

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по курсу

«Data Science Pro»

**Тема: «Прогнозирование конечных свойств новых материалов
(композиционных материалов)»**

Слушатель

Филимонов Фёдор Игоревич

Москва, 2024

СОДЕРЖАНИЕ

	Введение.....	1
1.	Аналитическая часть	3
1.1.	Постановка задачи	3
1.2.	Описание используемых методов.....	4
1.3.	Разведочный анализ данных.	12
2.	Практическая часть	16
2.1.	Предобработка данных.	16
2.2.	Разработка и обучение модели.	19
2.3.	Тестирование модели	20
2.4.	Написать нейронную сеть, которая будет рекомендовать соотношение «матрица – наполнитель».....	21
2.5.	Разработка приложения, которая будет рекомендовать соотношение «матрица – наполнитель».....	24
	Заключение.....	27
	Список используемой литературы и веб ресурсы.....	28

Введение

Тема данной работы - прогнозирование конечных свойств новых материалов (композиционных материалов).

Композиционными называются материалы, в которых имеет место сочетание двух (или более) химически разнородных компонентов (фаз) с четкой границей раздела между ними. Это неоднородные по химическому составу и структуре материалы.

Структура композиционных материалов представляет собой матрицу (основной компонент), содержащую в своем объеме или армирующие элементы, часто называемые наполнителем. Матрица и наполнитель разделены границей (поверхностью) раздела. Наполнитель равномерно распределен в матрице и имеет заданную пространственную ориентацию.

Композиционные материалы характеризуются совокупностью свойств, не присущих каждому в отдельности взятому компоненту. За счет выбора армирующих элементов, варьирования их объемной доли в матричном материале, а также размеров, формы, ориентации и прочности связи по границе «матрица-наполнитель», свойства композиционных материалов можно регулировать в значительных пределах.

Возможно получить композиты с уникальными эксплуатационными свойствами. Этим обусловлено широкое применение композиционных материалов в различных областях техники. Композиционные материалы используются:

- в авиационной, ракетной и космической технике;
- в металлургии;
- в горнорудной промышленности;
- в химической промышленности;
- в автомобильной промышленности;
- в сельскохозяйственном машиностроении;
- в электротехнической промышленности;
- в ядерной технике;

- в машиностроительной отрасли;
- в сварочной технике;
- в судостроительной промышленности;
- в медицинской промышленности;
- в строительстве;
- в бытовой технике.

Учитывая такое широкое распространение и высокую потребность в новых материалах, тема данной работы является очень актуальной. Поскольку стоимость производства композитного материала высока. Зная характеристики компонентов, невозможно рассчитать свойства композита. Значит для получения заданных свойств требуется большое количество испытаний различных комбинаций. Сократить время и затраты на создание определенного материала могла бы помочь система поддержки производственных решений, построенная на принципах машинного обучения.

Аналитическая часть

1.1. Постановка задачи.

Для исследовательской работы были даны 2 файла: X_br.xlsx содержит (1023 строки, 11 столбца) и X_nup.xlsx (1040 строки, 4 столбца).

```
Загрузка данных из исходных excel таблиц (X_br и X_nup)
```

```
DS1=pd.read_excel('X_br.xlsx')
DS2=pd.read_excel('X_nup.xlsx')
```

[2] ✓ 0.8s Python

Рисунок 1 – загрузка excel файлов X_br.xlsx, X_nup.xlsx

Цель работы разработать модели для прогноза модуля упругости при растяжении и прочности при растяжении, а также, соотношения «матрица-наполнитель». Для этого нужно объединить два этих файла. Часть информации из X_nup.xlsx (17 строк) не имеют соответствующих строк в файле X_br.xlsx, поэтому будут удалены в момент объединения удалены.

Во время ознакомления с данными в обеих таблицах предположил, что информация по столбцу "Unnamed: 0" не будет для нас информативной и удаляю её из обеих таблиц.

Из представленной выше информации делаю вывод, что информация по столбцу "Unnamed: 0" не будет для нас информативной и предлагаю удалить её из обеих таблиц

```
DS1.drop(['Unnamed: 0'],axis=1, inplace=True)
```

```
DS1.tail(3) # вывод 3 последних строк
```

[9] ✓ 0.0s Python

	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
1020	3.280604	1972.372865	416.836524	110.533477	23.957502	248.423047	740.142791	74.734344	2662.906040	236.606764
1021	3.705351	2066.799773	741.475517	141.397963	19.246945	275.779840	641.468152	74.042708	2071.715856	197.126067
1022	3.808020	1890.413468	417.316232	129.183416	27.474763	300.952708	758.747882	74.309704	2856.328932	194.754342

```
DS2.drop(['Unnamed: 0'],axis=1, inplace=True)
```

```
DS2.tail(3) # вывод 3 последних строк
```

[10] ✓ 0.0s Python

	Угол нашивки, град	Шаг нашивки	Плотность нашивки
1037	90	9.800926	72.858286
1038	90	10.079859	65.519479
1039	90	9.021043	66.920143

Рисунок 2 – удаление не информативной информации из обеих таблиц. Столбец "Unnamed: 0".

Затем необходимо провести разведочный анализ данных, или EDA, нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек. Для каждой колонки получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков; пред обработать данные: удалить шумы и выбросы, сделать нормализацию и стандартизацию. Обучить несколько моделей для прогноза модуля упругости при растяжении и прочности при растяжении. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель. Разработать приложение с графическим интерфейсом, которое будет выдавать прогноз соотношения «матрица-наполнитель». Оценить точность модели на тренировочном и тестовом датасете. Оформить файл README.

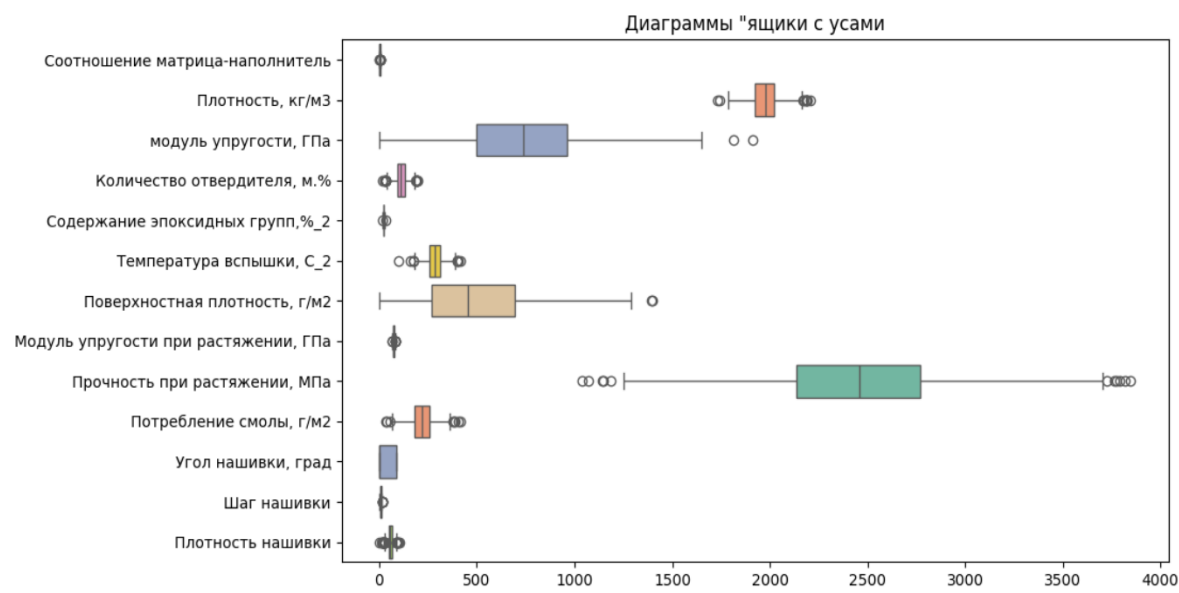


Рисунок 3 – Пример диаграммы «ящик с усами» объединённых данных.

1.2. Описание используемых методов.

Данная задача в рамках классификации категорий машинного обучения относится к машинному обучению с учителем. Обучение с учителем (Supervised Learning) — это один из основных методов машинного обучения, при котором модель обучается на размеченных данных. В процессе обучения модель получает

входные данные и соответствующие им правильные ответы (метки), что позволяет ей научиться делать предсказания на новых, ранее не виденных данных. Для решения в процессе исследования были применены следующие методы:

- а. 'Линейная регрессия': `LinearRegression()`;
- б. 'Хребет': `Ridge()`;
- в. 'Опорные вектора': `SVR()`;
- г. 'Случайный лес': `RandomForestRegressor()`;
- д. 'Градиентный бустинг ': `GradientBoostingRegressor()`.

Рассмотри положительные и отрицательные стороны каждого метода.

- а. 'Линейная регрессия': `LinearRegression()`.

Положительные стороны линейной регрессии:

- Простота. Алгоритм легко понять и реализовать;
- Интерпретируемость. Уравнение модели предоставляет чёткие коэффициенты, которые объясняют влияние каждой независимой переменной на зависимую переменную;
- Эффективность с точки зрения вычислений. Метод может обрабатывать большие наборы данных;
- Устойчивость к выбросам. Выбросы оказывают меньшее влияние на общую производительность модели.

Отрицательные стороны линейной регрессии:

- Допущение линейной зависимости. Если зависимость не линейная, модель может работать некорректно;
- Необходимость, чтобы объекты уже имели форму, подходящую для модели. Может потребоваться преобразование объектов в формат, который может эффективно использоваться моделью;
- Подверженность переобучению и недостаточному переобучению. Переобучение происходит, когда модель слишком хорошо запоминает обучающие данные и не может обобщить их на невидимые данные. Недостаточная подгонка возникает, когда модель слишком проста, чтобы уловить лежащие в основе взаимосвязи в данных;
- Ограниченная объяснительная способность для сложных взаимосвязей между переменными. Для более глубокого понимания могут потребоваться более продвинутые методы машинного обучения.

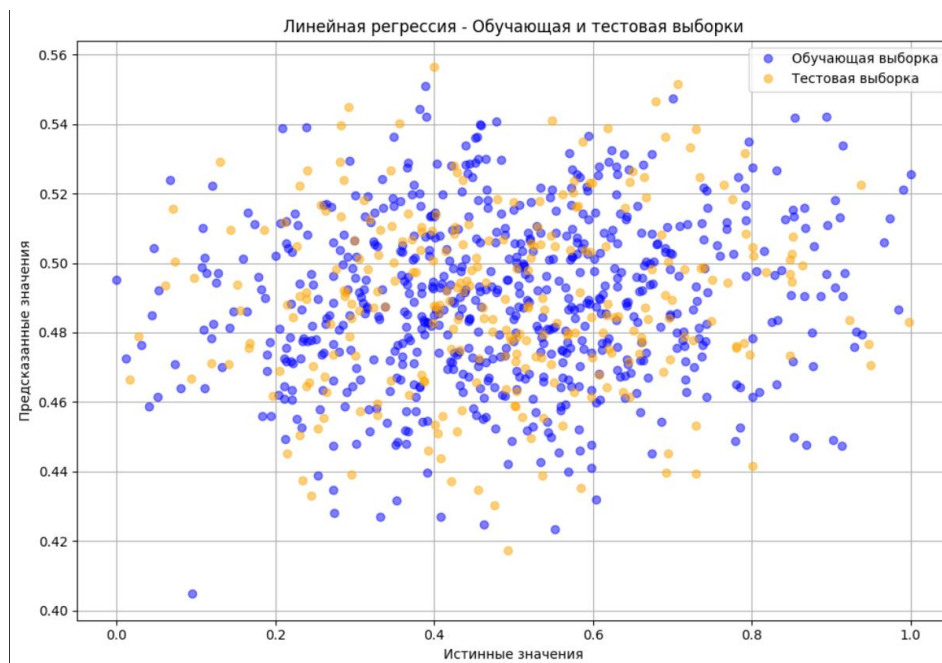


Рисунок 4 - график метода Линейной регрессии для 'Модуль упругости при растяжении, ГПа'.

