

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

Пахомов Игорь Александрович

Москва, 2023

Содержание

Введение	3
1. Анализ исходных данных и вбор методов решений	5
1.1. Описание задачи и исходных данных	5
1.2. Описание используемых методов.....	7
1.3. Разведочный анализ и визуализация исходных данных.....	13
2. Практическая часть	22
2.1. Предобработка данных	22
2.2 Разработка и обучение моделей.....	24
2.3 Тестирование моделей	26
2.4. Разработка нейронной сети	33
3. Разработка приложения	39
4. Создание удаленного репозитория	40
Заключение	41
Список используемой литературы.....	42

Введение

Современный мир требует от нас постоянного развития, и наука и технологии являются главными средствами, позволяющими нам это делать. В частности, разработка новых материалов – ключевой фактор, определяющий технологический прогресс, который приводит к созданию новых устройств и техники, которые недавно еще были невозможными. Однако, создание новых материалов – сложный процесс, который требует больших затрат времени и ресурсов. В связи с этим возникает необходимость в разработке методов и технологий, которые позволят сократить время и средства, необходимые для создания новых материалов. Одним из основных методов является прогнозирование конечных свойств новых материалов. Это позволяет ускорить процесс исследований, уменьшить затраты на эксперименты и сделать процесс создания новых материалов более эффективным и безопасным.

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

Композитные материалы – многокомпонентные материалы, изготовленные из двух (или более) компонентов с существенно различающимися физическими и/или химическими свойствами, которые, в сочетании, приводят к появлению нового материала с характеристиками, отличными от характеристик отдельных компонентов и не являющимися простой их суперпозицией

В составе композита принято выделять матрицу/матрицы и наполнитель/наполнители. Варьируя состав матрицы и наполнителя, их соотношения, ориентацию наполнителя, можно получить материалы с требуемым сочетанием эксплуатационных и технологических свойств. Многие композиты превосходят традиционные материалы и сплавы по своим механическим свойствам и в то же время они легче. Использование композитов обычно позволяет уменьшить массу конструкции при сохранении или улучшении её механических характеристик.

Композиты, в которых матрицей служит полимерный материал, являются одним из самых многочисленных и разнообразных видов материалов. Их применение в различных областях даёт значительный экономический эффект.

При использовании феноменологического подхода при расчете и проектировании элементов конструкций из композитов даже одного вида материала, сопряжено с проведением трудоемких экспериментальных исследований, из этого следует, что полученные таким образом данные представляют высокую ценность, а их обработка требует дополнительного анализа.

Вопрос построения математической модели состоит в поиске достаточно надежного описания количественных взаимосвязей между свойствами компонентов композита и композитного материала при различных способах их сочетания.

Современным подходом к решению задач такого типа является применение технологий машинного обучения в целях исследования влияния одной или нескольких независимых переменных на зависимую переменную.

Актуальность решения задачи обусловлена широким использованием композитных материалов практически во всех областях производства.

Прогнозирование модели может существенно сократить количество проводимых испытаний, а также пополнить базу данных материалов новыми свойствами материалов, и цифровыми двойниками новых композитов.

1. Аналитическая часть

1.1. Описание задачи и исходных данных

Предметом настоящей работы является построение при помощи методов машинного обучения моделей прогнозирования характеристик «модуль упругости при растяжении» и «прочность при растяжении», рекомендации «соотношение матрица-наполнитель».

Исходные данные о свойствах композиционных материалов и способах их компоновки получены структурным подразделением МГТУ им. Н.Э. Баумана – Центр НТИ «Цифровое материаловедение: новые материалы и вещества» в рамках решения производственных задач.

Датасет состоит из двух файлов: X_br.xlsx (признаки базальтопластика) и X_nur.xlsx (признаки углепластика) (рисунке 1.):

- Файл X_br.xlsx -содержит 1023 строки, индекс и 10 признаков.
- Файл X_nur.xlsx -содержит 1040 строк индекс и 3 признака.

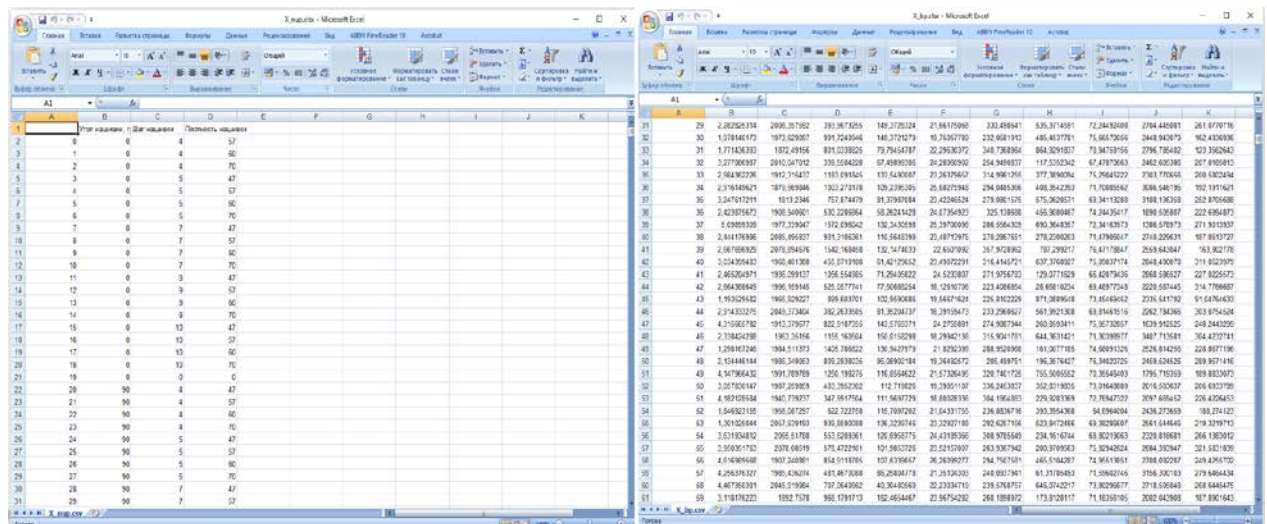


Рисунок 1 – Значения в файле X_br.xlsx и в файле X_nur.xlsx

Соотношения и свойства используемых компонентов композитов (6 входных переменных вещественного типа), а также интересующие выходные характеристики композитов (3 выходных переменных вещественного типа и 7 вход-

ных переменных вещественного типа), представлены в виде excel-таблицы, которая содержит 1023 строки и 10 столбцов с данными.

Способы компоновки материалов композитов (3 входных переменных вещественного типа) представлены в виде в виде excel-таблицы, которая содержит 1040 строк и 3 столбца с данными.

Данные таблицы имеют колонку с целочисленным индексом, не являющимся входным или выходным переменным, служащим для сопоставления таблиц данных.

Поставленная задача в рамках классификации категорий машинного обучения относится к машинному обучению с учителем, задача регрессии.

Цель работы разработать модели для прогноза модуля упругости при растяжении, прочности при растяжении и соотношения «матрица-наполнитель». Для этого нужно объединить 2 файла. Часть информации не имеют соответствующих строк в таблице соотношений и свойств используемых компонентов композитов, поэтому были удалены.

Затем провести разведочный анализ данных, нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек.

Обучить несколько моделей для прогноза модуля упругости при растяжении и прочности при растяжении. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель. Разработать приложение с графическим интерфейсом, которое будет выдавать прогноз соотношения «матрица-наполнитель». Оценить точность модели на тренировочном и тестовом датасете. Создать репозиторий в GitHub и разместить код исследования.

Анализ, предобработка данных, построение моделей выполнены посредством языка программирования Python с использованием библиотек Pandas, Matplotlib и Sklearn.

1.2. Описание используемых методов

Данная задача в рамках классификации категорий машинного обучения относится к машинному обучению с учителем и традиционно это задача регрессии. Цель любого алгоритма обучения с учителем — определить функцию потерь и минимизировать её, поэтому для наилучшего решения в процессе исследования были применены следующие методы:

- метод опорных векторов;
- случайный лес;
- линейная регрессия;
- градиентный бустинг;
- К-ближайших соседей;
- дерево решений;
- стохастический градиентный спуск;
- многослойный перцептрон;
- Лассо;

Метод опорных векторов (Support Vector Regression) – этот бинарный линейный классификатор был выбран, потому что он хорошо работает на небольших датасетах. Данный алгоритм – это алгоритм обучения с учителем, использующихся для задач классификации и регрессионного анализа, это контролируемое обучение моделей с использованием схожих алгоритмов для анализа данных и распознавания шаблонов. Учитывая обучающую выборку, где алгоритм помечает каждый объект, как принадлежащий к одной из двух категорий, строит модель, которая определяет новые наблюдения в одну из категорий.

Модель метода опорных векторов – отображение данных точками в пространстве, так что между наблюдениями отдельных категорий имеется разрыв, и он максимален.

Каждый объект данных представляется как вектор (точка) в p -мерном пространстве. Он создаёт линию или гиперплоскость, которая разделяет данные на классы.

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

Недостатки метода: неустойчивость к шуму, поэтому в работе была проведена тщательнейшая работа с выбросами, иначе в обучающих данных шумы становятся опорными объектами-нарушителями и напрямую влияют на построение разделяющей гиперплоскости; для больших наборов данных требуется долгое время обучения; достаточно сложно подбирать полезные преобразования данных; параметры модели сложно интерпретировать, поэтому были рассмотрены и другие методы.

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

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

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

Линейная регрессия (Linear regression) — это алгоритм машинного обучения, основанный на контролируемом обучении, рассматривающий зависимость между одной входной и выходными переменными. Это один из самых простых и эффективных инструментов статистического моделирования. Она определяет зависимость переменных с помощью линии наилучшего соответствия. Модель регрессии создаёт несколько метрик. R^2 , или коэффициент детерминации, позволяет измерить, насколько модель может объяснить дисперсию данных. Если R-квадрат равен 1, это значит, что модель описывает все данные. Если же R-квадрат равен 0,5, модель объясняет лишь 50 процентов дисперсии данных. Оставшиеся отклонения не имеют объяснения. Чем ближе R^2 к единице, тем лучше.

Достоинства метода: быстр и прост в реализации; легко интерпретируем; имеет меньшую сложность по сравнению с другими алгоритмами;

Недостатки метода: моделирует только прямые линейные зависимости; требует прямую связь между зависимыми и независимыми переменными; выбросы оказывают огромное влияние, а границы линейны.

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

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

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

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

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

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

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

го вывода. Дерево решений один из вариантов решения регрессионной задачи, в случае если зависимость в данных не имеет очевидной корреляции.

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

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

Стохастический градиентный спуск (SGDRegressor) — это простой, но очень эффективный подход к подгонке линейных классификаторов и регрессоров под выпуклые функции потерь. Этот подход подразумевает корректировку весов нейронной сети, используя аппроксимацию градиента функционала, вычисленную только на одном случайном обучающем примере из выборки.

Достоинства метода: эффективен; прост в реализации; имеет множество возможностей для настройки кода; способен обучаться на избыточно больших выборках.

Недостатки метода: требует ряд гиперпараметров; чувствителен к масштабированию функций; может не сходиться или сходиться слишком медленно; функционал многоэкстремален; процесс может "застрять" в одном из локальных минимумов; возможно переобучение.

