

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»

Тема:

**Прогнозирование конечных свойств новых материалов
(композиционных материалов)**

Слушатель

Несмиянов Сергей Витальевич

Москва, 2023

Содержание

Содержание.....	2
Введение.....	3
1 Аналитическая часть.....	5
1.1 Постановка задачи	5
1.2 Описание используемых методов	10
1.2.1 Линейная регрессия	11
1.2.2 Регрессия лассо	11
1.2.3 Метод опорных векторов	12
1.2.4 Дерево решений	13
1.2.5 Случайный лес.....	13
1.2.6 Градиентный бустинг	14
1.2.7 Метод k-ближайших соседей	15
1.3 Разведочный анализ данных	16
2 Практическая часть	19
2.1 Предобработка данных	19
2.1.1 Для прогнозирования модуля упругости при растяжении	21
2.1.2 Для прогнозирования прочности при растяжении	22
2.1.3 Для прогнозирования соотношения матрица-наполнитель	23
2.2 Разработка и обучение моделей.....	24
2.3 Создание нейронной сети, рекомендующей соотношение матрица-наполнитель	29
2.4 Оценка точности работы моделей	34
Заключение	37
Библиографический список.....	39

Введение

Композиционный или композитный материал (далее – композит, КМ) – многокомпонентный материал, изготовленный природой или человеком из двух или более компонентов с существенно различающимися физическими и химическими свойствами, которые, в сочетании приводят к появлению нового материала с характеристиками, отличными от характеристик исходных компонентов.

Композит как правило состоит из матрицы и наполнителя, который обычно выполняет функцию армирования (по аналогии с арматурой в таком композиционном строительном материале, как железобетон). Сочетание разных компонентов позволяет улучшить характеристики материала и делает его одновременно лёгким и прочным. Многие композиты превосходят традиционные материалы и сплавы по своим механическим свойствам. Так, на примере железобетона: бетон отлично сопротивляется сжатию и хуже – растяжению и изгибу, а стальная арматура внутри бетона компенсирует его недостатки, формируя тем самым новые, уникальные свойства. При этом стоит отметить, что бетон в свою очередь сам является истинным композитом так как он состоит из гравия и песка, которые связаны между собой при помощи цемента, а металлическая арматура обычно добавляется для усиления прочности бетона. Таким образом, при изменении состава матрицы и наполнителя, их соотношения и ориентации возможно получить широкий спектр материалов с требуемым набором свойств.

В настоящее время остается актуальной проблема прогнозирования свойств конечного композитного материала даже при знании свойств исходных компонентов. Одним из решений этой проблемы может выступать моделирование репрезентативного элемента композитного объёма на основе данных о свойствах входящих компонентов (связующего и армирующего компонента).

В ходе выполнения работы предполагается разработка моделей, способных прогнозировать модули упругости и прочности при растяжении, а также создание нейронной сети, предлагающей соотношение «матрица – наполнитель». Созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

1 Аналитическая часть

1.1 Постановка задачи

Задачей проведения настоящей работы является исследование предложенных датасетов с информацией о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.). с целью разработки модели для прогноза модуля упругости при растяжении, прочности при растяжении и соотношения «матрица-наполнитель».

Представленный датасет состоит из двух файлов: X_br.xlsx (с данными о параметрах базальтопластика), состоящий из 1024 строк и 11 столбцов, и X_npr.xlsx (с данными нашивок из углепластика), состоящий из 1041 строки и 4 столбцов.

В соответствии с заданием указанные файлы требуют объединения с типом INNER по индексу. После осуществления объединения часть строк из файла X_npr была отброшена, а дальнейшие исследования проводятся с объединенным датасетом, состоящим из 13 признаков и 1023 строк или объектов.

