этот метод особенно важен для алгоритмов машинного обучения, которые требуют числовых входных данных, таких как линейная регрессия, нейронные сети и другие. В One-Hot Encoding каждая категория категориальной переменной представляется отдельным столбцом, где для каждого объекта устанавливается значение 1, если он принадлежит данной категории, и 0, если не принадлежит.

Остальные переменные были отнесены к списку **numeric\_variables**, в который по условию добавляются только те столбцы данных, которые имеют числовой тип и содержат больше двух уникальных значений. Таким образом, был отсеян категориальный признак "Угол нашивки, град", который мы исследовали ранее и анализ распределения которого не имеет практического смысла.

Для анализа распределения значений числовых признаков были построены графики Boxplot (диаграмма с усами). Эти графики используются для визуализации распределения данных и выявления выбросов. Они отображают медиану, квартильные значения (Q1 и Q3) и возможные выбросы (данные, которые лежат за пределами 1.5 IQR от первого и третьего квартиля). Для каждого числового признака построен boxplot с помощью библиотеки seaborn. На рисунках 1 и 2 все графики размещены в одной матрице графиков, что помогает увидеть распределение всех числовых переменных одновременно.

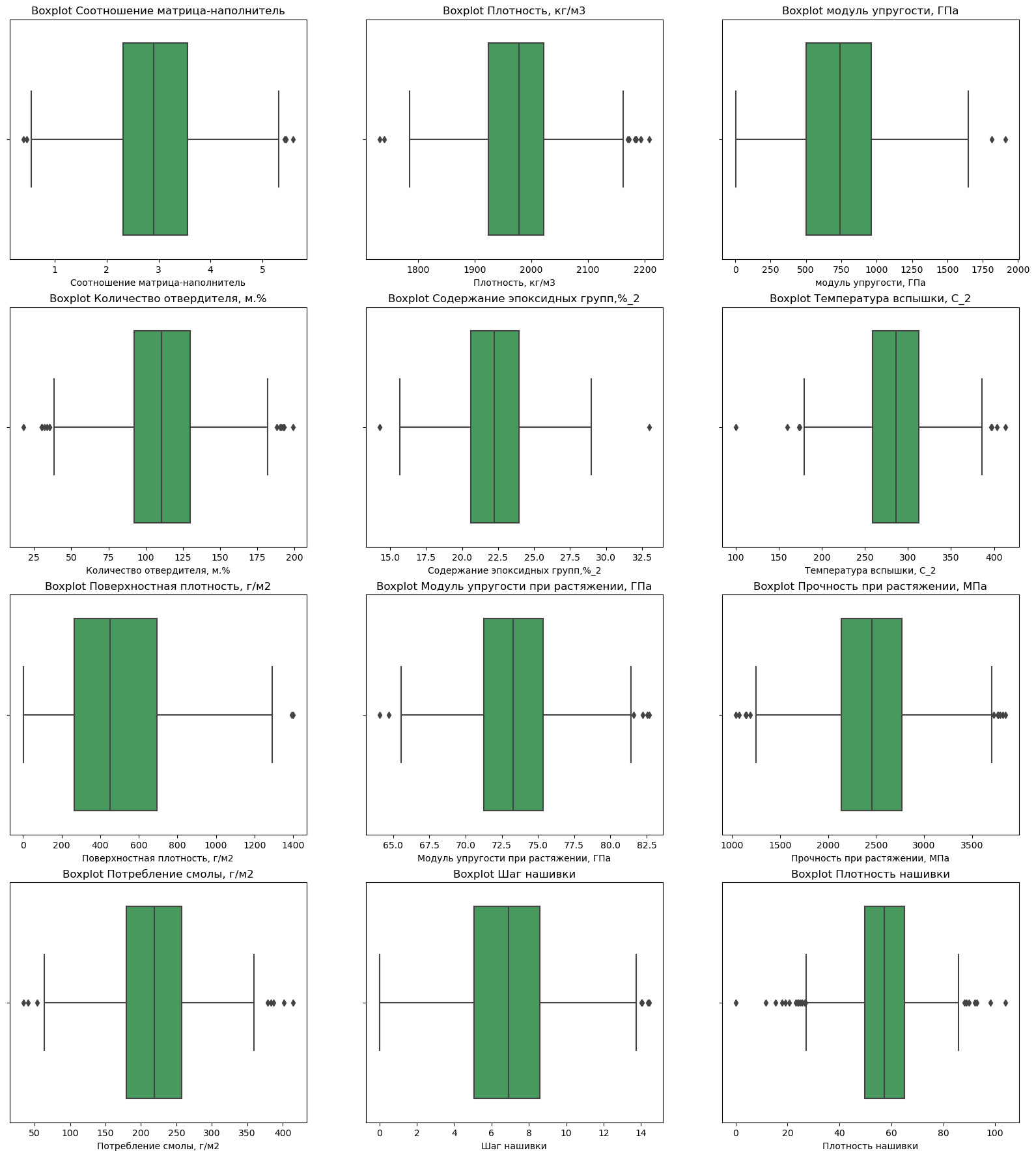


Рисунок 1- Boxlots по признакам обучающей выборки

Нормальным называется распределение вероятностей, которое для одномерного случая задается функцией Гаусса. Множество алгоритмов машинного обучения и статистических методов основываются на предположении, что данные или ошибки следуют нормальному распределению. В частности, это касается **линейных моделей** (например, линейной регрессии) и **методов, основанных на вероятностных оценках.** На всех графиках видно, что данные имеют тенденцию к нормальному распределению, но на каждом из них присутствуют **выбросы**, что требует дальнейшей работы с ними.

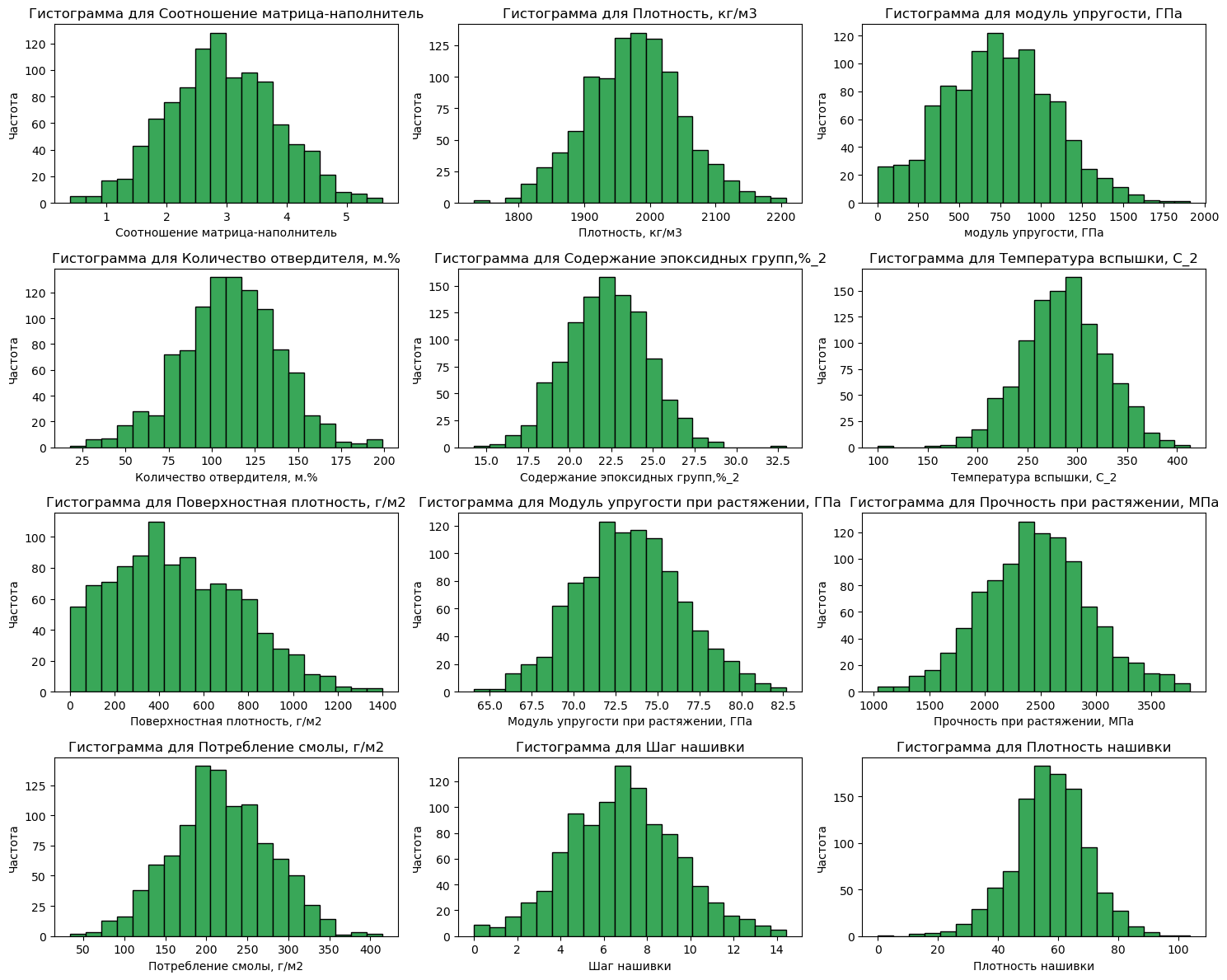


Рисунок 2 - Гистограммы по признакам обучающей выборки

Чтобы обнаружить выбросы, были использованы два метода:

1. метод 3-сигм, где выбросы определяются как значения, которые лежат за пределами трех стандартных отклонений от среднего значения (μ ± 3σ).
2. метод межквартильного расстояния (IQR), который помогает определить диапазон, в пределах которого должны лежать нормальные данные. IQR определяется как разница между третьим квартилем (Q3) и первым квартилем (Q1). Затем устанавливаются границы выбросов по формулам (1) и (2).