Многослойный персептрон (MLPRegressor) — это алгоритм обучения с учителем, который изучает функцию $f(\cdot): R_m \rightarrow R_o$ обучением на наборе данных, где m — количество измерений для ввода и o — количество размеров для вывода. Это искусственная нейронная сеть, имеющая 3 или более слоёв персептро-

нов. Эти слои - один входной слой, 1 или более скрытых слоёв и один выходной слой персептронов.

Достоинства метода: построение сложных разделяющих поверхностей; возможность осуществления любого отображения входных векторов в выходные; легко обобщает входные данные; не требует распределения входных векторов; изучает нелинейные модели.

Недостатки метода: имеет невыпуклую функцию потерь; разные инициализации случайных весов могут привести к разной точности проверки; требует настройки ряда гиперпараметров; чувствителен к масштабированию функций.

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

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

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

Немного расскажем об используемых метриках качества моделей: R^2 или коэффициент детерминации измеряет долю дисперсии, объяснённую моделью, в общей дисперсии целевой переменной.

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

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

1.3. Разведочный анализ и виртуализация исходных данных

Прежде чем передать данные в работу моделей машинного обучения, необходимо обработать и очистить их. Очевидно, что «грязные» и необработанные данные могут содержать искажения и пропущенные значения – это ненадежно, поскольку способно привести к крайне неверным результатам по итогам моделирования. Но безосновательно удалять что-либо тоже неправильно. Именно поэтому сначала набор данных надо изучить

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

Разведочный анализ данных в рамках данной задачи проведен над датафреймом Pandas, полученным путем импорта (и объединения по типу INNER по полю индекса) таблиц исходных данных.

В качестве инструментов разведочного анализа используется: оценка статистических характеристик датасета; гистограммы распределения каждой из переменной (несколько различных вариантов); диаграммы ящика с усами (несколько интерактивных вариантов); попарные графики рассеяния точек (несколько вариантов); график «квантиль-квантиль»; тепловая карта (несколько вариантов); описательная статистика для каждой переменной; анализ и полное

исключение выбросов (5 повторных итераций); проверка наличия пропусков и дубликатов.

Таблица 1 – Наименования параметров

№п/п	Наименование параметра	Входной/Выходной параметр
1	Соотношение матрица-наполнитель	Выходной
2	Модуль упругости при растяжении, ГПа	Выходной
3	Прочность при растяжении, МПа	Выходной
4	Плотность, кг/м3	Входной
5	модуль упругости, ГПа	Входной
6	Количество отвердителя, м.%	Входной
7	Содержание эпоксидных групп,%_2	Входной
8	Температура вспышки, С_2	Входной
9	Поверхностная плотность, г/м2	Входной
10	Потребление смолы, г/м2	Входной
11	Угол нашивки, град	Входной
12	Шаг нашивки	Входной
13	Плотность нашивки	Входной

Сформированный исходный датафрейм содержит 1023 записи с 9 входными параметрами и 3 выходными параметрами вещественного типа, пропуски значений отсутствуют.

Показатели описательной статистики и визуализация гистограмм и/или диаграмм размаха («ящик с усами») позволяют получить наглядное представление о характерах распределений переменных.

Описательная статистика исходных данных описана на рисунке 1.

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
count	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000
mean	2.930366	1975.734888	739.923233	110.570769	22.244390	285.882151	482.731833	73.328571	2466.922843	218.423144	44.252199	6.899222	57.153929
std	0.913222	73.729231	330.231581	28.295911	2.406301	40.943260	281.314690	3.118983	485.628006	59.735931	45.015793	2.563467	12.350969
min	0.389403	1731.764635	2.436909	17.740275	14.254985	100.000000	0.603740	64.054061	1036.856605	33.803026	0.000000	0.000000	0.000000
25%	2.317887	1924.155467	500.047452	92.443497	20.608034	259.066528	266.816645	71.245018	2135.850448	179.627520	0.000000	5.080033	49.799212
50%	2.906878	1977.621657	739.664328	110.564840	22.230744	285.896812	451.864365	73.268805	2459.524526	219.198882	0.000000	6.916144	57.341920
75%	3.552660	2021.374375	961.812526	129.730366	23.961934	313.002106	693.225017	75.356612	2767.193119	257.481724	90.000000	8.586293	64.944961
max	5.591742	2207.773481	1911.536477	198.953207	33.000000	413.273418	1399.542362	82.682051	3848.436732	414.590628	90.000000	14.440522	103.988901

Рисунок 2- Описательная статистика исходных данных

Гистограммы используются для изучения распределений частот значений переменных.

Построим гистограммы распределения по каждой переменной для оценки повторяющихся значений в многомерном пространстве.

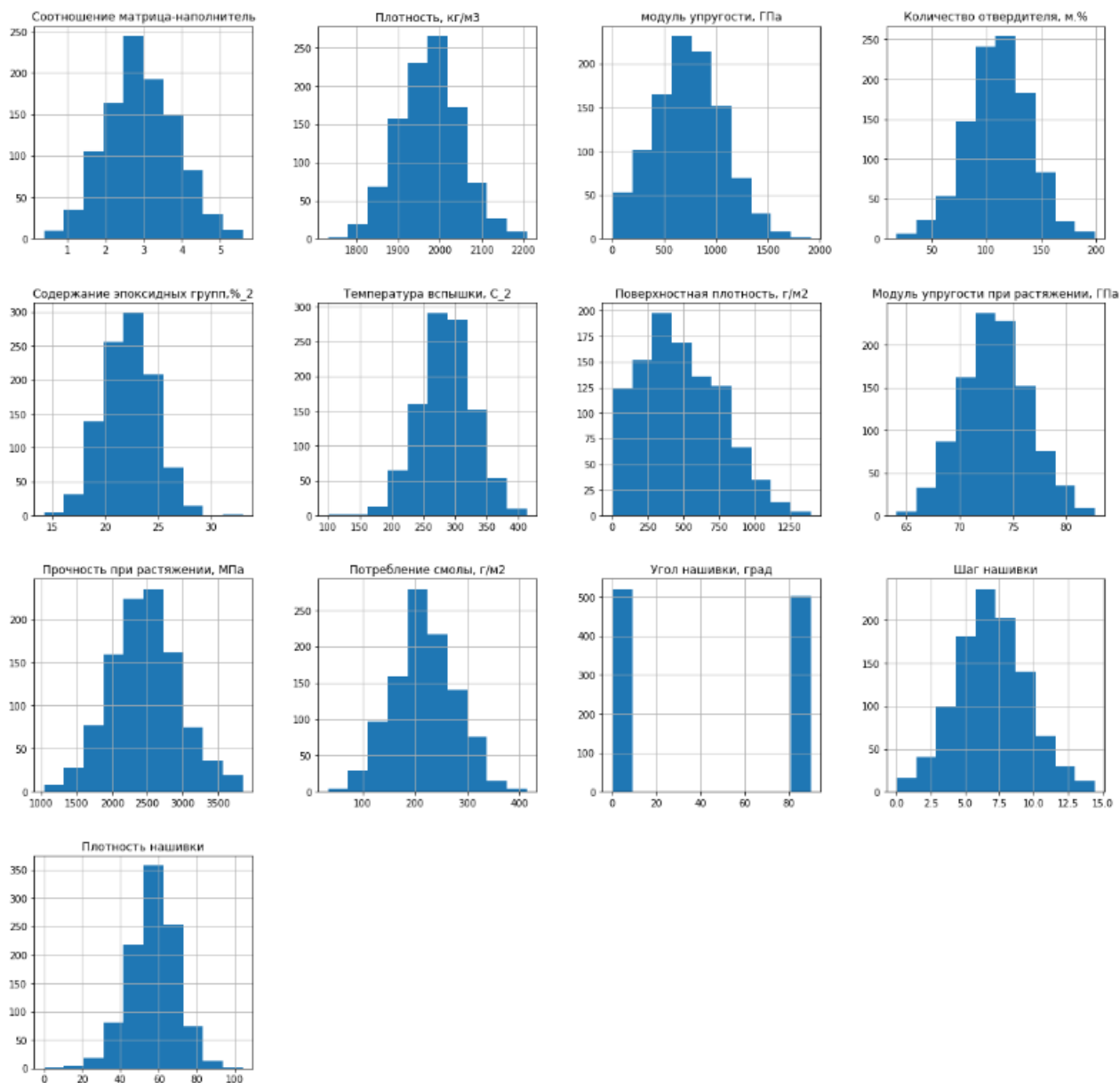


Рисунок 3- Гистограммы распределения

Мы видим очень слабую корреляцию между переменными.

Построим для каждой переменной диаграммы размаха для определения наличия выбросов в данных. Шкалы приведем к величинам в диапазоне $[0,1]$, чтобы "ящики с усами" были одного масштаба. Это поможет нам определить все выбросы и избавиться от них в дальнейшем для того, чтобы набор данных имел более сглаженный вид с точки зрения нормализации.

Построим для каждой переменной диаграммы размаха для определения наличия выбросов в данных. Шкалы приведем к величинам в диапазоне $[0,1]$, чтобы "ящики с усами" были одного масштаба. Это поможет нам определить все выбросы и избавиться от них в дальнейшем для того, чтобы набор данных имел более сглаженный вид с точки зрения нормализации.