б. 'Хребет': Ridge().

Положительные стороны метода Хребет:

- Защита от переобучения. Метод предотвращает переоснащение модели за счёт добавленных штрафов за сложность;
- Обработка мультиколлинеарности. Метод балансирует влияние коррелированных переменных, более справедливо распределяя влияние связанных функций;
- Стабильные результаты. По сравнению с обычными линейными классификаторами, Хребет выдаёт более надёжные и последовательные результаты.

Отрицательные стороны метода Хребет:

- Компромисс между смещением и дисперсией. Может быть трудно найти идеальный баланс между отклонениями и предвзятостью;
- Вычислительная сложность. Вычислительные затраты могут быть значительными, особенно при работе с многомерными наборами данных;
- Не подходит для разреженных данных. Регуляризация гребня менее подходит для наборов данных с большим количеством нерелевантных или разреженных объектов, поскольку предполагает, что каждый объект вносит свой вклад.

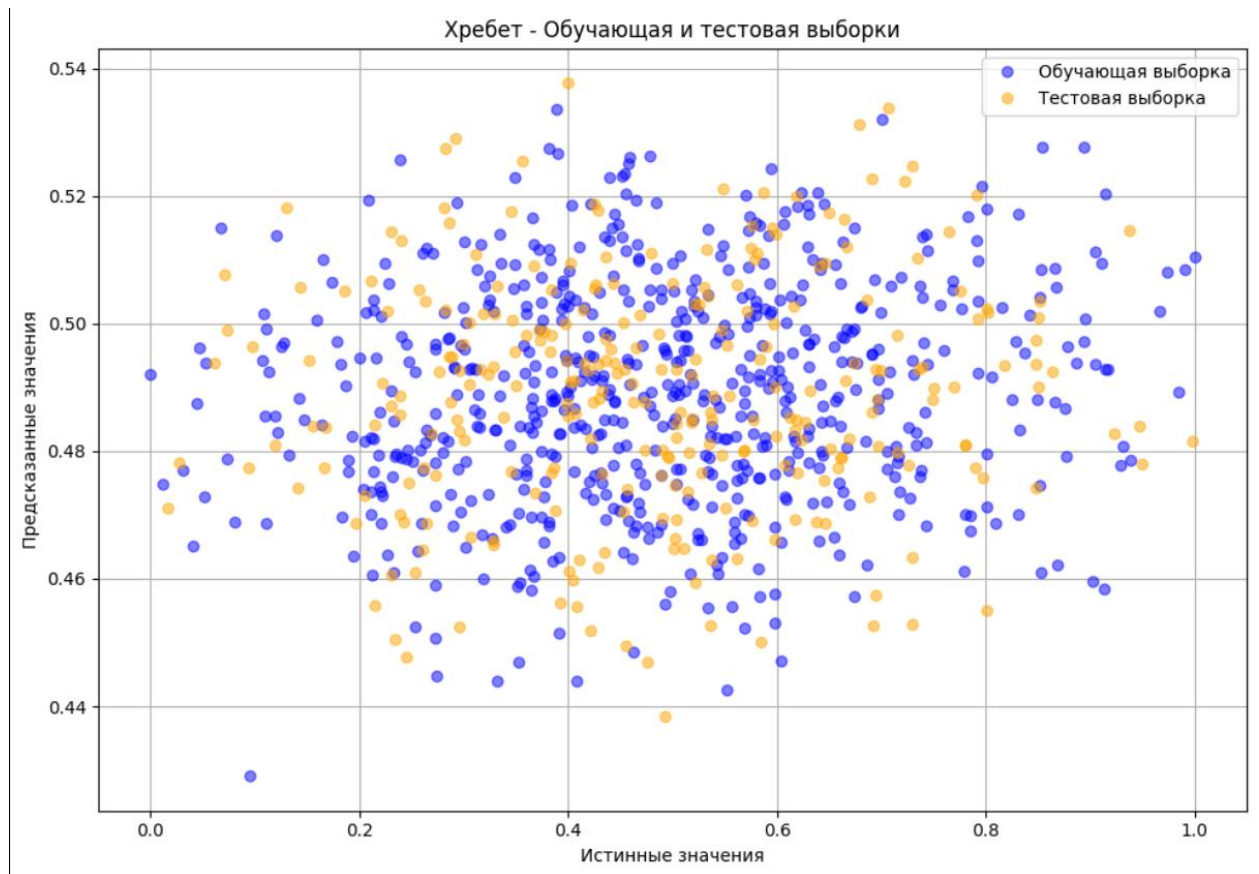


Рисунок 5 - график метода Хребта для 'Модуль упругости при растяжении, ГПа'.

в. 'Опорные вектора': SVR().

Положительные стороны метода опорных векторов (SVR):

- Высокая точность прогнозов;
- Возможность обработки многомерных данных без их предварительного преобразования или понижения размерности;
- Адаптация к обнаружению аномалий в данных;
- Эффективность в пространствах высокой размерности и в случаях, когда число измерений больше числа выборок.

Отрицательные стороны метода опорных векторов (SVR):

- Низкая скорость работы на данных большого размера;
- Чувствительность к шуму и нестабильная работа с сильно перекрывающимися классами в данных;
- Если число признаков намного больше числа выборок, то при выборе функционального ядра и термина регуляризации следует избегать переобучения;
- Не предоставляет оценки вероятностей напрямую, они вычисляются с помощью дорогостоящей пятикратной кросс-валидации.

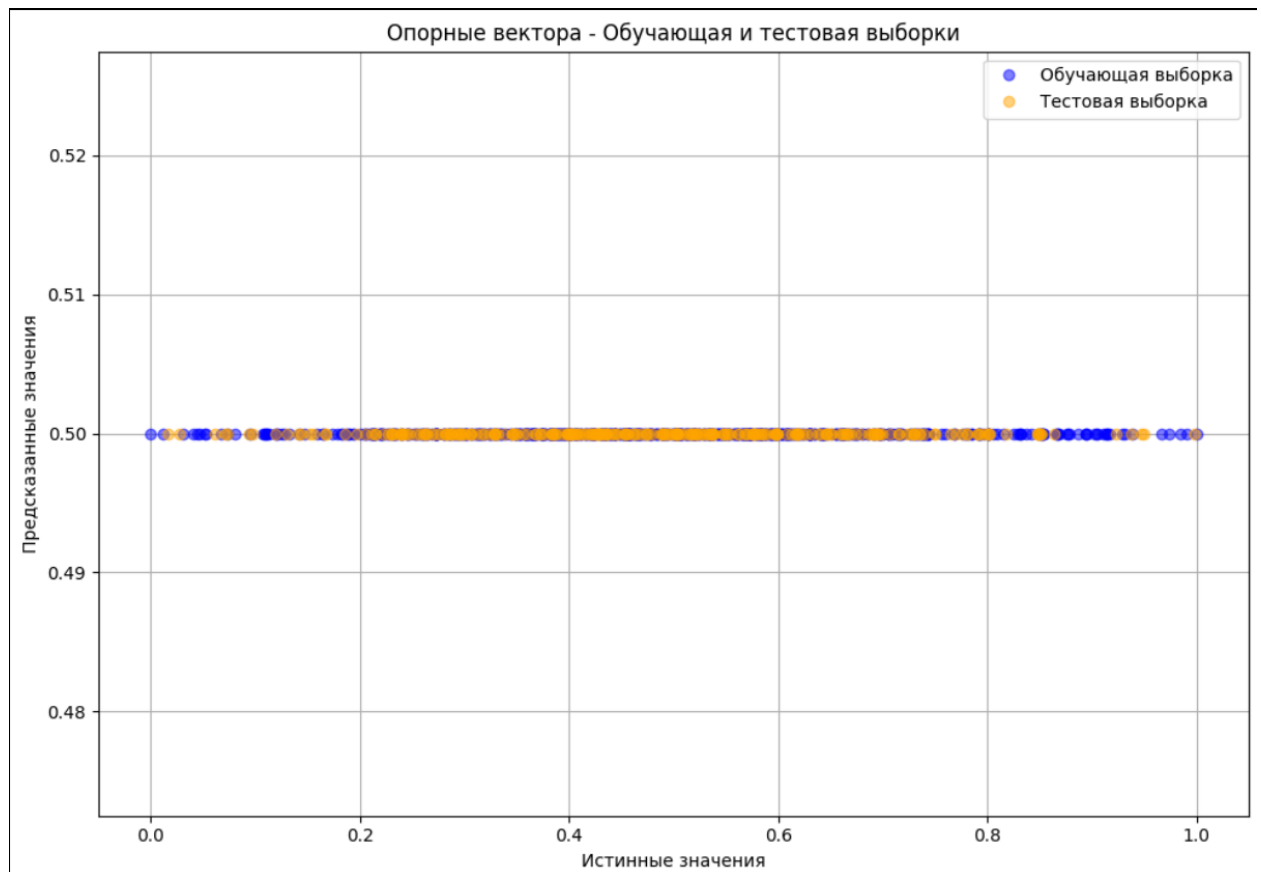


Рисунок 6 - график метода Опорные вектора для 'Модуль упругости при растяжении, ГПа'.

г. 'Случайный_лес': RandomForestRegressor.

Положительные стороны метода Случайный лес:

- Высокая точность предсказания;
- Не требует тщательной настройки параметров, хорошо работает «из коробки»;
- Практически не чувствителен к выбросам в данных;
- Не чувствителен к масштабированию и к другим монотонным преобразованиям значений признаков;
- Редко переобучается;
- Способен эффективно обрабатывать данные с большим числом признаков и классов;
- Хорошо работает с пропущенными данными — сохраняет хорошую точность даже при их наличии;
- Одинаково хорошо обрабатывает как непрерывные, так и дискретные признаки.

Отрицательные стороны метода Случайный лес:

- Для реализации алгоритма требуется значительный объём вычислительных ресурсов;
- Большой размер моделей;
- Построение случайного леса отнимает больше времени, чем деревья решений или линейные алгоритмы;
- Алгоритм склонен к переобучению на зашумлённых данных;
- В отличие от более простых алгоритмов, результаты случайного леса сложнее интерпретировать;
- Когда в выборке очень много разреженных признаков, таких как тексты или наборы слов, алгоритм работает хуже, чем линейные методы.