(1)

(2)

Весь ряд данных, который выходит за пределы этих границ, считается выбросом. Таким образом были получены данные по выбросам, вычисленные указанными способами (рисунок 3).

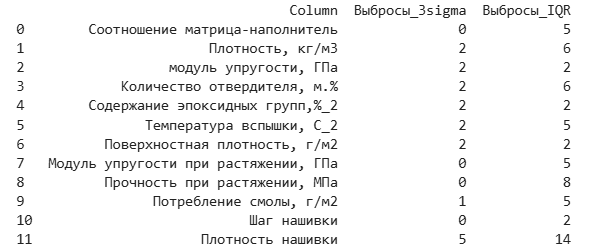


Рисунок 3 - Результаты проверки выбросов по методам 3-х сигм и IQR

После исключения выбросов методом межквартильного расстояния (IQR) были удалены не все выбросы, что видно на рисунке 4.

Визуализация данных после повторного удаления выбросов показывает в значительной стпени очищенные от шума показатели без существенных выбросов (см. рисунок 5).

* 1. **Описание используемых методов.**

1) Линейная регрессия (Linear Regression)

Линейная регрессия — это одна из самых простых и распространенных моделей машинного обучения для предсказания числовых значений. Она основана на предположении, что зависимость между входными переменными (признаками) и выходной переменной (целевой) является линейной. Формула (3) линейной регрессии выглядит следующим образом:

(3)

где y — это целевая переменная, 𝑥1,𝑥2,…,𝑥𝑛 — признаки, 𝑤0,𝑤1,…,𝑤𝑛 — параметры модели (веса).

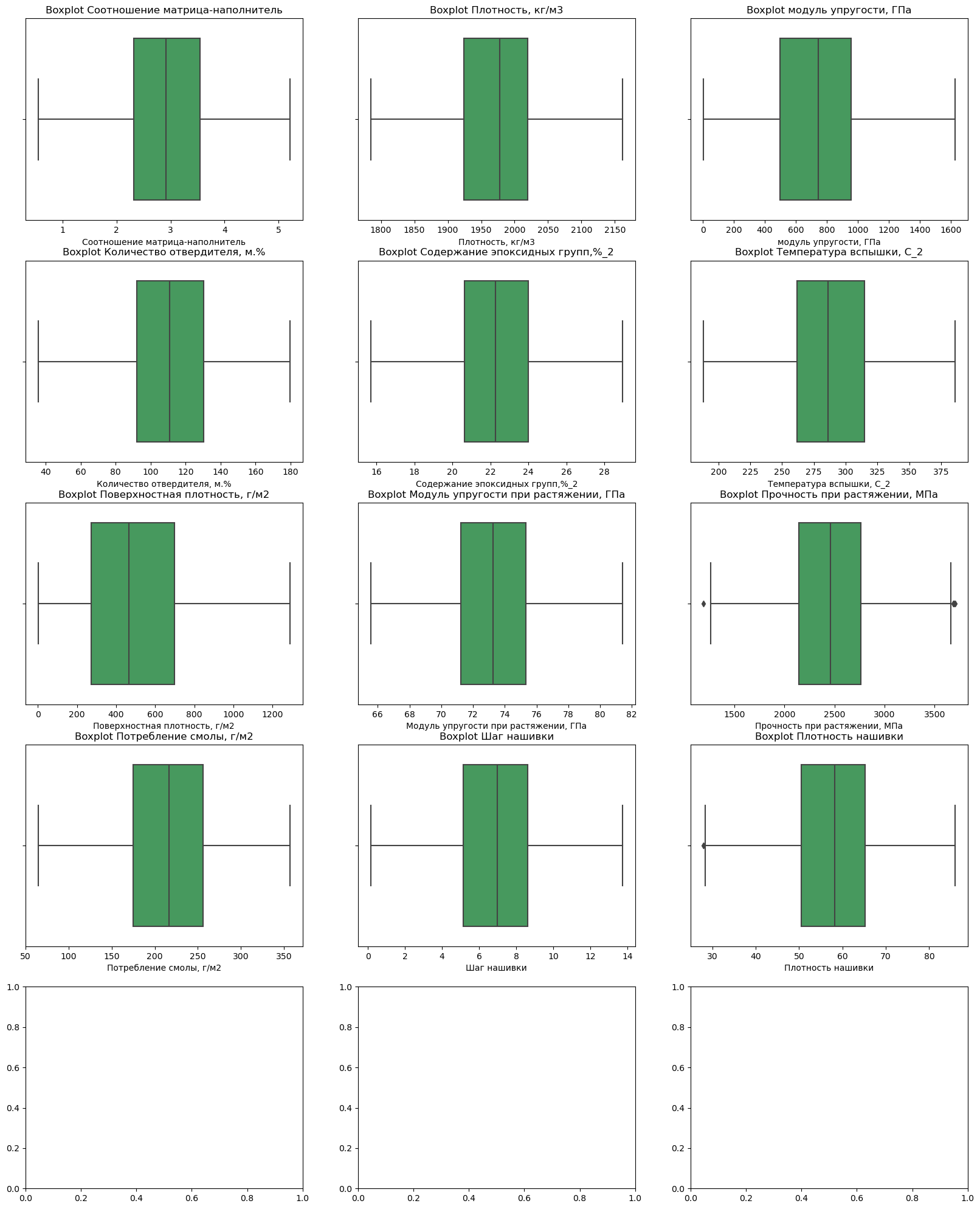


Рисунок 4 - Первичное удаление выбросов

Задача линейной регрессии — найти такие веса, которые минимизируют ошибку модели. Основная цель — минимизация функции потерь, например, среднеквадратичной ошибки, которая вычисляется по формуле (4):

(4)

где 𝑦𝑖  — реальные значения, а — предсказанные значения. Линейная регрессия эффективна при наличии линейных зависимостей между признаками и целевой переменной, но может плохо работать в случае сильных нелинейных зависимостей.

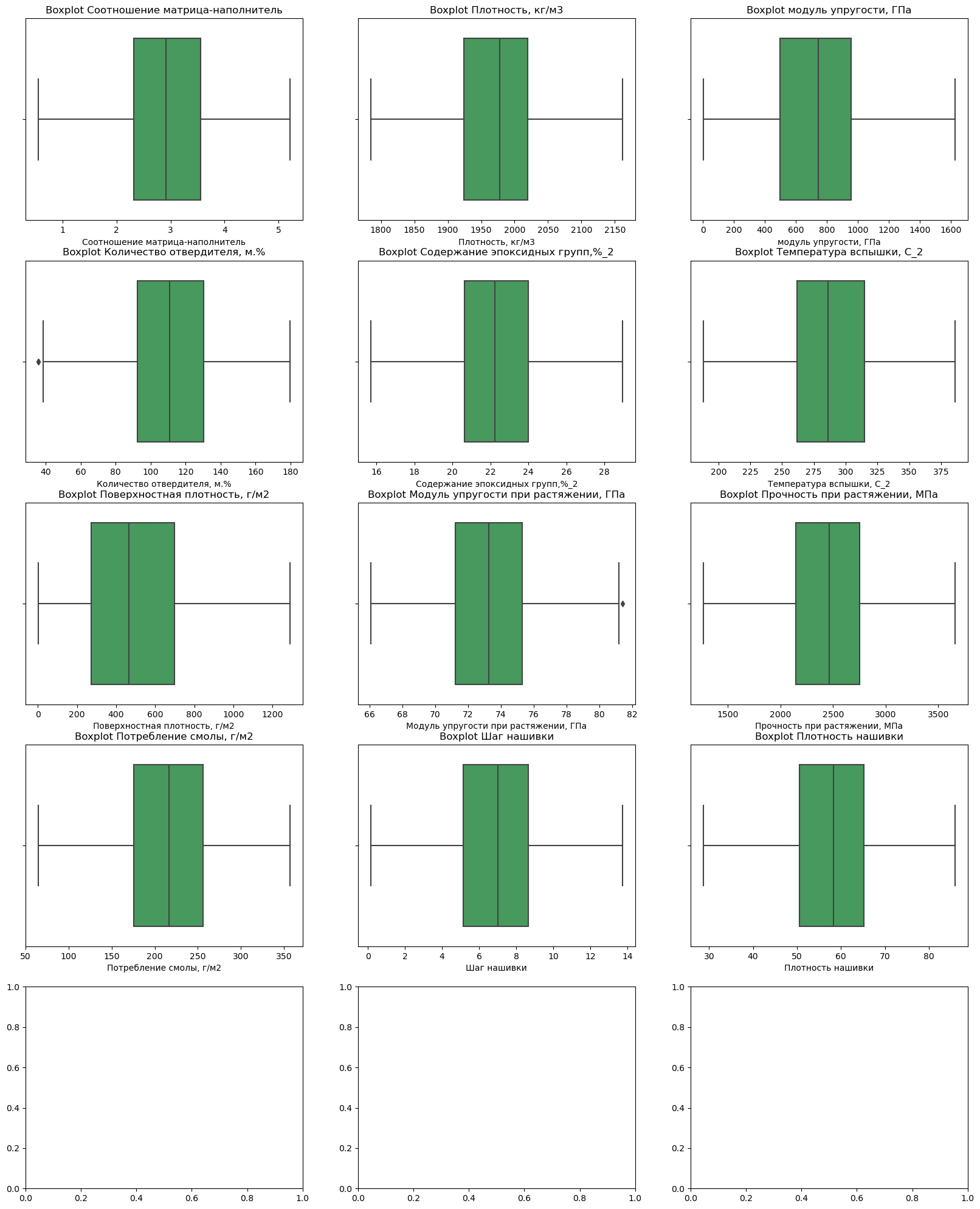


Рисунок 5 – Очищенные от выбросов показатели

2) Случайный лес (Random Forest)

Случайный лес (Random Forest) — это ансамблевый метод, который создает несколько решающих деревьев и объединяет их прогнозы. Каждое дерево строится на случайной подвыборке данных и с использованием случайных признаков при разделении на каждом узле. Это снижает вероятность переобучения (overfitting) и повышает устойчивость модели. Прогноз модели — это агрегированный результат прогнозов всех деревьев. В случае задачи регрессии прогноз рассчитывается как среднее значение предсказаний всех деревьев по формуле (5):

(5)

где — предсказания различных деревьев.