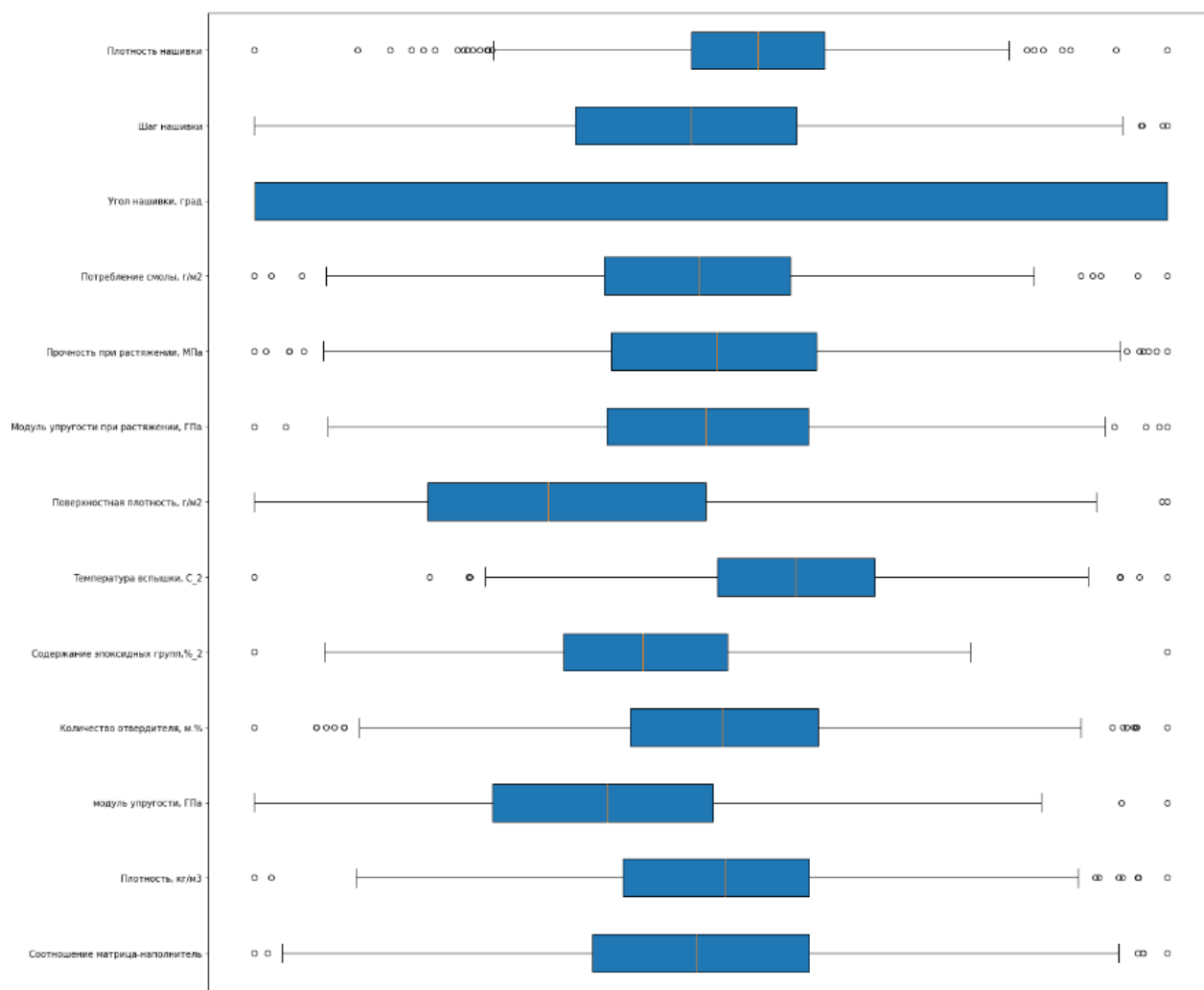


Рисунок 4- Диаграммы размаха

Как видим на рисунке 3 все параметры, кроме «Угла нашивки», представленного всего двумя значениями (0 градусов и 90 градусов), сильнее прочих также выделяется «Поверхностная плотность, г/м²», медиана которого разнится с выборочным средним сильнее, чем у других параметров, а форма распределения менее других походит на нормальное.

Также «ящики с усами» по каждому из параметров, кроме «Угол нашивки, град», показывают наличие некоторого количества значений, находящихся за

пределами полутора межквартильных расстояний от первого и третьего квартилей.

Принимая во внимание, во-первых, источник формирования данных – решение производственных задач (данные измерений), во-вторых то, что нетипичные значения параметров, хотя и находятся за пределами «усов», не демонстрируют экстремально больших отклонений, и в-третьих то, что такие значения присутствуют в том числе и у целевых параметров (а задача исследований в данной области – получение композитов с уникальными свойствами), такие значения вне дополнительных уточнений не следует трактовать как выбросы, по крайней мере, до тех пор, пока их наличие в обучающей и тестовых выборках не будет негативно сказываться на точности предсказаний модели.

Менее радикальным способом оценки качества исходных данных является применение «правила трех сигм».

Описательная статистика и диаграммы размаха датасета после удаления выбросов с применением «правила трех сигм» представлены на рисунках 4 - 6.

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп, %_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град
count	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000
mean	0.000888	0.587569	0.216978	0.032887	0.006601	0.085021	0.142062	0.021821	0.742763	0.064855	0.013891
std	0.000275	0.056993	0.092124	0.008474	0.000926	0.014082	0.081065	0.002203	0.056655	0.017766	0.013515
min	0.000163	0.444650	0.000709	0.011339	0.004113	0.049402	0.001902	0.016105	0.590461	0.021630	0.000000
25%	0.000679	0.546948	0.151021	0.027292	0.005925	0.075135	0.078825	0.020292	0.706068	0.052063	0.000000
50%	0.000857	0.585227	0.219229	0.032910	0.006589	0.063934	0.138593	0.021720	0.747345	0.064468	0.022461
75%	0.001052	0.626059	0.280808	0.036817	0.007208	0.094462	0.199600	0.023319	0.782554	0.076808	0.026792
max	0.001593	0.743130	0.476145	0.055088	0.009122	0.123083	0.368343	0.027834	0.877580	0.114133	0.034285

Рисунок 5- Описательная статистика датасета после очистки выбросов

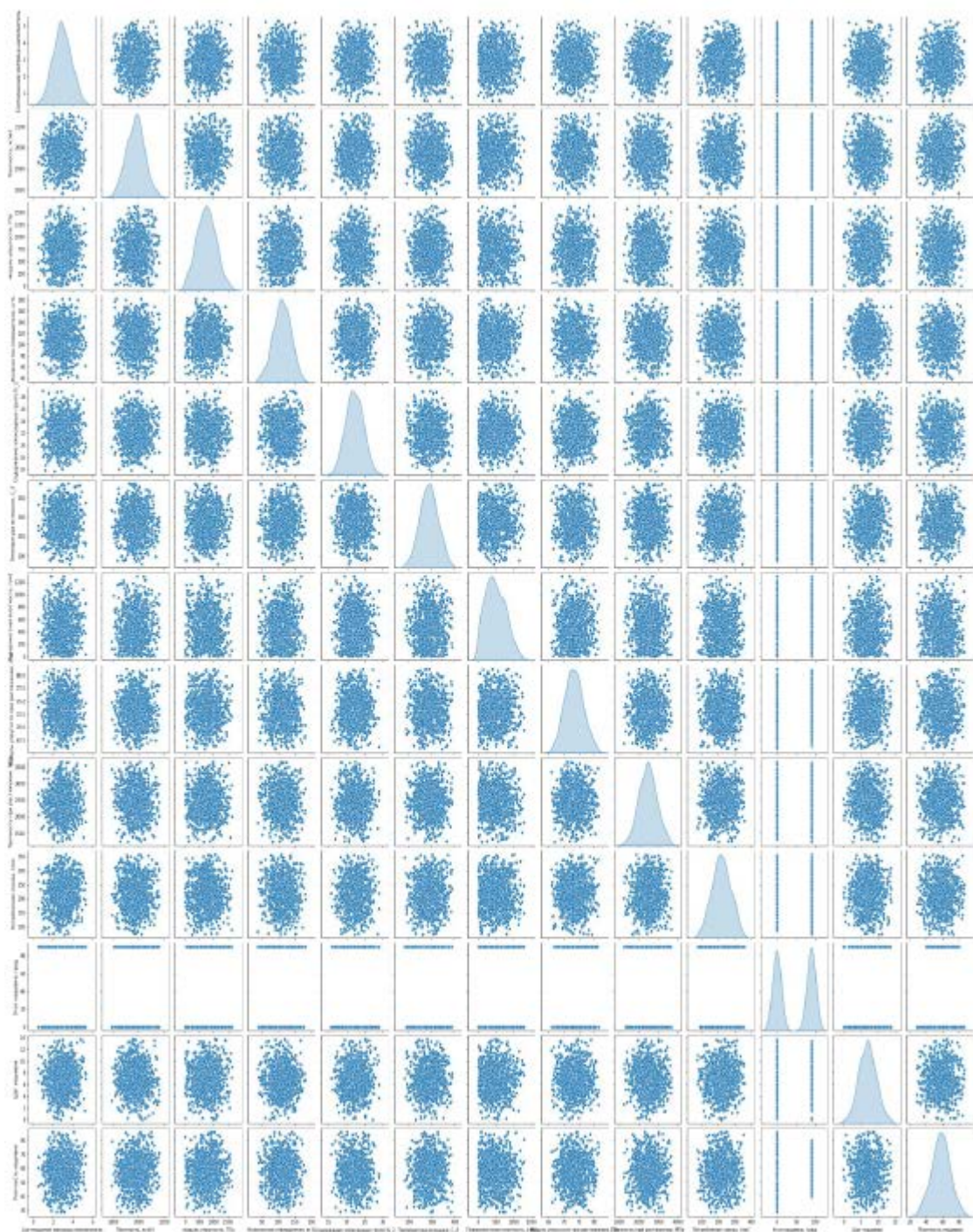


Рисунок 6- Гистограммы рассеяния после очистки выбросов

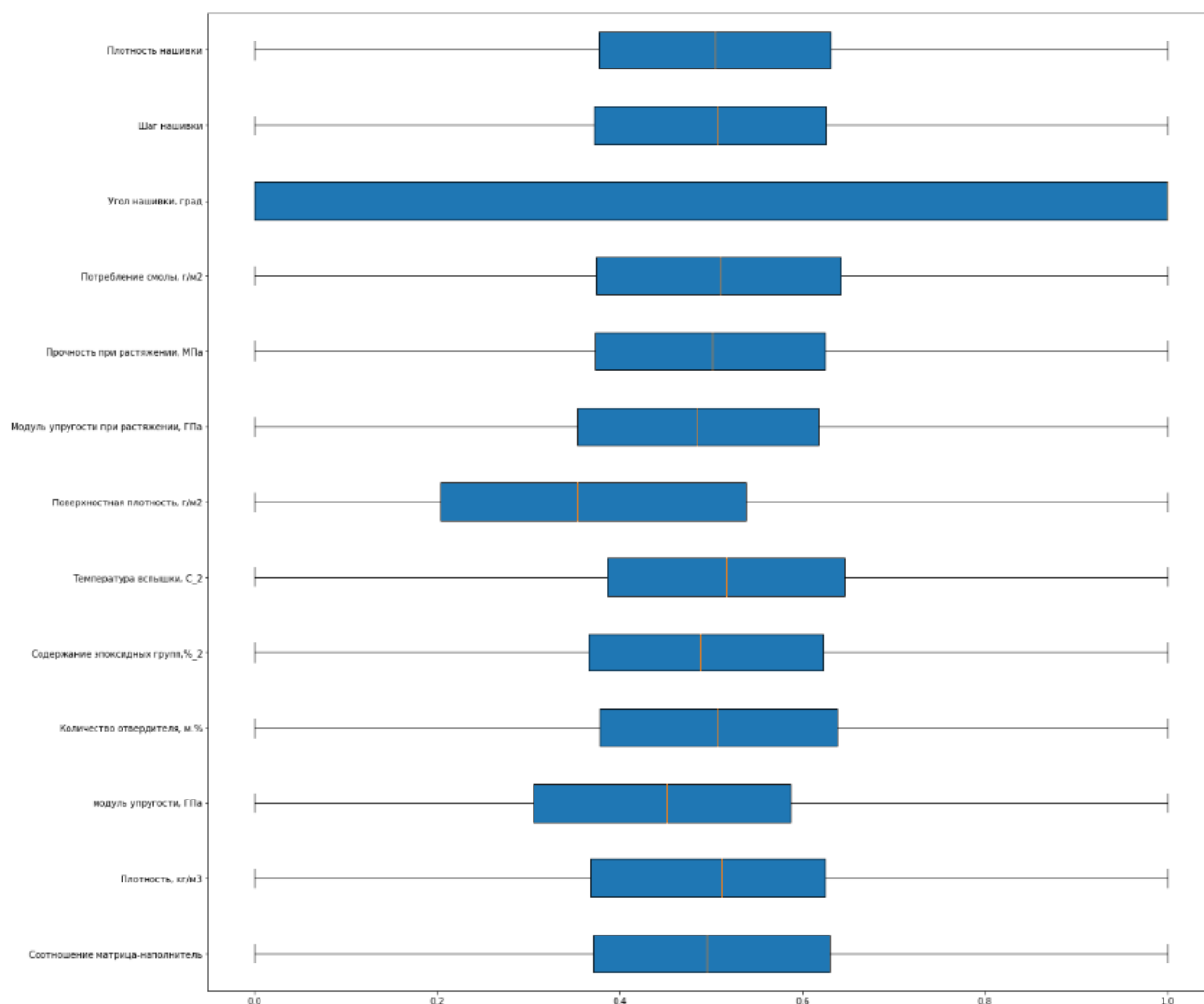


Рисунок 7 - Диаграммы размаха после очистки выбросов

После очистки выбросов по «правилу трех сигм» медианы и выборочные средние параметров «подтянулись» ближе друг к другу, за исключением параметра «Поверхностная плотность, г/м²», чья форма распределения, отличная от нормального, также сохранилась.