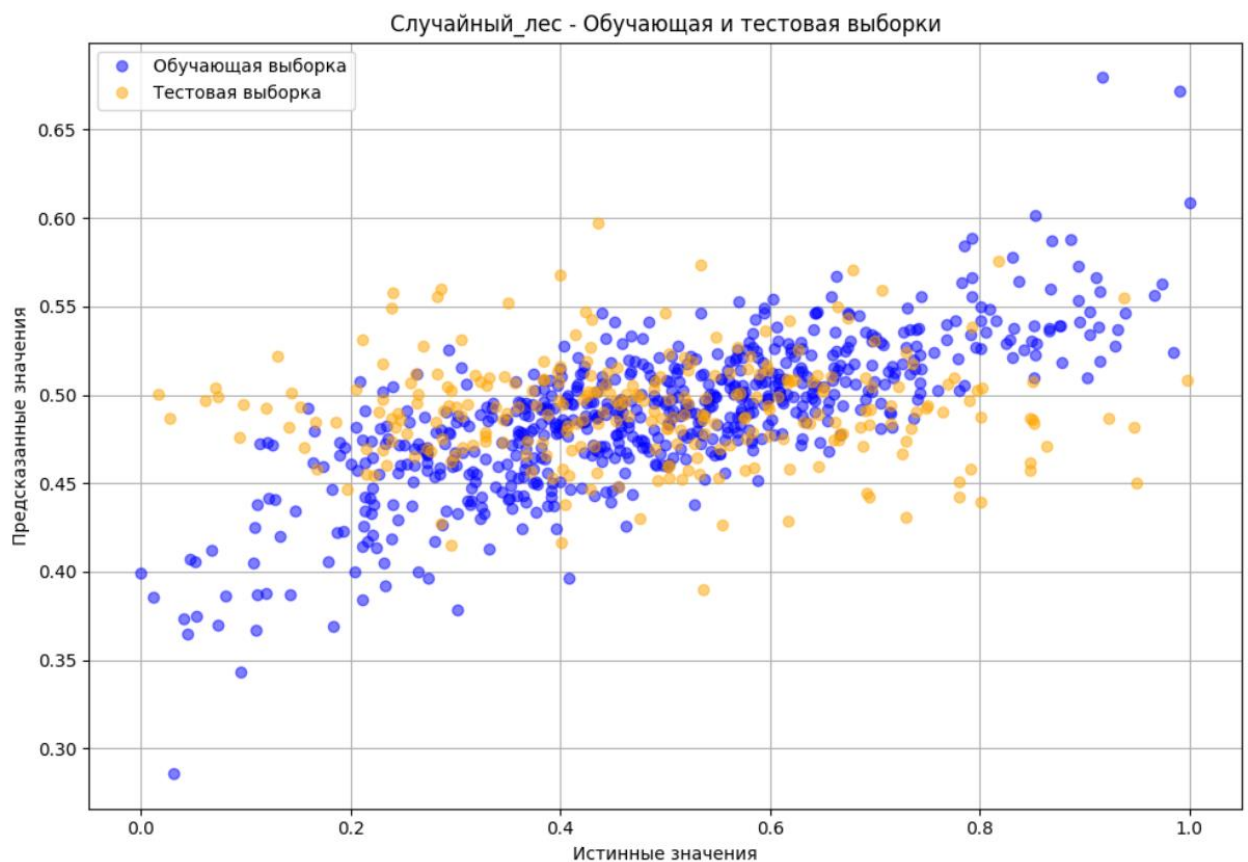


Рисунок 7 - график метода Смешанный лес для 'Модуль упругости при растяжении, ГПа'.

д. 'Градиентный бустинг': GradientBoostingRegressor().

Положительные стороны метода Градиентный бустинг:

- Работает с любыми функциями потерь;
- Предсказания в среднем лучше, чем у других алгоритмов;
- Самостоятельно справляется с пропущенными данными.

Отрицательные стороны:

- Алгоритм крайне чувствителен к выбросам и при их наличии будет тратить огромное количество ресурсов на эти моменты;
- Модель будет склонна к переобучению при слишком большом количестве деревьев;
- Вычисления могут занять много времени;
- Также при увеличении количества деревьев в методе Градиентный бустинг могут появляться отрицательные предсказания, даже если в обучающем или тестовом наборе данных нет отрицательных значений;
- Выбор между положительными и отрицательными сторонами метода зависит от конкретной задачи и условий её решения.

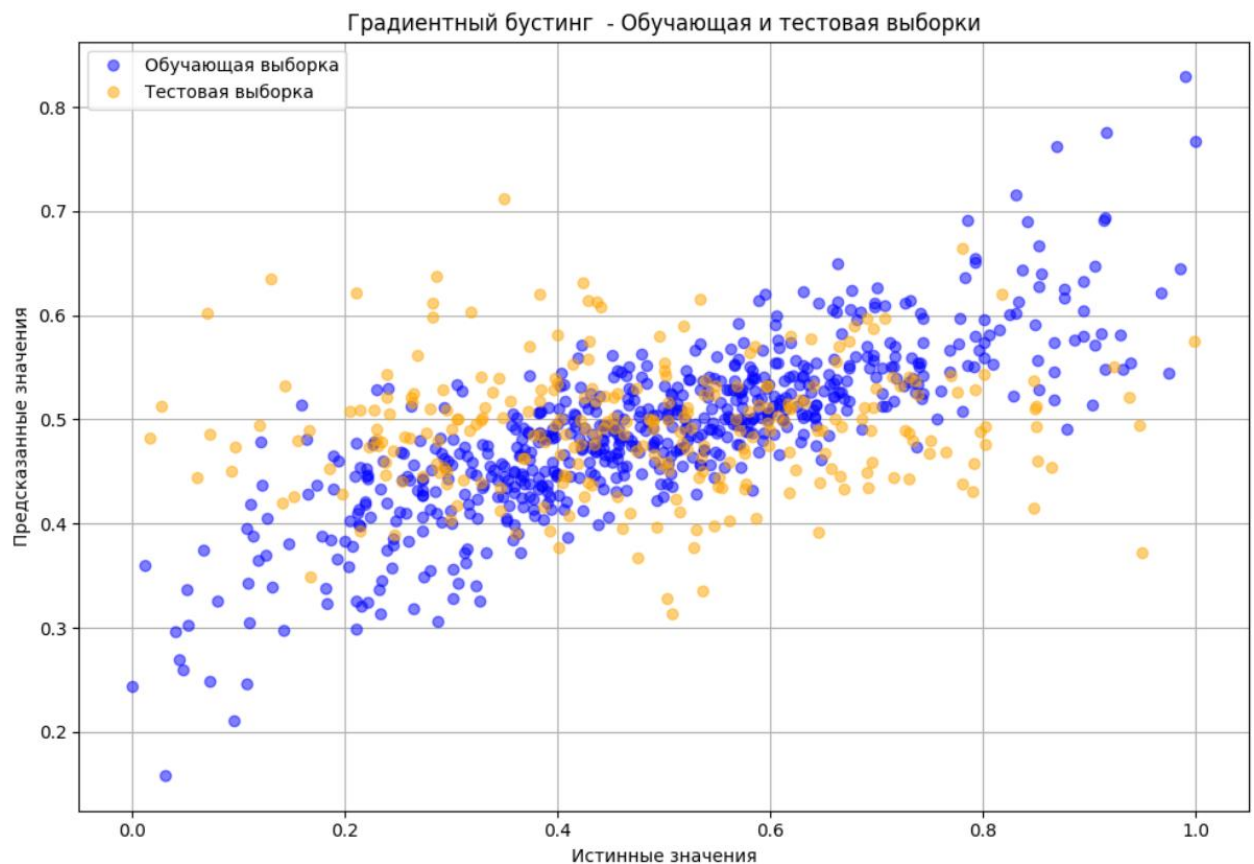


Рисунок 8 - график метода Градиентный бустинг для 'Модуль упругости при растяжении, ГПа'.

Для оценки модели было использовано пять методов, представленных на рисунке 9.

```

# Сохранение результатов
results[name] = {
    'R^2': round(r2, 2),
    'MAE': round(mae, 2),
    'MSE': round(mse, 2),
    'MAPE': round(mape, 2),
    'Test Score': round(test_score, 2)
}

```

Рисунок 9 – Часть кода, которая использовалась для оценки 'Модуль упругости при растяжении, ГПа'.

Описание методов оценки:

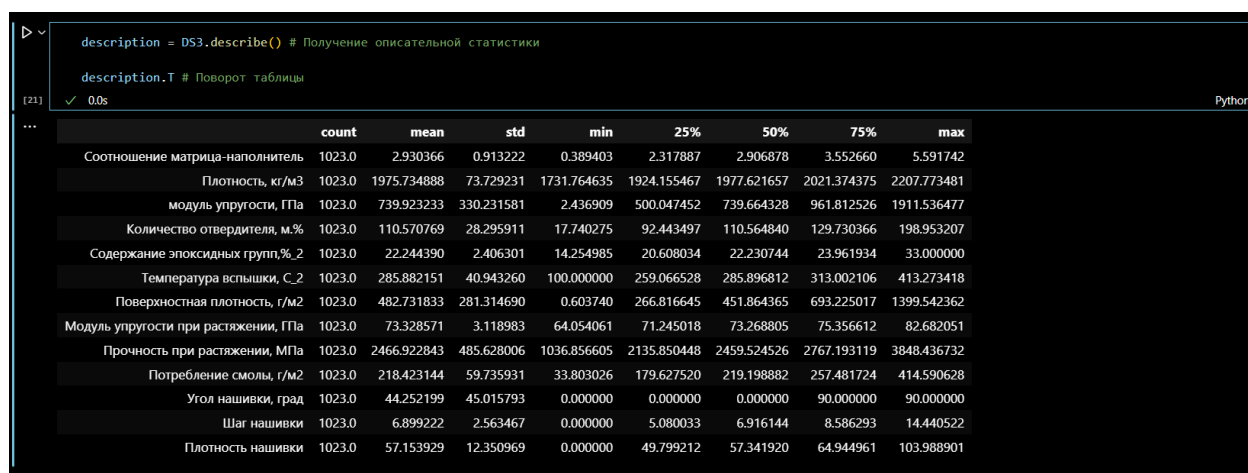
- а. Метод оценки R^2 (коэффициента детерминации) заключается в измерении доли вариации зависимой переменной, которую можно объяснить независимыми переменными в регрессионной модели. Проще говоря, он показывает, насколько хорошо модель соответствует данным. Значение R^2 должно находиться в диапазоне от нуля до единицы. Модель считается более качественной, если значение коэффициента детерминации близко к 1;
- б. Средняя абсолютная ошибка (MAE) — это метрика, которая измеряет среднее абсолютное отклонение каждого прогноза от соответствующего фактического значения. Метод оценки MAE заключается в расчёте среднего абсолютных разностей между целевыми значением и значением, предсказанным моделью на данном обучающем примере в процессе обучения. Если MAE равна 0, это указывает на идеальное соответствие прогнозов реальным значениям. Чем выше значение MAE, тем больше ошибок в прогнозах модели;
- в. Метод оценки MSE (среднеквадратической ошибки) заключается в следующем. Посчитав разницу между прогнозом и реальным значением для каждого объекта, возвести каждую разницу в квадрат. Сложить результаты и разделить на число объектов. Чем меньше значение MSE, тем лучше. Это свидетельствует о том, что модель лучше приближает фактические данные. Высокое значение MSE указывает на большие различия между фактическими и предсказанными значениями, что может свидетельствовать о неэффективности модели.

г. Результат теста (Оценка теста). Этот показатель может определяться в зависимости от контекста, но обычно он представляет собой оценку производительности модели на тестовом наборе данных.

Каждый из этих методов обеспечивает необходимую информацию о производительности модели и может использоваться в зависимости от конкретных требований.

1.3. Разведочный анализ данных.

Прежде чем приступить к обучению моделей, необходимо обработать и очистить их. Очевидно, что «грязные» и необработанные данные могут содержать искажения и пропущенные значения – это ненадёжно, поскольку способно привести к крайне неверным результатам по итогам моделирования. Для этого сначала набор данных надо изучить.



```
description = D53.describe() # Получение описательной статистики
description.T # Поворот таблицы
```

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660	5.591742
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375	2207.773481
модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	500.047452	739.664328	961.812526	1911.536477
Количество отвердителя, м.%	1023.0	110.570769	28.295911	17.740275	92.443497	110.564840	129.730366	198.953207
Содержание эпоксидных групп, %_2	1023.0	22.244390	2.406301	14.254985	20.608034	22.230744	23.961934	33.000000
Температура вспышки, C_2	1023.0	285.882151	40.943260	100.000000	259.066528	285.896812	313.002106	413.273418
Поверхностная плотность, г/м2	1023.0	482.731833	281.314690	0.603740	266.816645	451.864365	693.225017	1399.542362
Модуль упругости при растяжении, ГПа	1023.0	73.328571	3.118983	64.054061	71.245018	73.268805	75.356612	82.682051
Прочность при растяжении, МПа	1023.0	2466.922843	485.628006	1036.856605	2135.850448	2459.524526	2767.193119	3848.436732
Потребление смолы, г/м2	1023.0	218.423144	59.735931	33.803026	179.627520	219.198882	257.481724	414.590628
Угол нашивки, град	1023.0	44.252199	45.015793	0.000000	0.000000	0.000000	90.000000	90.000000
Шаг нашивки	1023.0	6.899222	2.563467	0.000000	5.080033	6.916144	8.586293	14.440522
Плотность нашивки	1023.0	57.153929	12.350969	0.000000	49.799212	57.341920	64.944961	103.988901

Рисунок 10 – Описательная статистика данных.