Random Forest хорошо справляется с многомерными данными и может моделировать как линейные, так и нелинейные зависимости. Однако, несмотря на свою эффективность, он может быть вычислительно дорогим, особенно при большом количестве деревьев.

1. Метод опорных векторов (SVR)

Support Vector Regressor (SVR) основан на методе опорных векторов (SVM), который широко используется для классификации и регрессии. SVR находит гиперплоскость, которая наилучшим образом аппроксимирует данные, минимизируя ошибку, но при этом учитывает некоторую погрешность. Важная особенность SVR — использование так называемого «эпсилон-неактивного» промежутка, где ошибки модели считаются несущественными, если они меньше заданного значения 𝜖. Формула (7) функции потерь для SVR выглядит следующим образом:

(7)

В случае значительных отклонений от этой гиперплоскости, применяется штраф. Это позволяет SVR быть устойчивым к выбросам, но модель может плохо работать на данных с высокой шумностью. SVR подходит для задач, где важно точное аппроксимирование данных, но требует тщательной настройки параметров.

1. Стохастический градиентный спуск (SGD Regressor)

Стохастический градиентный спуск (SGD) — это метод оптимизации, используемый для обучения различных моделей, включая линейную регрессию, логистическую регрессию и другие. В отличие от традиционного градиентного спуска, который использует весь набор данных для вычисления градиента, стохастический градиентный спуск обновляет параметры модели после каждого отдельного примера, что значительно ускоряет обучение. Обновление параметров происходит по следующей формуле (8):

*(8)*

где η — скорость обучения, ∇𝑤𝐿(𝑤) — градиент функции потерь. Стохастический градиентный спуск эффективен на больших объемах данных и позволяет избежать переобучения, но требует аккуратной настройки гиперпараметров, таких как скорость обучения и регуляризация.

1. Многослойный перцептрон (MLP Regressor)

Многослойный перцептрон (MLP) — это искусственная нейронная сеть с несколькими скрытыми слоями. MLP может моделировать сложные нелинейные зависимости в данных, что делает его мощным инструментом для регрессионных задач. Входной слой принимает признаки, а выходной слой генерирует предсказания. Каждый скрытый слой содержит несколько нейронов, которые обучаются с использованием функции активации, такой как ReLU или сигмоида. Формула (9) для нейронной сети с одним скрытым слоем:

*(9)*

где σ — функция активации, 𝑊1 и 𝑊2 — веса, 𝑏1 и 𝑏2 — смещения.

MLP может эффективно обучаться с использованием алгоритма обратного распространения ошибки и оптимизации градиентным спуском. Однако модели с большим количеством слоев могут быть подвержены переобучению, и необходимо правильно настроить гиперпараметры.

1. Градиентный бустинг (Gradient Boosting)

Градиентный бустинг (GB) — это ансамблевый метод, который строит несколько деревьев решений по принципу последовательного улучшения ошибок предыдущих деревьев. Каждое новое дерево исправляет ошибки модели, обучаясь на остатках предыдущего дерева. Это позволяет достичь высокой точности, несмотря на простоту отдельных деревьев. Важная особенность GB заключается в том, что в отличие от случайного леса, где деревья обучаются параллельно, деревья в градиентном бустинге строятся последовательно. Прогноз модели рассчитывается как взвешенная сумма предсказаний деревьев. Формула (10) для обучения на каждом шаге:

(10)

где 𝑦𝑡 — предсказания после t-го дерева, 𝑓𝑡(𝑥) — предсказания t-го дерева, η — коэффициент обучения.

GB требует тщательной настройки гиперпараметров, таких как количество деревьев, максимальная глубина деревьев и коэффициент обучения.

8. Lasso (Lasso Regression)

Lasso (Least Absolute Shrinkage and Selection Operator) — это метод линейной регрессии с добавлением L1-регуляризации, которая позволяет не только улучшить обобщающие способности модели, но и производить отбор признаков. Lasso применяет штраф в виде суммы абсолютных значений коэффициентов модели по формуле (8):

(8)

где λ — коэффициент регуляризации, 𝑤𝑖 — веса модели. Lasso помогает устранить незначимые признаки, приводя их веса к нулю, что делает модель более интерпретируемой. Это полезно в случаях, когда имеется большое количество признаков, многие из которых могут быть несущественными для задачи.

Преимущества, недостатки и априорные предпосылки для работоспособности указаны в таблице 3.

Таблица 3 – Сравнительые характеристики используемых моделей машинного обучения

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Преимущества | Недостатки | Априорные предпосылки для работоспособности |
| Линейная регрессия (Linear Regression) | Простой и легко интерпретируемый метод. Быстро обучается и работает с малым количеством данных. | Не подходит для сложных, нелинейных зависимостей. Подвержена переобучению при многократной корреляции признаков. | Подразумевает линейную зависимость между признаками и целевой переменной. Признаки должны быть независимыми. Подходит для линейных моделей. |
| Случайный лес (Random Forest) | Хорошо справляется с большими объемами данных и сложными зависимостями. Работает хорошо с неполными данными. | Может быть вычислительно дорогим, особенно при большом количестве деревьев. | Требует достаточного количества данных для построения ансамбля. Хорошо работает с категориальными и числовыми признаками. Не требует линейности зависимостей между признаками и целевой переменной. |
| Метод опорных векторов (SVR) | Подходит для данных с малым количеством признаков и для задач с высоко чувствительными данными. | Трудно масштабируется для больших наборов данных, может требовать длительного времени обучения. | Эффективен для данных с четкими границами и когда важно точно аппроксимировать данные. Хорошо работает с высокоразмерными данными, но требует тщательной настройки параметров. |
| Стохастический градиентный спуск (SGD) | Эффективен для больших данных и может обучать как линейные, так и нелинейные модели. Высокая скорость обучения. | Требует настройки гиперпараметров (например, скорости обучения). Может сильно колебаться при плохой настройке. | Подходит для больших объемов данных. Требует масштабирования признаков. Хорошо работает для задач с большим количеством данных и признаков. |
| Многослойный перцептрон (MLP) | Может моделировать сложные нелинейные зависимости. Хорошо работает с большими объемами данных. | Может легко переобучиться при малом объеме данных или неправильной настройке. Требует много времени на обучение. | Подходит для задач, где присутствуют сложные нелинейные зависимости. Хорошо работает с большими наборами данных, но может быть склонен к переобучению без правильной регуляризации. |
| Градиентный бустинг (Gradient Boosting) | Отлично справляется с задачами, где важны сложные зависимости. Часто дает лучшие результаты среди моделей. | Высокая вычислительная сложность, особенно на больших данных. Может быть подвержен переобучению. | Подходит для задач с большими объемами данных, хорошо работает с отсутствующими значениями и сильно неотфильтрованными данными. Может быть чувствителен к настройке гиперпараметров. |
| Lasso | Помогает при отборе признаков, сокращая их количество, что улучшает интерпретируемость модели. | Могут возникнуть проблемы с переобучением при слишком сильной регуляризации. | Подходит для линейных зависимостей. Эффективен, если данные содержат много ненужных признаков. Линейная зависимость между признаками и целевой переменной. |
| Стэкинг (Stacking) | Использует сильные стороны различных моделей для улучшения итогового результата. | Модель может быть сложной для понимания и требует много вычислительных ресурсов. | Подходит для комплексных задач, где требуется комбинация нескольких моделей для получения более точных предсказаний. Эффективен, если используемые модели хорошо обучены на основе данных. |

## 1.3 Разведочный анализ данных

Разведочный анализ данных (EDA) — это важный этап в процессе анализа данных, который помогает лучше понять структуру, свойства и возможные проблемы в наборе данных. В ходе EDA исследуются взаимосвязи между переменными, их распределение, наличие выбросов, пропусков и прочих аномалий. Методы EDA играют ключевую роль в подготовке данных для последующего моделирования. Ниже перечислены основные методы и техники, которые часто используются в EDA:

1) Описание данных

Статистические характеристики: Перед началом анализа важно изучить основные статистические параметры данных. Для этого используется метод describe() в pandas, который выводит статистику для каждого числового столбца: среднее значение, стандартное отклонение, минимальное и максимальное значения, квартильные значения и количество непустых значений.

Этот метод помогает быстро оценить распределение и масштабы значений переменных, а также выявить возможные проблемы, такие как выбросы или пропуски.

2) Проверка на пропуски и аномалии

Пропуски: Для выявления пропущенных значений в датасете используется метод isnull(). Он позволяет определить, какие ячейки содержат пропуски. Данные с пропущенными значениями могут быть обработаны с помощью методов заполнения или удаления.

Аномалии: Аномалии или выбросы — это значения, которые значительно отклоняются от других наблюдений. Их можно обнаружить с помощью визуализаций (например, boxplot или гистограммы) или статистических методов (например, Z-оценка или IQR).

3) Визуализация данных

Гистограммы: Гистограммы отображают распределение данных по интервалам. Это один из самых простых способов визуализировать распределение данных, обнаружить его асимметричность, нормальность и наличие выбросов.

Boxplot (ящик с усами): Это график, который помогает визуализировать медиану, квартали, а также выявить выбросы. Boxplot отображает межквартильный размах (IQR) и дает информацию о центральных тенденциях.