Построение матрицы корреляции и/или визуализация матрицы рассеивания позволяют получить представление о том, как попарно связаны между собой те или иные параметры.

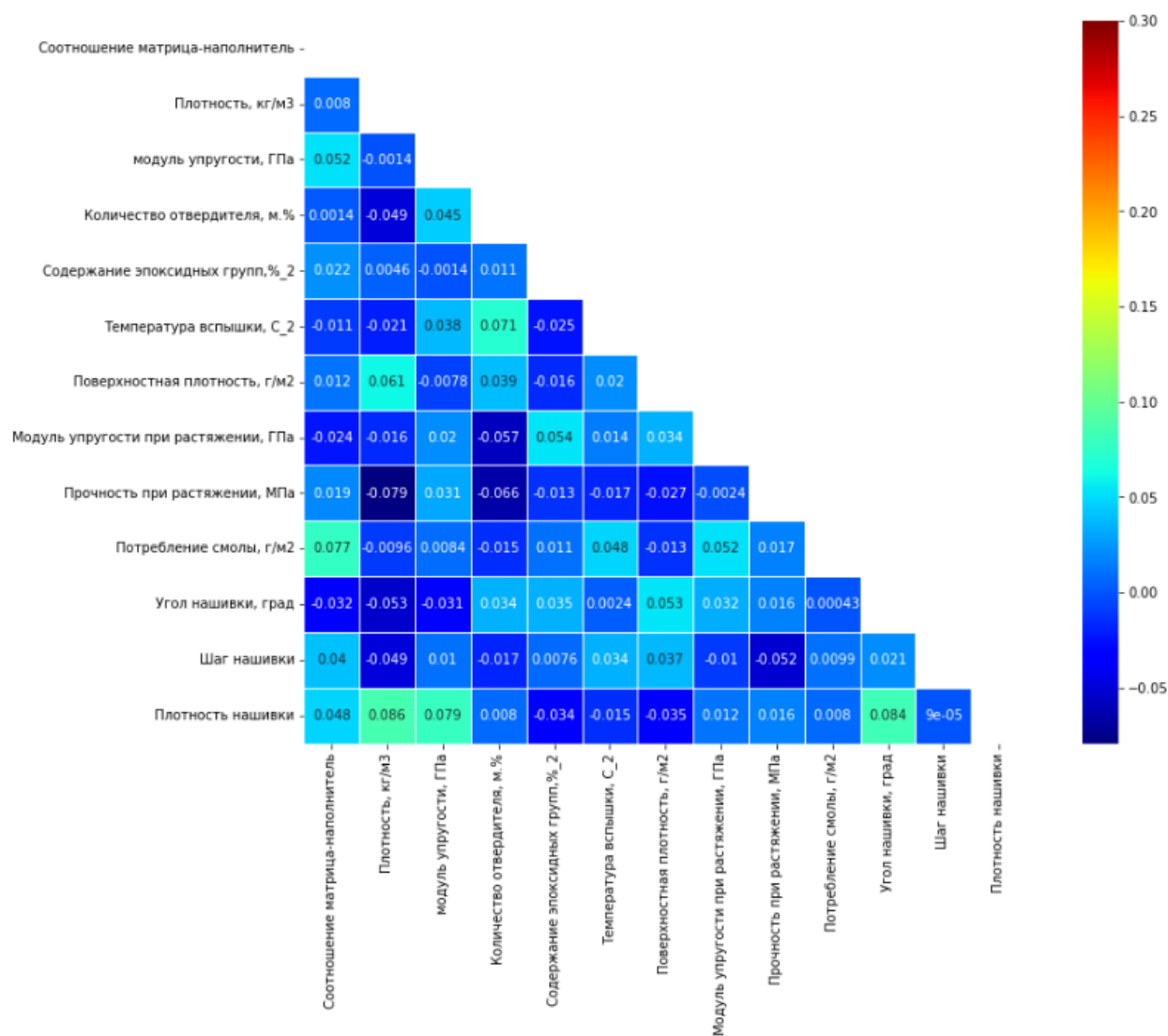


Рисунок 8 - Матрица корреляции датасета

Визуализация матрицы корреляции и матрицы рассеяния исходных данных данной задачи, представленная на рисунке 8, показывает около нулевую попарную корреляцию между параметрами и, соответственно, указывают на нелинейный характер связей между ними

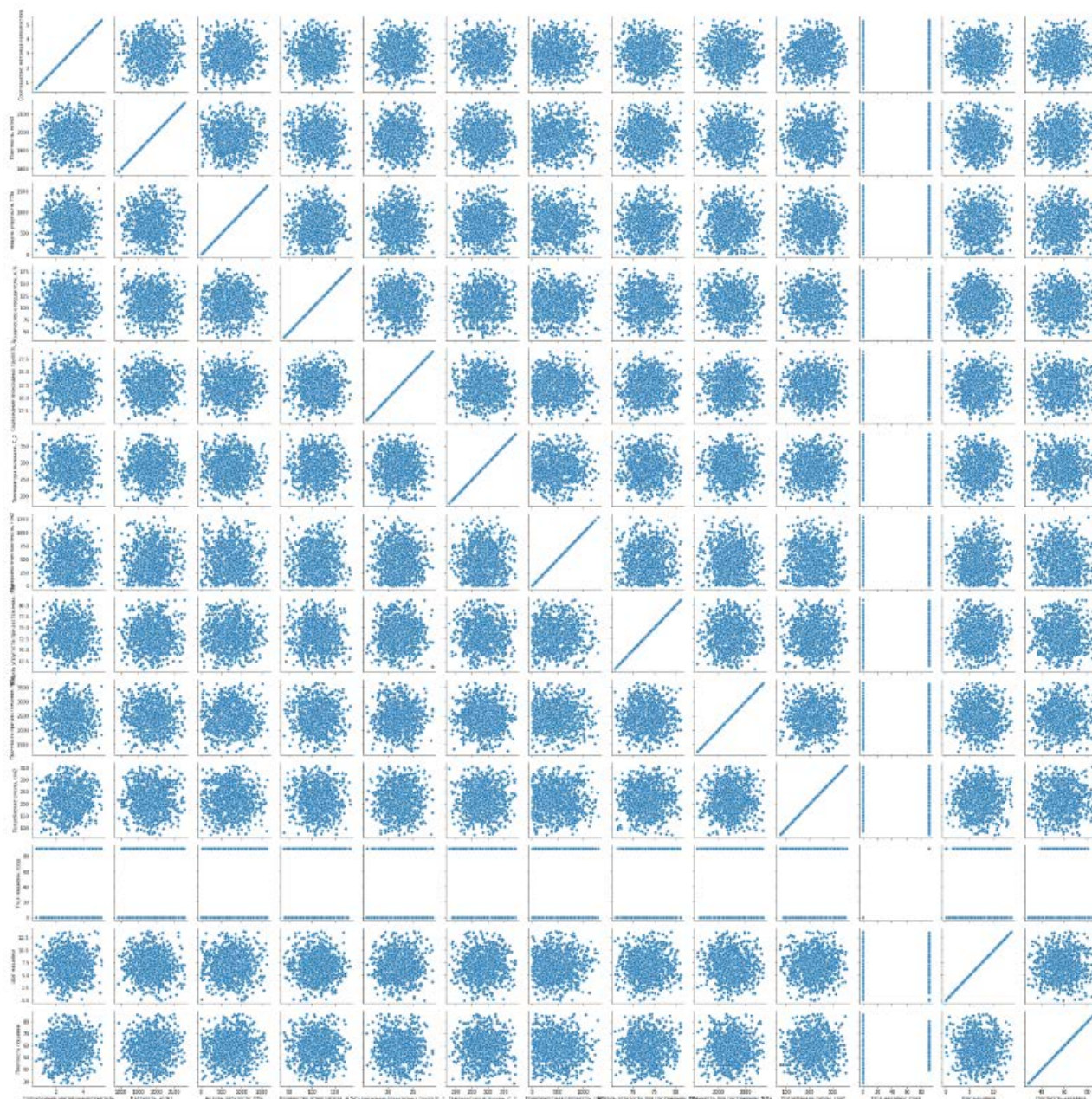


Рисунок 9 - Матрица попарной зависимости датасета

В условиях нелинейных зависимостей, для возможности «взглянуть» на данные в совокупности признаков и попытаться проследить какие-то взаимосвязи, следует обратиться к методам обучения на базе многообразий – класс оценщиков без учителя, нацеленных на описание наборов данных, как низко-размерных многообразий, вложенных в пространство большей размерности.

Корреляция между всеми параметрами очень близка к 0, корреляционные связи между переменными не наблюдаются.

2 Разработка моделей машинного обучения

2.1 Предобработка данных

Для предобработки данных использовались следующие процедуры:

1. Анализ датасета на пропуски, дубликаты и удаление пропусков, с помощью методов `info()`, `duplicated()` и `describe()`.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 936 entries, 1 to 1022
Data columns (total 13 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Соотношение матрица-наполнитель          936 non-null    float64
1   Плотность, кг/м3                          936 non-null    float64
2   модуль упругости, ГПа                     936 non-null    float64
3   Количество отвердителя, м.%               936 non-null    float64
4   Содержание эпоксидных групп,%_2          936 non-null    float64
5   Температура вспышки, С_2                  936 non-null    float64
6   Поверхностная плотность, г/м2             936 non-null    float64
7   Модуль упругости при растяжении, ГПа      936 non-null    float64
8   Прочность при растяжении, МПа             936 non-null    float64
9   Потребление смолы, г/м2                   936 non-null    float64
10  Угол нашивки, град                         936 non-null    float64
11  Шаг нашивки                               936 non-null    float64
12  Плотность нашивки                         936 non-null    float64
dtypes: float64(13)
memory usage: 102.4 KB
```

Рисунок 10 - Анализ датасета на пропуски

2. Удаление выбросов из датасета, замена данных, за пределами второго и третьего квантиля на пустые, затем удаление строк, содержащие пустые значения.