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

В качестве инструментов разведочного анализа используется: оценка статистических характеристик данных, гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек, тепловая

карта, анализ и полное исключение выбросов, проверка наличия пропусков и дубликатов.

Данные объединённые данные не имеют чётко выраженной зависимости.

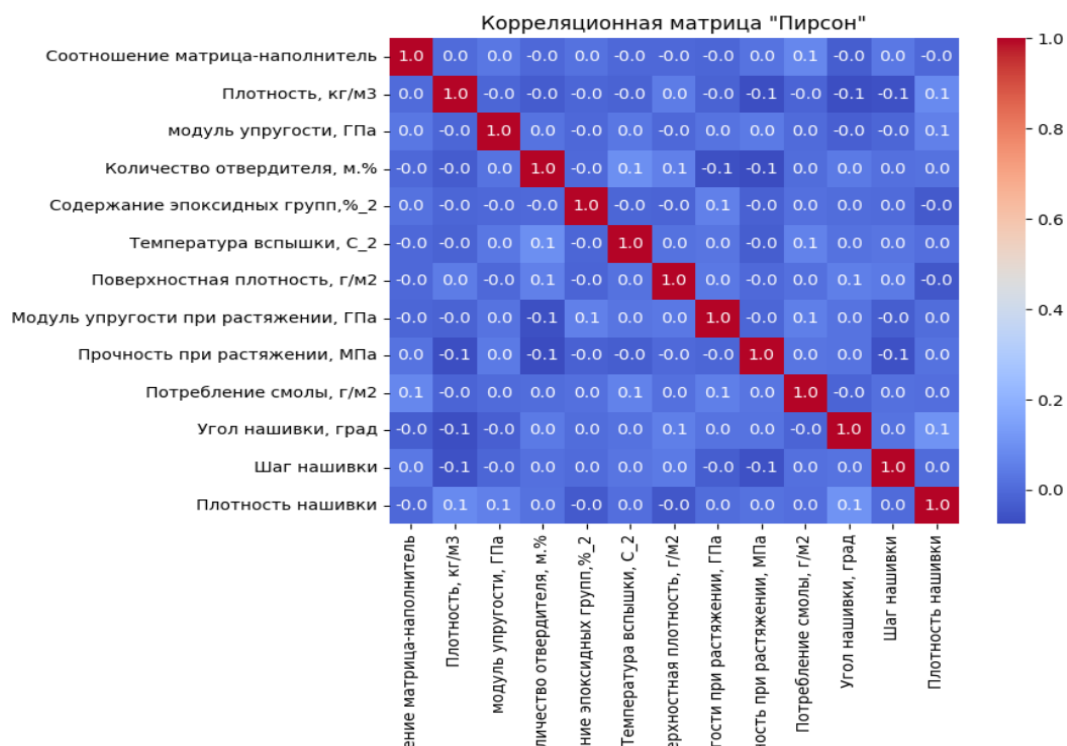


Рисунок 11 – Пример графика корреляции (Тепловая карта) по Пирсану.

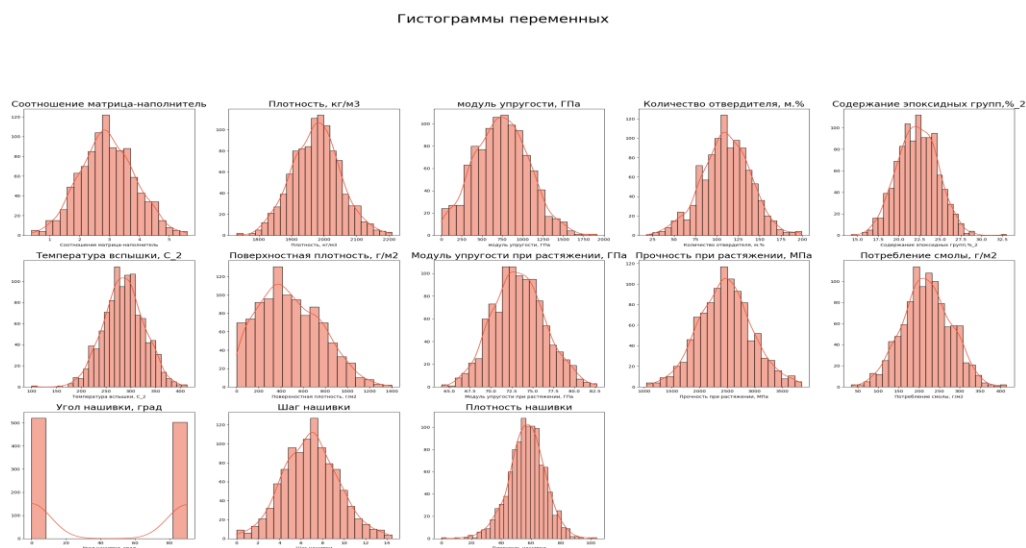


Рисунок 12 – гистограммы распределения.

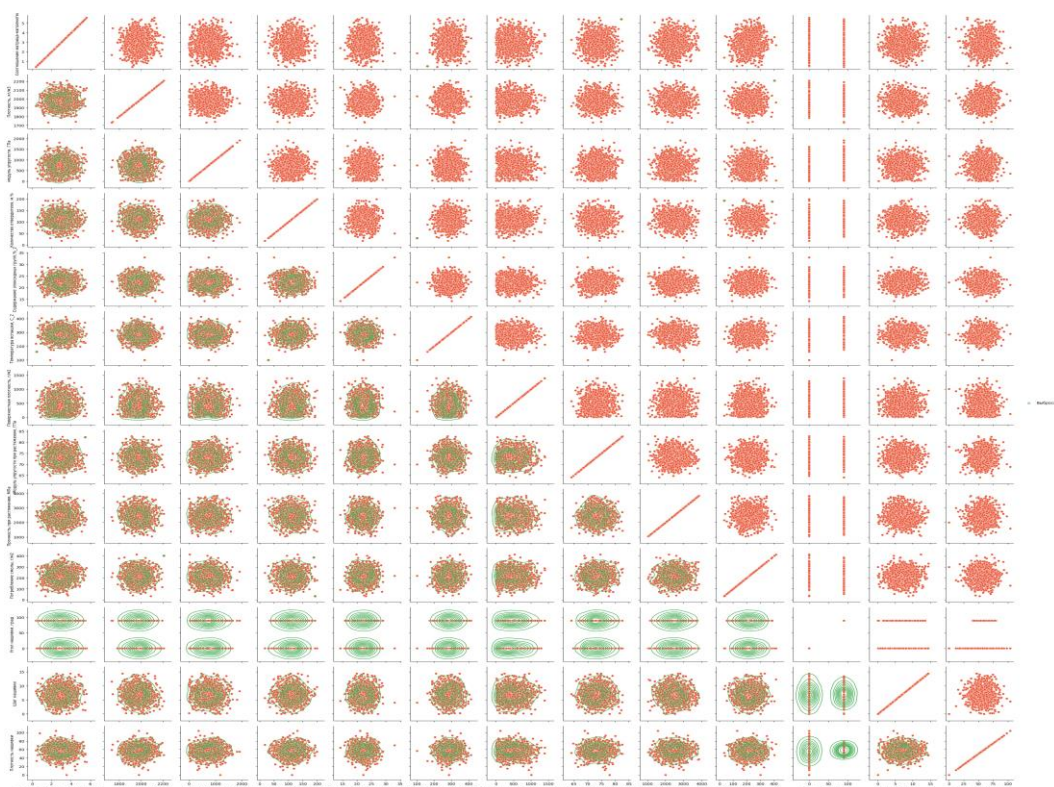


Рисунок 13 - попарные графики рассеяния точек.

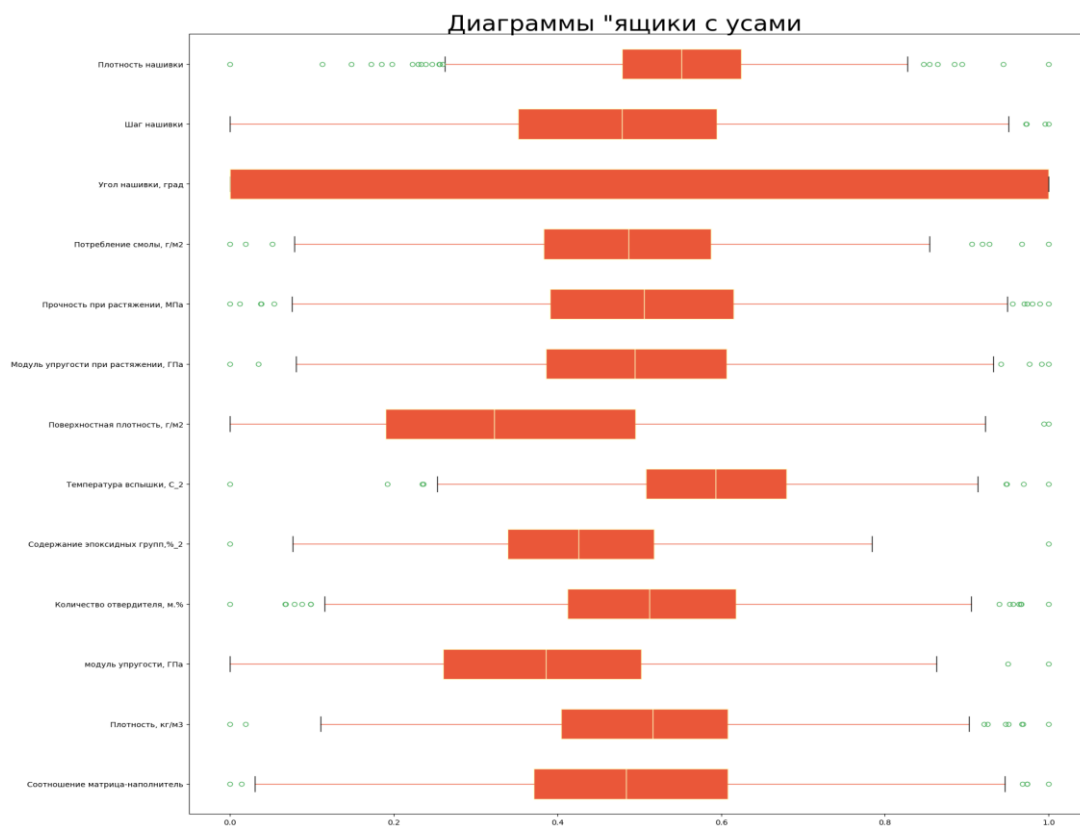


Рисунок 14 – Диаграмма «ящик с усами» для визуализации выбросов.

После обнаружения выбросов данные, значительно отличающиеся от выборки, будут полностью удалены. Для расчёта этих данных мы будем использовать несколько методов.

	Column	IQR	Standard Deviation	Z-Score
0	Соотношение матрица-наполнитель	6	0	0
1	Плотность, кг/м3	9	3	3
2	модуль упругости, ГПа	2	2	2
3	Количество отвердителя, м.%	14	2	2
4	Содержание эпоксидных групп,%_2	2	2	2
5	Температура вспышки, С_2	8	3	3
6	Поверхностная плотность, г/м2	2	2	2
7	Модуль упругости при растяжении, ГПа	6	0	1
8	Прочность при растяжении, МПа	11	0	0
9	Потребление смолы, г/м2	8	3	3
10	Угол нашивки, град	0	0	0
11	Шаг нашивки	4	0	0
12	Плотность нашивки	21	7	7

Рисунок 15 – Пример определённых выбросов разными методами.