Диаграмма рассеяния (scatter plot): Используется для анализа зависимостей между двумя числовыми переменными. Этот метод помогает визуализировать возможные линейные и нелинейные связи.

Матрица корреляции: Для оценки силы и направления взаимосвязи между переменными используется матрица корреляций. Это позволяет определить, какие переменные могут быть сильнее связаны друг с другом.

4) Обнаружение выбросов

Метод межквартильного размаха (IQR): Один из популярных методов обнаружения выбросов. Он вычисляет диапазон, в котором должны располагаться данные. Значения, выходящие за пределы 1.5 \* IQR от первого и третьего квартилей, считаются выбросами.

Метод Z-оценки: Этот метод использует стандартное отклонение для измерения отклонений каждого значения от среднего. Обычно значения с Z-оценкой больше 3 или меньше -3 считаются выбросами.

Методы разведочного анализа данных (EDA) помогают выявить важные аспекты данных, такие как распределение переменных, наличие выбросов, зависимостей и других аномалий. Этап EDA является неотъемлемой частью процесса подготовки данных для последующего построения моделей машинного обучения и помогает улучшить качество прогнозирования.

Применение этих методов к нашей задаче описано выше в разделе 1.1.

После удаления выбросов мы можем приступить к поиску зависимостей между переменными.

Корреляция — это статистический показатель, который измеряет степень линейной зависимости между двумя переменными. Корреляция показывает, как изменение одной переменной связано с изменением другой, и выражается числовым значением, которое варьируется от -1 до +1. Корреляция, близкая к нулю (от -0.1 до +0.1), указывает на отсутствие или слабую линейную зависимость между переменными. Корреляция обычно измеряется с помощью коэффициента корреляции Пирсона.

Корреляционная матрица — это таблица, которая показывает коэффициенты корреляции между всеми возможными парами переменных в наборе данных. Она полезна для анализа взаимосвязей между признаками и позволяет выявить сильные и слабые связи между ними.

Корреляционные матрицы являются важным инструментом для анализа данных, так как они помогают:

* Выявить мультиколлинеарность — ситуацию, когда два или более признака сильно коррелируют между собой. Это может привести к проблемам в моделях машинного обучения, например, в линейной регрессии, где мультиколлинеарность увеличивает нестабильность коэффициентов.
* Выбрать значимые признаки — если два признака сильно коррелируют, можно оставить только один из них, тем самым упрощая модель и уменьшая риск переобучения.
* Определить зависимости между переменными — в задаче прогнозирования свойств материалов корреляция помогает понять, как изменения в одном свойстве материала могут влиять на другие свойства.

Как видно из рисунка 6, в нашей задаче не удалось выявить никаких линейных зависимостей между переменными.

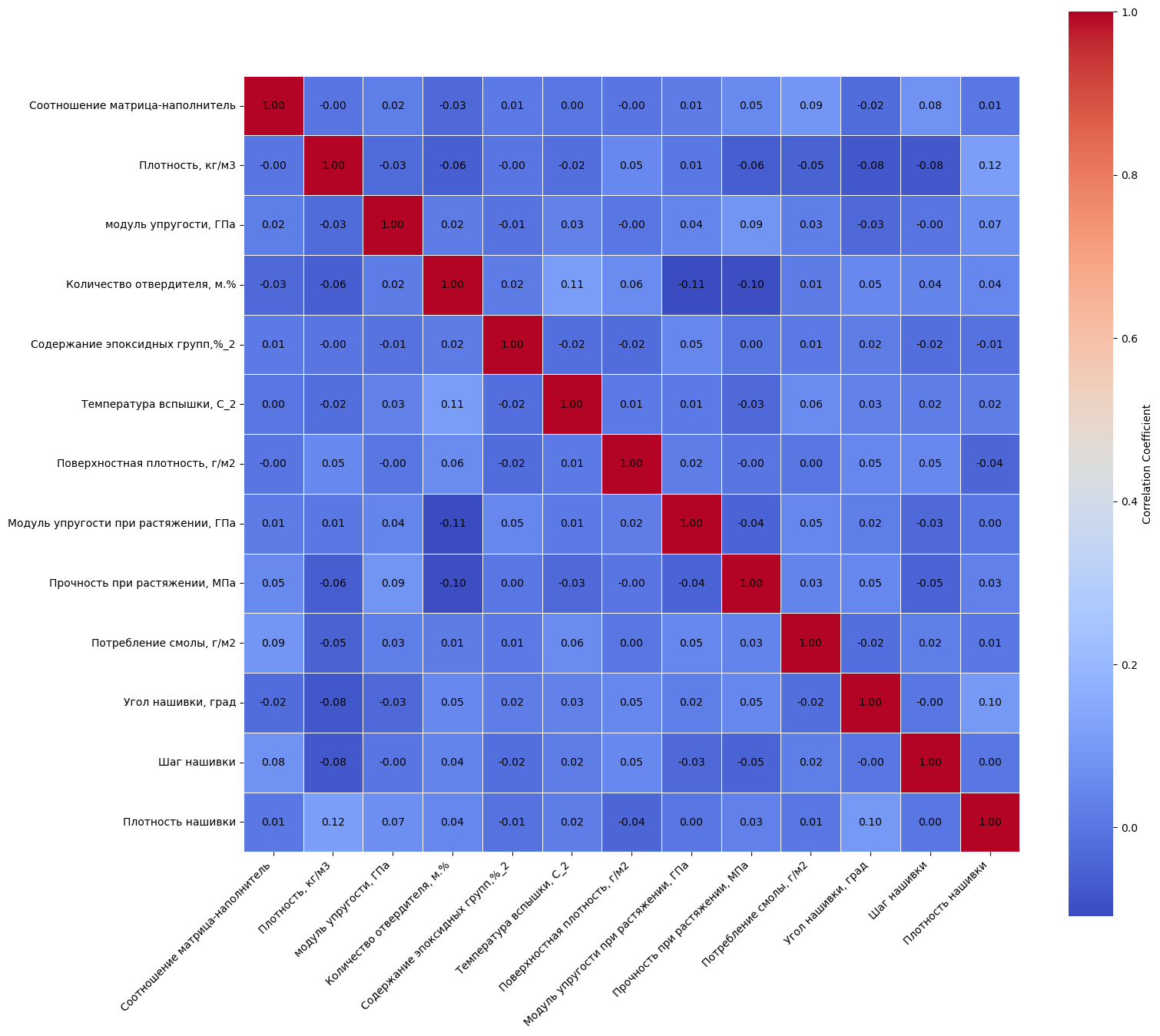


Рисунок 6 – Матрица корреляции показателей

# 2. Практическая часть

## 2.1 Предобработка данных

Для предобработки данных используются две основные техники - нормализация и стандартизация. Обе эти техники помогают решить проблемы, связанные с масштабированием данных, а также обеспечивают более эффективное обучение моделей, таких как линейные регрессии, поддерживающие вектора, нейронные сети и многие другие.

1) Нормализация данных

Нормализация данных (или min-max масштабирование) — это процесс преобразования всех признаков в диапазон от 0 до 1 (или в другой заданный диапазон). Это позволяет привести данные к общему масштабу, что особенно важно для алгоритмов, чувствительных к масштабу данных, таких как нейронные сети и алгоритмы кластеризации (формула 9).

(9)

где: X — исходное значение признака, 𝑋min — минимальное значение признака в данных, 𝑋max — максимальное значение признака в данных.

Этот метод полезен, когда данные имеют разные масштабы, и важно привести их к единому виду. Например, переменные, измеренные в разных единицах (граммы, метры, проценты), могут существенно влиять на обучение модели, если не применить нормализацию.

В нашем случае задача заключается в прогнозировании различных характеристик композиционных материалов, таких как модуль упругости при растяжении или прочность при растяжении. Эти характеристики могут значительно отличаться по величине, и признаки, такие как плотность, температура вспышки и другие, могут иметь различные диапазоны значений. Применение нормализации позволит привести все признаки к одинаковому масштабу и устранить влияние масштаба на обучение модели.

Нормализация полезна для алгоритмов, основанных на вычислениях расстояний (например, для кластеризации или алгоритмов, использующих эвклидово расстояние).

Она может быть чувствительна к выбросам, так как данные, выходящие за пределы нормализованного диапазона, могут существенно влиять на результат.

2) Стандартизация данных

Стандартизация (или z-преобразование) — это процесс преобразования данных таким образом, чтобы они имели нулевое среднее и стандартное отклонение, равное единице. Этот метод используется, когда данные имеют разные масштабы или распределение, не приближенное к нормальному. Стандартизация устраняет влияние масштаба, но сохраняет распределение данных, что важно для многих алгоритмов машинного обучения, таких как линейная регрессия, SVM и нейронные сети (формула 10).

(10)

где: X — исходное значение признака, μ — среднее значение признака в данных, σ — стандартное отклонение признака.

При стандартизации данные преобразуются так, что они имеют распределение с нулевым средним и единичным стандартным отклонением.

В задаче прогнозирования свойств композиционных материалов стандартизация может быть полезной для алгоритмов, которые чувствительны к распределению данных. Например, если данные имеют большую дисперсию или асимметричное распределение, стандартизация позволяет устранить эти проблемы и привести все признаки к нормальному виду. Для переменных, таких как температура вспышки или модуль упругости при растяжении, стандартизация позволит привести их к общему виду, независимо от их изначальной шкалы. Стандартизированные данные будут иметь нулевое среднее, и алгоритм будет рассматривать признаки на равных.

Стандартизация используется для алгоритмов, которые предполагают нормальное распределение данных или чувствительны к различиям в масштабе (например, линейные модели, SVM и нейронные сети).

Она не зависит от минимальных и максимальных значений и может быть полезной, если данные имеют выбросы, которые нормализация может исказить.

Таким образом, в процессе предобработки данных помимо исходного датасета, очищенного от выбросов, были сформированы нормализованный и стандартизированный датасеты..