Описание признаков объединенного датасета приведено в таблице 1. Все признаки кроме «Угол нашивки, град» имеют тип float64, то есть вещественный. Пропусков в данных нет. Все признаки за исключением вышеуказанного, являются непрерывными, количественными. «Угол нашивки, град» принимает только два значения (0 и 90) в связи с чем он может быть преобразован в бинарный признак и будет рассматриваться как категориальный.

Таблица 1 – Описание признаков объединенного набора данных

Название	Исходный файл	Тип данных	Количество пропусков	Количество уникальных значений
Соотношение матрица-наполнитель	X_br	float64	-	1014
Плотность, кг/м ³	X_br	float64	-	1013

Продолжение таблицы 1

Модуль упругости, ГПа	X_bp	float64	-	1020
Количество отвердителя, м %	X_bp	float64	-	1005
Содержание эпоксидных групп,%_2	X_bp	float64	-	1004
Температура вспышки, С 2	X_bp	float64	-	1003
Поверхностная плотность, г/м2	X_bp	float64	-	1004
Модуль упругости при растяжении, ГПа	X_bp	float64	-	1004
Прочность при растяжении, МПа	X_bp	float64	-	1004
Потребление смолы, г/м ²	X_bp	float64	-	1003
Угол нашивки, град	X_nup	int64	-	2
Шаг нашивки	X_nup	float64	-	989
Плотность нашивки	X_nup	float64	-	988

По представленным на рисунках 1-2 гистограммам распределения переменных и диаграммам «ящик с усами» видно, что все признаки, кроме «Угол нашивки, град», имеют нормальное распределение и принимают неотрицательные значения.