Практическая часть

2.1. Предобработка данных.

По условиям задания нормализуем значения. Для этого попробуем несколько методов: MinMaxScaler, Normalizer, MaxAbsScaler, RobustScaler.

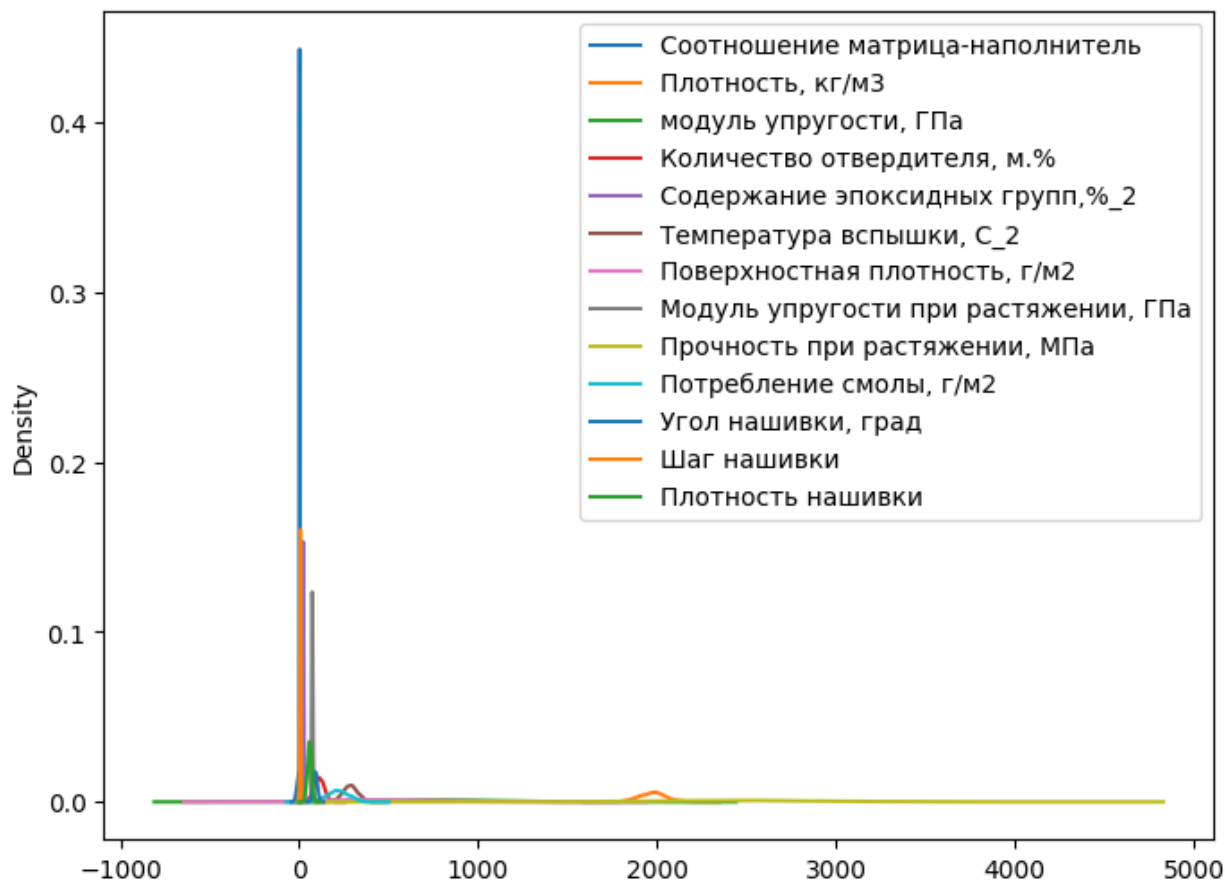


Рисунок 16 – Изначальные данные находятся в разных диапазонах.

Первый вариант: MinMaxScaler- этот метод является простым и эффективным инструментом для нормализации данных, который может значительно улучшить результаты машинного обучения моделей.

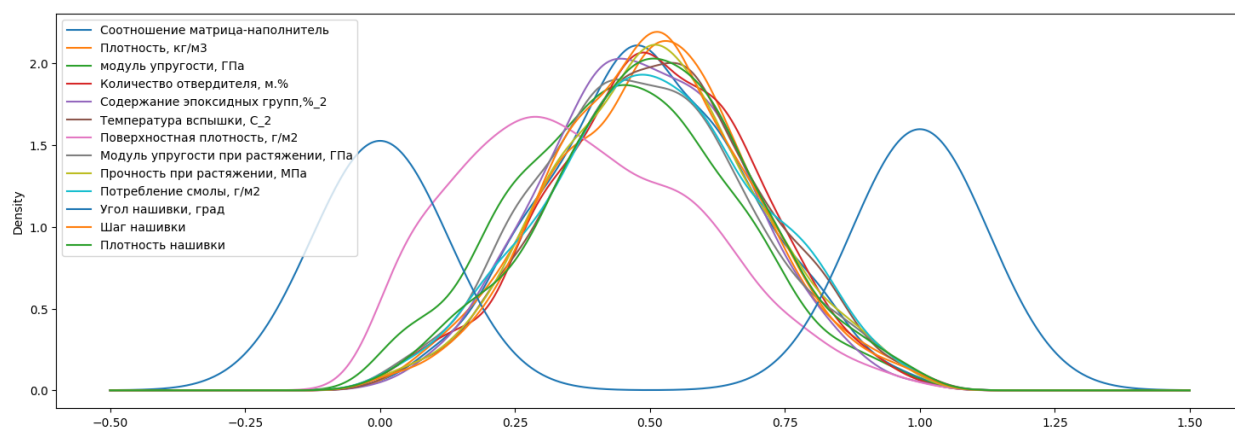


Рисунок 17 – Первый вариант: MinMaxScaler.

Второй вариант: RobustScaler- Этот метод использует медиану и межквартильный размах для нормализации, что делает его устойчивым к выбросам. Это может быть полезно, если наши данные содержат аномалии.

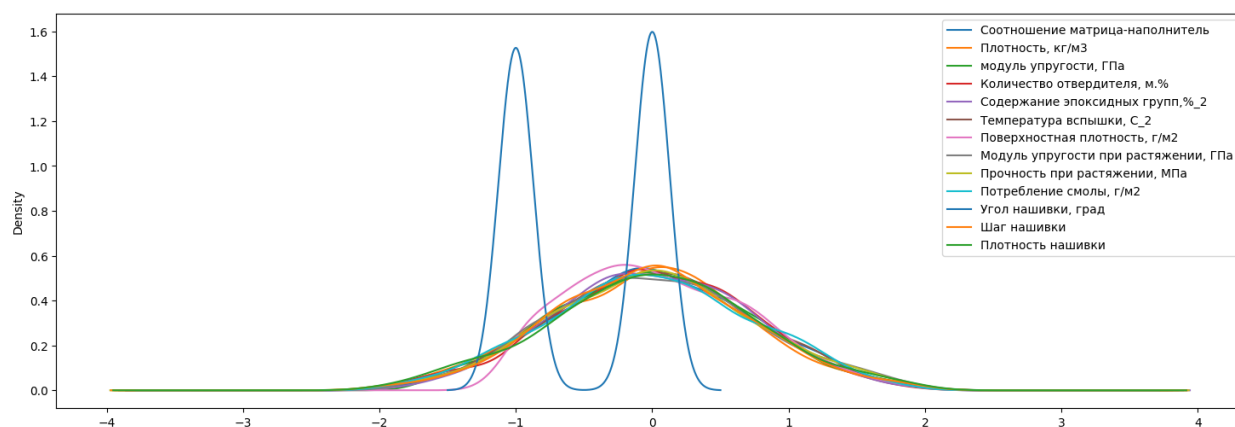


Рисунок 18 – Второй вариант: RobustScaler.

Третий вариант: MaxAbsScaler - Этот метод масштабирует данные, обеспечивая их максимальное значение, сохраняя разреженность данных. Это может быть полезно для разреженных матриц.

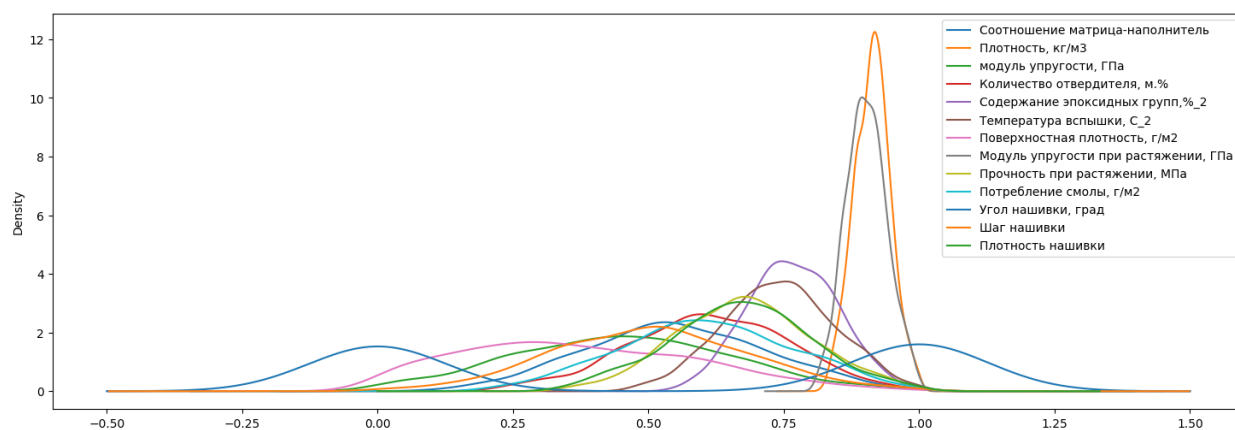


Рисунок 19 – Третьей вариант: MaxAbsScaler.

Четверной вариант: Normalizer - Этот метод нормализует данные по одной длине, что может быть полезно при решении задач, связанных с расстояниями, например, при кластеризации.

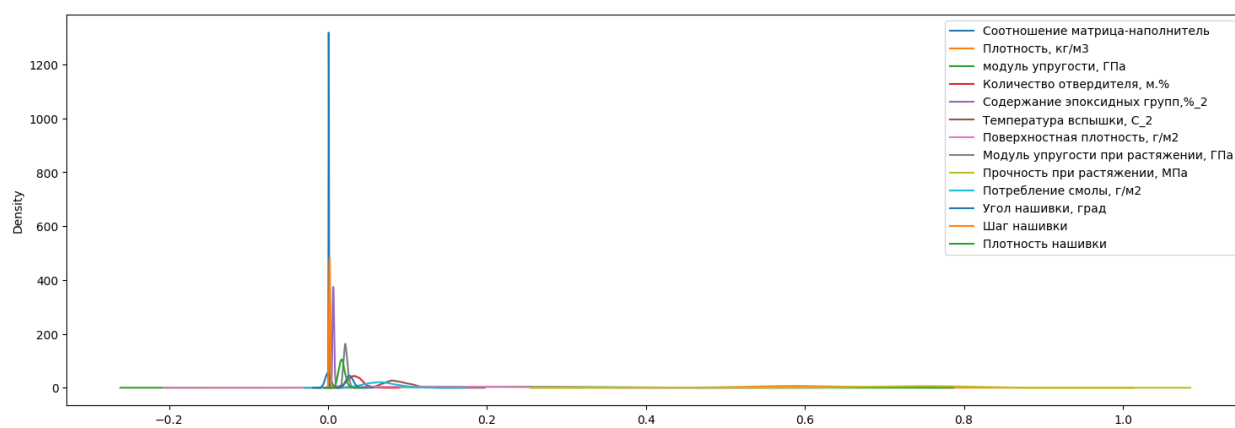


Рисунок 20 – Четверной вариант: Normalizer.