## 2.2 Разработка и обучение модели.

Для разработки модели предсказания модуля упругости и прочности при растяжении были использованы поочередно следующие методы (рисунок 7):

1. Линейная регрессия (Linear Regression)

2. Случайный лес (Random Forest)

3. Гребневая регрессия (Ridge Regression)

4. Поддерживающий вектор регрессор (SVR)

5. Стохастический градиентный спуск (SGD Regressor)

6. Многослойный перцептрон (MLP Regressor)

7. Градиентный бустинг (Gradient Boosting)

8. Lasso (Lasso Regression)

Теоретическое описание этих методов было приведено в разделе 1.2.



Рисунок 7 – Часть кода на Python для обучения и оценки моделей

## 2.3 Тестирование модели.

Для оценки моделей были выбраны три метрики:

1) MAE (Mean Absolute Error) — Средняя абсолютная ошибка между фактическими и предсказанными значениями.

MAE измеряет среднюю величину отклонений предсказанных значений от реальных, при этом не делает различий между большими и малыми ошибками. Это дает интуитивно понятное представление о том, насколько точно модель предсказывает значения.

Этот показатель устойчив к выбросам в меньшей степени, чем MSE, поскольку не возводит ошибки в квадрат, а также он ее учитывает сильные выбросы, так как все ошибки одинаково важны.

2) MSE (Mean Squared Error) — Средняя квадратичная ошибка между фактическими и предсказанными значениями.

MSE также измеряет среднее отклонение между фактическими и предсказанными значениями, но делает это, возводя ошибки в квадрат. Это приводит к большему штрафу за большие ошибки.

Этот показатель подходит для оценки моделей, где важно минимизировать большие ошибки (выбросы). MSE сильно зависит от выбросов, так как квадратичные отклонения делают большую ошибку еще более значимой.

3) R² (коэффициент детерминации)

R² измеряет долю вариации зависимой переменной, объясненную моделью, указывает на то, сколько процентов вариации целевой переменной объясняется моделью. Если R² равно 1, модель идеально предсказывает данные, если R² равно 0, модель не объясняет данные лучше, чем простое среднее значение.

R² не всегда является хорошей метрикой для оценки моделей, если данные имеют нелинейные зависимости.

* + 1. Прогнозирование модуля упругости при растяжении

1. Базовые модели

После обучения и оценки моделей были получены результаты, представленные в таблице 4.

Таблица 4 – Оценка моделей для прогнозирования модуля упругости при растяжении

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Train MAE** | **Test MAE** | **Train MSE** | **Test MSE** | **Train R²** | **Test R²** |
| Linear Regression | 0,16 | 0,16 | 0,04 | 0,04 | 0,02 | -0,02 |
| Random Forest | 0,06 | 0,17 | 0,01 | 0,04 | 0,85 | -0,06 |
| Support Vector Regressor | 0,11 | 0,18 | 0,02 | 0,05 | 0,49 | -0,23 |
| SGD Regressor | 0,16 | 0,16 | 0,04 | 0,04 | -0,04 | -0,02 |
| MLP Regressor | 0,15 | 0,17 | 0,04 | 0,04 | 0,07 | -0,06 |
| Gradient Boosting | 0,11 | 0,17 | 0,02 | 0,05 | 0,49 | -0,12 |
| Lasso | 0,16 | 0,16 | 0,04 | 0,04 | 0,00 | 0,00 |
| Stacking | 0,15 | 0,16 | 0,03 | 0,04 | 0,09 | -0,01 |

1. Linear Regression

Модель линейной регрессии показывает хорошие результаты на обучающих данных, но на тестовой выборке её метрики ухудшаются. Очень низкие значения R² указывают на слабую способность модели объяснять вариацию данных, как на обучающей, так и на тестовой выборке. Это может быть связано с тем, что линейная регрессия не способна захватывать сложные зависимости в данных.

2. Random Forest

Random Forest показывает отличные результаты на обучающих данных с очень низким MAE и MSE, а также высоким R². Однако на тестовых данных модель значительно ухудшает свою производительность, что указывает на возможное переобучение. Модель слишком хорошо подстроилась под обучающие данные, но не обобщает на новые данные.

3. Support Vector Regressor

Support Vector Regressor показывает довольно хорошие результаты на обучающих данных с относительно низким MAE и MSE, а также хорошим R². Однако на тестовых данных результаты значительно ухудшаются, что также указывает на переобучение. Модель плохо обобщает на новые данные.

3. SGD Regressor

SGD Regressor дает стабильные результаты как на обучающих, так и на тестовых данных. Хотя метрики MAE и MSE на уровне других моделей, очень низкие значения R² показывают, что модель не объясняет данные должным образом и плохо обобщает.

5. MLP Regressor (Multi-layer Perceptron)

MLP Regressor, как и другие нейронные сети, показывает сильное ухудшение на тестовых данных. Высокие значения MAE и MSE на тесте и низкие значения R² говорят о плохой способности нейронной сети обобщать на новые данные.

6. Gradient Boosting

Gradient Boosting работает хорошо на обучающих данных, показывая низкие значения MAE и MSE с довольно высоким R². Однако на тестовых данных результаты значительно ухудшаются, что указывает на сильное переобучение модели.

7. Lasso

Lasso демонстрирует стабильность в метриках на обучающих и тестовых данных. Однако её R² близко к нулю, что указывает на слабую объясняющую способность модели.

8. Stacking

Stacking показывает хорошие результаты по MAE и MSE на тестовой выборке, но её R² остается отрицательным как на обучении, так и на тесте, что указывает на слабую способность модели объяснять вариацию данных.

Выводы:

* Лучшая модель на обучении: Random Forest показывает результаты с высоким R² (0.852360) и низкими значениями MAE и MSE. Это делает её лучшей моделью для обучающих данных.
* Лучшая модель на тестовых данных: Lasso имеет наименьшие MAE и MSE на тестовой выборке, а также лучшую модель среди всех с самым близким к нулю значением R².
* Однако, стоит отметить, что все модели показывают довольно низкие значения R², что говорит о сложности задачи или о плохом качестве данных.

1. Попытка улучшить Random Forest

В связи с тем, что модель Random Forest хорошо показала себя на обучающей выборке, было решено выявить наиболее важные для модели признаки. Топ-5 признаков по важности: Количество отвердителя, м.%, 'Температура вспышки, С\_2, Потребление смолы, г/м2, Плотность, кг/м3, Прочность при растяжении, Мпа.

Их перечисленных признаков был сформирован новый датасет и запущено обучение Случайного леса.

К сожалению, улучшить модель таким способом не удалось. R² на тестовой выборке снова меньше 0.

1. Попытка улучшить Lasso

Использованные методы:

* LASSO коэффициенты - веса признаков, которые определяют их вклад в предсказания модели. Некоторые коэффициенты становятся равными нулю из-за регуляризации. Это позволяет выбирать только важные признаки.
* Поиск гиперпараметров по сетке (GridSearchCV) - это метод для поиска оптимального параметра модели. В данном случае ищется alpha – параметр регуляризации, который контролирует штраф. С помощью GridSearchCV происходит кросс-валидация по сетке значений параметров.

Вышеперечисленные методы не помогли улучшить модель. R² остался на уровне нуля.

* + 1. Прогнозирование прочности при растяжении

После обучения и оценки моделей были получены результаты, представленные в таблице 5.

Таблица 5 - Оценка моделей для прогнозирования прочности при растяжении

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Train MAE** | **Test MAE** | **Train MSE** | **Test MSE** | **Train R²** | **Test R²** |
| Linear Regression | 0,15 | 0,16 | 0,03 | 0,04 | 0,04 | -0,04 |
| Random Forest | 0,06 | 0,16 | 0,01 | 0,04 | 0,86 | 0,01 |
| Support Vector Regressor | 0,11 | 0,18 | 0,02 | 0,05 | 0,52 | -0,26 |
| SGD Regressor | 0,15 | 0,17 | 0,04 | 0,04 | -0,05 | -0,10 |
| MLP Regressor | 0,15 | 0,17 | 0,03 | 0,04 | 0,06 | -0,09 |
| Gradient Boosting | 0,10 | 0,17 | 0,02 | 0,04 | 0,55 | -0,08 |
| Lasso | 0,15 | 0,16 | 0,04 | 0,04 | 0,00 | 0,00 |
| Stacking | 0,15 | 0,16 | 0,03 | 0,04 | 0,04 | -0,02 |

1. Linear Regression

Модель линейной регрессии имеет низкие значения MAE и MSE как для обучающих, так и для тестовых данных. Однако R² остаётся низким (около 0 на обучающих данных и отрицательным на тесте), что говорит о слабой способности модели объяснять вариацию данных. Линейная регрессия может не быть лучшим выбором для данной задачи.

2. Random Forest

Random Forest показывает отличные результаты на обучающих данных, с очень низким MAE и MSE и высоким R² (0.86). Однако на тестовых данных модель демонстрирует ухудшение, что указывает на переобучение. Несмотря на это, она остаётся одной из лучших моделей для обучения, хотя обобщение на тестовых данных является её слабостью.

3. Ridge Regression

Ridge Regression имеет похожие результаты с линейной регрессией, показывая те же значения MAE и MSE. Модель не объясняет данные хорошо, и её производительность ухудшается на тестовых данных. Возможно, Ridge не подходит для данной задачи.

4. Support Vector Regressor

Support Vector Regressor показывает хорошие результаты на обучающих данных с низкими значениями MAE и MSE, но сильно ухудшается на тестовых данных. R² также падает на тесте, что указывает на переобучение. Это также не является идеальной моделью для данной задачи.

5. SGD Regressor

SGD Regressor демонстрирует стабильные значения MAE и MSE как на обучающих, так и на тестовых данных, но его R² остаётся низким на обоих наборах данных, что свидетельствует о низкой объясняющей способности модели.