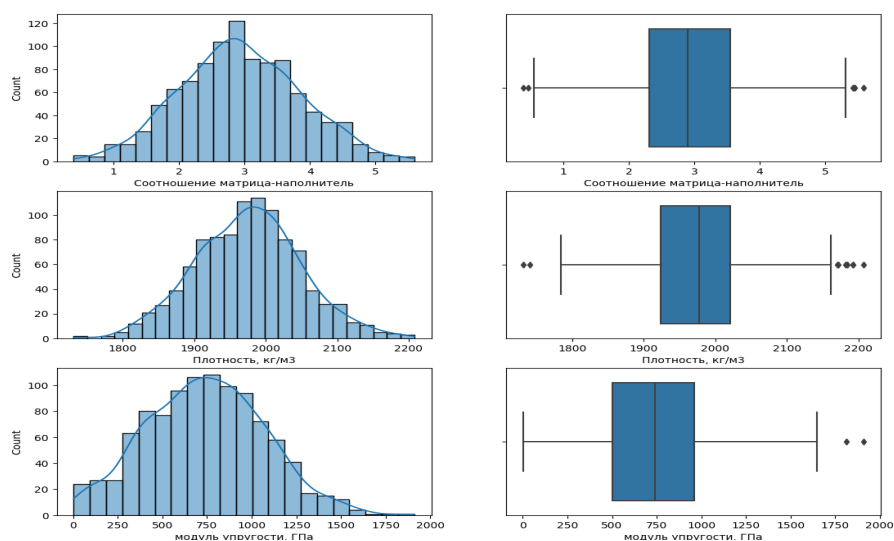


Рисунок 1 – Визуальное отображение распределения данных

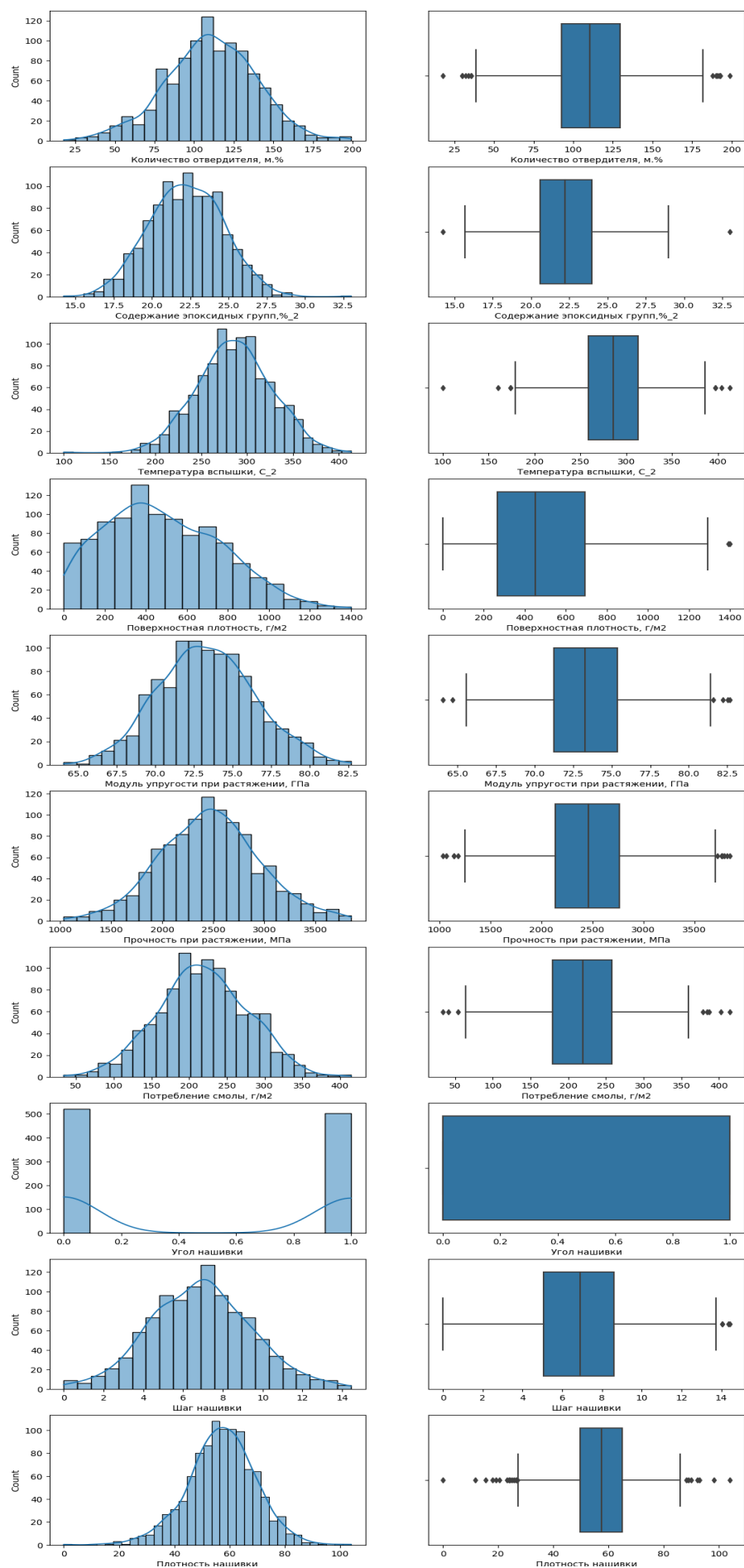


Рисунок 2 – Визуальное отображение распределения данных

Известно, что представленный набор данных был предварительно подготовлен, поэтому отсутствие пропусков вопросов не вызывает.

Для численного отображения информации из приведенных выше рисунков в таблице 2 приводится описательная характеристика данных.