Опробовав несколько методов делаем вывод, что нам могут подойти MinMaxScaler и RobustScaler. MaxAbsScaler показал "рассеивание" по графику, а Normalizer показал выбросы. Проведя Нормализацию и сравнив два вариант (рис. 21) я склоняюсь к методу MinMaxScaler, т.к. он является простым и эффективным инструментом для нормализации данных, который может значительно улучшить результаты машинного обучения ваших моделей. Предварительно, мы очистили данные от "выбросов" и подготовили данные. Последующая стандартизация может не вернуть данные к их исходному распределению. Это может привести к потере информации об изменениях в данных. Поэтому проводить её не будем.

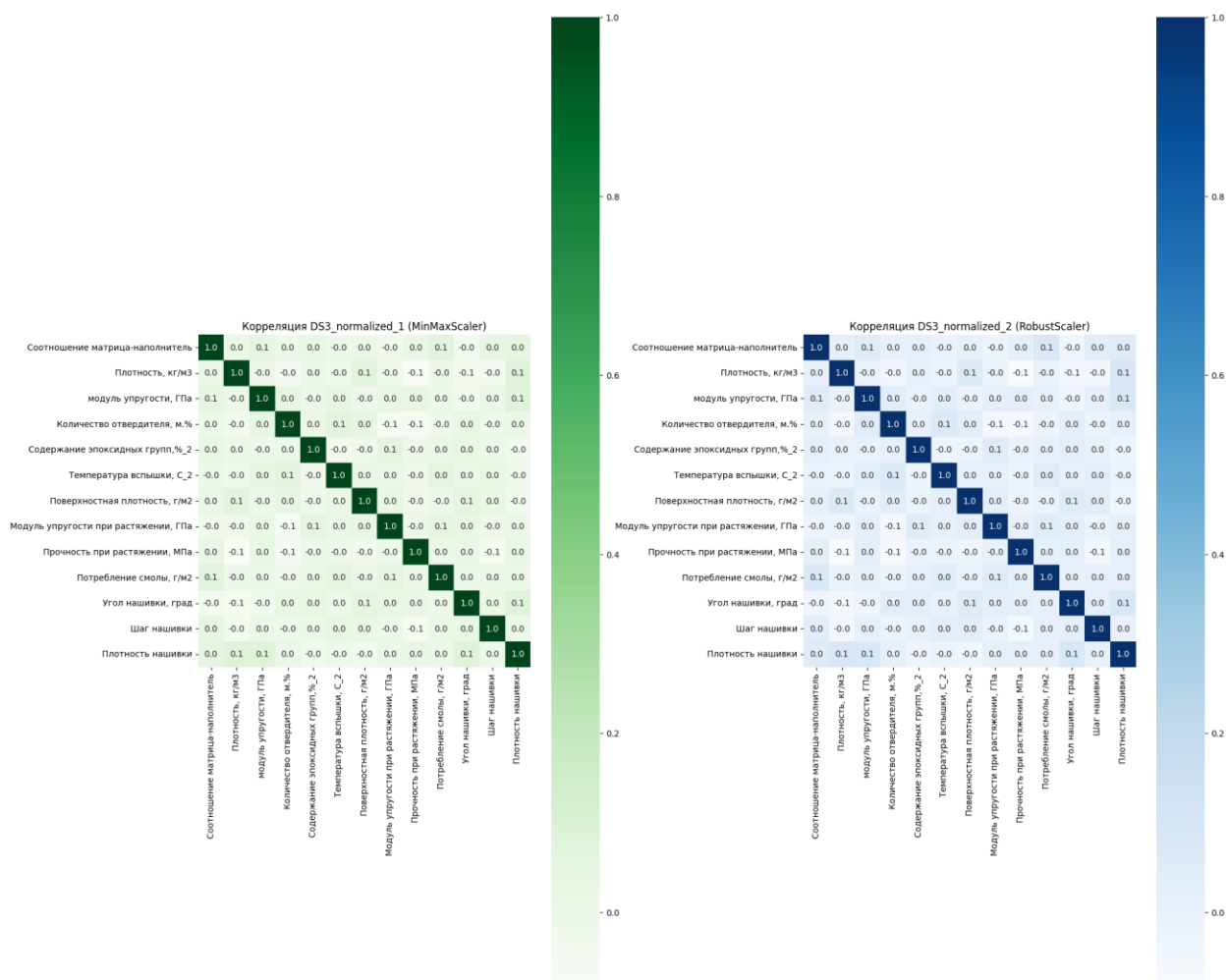


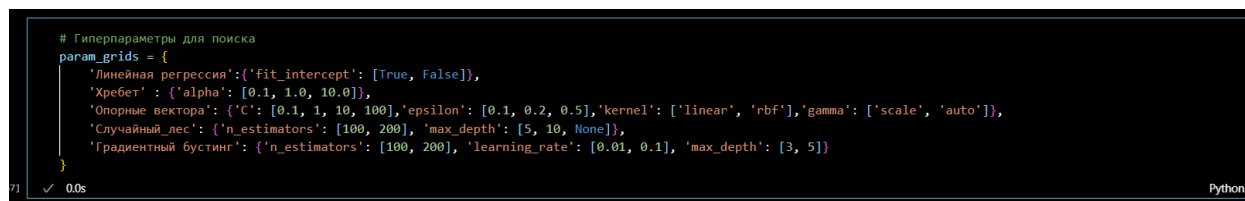
Рисунок 21 – Сравнение Корреляция MinMaxScaler и RobustScaler.

2.2. Разработка и обучение модели.

Разработка и обучение моделей машинного обучения осуществлялась для двух выходных параметров: «Прочность при растяжении» и «Модуль упругости при растяжении» отдельно. Для решения применим методы, описанные выше, напомним их ещё раз:

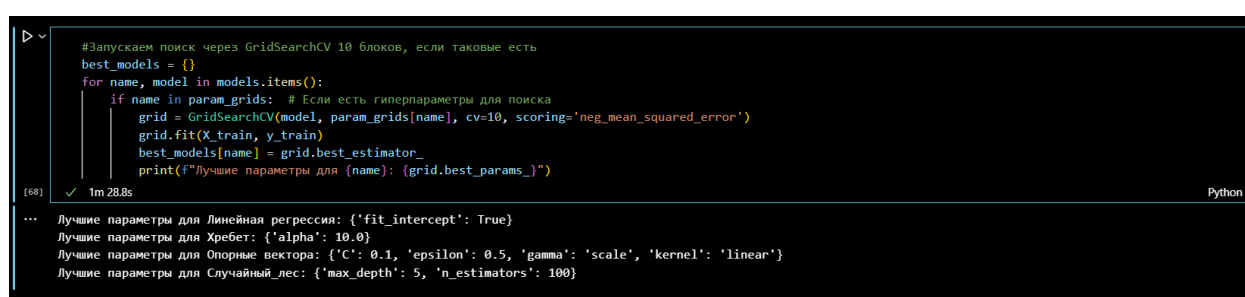
- 'Линейная регрессия': LinearRegression();
- 'Хребет': Ridge();
- 'Опорные вектора': SVR();
- 'Случайный лес': RandomForestRegressor();
- 'Градиентный бустинг ': GradientBoostingRegressor().

Согласно заданию, необходимо 30% данных оставить на тестирование модели, на остальных происходит обучение моделей. При построении моделей провести поиск гиперпараметров моделей с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10.



```
# Гиперпараметры для поиска
param_grids = {
    'Линейная регрессия': {'fit_intercept': [True, False]},
    'Хребет': {'alpha': [0.1, 1.0, 10.0]},
    'Опорные вектора': {'C': [0.1, 1, 10, 100], 'epsilon': [0.1, 0.2, 0.5], 'kernel': ['linear', 'rbf'], 'gamma': ['scale', 'auto']},
    'Случайный лес': {'n_estimators': [100, 200], 'max_depth': [5, 10, None]},
    'Градиентный бустинг': {'n_estimators': [100, 200], 'learning_rate': [0.01, 0.1], 'max_depth': [3, 5]}
}
```

Рисунок 22 – код определения гиперпараметров.



```
#Запускаем поиск через GridSearchCV 10 блоков, если таковые есть
best_models = {}
for name, model in models.items():
    if name in param_grids: # Если есть гиперпараметры для поиска
        grid = GridSearchCV(model, param_grids[name], cv=10, scoring='neg_mean_squared_error')
        grid.fit(X_train, y_train)
        best_models[name] = grid.best_estimator_
        print(f"Лучшие параметры для {name}: {grid.best_params_}")
```

Лучшие параметры для Линейная регрессия: {'fit_intercept': True}
 Лучшие параметры для Хребет: {'alpha': 10.0}
 Лучшие параметры для Опорные вектора: {'C': 0.1, 'epsilon': 0.5, 'gamma': 'scale', 'kernel': 'linear'}
 Лучшие параметры для Случайный лес: {'max_depth': 5, 'n_estimators': 100}

Рисунок 23 – код определения наилучших показателей для гиперпараметров.

2.3. Тестирование модели.

После обучения моделей была проведена оценка точности этих моделей на обучающей и тестовых выборках. Для обоих входных параметров использовались одинаковые методы, что объясняет такие оценки, представленные ниже.

...	R ²	MAE	MSE	MAPE	Test Score
Линейная регрессия	-0.00	0.16	0.04	64.45	-0.00
Хребет	0.00	0.16	0.04	64.51	0.00
Опорные вектора	-0.01	0.16	0.04	66.38	-0.01
Случайный лес	-0.02	0.16	0.04	65.70	-0.02
Градиентный бустинг	-0.06	0.16	0.04	66.59	-0.06

Рисунок 24 – Результаты для оценки моделей для 'Модуль упругости при растяжении, ГПа'.

...	R ²	MAE	MSE	MAPE	Test Score
Линейная регрессия	0.00	0.16	0.04	inf	0.00
Хребет	0.00	0.16	0.04	inf	0.00
Опорные вектора	-0.00	0.16	0.04	inf	-0.00
Случайный_лес	0.01	0.16	0.04	inf	0.01
Градиентный бустинг	-0.02	0.16	0.04	inf	-0.02

Рисунок 25 – Результаты для оценки моделей для 'Прочность при растяжении'.

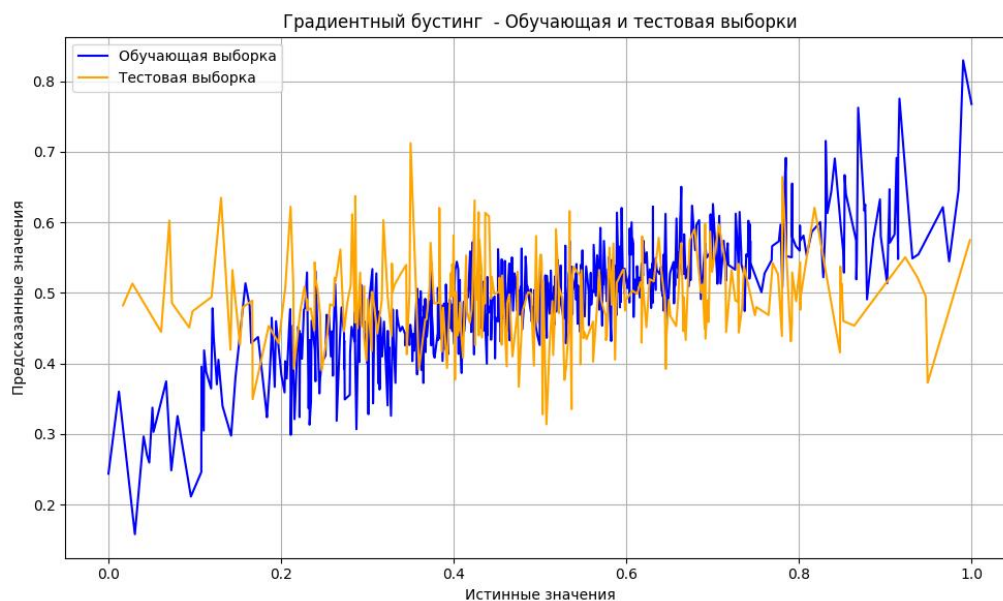


Рисунок 26 – пример графика сравнения обучающей выборки и тестовой.