6. MLP Regressor

MLP Regressor (нейронная сеть) показывает неплохие результаты на обучающих данных, но её производительность значительно ухудшается на тесте. Низкие значения R² указывают на проблемы с обобщением модели.

7. Gradient Boosting

Gradient Boosting показывает хорошие результаты на обучающих данных, но также сильно ухудшается на тестовых, что указывает на возможное переобучение. Модель показывает стабильные метрики, но её обобщение на новых данных оставляет желать лучшего.

8. Lasso

Lasso даёт стабильные результаты по MAE и MSE, но её способность объяснять данные (по R²) минимальна. Это может быть связано с высокой регуляризацией, ограничивающей способность модели для более сложных зависимостей.

9. Stacking

Stacking (комбинированная модель) имеет схожие результаты с другими моделями, но её R² остаётся отрицательным как на обучении, так и на тесте, что говорит о её слабой способности объяснять данные.

Выводы по выбору лучшей модели:

Лучшая модель на обучении: Random Forest имеет наилучший R² (0.86) и низкие значения MAE и MSE, что указывает на хорошее подстраивание под данные обучающей выборки. Однако она переобучается на тестовой выборке.

Лучшая модель на тесте: Random Forest также имеет наименьшее MAE на тестовых данных среди всех моделей (0.16), но её R² существенно снижается на тестовой выборке. Наиболее стабильными остаются Linear Regression и Lasso, но их R² остаются близкими к нулю, что свидетельствует о слабой способности модели объяснять зависимость.

## 2.4 Создание нейронной сети для рекомендации соотношения матрицы.

Модель нейронной сети, построенная с использованием Keras (пакет для глубокого обучения, который является частью TensorFlow), представляет собой последовательную модель (Sequential). Это означает, что каждый слой нейронной сети идет один за другим, без ветвлений.

Количество нейронов: Входной слой содержит 64 нейрона. Количество нейронов в этом слое определяет размерность входных данных, которые поступают в нейронную сеть.

Размер входных данных - это количество признаков (фичей) в обучающих данных. То есть, если в данных 10 признаков, входной слой будет содержать 10 нейронов.

Функция активации: Для входного слоя выбрана функция активации ReLU (Rectified Linear Unit). Эта функция активации позволяет нейрону передавать только положительные значения, и в случае отрицательных значений она будет передавать 0. ReLU хорошо работает на больших объемах данных и помогает избежать проблемы исчезающего градиента.

Скрытые слои (Hidden Layers):

Количество нейронов: В этой модели добавлены два скрытых слоя. Первый скрытый слой имеет 64 нейрона, второй — 32 нейрона. Количество нейронов в скрытых слоях обычно выбирается эксперименты, но уменьшение числа нейронов на каждом слое помогает снизить вычислительные затраты и уменьшить переобучение.

Функция активации: Для этих скрытых слоев также используется ReLU. Это распространенная функция активации для скрытых слоев, так как она позволяет моделям обучаться быстрее и делает их менее склонными к переобучению.

Выходной слой (Output Layer):

Количество нейронов: В выходном слое только 1 нейрон, так как предсказывается одно значение (например, прочность при растяжении). Для задач регрессии на выходе обычно используется один нейрон.

Нейрон будет использовать линейную активацию по умолчанию. Для задач регрессии такой выход является оптимальным, так как предсказывается непрерывное число.

Компиляция модели:

Оптимизатор: Для обучения используется Adam — один из наиболее популярных оптимизаторов для нейронных сетей, который адаптирует скорость обучения для каждого параметра.

Для задачи регрессии выбрана среднеквадратичная ошибка (MSE, Mean Squared Error). Эта функция потерь измеряет средний квадрат разности между предсказанными и реальными значениями. Чем меньше значение MSE, тем лучше модель.

Для мониторинга процесса обучения используется средняя абсолютная ошибка (MAE, Mean Absolute Error). Эта метрика показывает, насколько в среднем ошибается модель, игнорируя знак ошибки, и часто используется для регрессионных задач, когда важен масштаб ошибки.

Модель обучается на протяжении 20 эпох.

Таблица 6 - Метрики оценки нейронной сети

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **MAE** | **MSE** | **R²** |
| NeuralNetwork Train | 0,14 | 0,03 | 0,23 |
| NeuralNetwork Test | 0,17 | 0,05 | -0,13 |

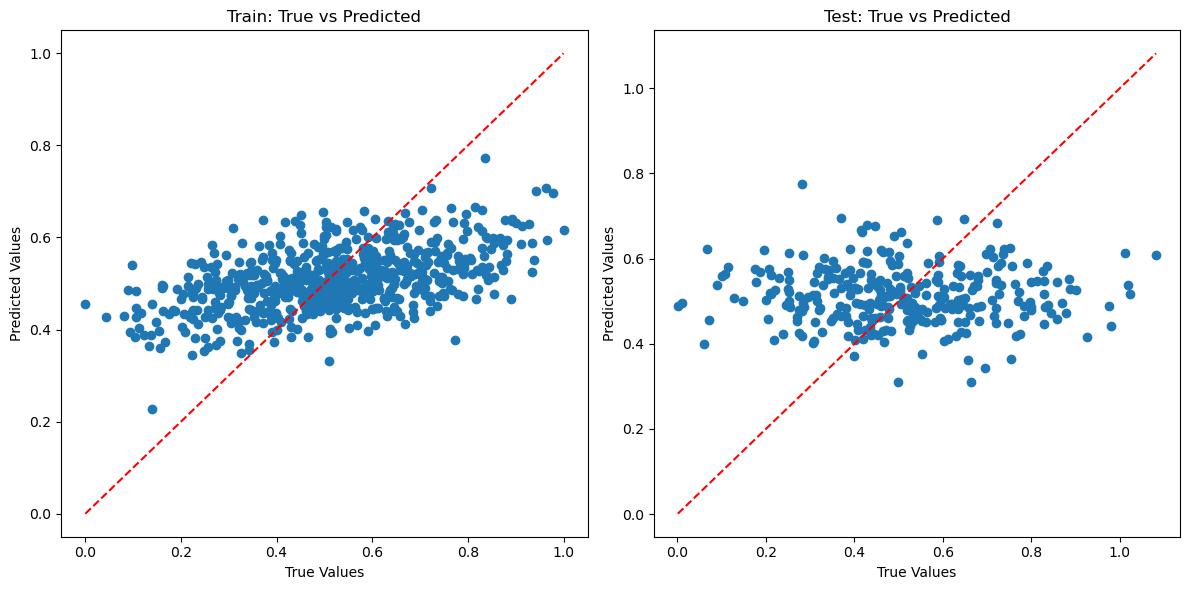


Рисунок 8 – Визуализация предсказаний нейронной сети

Как видно из рисунка 8, результаты модели неудовлетворительны, поэтому решено было преобразовать нейронную сеть. Архитектура второго варианта нейронной сети выглядит следующим образом:

Входной слой:

* 128 нейронов
* Активация: LeakyReLU
* Нормализация: BatchNormalization
* Дроп-аут: 20%

Скрытые слои:

* 64 нейрона, активация: LeakyReLU, нормализация: BatchNormalization, дроп-аут: 20%
* 32 нейрона, активация: LeakyReLU, нормализация: BatchNormalization, дроп-аут: 20%

Выходной слой:

* 1 нейрон (для регрессии)
* Без активации (линейная функция)

Компиляция:

* Оптимизатор: Adam
* Функция потерь: Huber
* Метрики: mean\_absolute\_error

Обучение:

* Эпохи: 100
* Мини-батчи: 32
* Валидация: тестовая выборка

Архитектура направлена на регрессию с использованием обычных скрытых слоев и модификаций для предотвращения переобучения (дроп-аут, нормализация).

В таблице 7 представлены результаты оценки предсказаний модели на обучающей и тестовой выборках. Визуализация продемонстрирована на рисунках 8 и 9. К сожалению, модель также оказалась нерезультативна.

Таблица 7 – Метрики оценки предсказаний нейронной сети

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **MAE** | **MSE** | **R²** |
| Neural Network Train | 0,14 | 0,03 | 0,12 |
| Neural Network Test | 0,16 | 0,04 | -0,04 |