```

Соотношение матрица-наполнитель      6
Плотность, кг/м3                       9
модуль упругости, ГПа                  2
Количество отвердителя, м.%            14
Содержание эпоксидных групп,%_2       2
Температура вспышки, С_2               8
Поверхностная плотность, г/м2         2
Модуль упругости при растяжении, ГПа   6
Прочность при растяжении, МПа          11
Потребление смолы, г/м2                8
Угол нашивки, град                     0
Шаг нашивки                            4
Плотность нашивки                      21
dtype: int64

```

Рисунок 11 - Количество выбросов по каждому из столбцов

3. Нормализация данных с помощью метода MinMaxScaler и Normalizer из библиотеки sklearn.

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0	0.274768	0.651097	0.452951	0.079153	0.607435	0.509164	0.162230	0.272962	0.727777	0.514688	0.0	0.289334	0.546433
1	0.274768	0.651097	0.452951	0.630983	0.418887	0.583596	0.162230	0.272962	0.727777	0.514688	0.0	0.362355	0.319758
2	0.466552	0.651097	0.461725	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.0	0.362355	0.494123
3	0.465836	0.571539	0.458649	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.0	0.362355	0.546433
4	0.424236	0.332865	0.494944	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.0	0.362355	0.720799
...
917	0.361662	0.444480	0.560064	0.337550	0.333908	0.703458	0.161609	0.473553	0.472912	0.183151	1.0	0.660014	0.320103
918	0.607674	0.704373	0.272088	0.749605	0.294428	0.362087	0.271207	0.462512	0.461722	0.157752	1.0	0.768759	0.437468
919	0.573391	0.498274	0.254927	0.501991	0.623085	0.334063	0.572959	0.580201	0.587558	0.572648	1.0	0.301102	0.679468
920	0.662497	0.748688	0.454635	0.717585	0.267818	0.466417	0.496511	0.535317	0.341643	0.434855	1.0	0.458245	0.516112
921	0.684036	0.280923	0.255222	0.632264	0.888354	0.588206	0.587373	0.552644	0.668015	0.426577	1.0	0.441137	0.850430

922 rows x 13 columns

Рисунок 12 - Нормализация данных с помощью метода MinMaxScaler

Итоговая выборка представляет собой датасет с 922 уникальными строками.

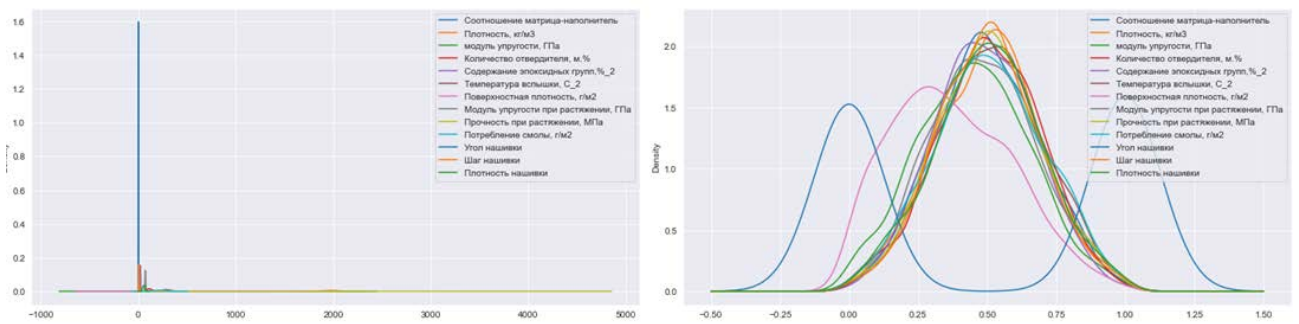


Рисунок 13 - Визуализированные данные до и после нормали

2.2 Разработка и обучение моделей

В данной работе разработка и обучение моделей машинного обучения осуществляется для двух выходных параметров: «Прочность при растяжении» и «Модуль упругости при растяжении». Для каждого признака построение моделей осуществляется отдельно.

Для признака «Прочность при растяжении» были разработаны и обучены и применены все методы. Порядок разработки модели для каждого параметра и для каждого выбранного метода можно разделить на следующие этапы:

- 1) Разделение нормализованных данных на обучающую и тестовую выборки (в соотношении 70 на 30%, согласно поставленной задаче)
- 2) Задание сетки гиперпараметров, по которым будет происходить оптимизация модели. В качестве параметра оценки выбран коэффициент детерминации (R^2).

Зададим сетку параметров, по которым будем оптимизировать модель

```
1 t_search_1 = {'weights': ['uniform', 'distance'],
2               'n_neighbors': list(np.linspace(5, 100, 10, dtype = int)),
3               'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
4               'leaf_size': list(np.linspace(10, 100, 10, dtype = int))}
5 #В качестве первой модели будем использовать метод ближайших соседей
6 clf_1 = KNeighborsRegressor()
7
8
```

Зададим сетку параметров, по которым будем оптимизировать модель

```
1 t_search_2 = {'weights': ['uniform', 'distance'],
2               'n_neighbors': list(np.linspace(5, 100, 10, dtype = int)),
3               'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
4               'leaf_size': list(np.linspace(10, 100, 10, dtype = int))}
5 #В качестве первой модели будем использовать метод ближайших соседей
6 clf_2 = KNeighborsRegressor()
7
```

Рисунок 13 - Пример определения сетки параметров для модели k ближайших соседей.

3) Оптимизация подбора гиперпараметров модели с помощью выбора по сетке и перекрестной проверки.

4) Подстановка оптимальных гиперпараметров в модель и обучение модели на тренировочных данных.

```
# Проведем поиск по сетке гиперпараметров с перекрестной проверкой, количество блоков равно 10 (cv = 10), для
# модели случайного леса - Random Forest Regressor - 2

parameters = { 'n_estimators': [200, 300],
                'max_depth': [9, 15],
                'max_features': ['auto'],
                'criterion': ['mse'] }

grid = GridSearchCV(estimator = rfr, param_grid = parameters, cv = 10)
grid.fit(x_train_1, y_train_1)

GridSearchCV(cv=10,
             estimator=RandomForestRegressor(max_depth=7, n_estimators=15,
                                             random_state=33),
             param_grid={'criterion': ['mse'], 'max_depth': [9, 15],
                         'max_features': ['auto'], 'n_estimators': [200, 300]})

grid.best_params_

{'criterion': 'mse',
 'max_depth': 15,
 'max_features': 'auto',
 'n_estimators': 200}

#Выводим гиперпараметры для оптимальной модели
print(grid.best_estimator_)
knr_upr = grid.best_estimator_
print(f'R2-score RFR для прочности при растяжении, МПа: {knr_upr.score(x_test_1, y_test_1).round(3)}')

RandomForestRegressor(criterion='mse', max_depth=15, n_estimators=200,
                      random_state=33)
R2-score RFR для прочности при растяжении, МПа: 0.963

#подставим оптимальные гиперпараметры в нашу модель случайного леса
rfr_grid = RandomForestRegressor(n_estimators = 200, criterion = 'mse', max_depth = 15, max_features = 'auto')
#Обучаем модель
rfr_grid.fit(x_train_1, y_train_1)

predictions_rfr_grid = rfr_grid.predict(x_test_1)
#Оцениваем точность на тестовом наборе
mae_rfr_grid = mean_absolute_error(predictions_rfr_grid, y_test_1)
mae_rfr_grid

67.60356685553326

new_row_in_mae_df = {'Perpeccop': 'RandomForest_GridSearchCV', 'MAE': mae_rfr_grid}
mae_df = mae_df.append(new_row_in_mae_df, ignore_index=True)
```

Рисунок 1- Поиск гиперпараметров

Модель после настройки гиперпараметров показала результат немного лучше. Однако, ниже, чем базовая модель. Прочность при растяжении и модуль упругости не имеет линейной зависимости. Все использованные модели не справились с задачей. Результат неудовлетворительный. Свойства композитных материалов в первую очередь зависят от используемых материалов.

2.3 Тестирование моделей

После обучения моделей была проведена оценка точности этих моделей на обучающей и тестовых выборках. В качестве параметра оценки модели использовалась средняя абсолютная ошибка (MAE). Для большей наглядности результатов работы модели на тестовых данных, были построены диаграммы рассеяния тестовых данных (реальные данные) и значений, полученных в качестве прогноза.

Подставляем оптимальные гиперпараметры в модель

```
1 model_base_1 = KNeighborsRegressor(algorithm='brute', leaf_size=10, n_neighbors=100, weights='distance')
2 #Обучаем модель
3 model_base_1.fit(Xtrain1_1,Ytrain1_1)
4 #Оцениваем точность на тренировочном наборе
5 base_accuracy = evaluate(model_base_1, Xtrain1_1,Ytrain1_1)
6 #Оцениваем точность на тестовом наборе
7 base_accuracy = evaluate(model_base_1, Xtest1_1,Ytest1_1)
```

Средняя абсолютная ошибка: 0.0000

Средняя абсолютная ошибка: 0.1546

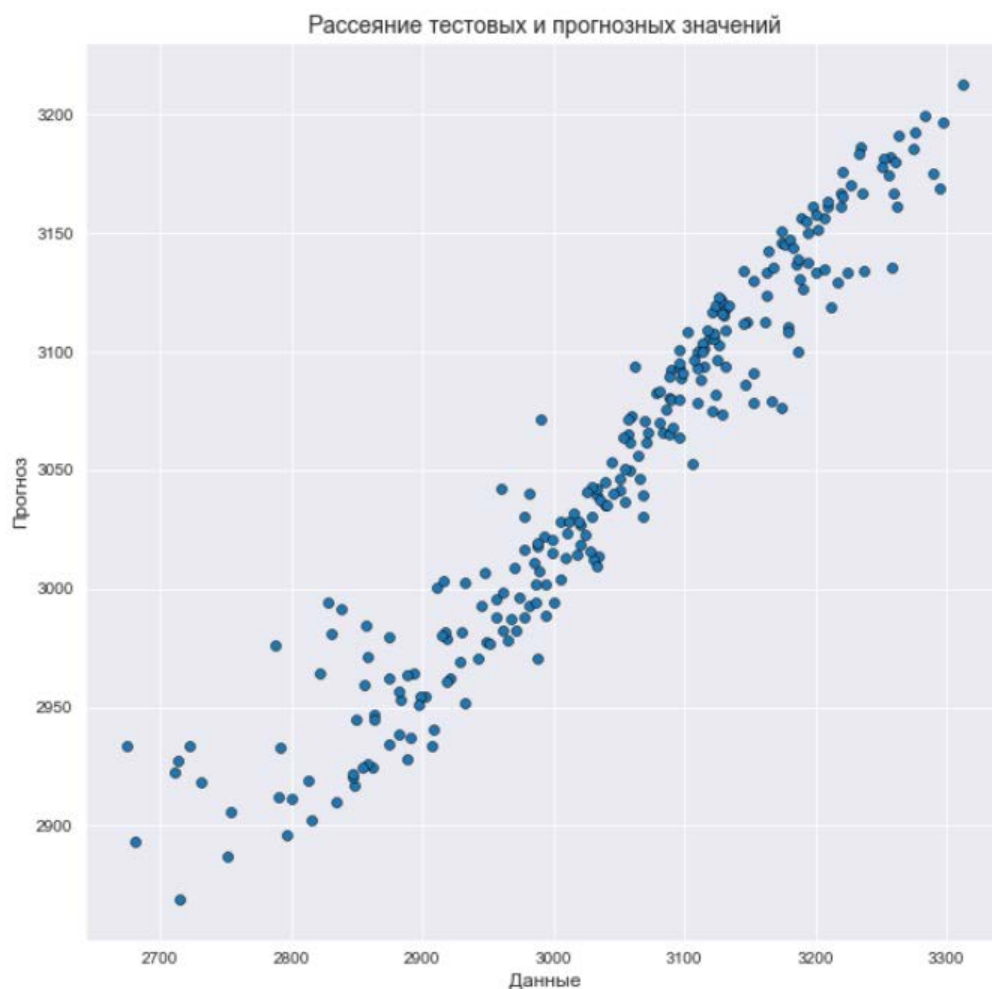
Подставляем оптимальные гиперпараметры в модель