Таблица 2 – Описательная характеристика признаков выборки

Название признака	Среднее значение	Стандартное отклонение	Минимум/ Медиана/ Максимум
Соотношение матрица-наполнитель	2.9304	0.9132	0.3894/ 2.9069/ 5.5917
Плотность, кг/м ³	1975.7349	73.7292	1731.7646/ 1977.6217/ 2207.7735
Модуль упругости, ГПа	739.9232	330.2316	2.4369/ 739.6643/ 1911.5365
Количество отвердителя, м_%	110.5708	28.2959	17.7403/ 110.5648/ 198.9532
Содержание эпоксидных групп,%_2	22.2444	2.4063	14.2550/ 22.2307/ 33.0000
Температура вспышки, С_2	285.8822	40.9433	100.0000/ 285.8968/ 413.2734
Поверхностная плотность, г/м ²	482.7318	281.3147	0.6037/ 451.8644/ 1399.5424
Модуль упругости при растяжении, ГПа	73.3286	3.1190	64.0541/ 73.2688/ 82.6821
Прочность при растяжении, МПа	2466.9228	485.6280	1036.8566/ 2459.5245/ 3848.4367
Потребление смолы, г/м ²	218.4231	59.7359	33.8030/ 219.1989/ 414.5906
Угол нашивки, град	44.2522	45.0158	0.0000/ 0.0000/ 90.0000

Продолжение таблицы 2

Шаг нашивки	6.8992	2.5635	0.0000/ 6.9161/ 14.4405
Плотность нашивки	57.1539	12.3510	0.0000/ 57.3419/ 103.9889

Для поиска выбросов целесообразно произвести вывод попарных графиков рассеяния точек (рисунок 3). На изображении видно, что некоторые точки отделяются далеко от общего облака, – это и есть выбросы, аномальные, некорректные значения данных, выходящие за пределы допустимых значений признака.