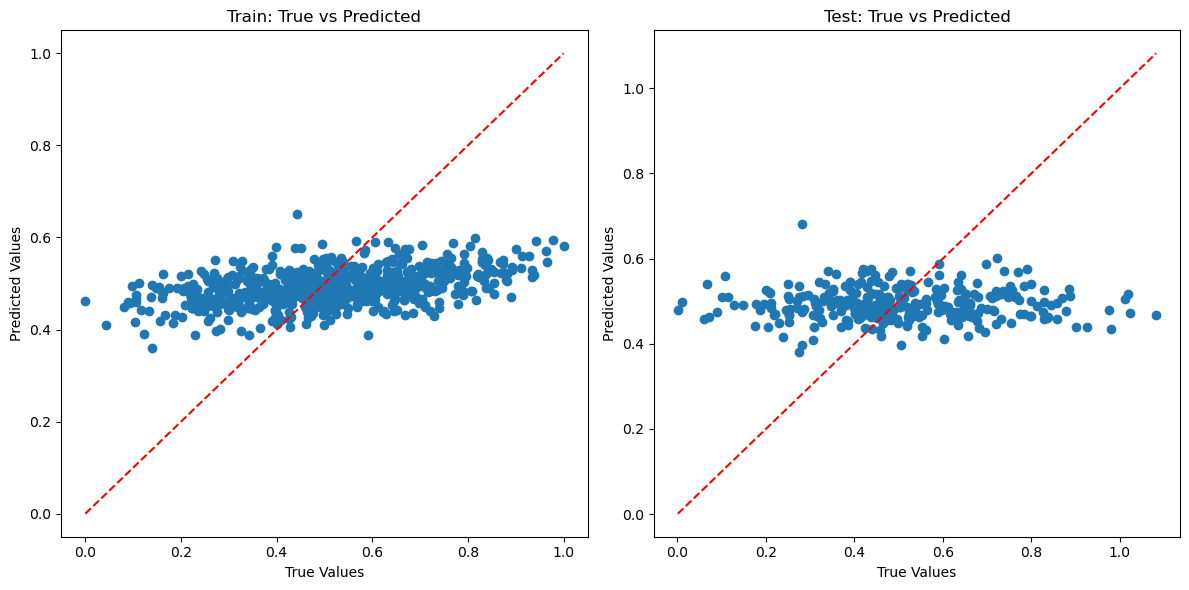


Рисунок 9 - Визуализация предсказаний обновленной нейронной сети

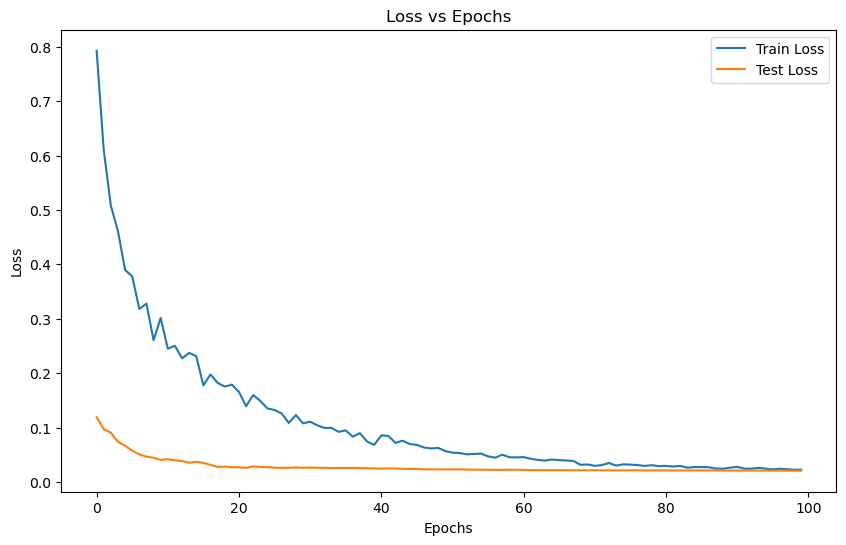


Рисунок 10 - График ошибки обновленной нейронной сети

**2.4.1 Дополнительные варианты архитектуры нейронной сети**

1 вариант:

Входной слой:

128 нейронов.

LeakyReLU (с параметром alpha=0.1),

Batch Normalization

Dropout (0.3)

Первый скрытый слой:

64 нейрона.

LeakyReLU

Batch Normalization

Dropout (0.3)

Второй скрытый слой:

32 нейрона

LeakyReLU

Batch Normalization

Dropout (0.3)

Выходной слой:

1 нейрон

ReLU

Оптимизатор: RMSprop Функция потерь: Huber Loss

Метрика: mean\_absolute\_error (MAE)

1. вариант:

Входной слой:

256 нейронов

ReLU

BatchNormalization

Dropout (0.3): регуляризация для предотвращения переобучения.

Первый скрытый слой:

128 нейронов

ReLU

BatchNormalization

Dropout (0.3)

Второй скрытый слой

64 нейрона

ReLU

Выходной слой:

1 нейрон с активацией linear

1. вариант:

Предварительное преобразование Y в логарифмический масштаб для стабилизации обучения

Входной слой:

128 нейронов

LeakyReLU с alpha=0.1 (предотвращает "затухание градиентов").

BatchNormalization

Dropout (0.3)

Первый скрытый слой:

64 нейрона

LeakyReLU с alpha=0.1.

BatchNormalization.

Dropout (0.3)

Второй скрытый слой:

32 нейрона

LeakyReLU с alpha=0.1.

BatchNormalization.

Dropout (0.3)

Выходной слой:

1 нейрон

ReLU – гарантирует неотрицательные

значения на выходе.

Оптимизатор: RMSprop с learning\_rate=0.0005.

Функция потерь: Huber loss – устойчива к выбросам.

Метрики: mean\_absolute\_error (MAE).

EarlyStopping: прекращает обучение, если валидационная ошибка не уменьшается 20 эпох подряд, восстанавливает лучшие веса.

ReduceLROnPlateau: снижает learning rate вдвое, если ошибка на валидации не уменьшается в течение 10 эпох.

Вывод: как видно из таблицы ни одна из моделей не улучшила метрики

Таблица 8 - Сравнение метрик по четырем реализованным архитектурам нейросети

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Train MAE** | **Test MAE** | **Train MSE** | **Test MSE** | **Train R²** | **Test R²** |
| Neural Network 1 | 0,14 | 0,16 | 0,03 | 0,04 | 0,12 | -0,04 |
| Neural Network 2 | 0,14 | 0,16 | 0,03 | 0,04 | 0,12 | -0,04 |
| Neural Network 3 | 0,15 | 0,16 | 0,03 | 0,04 | 0,10 | -0,03 |
| Neural Network 4 | 0,67 | 0,78 | 0,70 | 0,95 | 0,15 | -0,09 |

## 2.5 Разработка приложения

В удобства пользователя в работе с моделью прогнозирования было разработано веб-приложение, реализованное с использованием Flask и обученной модели Random Forest. Оно позволяет прогнозировать модуль упругости при растяжении композитных материалов на основе введённых характеристик. Пользователь вводит значения параметров, после чего приложение делает предсказание с помощью модели машинного обучения и отображает результат (рисунок 11).

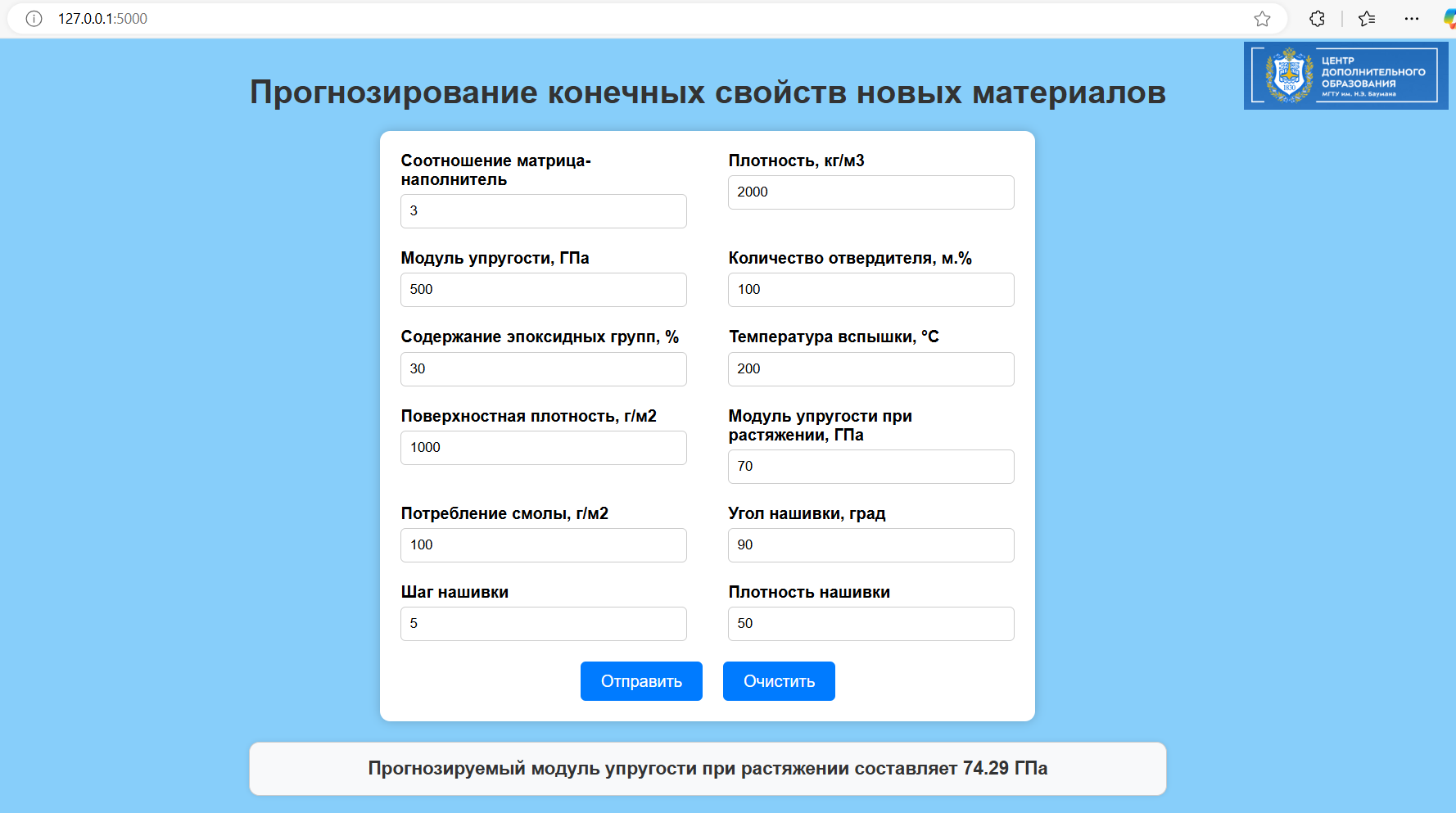


Рисунок 11 – Интерфейс приложения для прогнозирования модуля упругости при растяжении

Основные компоненты и их функции:

1. Веб-интерфейс (HTML + CSS) Реализован в файле index.html.