```
1 model_base_2 = KNeighborsRegressor(algorithm='brute', leaf_size=10, n_neighbors=100, weights='distance')
2 #Обучаем модель
3 model_base_2.fit(Xtrain2_1,Ytrain2_1)
4 #Оцениваем точность на тренировочном наборе
5 base_accuracy = evaluate(model_base_2, Xtrain2_1,Ytrain2_1)
6 #Оцениваем точность на тестовом наборе
7 base_accuracy = evaluate(model_base_2, Xtest2_1,Ytest2_1)
```

Средняя абсолютная ошибка: 0.0000

Средняя абсолютная ошибка: 0.0202

Рисунок 14 - Результаты модели k ближайших соседей для параметра «Прочность при растяжении»



	Данные	Прогноз
0	2731.187634	2918.466881
1	3005.029020	3003.927511
2	3095.598833	3100.544471
3	2910.996260	3000.398009
4	3118.143272	3105.850278
...
247	2985.209632	3010.800784
248	3039.836511	3045.180054
249	2874.154431	2962.153945
250	3067.883163	3030.528017
251	2987.594837	3019.108070

252 rows × 2 columns

Рисунок 15 - Результаты работы модели по оценке значений параметра на основе тестовых данных (Прочность при растяжении)

Оцениваем точность на тренировочном наборе

```
1 base_accuracy = evaluate_2(model_base_11, Xtrain1_2, Ytrain1_2)
2 #Оцениваем точность на тестовом наборе
3 base_accuracy = evaluate_2(model_base_11, Xtest1_2, Ytest1_2)
```

Средняя абсолютная ошибка: 0.1288

Средняя абсолютная ошибка: 0.1655

```
1 Подставляем оптимальные гиперпараметры в модель
```

```
1 model_base_22 = GradientBoostingRegressor(loss='lad', max_depth=2)
2 #Обучаем модель
3 model_base_22.fit(Xtrain2_2,np.ravel(Ytrain2_2))
```

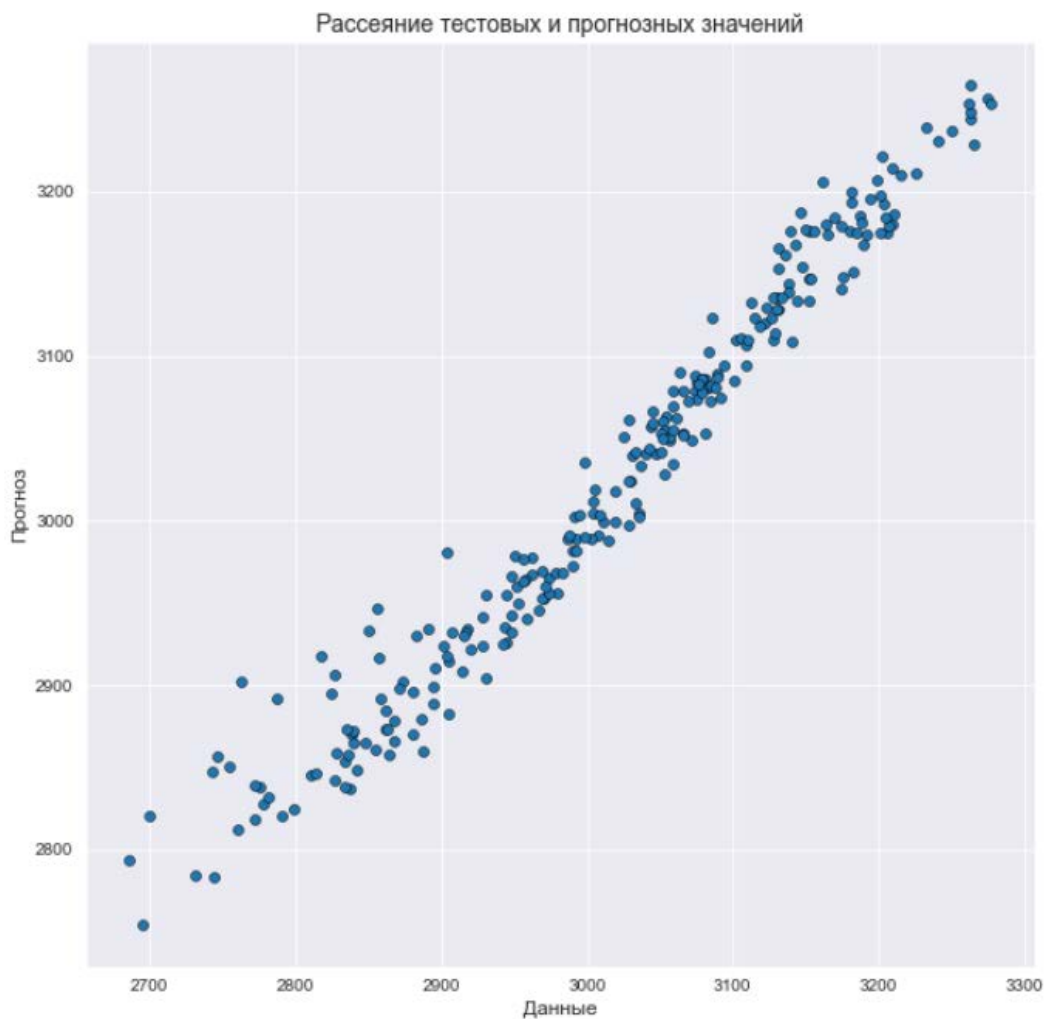
GradientBoostingRegressor(loss='lad', max_depth=2)

```
1 #Оцениваем точность на тренировочном наборе
2 base_accuracy = evaluate_2(model_base_22, Xtrain2_2, Ytrain2_2)
3 #Оцениваем точность на тестовом наборе
4 base_accuracy = evaluate_2(model_base_22, Xtest2_2, Ytest2_2)
```

Средняя абсолютная ошибка: 0.0050

Средняя абсолютная ошибка: 0.0085

Рисунок 16 - Результаты модели повышения градиента для параметра «Прочность при растяжении»



	Данные	Прогноз
0	3075.513003	3084.535920
1	3019.465343	2999.180338
2	3058.533799	3034.767054
3	3129.901407	3136.223928
4	3083.324200	3102.851986
...
247	2685.747892	2793.348055
248	3028.420736	3024.451303
249	3152.256939	3133.135592
250	3032.628666	3041.220577
251	3051.973752	3049.619002

252 rows × 2 columns

Рисунок 17 - Результаты работы модели повышения градиента для параметра «Прочность при растяжении»

```
1 best_estimator = model11.best_estimator_
2 #Выводим гиперпараметры для оптимальной модели
3 print(best_estimator)
4 #выводим точность оптимального трейнера
5 print(model11.best_score_)
```

```
LinearRegression()
-0.0261111116626460085
```

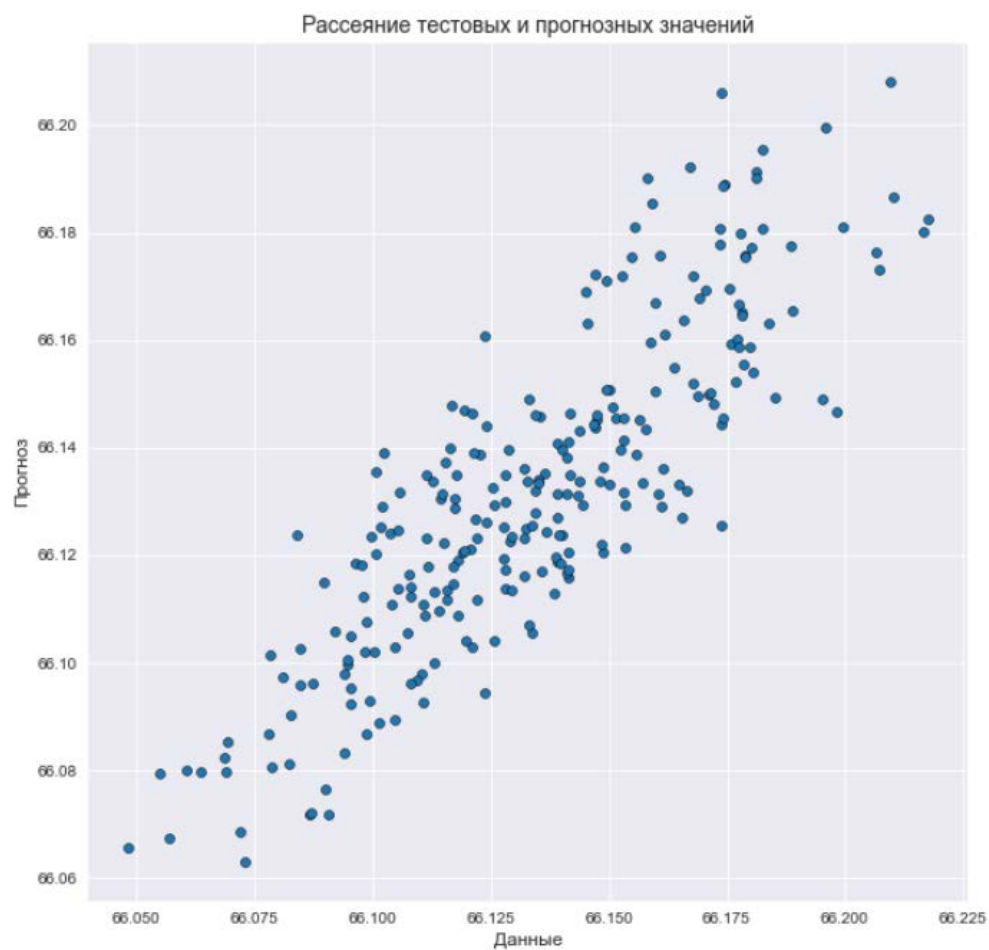
```
1 #Подставляем оптимальные гиперпараметры в модель
2 model_base11 = LinearRegression()
3 #Обучаем модель
4 model_base11.fit(Xtrain11,Ytrain11)
```

```
LinearRegression()
```

```
1 #Оцениваем точность на тренировочном наборе
2 base_accuracy = evaluate(model_base11, Xtrain11,Ytrain11)
3 #Оцениваем точность на тестовом наборе
4 base_accuracy = evaluate(model_base11, Xtest11, Ytest11)
```

```
Средняя абсолютная ошибка: 0.1608
Средняя абсолютная ошибка: 0.1509
```

Рисунок 18 Результаты модели LinearRegression для параметра «Модуль упругости при растяжении»



	Данные	Прогноз
0	66.104819	66.103029
1	66.160714	66.175785
2	66.159139	66.185353
3	66.127873	66.125390
4	66.177853	66.179916
...
247	66.210186	66.186612
248	66.111366	66.123212
249	66.129569	66.123408
250	66.135667	66.117103
251	66.139894	66.118540

252 rows × 2 columns

Рисунок 19 - Результаты работы модели LinearRegression для параметра «Модуль упругости при растяжении»