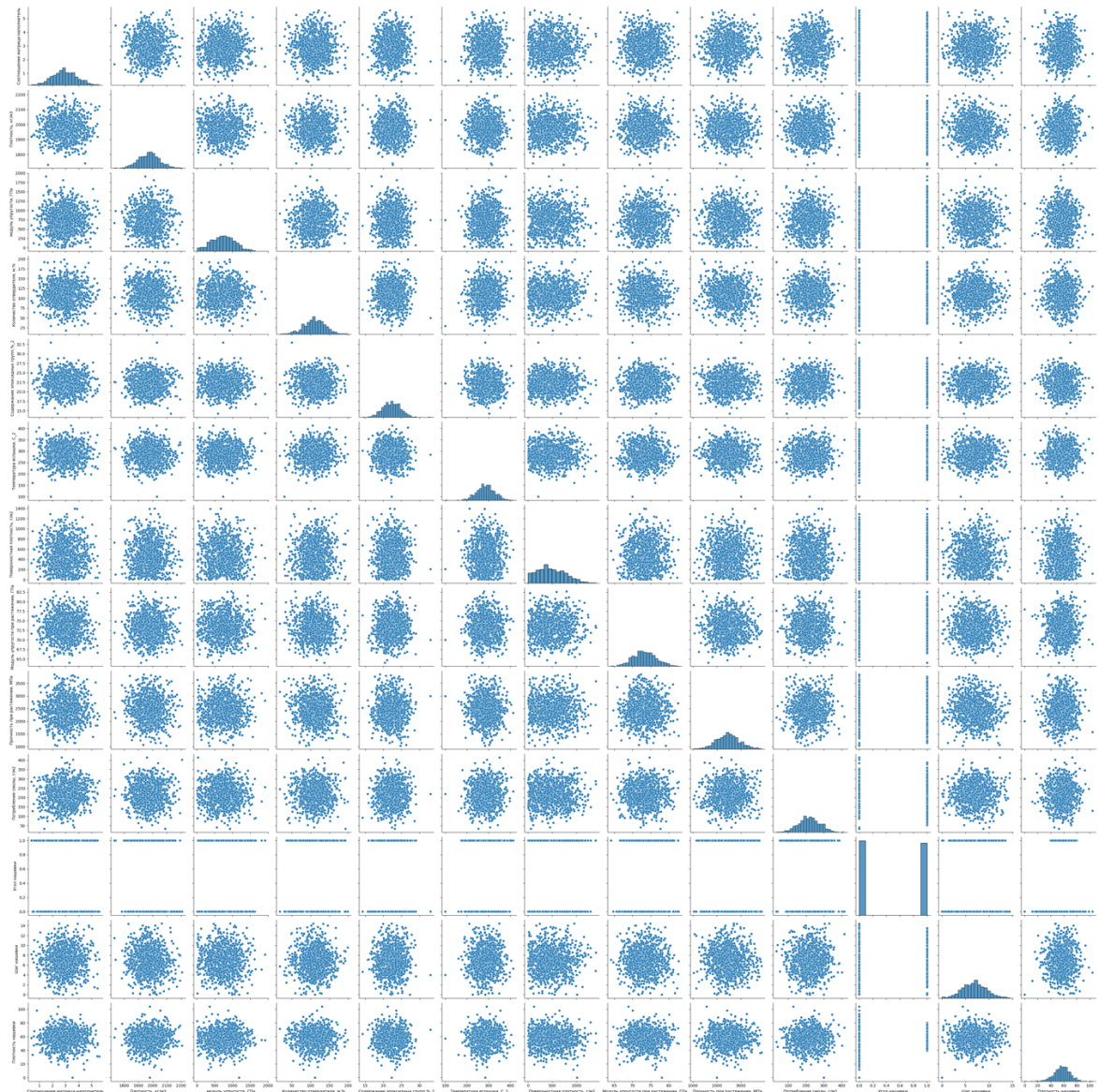


Рисунок 3 – Попарные графики рассеяния точек

Ни одна модель машинного обучения не выдаст осмысленных результатов, если предоставить ей такого типа данные. После формирования выборки их необходимо очистить. Выполнение этой задачи предполагается в подразделе 1.3 настоящей работы.

1.2 Описание используемых методов

Предсказание значений вещественной, непрерывной переменной — это задача регрессии. Эта зависимая переменная должна иметь связь с одной или несколькими независимыми переменными, называемыми также предикторами или регрессорами. Регрессионный анализ помогает понять, как «типичное» значение зависимой переменной изменяется при изменении независимых переменных.

В настоящее время разработано множество методов регрессионного анализа. Например, простая и множественная линейная регрессия. Эти модели являются параметрическими в том смысле, что функция регрессии определяется конечным числом неизвестных параметров, которые оцениваются на основе данных.

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

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

1.2.1 Линейная регрессия

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, который определяет зависимость переменных с помощью линии наилучшего соответствия. Модель регрессии создаёт несколько метрик. R^2 , или коэффициент детерминации и 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: быстрый и простой в реализации, легко интерпретируется и имеет меньшую сложность по сравнению с другими алгоритмами.

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

1.2.2 Регрессия лассо

Регрессия лассо (LASSO, Least Absolute Shrinkage and Selection Operator) – это вариант линейной регрессии, специально адаптированный для данных, которые имеют сильную корреляцию признаков друг с другом.

LASSO использует сжатие коэффициентов (shrinkage) и этим пытается уменьшить сложность данных, искривляя пространство, на котором они лежат. В этом процессе лассо автоматически помогает устранить или исказить сильно коррелированные и избыточные функции в методе с низкой дисперсией.

Достоинства метода: легко избавляется от шумов, быстро работает, не очень энергозатратно, способно полностью убрать признак из датасета, доступно обнуляет значения коэффициентов.

Недостатки метода: часто страдает качество прогнозирования, выбор модели не помогает и обычно вредит, периодически выдаёт ложный результат, случайным образом выбирает одну из коллинеарных переменных, не оценивает правильность формы взаимосвязи между независимой и зависимой переменными.