Пользователь вводит значения параметров композитного материала в удобной форме, которая состоит из 12 полей:

1. Соотношение матрица-наполнитель
2. Плотность, кг/м3
3. Модуль упругости, ГПа
4. Количество отвердителя, м.%
5. Содержание эпоксидных групп, %
6. Температура вспышки, °C
7. Поверхностная плотность, г/м2
8. Модуль упругости при растяжении, ГПа
9. Потребление смолы, г/м2
10. Угол нашивки, град
11. Шаг нашивки
12. Плотность нашивки

Кнопки управления:

* "Отправить": отправляет введённые данные на сервер для предсказания.
* "Очистить": очищает все поля формы с помощью JavaScript-функции.

Итоговое предсказание модели оформлено в виде отдельного блока. Если одно или несколько полей пустые, выводится сообщение об ошибке: "Все поля обязательны к заполнению"

2. Серверная часть (Flask) реализована в файле app.py. Логика обработки данных:

* Список характеристик и их значений считывается из формы, отправленной пользователем.
* Проверяется, что все поля заполнены.
* Данные преобразуются в формат, необходимый для модели машинного обучения.

Полученные данные передаются функции get\_prediction, которая загружает модель и выполняет предсказание. Если предсказание успешно, пользователю возвращается сообщение с результатом: "Прогнозируемый модуль упругости при растяжении, ГПа составляет {значение}". В случае ошибки выводится сообщение о возникшей проблеме.

3. Модель машинного обучения загружается в файле model\_loader.py.

Модель представляет собой обученный Random Forest Regressor, сохранённый в формате random\_forest\_model.pkl.

Функция get\_prediction:

* Принимает список числовых признаков.
* Преобразует их в формат, подходящий для предсказания
* Возвращает результат предсказания с округлением до двух знаков после запятой.

Схема работы приложения:

1. Пользователь открывает браузер по адресу http://127.0.0.1:5000/.
2. Вводит значения всех характеристик материала.
3. Нажимает кнопку "Отправить".
4. Сервер принимает данные и передаёт их модели Random Forest.
5. Модель возвращает предсказание, которое отображается на странице.

Кнопка "Очистить" позволяет обнулить все поля формы без перезагрузки страницы.

Преимущества приложения:

* Удобный пользовательский интерфейс: компактная форма с кнопками и визуальными блоками;
* Отображение ошибок: проверка на заполнение всех обязательных полей;
* Модель машинного обучения: предсказание выполняется на основе модели Random Forest;
* Гибкость и масштабируемость: можно легко добавить новые параметры или заменить модель.

## 2.6 Создание удаленного репозитория

Для данного исследования был создан удаленный репозиторий на GitHub, который находится по адресу: https://github.com/MaryShumk/Composits.

На него были загружены результаты работы: исследовательский notebook, код приложения.

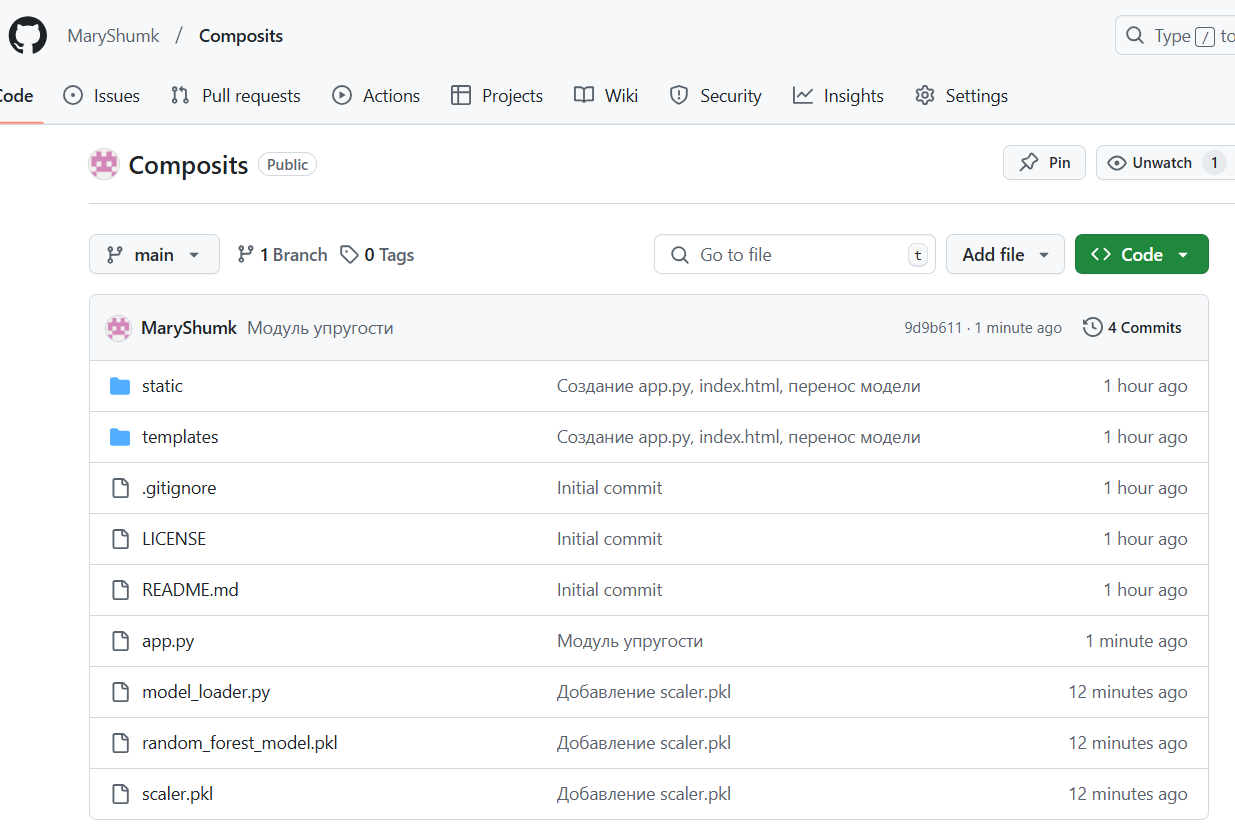


Рисунок 11 – Скриншот, подтверждающий размещение проекта на GitHub

# 

# Заключение

В рамках работы была проведена попытка решения задачи прогнозирования модуля упругости и прочности композиционных материалов при растяжении с использованием методов машинного обучения на основании данных о начальных свойствах компонентов композиционных материалов. Исследование началось с изучения и предобработки данных, включающих изучение типов данных, наличия пропусков, визуализацию распределения данных, выявление выбросов и их устранение, нормализацию и стандартизацию.

Были протестированы различные алгоритмы машинного обучения, включая линейную регрессию, случайный лес, градиентный бустинг и нейронные сети. К сожалению, модели не продемонстрировали высоких результатов на тестовой выборке. На основе сравнения метрик качества, таких как MAE, MSE и R², были выбраны наилучшие модели из представленных вариантов:

* Lasso – для прогнозирования модуля упругости при растяжении
* Random Forest - для прогнозирования прочности при растяжении.

Кроме того, была разработана и смоделирована архитектура нейронной сети для рекомендации соотношения матрица-наполнитель.

На основе полученных результатов было разработано и внедрено веб-приложение с использованием фреймворка Flask. Оно позволяет пользователю вводить параметры композитного материала и получать прогнозируемое значение прочности. Проект также размещен на GitHub.

В связи с тем, что в ходе работы не удалось разработать точный способ прогнозирования целевых переменных, исследование необходимо продолжить. Среди возможных направлений развития можно выделить:

* углубленное погружение в предметную область
* получение новых данных,
* корректировка исходных данных.

**Список использованной литературы**

1 Композиционные материалы : учебное пособие для вузов / Д. А. Иванов, А. И. Ситников, С. Д. Шляпин ; под редакцией А. А. Ильина. — Москва : Издательство Юрайт, 2019 — 253 с.

1. Вандер П. Д. Вандер Плас. Python для сложных задач: наука о данных и машинное обучение. – 2019.
2. Силен Дэви, Мейсман Арно, Али Мохамед. Основы Data Science и Big Data. Python и наука о данных. – СПб.: Питер, 2017. – 336 с.: ил.
3. Дауни А. Б. Основы Python: Научитесь думать как программист. – " Манн, Иванов и Фербер", 2021.
4. Любанович Б. Простой Python. Современный стиль программирования. – " Издательский дом"" Питер""", 2016.
5. Крис Э. Машинное обучение с использованием Python. Сборник рецептов: Пер. с англ. – БХВ-Петербург, 2020.
6. Рашка С. Python и машинное обучение. – Litres, 2022.
7. Черкасов Д. Ю., Иванов В. В. Машинное обучение //Наука, техника и образование. – 2018. – №. 5 (46).
8. Коротеев М. В. Обзор некоторых современных тенденций в технологии машинного обучения //E-Management. – 2018. – Т. 1. – №. 1. – С. 26-35.
9. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. – Litres, 2022.
10. Вьюгин В. Математические основы машинного обучения и прогнозирования. – Litres, 2022.
11. Шакла Н. Машинное обучение и TensorFlow. – Питер, 2023.
12. Документация по языку программирования python: – Режим доступа: <https://docs.python.org/3.8/index.html>.
13. Документация по библиотеке numpy: – Режим доступа: <https://numpy.org/doc/1.22/user/index.html#user>.
14. Документация по библиотеке pandas: – Режим доступа: <https://pandas.pydata.org/docs/user_guide/index.html#user-guide>.
15. Документация по библиотеке matplotlib: – Режим доступа: <https://matplotlib.org/stable/users/index.html>.
16. Документация по библиотеке seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>.
17. Документация по библиотеке sklearn: – Режим доступа: <https://scikitlearn.org/stable/user_guide.html>.
18. Документация по библиотеке keras: – Режим доступа: https://keras.io/ api/.
19. Руководство по быстрому старту в flask: – Режим доступа: https://flaskrussian-docs.readthedocs.io/ru/latest/quickstart.html.