```

1 best_estimator = model111.best_estimator_
2 #Выводим гиперпараметры для оптимальной модели
3 print(best_estimator)
4 #Выводим точность оптимального трейнера
5 print(model111.best_score_)

```

```

SVR(kernel='linear')
-0.0304111012748157

```

```

1 #Подставляем оптимальные гиперпараметры в модель
2 model_base111 = SVR(kernel='linear')
3 #Обучаем модель
4 model_base111.fit(Xtrain111,np.ravel(Ytrain111))

```

```

SVR(kernel='linear')

```

```

1 #Оцениваем точность на тренировочном наборе
2 base_accuracy = evaluate_2(model_base111, Xtrain111, Ytrain111)
3 #Оцениваем точность на тестовом наборе
4 base_accuracy = evaluate_2(model_base111,Xtest111, Ytest111)

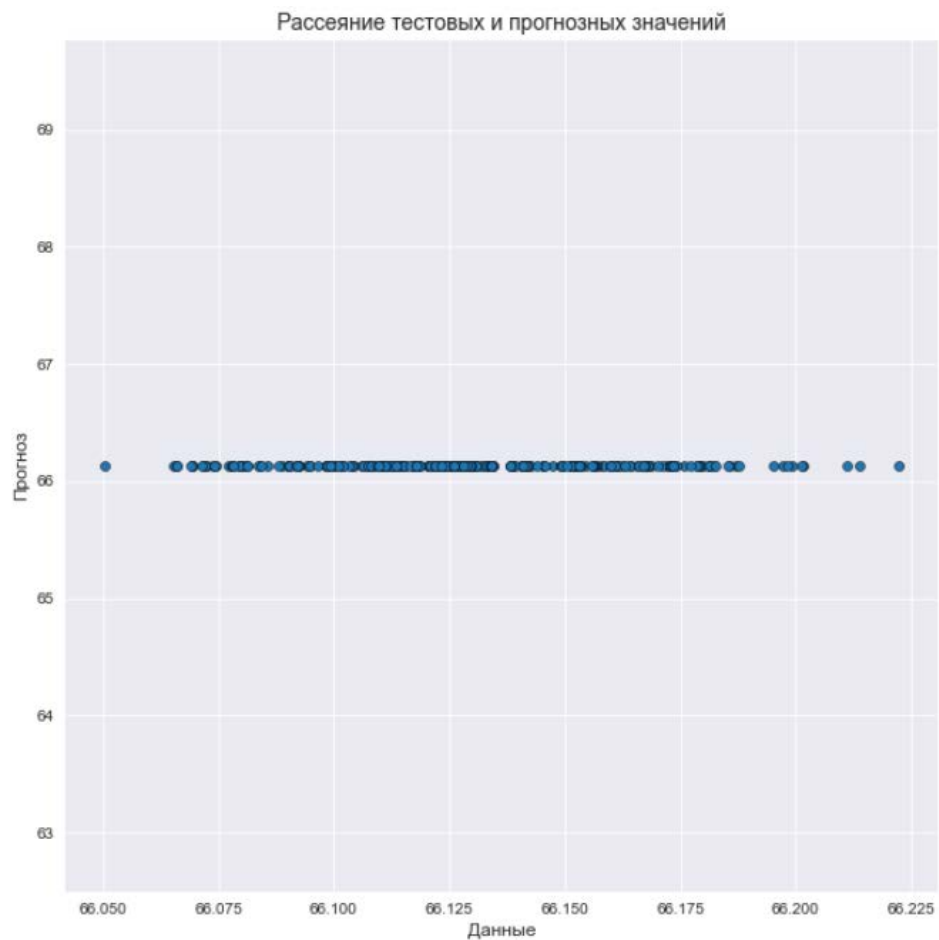
```

```

Средняя абсолютная ошибка: 0.1581
Средняя абсолютная ошибка: 0.1585

```

Рисунок 20 - Результаты модели регрессии опорных векторов (SVR) для параметра «Модуль упругости при растяжении»



	Данные	Прогноз
0	66.123800	66.132373
1	66.151416	66.132373
2	66.167024	66.132373
3	66.161214	66.132373
4	66.173399	66.132373
...
247	66.126000	66.132373
248	66.150265	66.132373
249	66.081114	66.132373
250	66.201315	66.132373
251	66.123064	66.132373

252 rows × 2 columns

Рисунок 21 - Результаты работы регрессии опорных векторов (SVR) для параметра «Модуль упругости при растяжении»

2.5 Разработка нейронной сети

Обучение нейронной сети — это такой процесс, при котором происходит подбор оптимальных параметров модели, с точки зрения минимизации функционала ошибки. Начнём строить нейронную сеть с помощью класса `keras.Sequential`.

```
# Сформируем входы и выход для модели

tv = df['Соотношение матрица-наполнитель']
tr_v = df.loc[:, df.columns != 'Соотношение матрица-наполнитель']

# Разбиваем выборки на обучающую и тестовую
x_train, x_test, y_train, y_test = train_test_split(tr_v, tv, test_size = 0.3, random_state = 14)

# Нормализуем данные

x_train_n = tf.keras.layers.Normalization(axis = -1)
x_train_n.adapt(np.array(x_train))
```

Рисунок 21 - создание нейронной сети при второй попытке

Определим параметры, поищем оптимальные параметры, посмотрим на результаты. С помощью `KerasClassifier` выйдем на наилучшие параметры для нашей нейронной сети и построим окончательную нейросеть.

Обучим и оценим модель, посмотрим на потери, зададим функцию для визуализации факт/прогноз для результатов моделей.

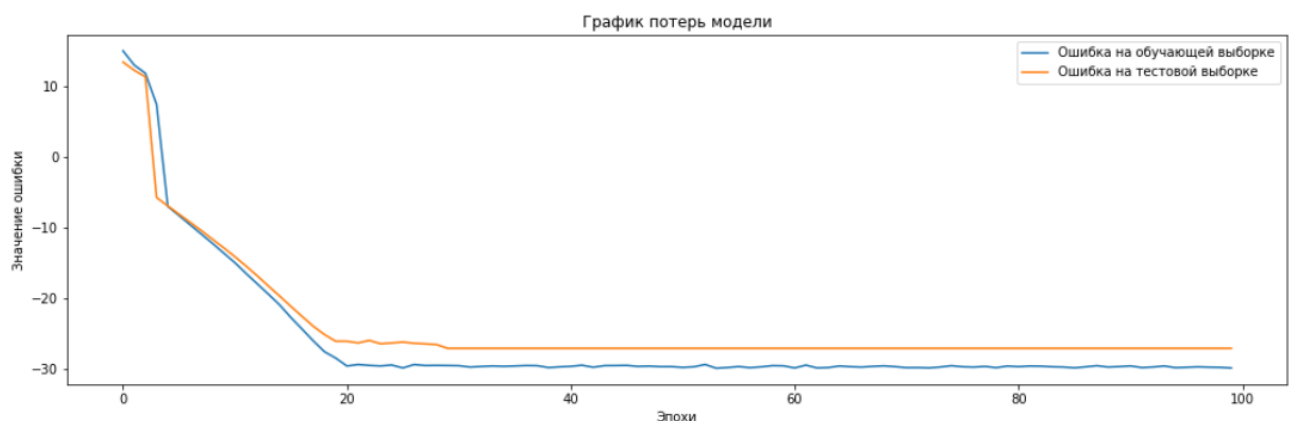


Рисунок 2 - График потерь модели при второй попытке



Рисунок 3 - тестовые и прогнозные значения модели при второй попытке

Для рекомендации соотношения «матрица-наполнитель» разработана простая модель глубокого обучения с помощью библиотеки Keras.

Модель состоит из трех скрытых уровней. Первый уровень содержит 64 нейрона, что немногим более чем в три раза превышает объем входных данных (10 входных переменных). Последующие скрытые уровни содержат 64 и 1 нейрон. Снижение числа нейронов на каждом уровне сжимает информацию, которую сеть обработала на предыдущих уровнях.

Скрытые уровни нейронной сети трансформируются функциями активации. Эти функции являются важными элементами сетевой инфраструктуры, так как они вносят в систему нелинейность.

Для эксперимента были выбраны три функции активации:

1. tanh (арктангенс),
2. relu (выпрямленная линейная единица),
3. sigmoid (сигмоида $1/(1+\exp(-x))$)