Памятуя о критериях оценки моделей, можно сделать вывод, что все использованные модели не справились с задачей.

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

Обучение нейронной сети — это процесс, в ходе которого модель искусственного интеллекта учится выполнять определённые задачи на основе предоставленных ей данных. Это может быть, например, распознавание образов или предсказание тенденций.

Различают два вида обучения нейронных сетей:

- Обучение с учителем. Это наиболее распространённый подход, при котором нейросеть обучается на основе предварительно размеченных данных. Эти данные включают в себя входные значения и соответствующие им целевые (ответы). Нейросеть обучается предсказывать последние на основе входных данных.
- Обучение без учителя. Это когда нейронная сеть обучается на неразмеченных данных. Главная её задача — найти скрытые закономерности, структуры или взаимосвязи без какой-либо предварительной информации о результатах.

В данном случае мы будем использовать обучение с учителем с размеченными данными используя класс `keras.Sequential` — это фундаментальный компонент библиотеки Keras, который широко используется для построения и обучения моделей глубокого обучения.

Этот класс позволяет создавать нейронные сети путём линейного наложения слоёв. Он особенно хорошо подходит для построения простых сетей прямой связи.

Особенности класса `Sequential`:

- Простота использования. Класс разработан так, чтобы быть удобным для начинающих, позволяя пользователям быстро определять и обучать модели без сложных конфигураций.
- Единый ввод и вывод. Он поддерживает модели с одним входным тензором и одним выходным тензором, что делает его идеальным для задач с одним источником входного сигнала и одним прогнозом выходного сигнала. 1
- Укладка слоёв. Слои можно добавлять один за другим с помощью метода `add()`. В результате получается простая и интуитивно понятная архитектура модели.

- Класс Sequential не поддерживает модели с несколькими входами или выходами, или архитектуры, требующие общих уровней или сложного ветвления.

```
# Выделение признаков и целевой переменной
X = df.drop(columns=['Соотношение матрица-наполнитель']) # Удаляем целевую переменную
y = df['Соотношение матрица-наполнитель']

# Выведем формы X и y, чтобы убедиться, что разделение выполнено корректно
X.shape, y.shape
[82] ✓ 0.0s Python
... ((921, 12), (921,))

# Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
[83] ✓ 0.0s Python

model = Sequential() # Создание модели

# Добавление слоев с регуляризацией и Dropout
model.add(Dense(64, activation='tanh', input_shape=(X_train.shape[1],), kernel_regularizer='l2'))
model.add(Dropout(0.3)) # Слой Dropout
model.add(Dense(64, activation='tanh', kernel_regularizer='l2'))
model.add(Dropout(0.3)) # Слой Dropout
model.add(Dense(32, activation='tanh', kernel_regularizer='l2'))
model.add(Dense(1, activation='sigmoid'))
[84] ✓ 0.2s Python
```

Рисунок 26 – пример кода с входными параметрами для нейросети.

Предполагая, что модель может переобучиться, было принято решение добавить слои Dropout. Которые в свою очередь случайным образом выключают (обнуляет) некоторые нейроны во время обучения с определённой вероятностью. Это приводит к уменьшению сложности модели и улучшению её обобщающей способности. Так же был применен метод EarlyStopping. Этот метод позволяет остановить процесс обучения, пока модель не стала слишком специализированной и не может хорошо обобщаться на новых данных. Early Stopping особенно полезен, когда не известно оптимальное количество эпох, необходимых для сходимости модели. В таких случаях метод позволяет автоматически остановить обучение на оптимальном количестве эпох. Также Early Stopping полезен при ограниченных вычислительных ресурсах, так как экономит время обучения, останавливая процесс после нахождения лучшей модели. После обучения модели мы получили оценку $MSE = 0.0345$. Такая оценка может считаться хорошим результатом, так как чем меньше значение MSE, тем лучше модель.

MSE измеряет среднее квадратичное отклонение предсказанных значений от истинных значений, и чем ближе значение к нулю, тем модель более точная.

```
test_loss, test_mse, = model.evaluate(X_test, y_test)
print(f'MSE: {test_mse:.4f}')

[063] ✓ 0.1s
... 9/9 0s 7ms/step - loss: 0.0344 - mse: 0.0344
MSE: 0.0345
```

Рисунок 27 – пример кода с оценкой модели.

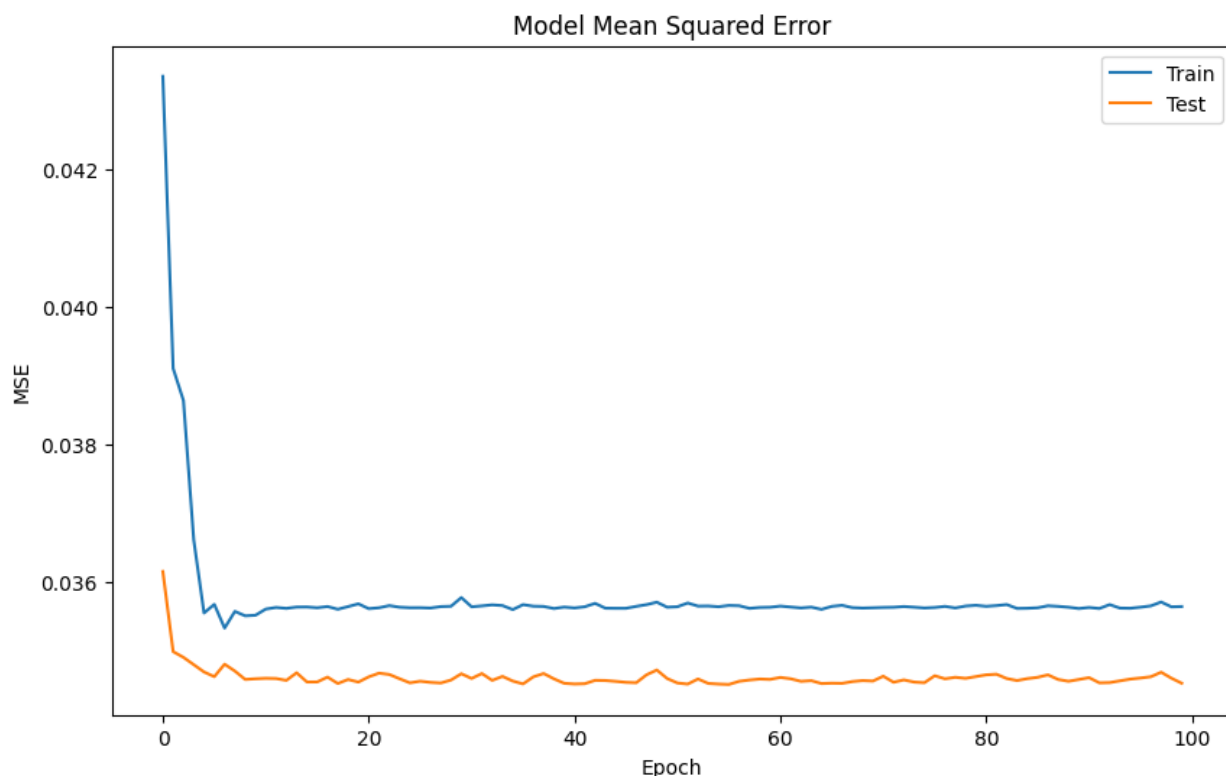


Рисунок 27 –график показывающий Среднеквадратичную ошибку модели на обучаемы и тестовых данных.

2.5. Разработка приложения, которая будет рекомендовать соотношение «матрица – наполнитель».

При разработке приложения я столкнулся с тем, что изначально использовал нормализованные данные в пределах (0 и 1) и никак не мог вернуть данные в «исходное» состояние. Было принято решение повторить путь по созданию новой нейросети с другим параметрами слоёв и данная модель была сохранена в дальнейшем для разработки графического приложения. Кроме того, поскольку я не помнил диапазон входных данных, решил добавить их сразу в виде описания к

полям для ввода данных. Так же добавлена функция предупреждающая об ошибочно введенных данных.

```
# Define the model
model = Sequential()
model.add(Dense(128, activation='relu', input_shape=(X_train.shape[1,]), kernel_regularizer='l2'))
model.add(Dropout(0.3))
model.add(Dense(128, activation='relu', kernel_regularizer='l2'))
model.add(Dropout(0.3))
model.add(Dense(64, activation='relu', kernel_regularizer='l2'))
model.add(Dense(1, activation='linear'))
```

Рисунок 28 – пример кода с входными параметрами для нейросети второй вариант

```
# Evaluate the model
test_loss, test_mse = model.evaluate(X_test, y_test)
print(f'MSE after normalization: {test_mse:.4f}')
```

9/9 0s 3ms/step - loss: 0.8594 - mse: 0.7992
MSE after normalization: 0.8009

Рисунок 29 – пример кода с оценкой модели второй вариант.

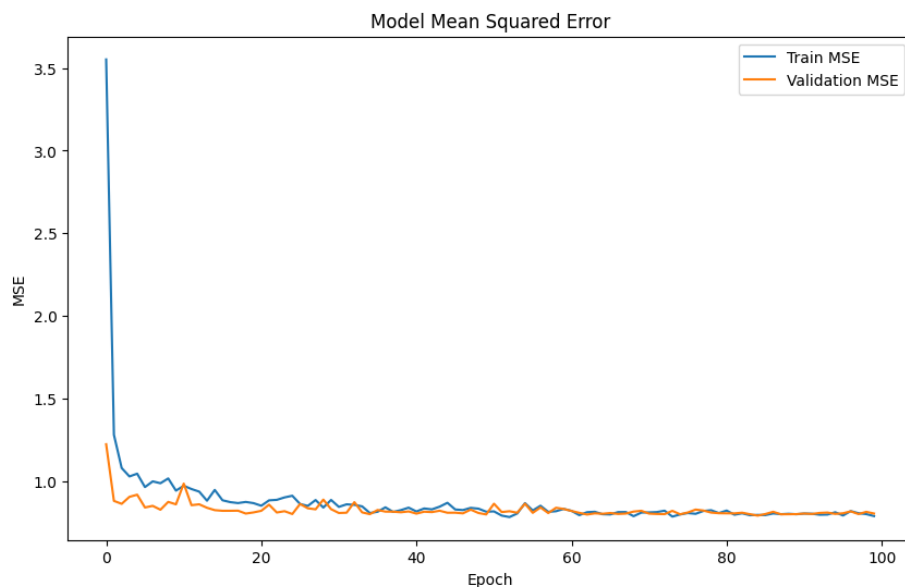


Рисунок 29 –график показывающий Среднеквадратичную ошибку модели на обучаеы и тестовых данных.