```

1 def build_model1():
2     model1=models.Sequential()
3     model1.add(layers.Dense(64, activation='tanh', input_shape=(X1trn1.shape[1],)))
4     model1.add(layers.Dense(64, activation='tanh'))
5     model1.add(layers.Dense(1))
6     model1.compile(optimizer='rmsprop', loss='mse', metrics=['mae'])
7     return model1

```

Рисунок 22 - Архитектура нейронной сети

Далее была определена функция стоимости сети, которая используется для генерации оценки отклонения между прогнозами сети и реальными резуль-

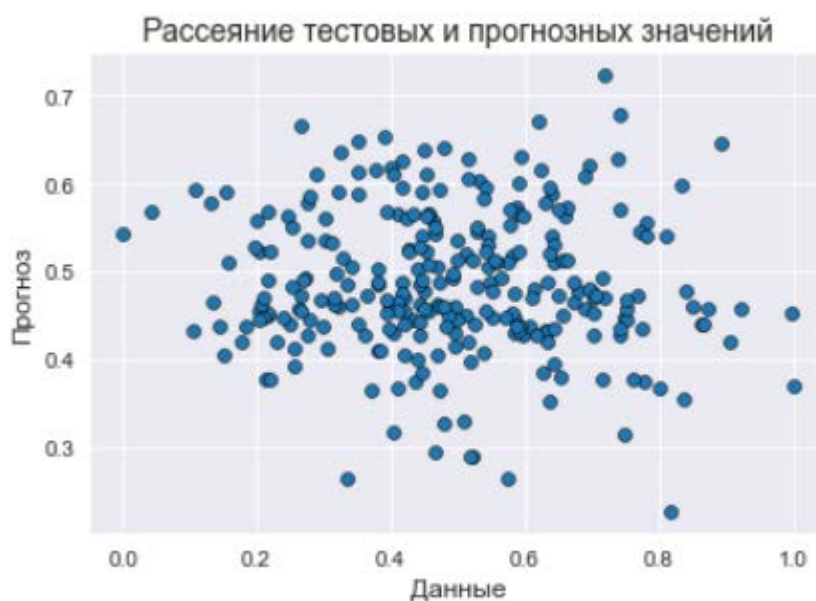
татами наблюдений в ходе обучения. Для решения проблем с регрессией используют функцию средней квадратичной ошибки (Mean Squared Error). Данная функция вычисляет среднее квадратичное отклонение между предсказаниями и целями.

В качестве оптимизатора использовался RMSprop-оптимизатор, алгоритм которого похож на метод градиентного спуска с импульсом. Оптимизатор RMSprop ограничивает колебания в вертикальном направлении

После обучения для модели нейронной сети была определена средняя абсолютная ошибка на тестовом наборе данных.

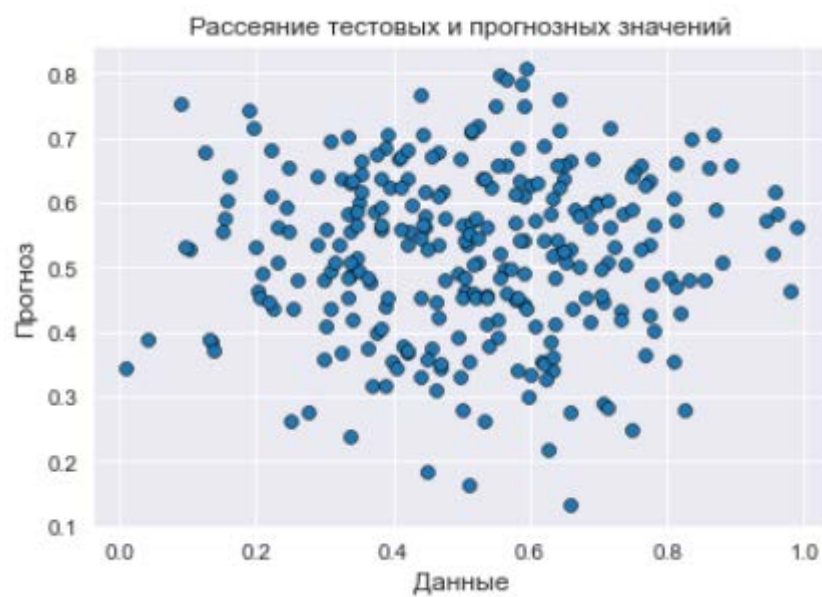
Средняя абсолютная ошибка: 0.171.

На рисунках 21-23 представлены результаты прогноза модели на тестовых данных, по аналогии с результатами для моделей машинного обучения, описанных в предыдущем разделе.



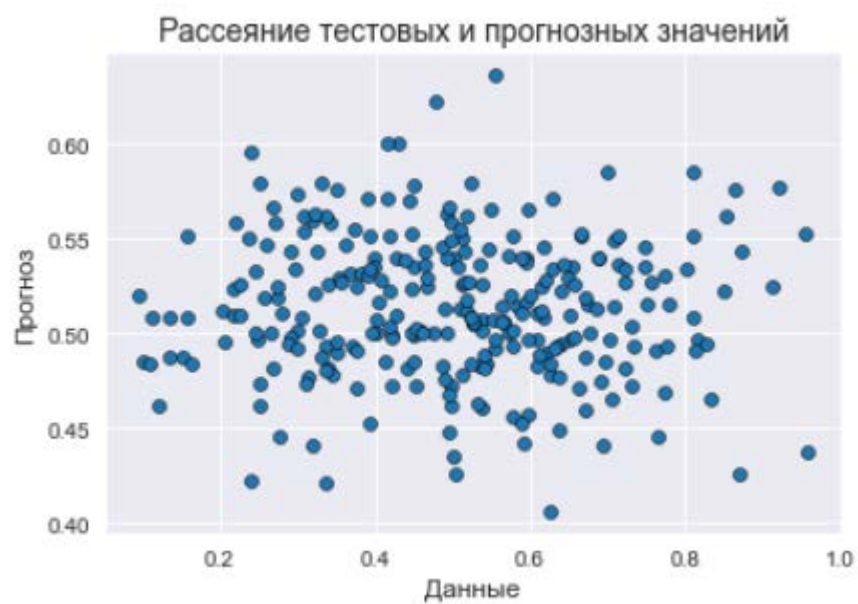
	Данные	Прогноз
0	0.439831	0.501854
1	0.638328	0.539216
2	0.274799	0.427661
3	0.381253	0.410576
4	0.455313	0.564241
...
272	0.648514	0.571335
273	0.419445	0.405875
274	0.341934	0.505846
275	0.526656	0.550951
276	0.590948	0.601100

Рисунок 23 - Прогнозные данные для модели с функцией tanh



	Данные	Прогноз
0	0.387258	0.438928
1	0.588735	0.784112
2	0.504202	0.463572
3	0.528359	0.638538
4	0.505854	0.534838
...
272	0.739697	0.503467
273	0.524701	0.544280
274	0.198417	0.533502
275	0.409386	0.625002
276	0.748254	0.639669

Рисунок 24 - Прогнозные данные для модели с функцией relu



	Данные	Прогноз
0	0.319138	0.441282
1	0.366300	0.531129
2	0.397774	0.540102
3	0.722423	0.481907
4	0.666214	0.550901
...
272	0.589733	0.510593
273	0.494837	0.467828
274	0.223971	0.509313
275	0.298759	0.492468
276	0.555986	0.496744

Рисунок 25 - Прогнозные данные для модели с функцией sigmoid

3. Разработка приложения

При разработке приложения нам понадобилась библиотека Flask, при помощи которой мы смогли создать и реализовать функционал нашего приложения. Помимо библиотеки Flask мы ещё использовали библиотеку tensorflow и pickle для внедрения наших моделей в приложение.

Приложение разработано с Web-интерфейсом и позволяет решать задачи прогнозирования целевой переменной на основе входных данных.

Прогнозирование для "соотношения матрица-наполнитель"

Плотность, кг/м3 (1700...2300)	1880.0
Модуль упругости, ГПа (2...2000)	622.0
Количество отвердителя, м. % (17...200)	111.86
Содержание эпоксидных групп, % 2 (14...34)	22.2678571428571
Температура всплытия, С 2 (100...414)	284.615384615384
Поверхностная плотность, г/м2 (0.6...1400)	470.0
Модуль упругости при растяжении, ГПа (64...83)	73.3333333333333
Прочность при растяжении, МПа (1036...3849)	2455.55555555555
Потребление смолы, г/м2 (33...414)	220.0
Угол нашивки, град (0...90)	90.0
Шаг нашивки (0...15)	4.0
Плотность нашивки (0...104)	60.0

Отправить

Входные переменные:

Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м. %	Содержание эпоксидных групп, % 2	Температура всплытия, С 2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0 1880.0	622.0	111.86	22.267857	284.615385	470.0	73.333333	2455.555556	220.0	90.0	4.0	60.0

Результат модели:

Соотношение матрица-наполнитель [2.9509137]
--

Рисунок 26 - Web-интерфейс приложения

После ввода всех переменных выдается прогнозное значение параметра «Соотношение матрица-наполнитель», сформированного моделью нейронной сети.

4. Создание удаленного репозитория и загрузка файлов в него

Страница создана на GitHub.

Адрес страницы: https://github.com/ComrGarry/DS_PahomovIAB

Репозитории находятся: файлы тетрадок Юпитера, наборы данных, модели, приложение, ВКР в текстовом формате.

Заключение

Теоретически разработанный метод определения надёжности изделий из композиционных материалов, основанный на использовании статистически достоверных характеристик материалов, полученных физическим и вычислительным экспериментом, позволяет оценивать уровень надёжности изделий как в отдельных точках, так и по всему объёму в целом.

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

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

Вывод: текущим набором алгоритмов задача не решается, возможно, решается трудно или не решается совсем.

Список использованных источников и 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. — (Высшее образование). — Текст : непосредственный.
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. ГрасД. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2021. - 416 с.: ил.
4. Абросимов Н.А.: Методика построения разрешающей системы уравнений динамического деформирования композитных элементов конструкций (Учебно-методическое пособие), ННГУ, 2010
5. Абу-Хасан Махмуд, Масленникова Л. Л.: Прогнозирование свойств композиционных материалов с учётом наноразмера частиц и акцепторных свойств катионов твёрдых фаз, статья 2006 год
6. Бизли Д. Python. Подробный справочник: учебное пособие. – Пер. с англ. – СПб.: Символ-Плюс, 2010. – 864 с., ил.
7. Гафаров, Ф.М., Галимянов А.Ф. Искусственные нейронные сети и приложения: учеб. пособие /Ф.М. Гафаров, А.Ф. Галимянов. – Казань: Издательство Казанского университета, 2018. – 121 с.
8. Грас Д. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2021. - 416 с.: ил.
9. Документация по языку программирования python: – Режим доступа: <https://docs.python.org/3.8/index.html>.
- 10 Документация по библиотеке numpy: – Режим доступа: <https://numpy.org/doc/1.22/user/index.html#user>.
11. Документация по библиотеке pandas: – Режим доступа: https://pandas.pydata.org/docs/user_guide/index.html#user-guide.

12. Документация по библиотеке matplotlib: – Режим доступа: <https://matplotlib.org/stable/users/index.html>.

13. Документация по библиотеке seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>.

14. Документация по библиотеке sklearn: – Режим доступа: https://scikit-learn.org/stable/user_guide.html.

15. Документация по библиотеке keras: – Режим доступа: <https://keras.io/api/>.

16. Руководство по быстрому старту в flask: – Режим доступа: <https://flask-russian-docs.readthedocs.io/ru/latest/quickstart.html>.

17. Loginom Вики. Алгоритмы: – Режим доступа: <https://wiki.loginom.ru/algorithms.html>.

18. Andre Ye. 5 алгоритмов регрессии в машинном обучении, о которых вам следует знать: – Режим доступа: <https://habr.com/ru/company/vk/blog/513842/>.

19. Alex Maszański. Метод k-ближайших соседей (k-nearest neighbour): – Режим доступа: <https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19>.

20. Yury Kashnitsky. Открытый курс машинного обучения. Тема 3. Классификация, деревья решений и метод ближайших соседей: – Режим доступа: <https://habr.com/ru/company/ods/blog/322534/>.

21. Yury Kashnitsky. Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес: – Режим доступа: <https://habr.com/ru/company/ods/blog/324402/>.

22. Alex Maszański. Машинное обучение для начинающих: алгоритм случайного леса (Random Forest): – Режим доступа: <https://proglib.io/p/mashinnoe-obuchenie-dlya-nachinayushchih-algoritm-sluchaynogo-lesa-random-forest-2021-08-12>

23. Реутов Ю.А.: Прогнозирование свойств полимерных композиционных материалов и оценка надёжности изделий из них, Диссертация на соискание учёной степени кандидата физико-математических наук, Томск 2016.

24. Роббинс, Дженнифер. HTML5: карманный справочник, 5-е издание.: Пер. с англ. - М.: ООО «И.Д. Вильямс»: 2015. - 192 с.: ил.
25. Скиена, Стивен С. С42 Наука о данных: учебный курс.: Пер. с англ. - СПб.: ООО "Диалектика", 2020. - 544 с. : ил.
26. Справочник по композиционным материалам: в 2 - х кн. Кн. 2 / Под ред. Дж. Любина; Пер. с англ. Ф. Б. Геллера, М. М. Гельмонта; Под ред. Б. Э. Геллера - М.: Машиностроение, 1988. - 488 с. : ил;
27. Траск Эндрю. Грокаем глубокое обучение. – СПб.: Питер, 2019. – 352 с.: ил.
28. Чун-Те Чен и Грейс Х. Гу. Машинное обучение для композитных материалов (март 2019г.) – Режим доступа: <https://www.cambridge.org/core/journals/mrs-communications/article/machine-learning-for-composite-materials/F54F60AC0048291BA47E0B671733ED15>. (дата обращения 02.06.2022)