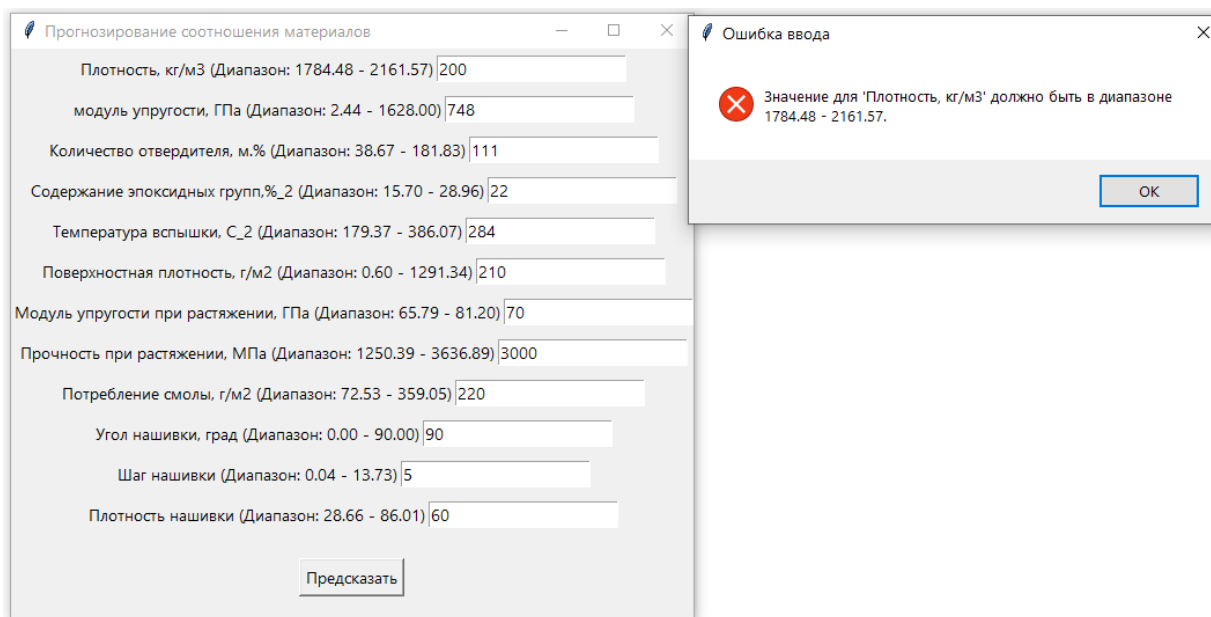


Рисунок 30 – скриншот пользовательского приложения

На выходе пользователь получает результат прогноза для значения параметра «Соотношение «матрица – наполнитель»». Пример с введёнными не корректными данными.

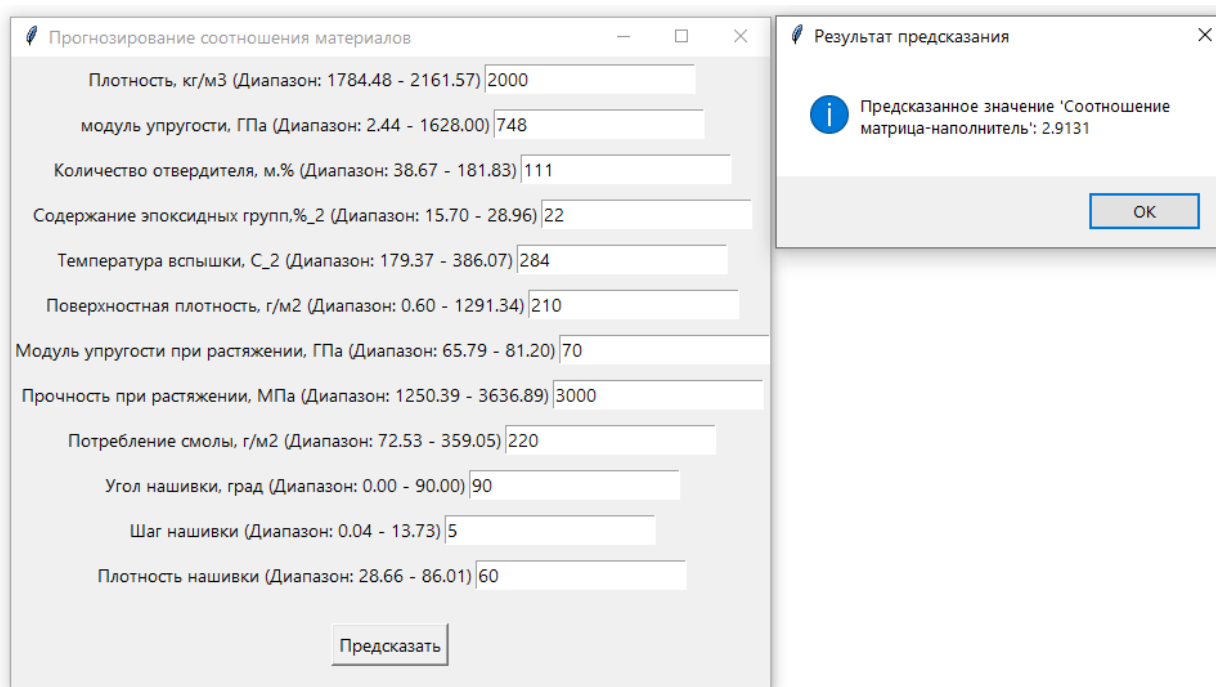


Рисунок 31 – скриншот пользовательского приложения

На выходе пользователь получает результат прогноза для значения параметра «Соотношение «матрица – наполнитель»». Пример с полученным прогнозом.

Заключение

В ходе выполнения данной работы мы прошли практически весь Pipelines, рассмотрели большую часть операций и задач, которые приходится выполнять специалисту по работе с данными.

Этот поток операций и задач включает:

- изучение теоретических методов анализа данных и машинного обучения;
- изучение основ предметной области, в которой решается задача;
- извлечение и трансформацию данных;
- проведение разведочного анализа данных статистическими методами;
- разделение имеющихся, в нашем случае размеченных, данных на обучающую и тестовую выборки;
- выполнение предобработки данных для обеспечения корректной работы моделей;
- построение аналитического решения. Это включает выбор алгоритма решения и модели, сравнение различных моделей, подбор гиперпараметров модели;
- визуализация модели и оценка качества решения;
- сохранение моделей;
- разработка и тестирование приложения.

В этой работе мы к сожалению, не смогли решить поставленную задачу - не получили моделей, которые бы описывали закономерности предметной области. Со своей стороны, я применил большую часть знаний, полученных в ходе прохождения курса.

Данная тема очень востребована. Для реализации исследования на практике потребуется как специфические нейросети обученные на химию процессов, так и на сопромат, чтобы моделировать физические свойства брусков/изделий композитов. Также, это исследование на практике потребует вычислительных ресурсов.

Поэтому, тут нужно идти на стажировку к металлургам или химикам для более глубокого понимания предмета по композитам.

Не исключаю, что что-то не так с исходными данными или неверный выбор процесс подготовки данных.

Список используемой литературы и веб ресурсы.

1. В.Г. Соловьева, В.Ф. Коровякова, О.А. Ларсен, Н.А. Гальцева Композиционные материалы в строительстве (Учебное пособие для обучающихся по направлению подготовки 08.03.01 Строительство), Национальный исследовательски Московский государственный строительный университет, 2020

2. Лейси Никола Python, например. — СПб.: Питер, 2021. — 208 с.: ил. — (Серия «Библиотека программиста»).

3. Jupyter Notebook. Часть 2: Режим доступа : <https://www.dmitrymakarov.ru/python/jupyter-notebook-14-02/> (дата обращения: 09.12.2024)

4. Композиционные материалы: Режим доступа : <https://e--plastic.ru.turbopages.org/e-plastic.ru/s/specialistam/composite/kompozicionnye-materialy/> (дата обращения: 10.12.2024)

5. Предварительный анализ и обработка данных: Режим доступа: <https://koroteev.site/pres/ml5/> (дата обращения: 11.12.2024)

6. Seaborn Heatmaps: 13 способов настроить визуализацию матрицы корреляции: Режим доступа: <https://datastart.ru/blog/read/seaborn-heatmaps-13-sposobov-nastroit-vizualizaciyu-matricy-korrelyacii> (дата обращения: 11.12.2024)

7. Пирсон против Спирмена против Кендалла: Режим доступа: https://translated.turbopages.org/proxy_u/en-ru.ru.35ab8a44-67611bb6-7ae978a7-74722d776562/https/datascience.stackexchange.com/questions/64260/pearson-vs-spearman-vs-kendall (дата обращения: 11.12.2024)

8. Определение выбросов в таблице статистики: Режим доступа: https://translated.turbopages.org/proxy_u/en-ru.ru.bdfc344d-67611c27-18829480-74722d776562/https/www.geeksforgeeks.org/identifying-outliers-in-statistics-

[worksheet/](#) (дата обращения: 12.12.2024)

9. Нормализация данных с помощью Python Scikit-Learn: Режим доступа: https://translated.turbopages.org/proxy_u/en-ru.ru.313df400-67611c8f-8214a187-74722d776562/https/www.geeksforgeeks.org/data-normalization-with-python-scikit-learn/ (дата обращения: 12.12.2024)

10. Метод MinMaxScaler: Режим доступа: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html> (дата обращения: 13.12.2024)

11. Метод Normalizer: Режим доступа: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Normalizer.html> (дата обращения: 13.12.2024)

12. Машинное обучение. Погружение в технологию, Артем Демиденко, 2023: Режим доступа: https://kartaslov.ru/%D0%BA%D0%BD%D0%B8%D0%B3%D0%B8/%D0%90%D1%80%D1%82%D0%B5%D0%BC_%D0%94%D0%B5%D0%BC%D0%B8%D0%B4%D0%B5%D0%BD%D0%BA%D0%BE_%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%9F%D0%BE%D0%B3%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B2_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8E/1 (дата обращения: 13.12.2024)

13. Делаем проект по машинному обучению на Python. Часть 2: Режим доступа: <https://habr.com/ru/companies/nix/articles/425907/> (дата обращения: 13.12.2024)

14. Обзор самых популярных алгоритмов машинного обучения: Режим доступа: <https://tproger.ru/translations/top-machine-learning-algorithms> (дата обращения: 13.12.2024)

15. Гиперпараметрический поиск и оптимизация моделей: Режим доступа: <https://habr.com/ru/companies/otus/articles/754402/> (дата обращения: 14.12.2024)

16. Настройка гиперпараметров модели: Режим доступа: https://scikit-learn.ru/stable/modules/grid_search.html (дата обращения: 15.12.2024)
17. Простыми словами про метрики в ИИ. Регрессия. MSE, RMSE, MAE, R-квадрат, MAPE: Режим доступа: <https://habr.com/ru/articles/820499/> (дата обращения: 15.12.2024)
18. Метрики качества линейных регрессионных моделей: Режим доступа: <https://loginom.ru/blog/quality-metrics> (дата обращения: 15.12.2024)
19. Построение искусственных нейронных сетей (ANN) с нуля: Режим доступа: https://translated.turbopages.org/proxy_u/en-ru.ru.877f6145-676120fa-3b9c48e6-74722d776562/https/www.geeksforgeeks.org/building-artificial-neural-networks-ann-from-scratch/ (дата обращения: 15.12.2024)
20. Dropout layer: Режим доступа: https://keras.io/api/layers/regularization_layers/dropout/ (дата обращения: 15.12.2024)
21. Использование ранней остановки для уменьшения переобучения в нейронных сетях: Режим доступа: https://translated.turbopages.org/proxy_u/en-ru.ru.aaf0dc93-67612227-642d4c9e-74722d776562/https/www.geeksforgeeks.org/using-early-stopping-to-reduce-overfitting-in-neural-networks/ (дата обращения: 15.12.2024)
22. tkinter— Интерфейс Python: Режим доступа: <https://docs.python.org/3/library/tkinter.html> (дата обращения: 16.12.2024)
23. Курс по библиотеке Tkinter языка Python: Режим Доступа: https://ru.wikiversity.org/wiki/%D0%9A%D1%83%D1%80%D1%81_%D0%BF%D0%BE_%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B5_Tkinter_%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0_Python (дата обращения: 16.12.2024)