1.2.3 Метод опорных векторов

Метод опорных векторов (SVM, Support vector machines) – один из алгоритмов бинарной классификации, основанный на обучении с учителем, использующих линейное разделение пространства признаков с помощью гиперплоскости. Основная идея метода заключается в отображении векторов пространства признаков, представляющих классифицируемые объекты, в пространство более высокой размерности. Это связано с тем, что в пространстве большей размерности линейная разделимость множества оказывается выше, чем в пространстве меньшей размерности. Причины этого интуитивно понятны: чем больше признаков используется для распознавания объектов, тем выше ожидаемое качество распознавания.

Достоинства метода: разделяющая гиперплоскость, построенная по вышеуказанному правилу, обеспечит наиболее уверенное разделение классов и минимизирует среднюю ошибку распознавания. Для классификации достаточно небольшого набора данных. При правильной работе модели, построенной на тестовом множестве, 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.

Недостатки метода: неустойчивость к шуму, для больших наборов данных требуется долгое время обучения, а также сложность интерпретации параметров модели.

1.2.4 Дерево решений

Дерево принятия решений (Decision trees) – классификатор, построенный на основе решающих правил вида «если, то», упорядоченных в древовидную иерархическую структуру. Дерево решений является линейным классификатором, т.е. производит разбиение объектов в многомерном пространстве плоскостями (в двумерном случае — линиями). В настоящее время деревья решений стали одним из наиболее популярных методов классификации в интеллектуальном анализе данных и бизнес-аналитике.

Широкая популярность деревьев решений обусловлена следующими их преимуществами:

- правила в них формируются практически на естественном языке, что делает объясняющую способность деревьев решений очень высокой;
- могут работать как с числовыми, так и с категориальными данными;
- требуют относительно небольшой предобработки данных, в частности, не требуют нормализации, создания фиктивных переменных, могут работать с пропусками;
- могут работать с большими объемами данных.

Вместе с тем, деревьям решений присущ ряд ограничений:

- неустойчивость — даже небольшие изменения в данных могут привести к значительным изменениям результатов классификации;
- поскольку алгоритмы построения деревьев решений являются жадными (на каждом шаге ищут локально-оптимальное решение, предполагая, что конечное общее решение также будет оптимальным), они не гарантируют построения оптимального дерева;
- склонность к переобучению.

1.2.5 Случайный лес

Случайный лес (RandomForest) — это множество решающих деревьев. Суть алгоритма заключается в использовании ансамбля решающих деревьев,

каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества получается необходимый результат.

Достоинства:

- способность эффективно обрабатывать данные с большим числом признаков и классов;
- нечувствительность к любым монотонным преобразованиям значений признаков;
- одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки;
- существуют методы оценивания значимости отдельных признаков;
- внутренняя оценка способности модели к обобщению;
- высокая параллелизуемость и масштабируемость;
- очень гибок и обладает высокой точностью.

Недостатки:

- большой размер получающихся моделей;
- построение леса сложнее и отнимает больше времени;
- чем больше объем, тем меньше интуитивное понимание.

1.2.6 Градиентный бустинг

Бустинг (Boosing) – метод построения ансамбля моделей, при котором базовые модели обучаются последовательно и каждая последующая модель ансамбля применяется к результатам на выходе предыдущей.

Логика, что стоит за градиентным бустингом, проста, ее можно понять интуитивно, без математического формализма. Первое предположение линейной регрессии, что сумма отклонений $= 0$, т.е. отклонения должны быть случайно распределены в окрестности нуля. Теперь давайте думать об отклонениях, как об ошибках, сделанных нашей моделью. Хотя в моделях, основанных на деревьях не делается такого предположения, если мы будем размышлять об этом предположении логически (не статистически), мы можем понять, что увидев

принцип распределения отклонений, сможем использовать данный паттерн для модели.

Итак, суть, стоящая за алгоритмом градиентного бустинга — итеративно применять паттерны отклонений и улучшать предсказания. Как только мы достигли момента, когда отклонения не имеют никакого паттерна, мы прекращаем дотраивать нашу модель (иначе это может привести к переобучению). Алгоритмически, мы минимизируем нашу функцию потерь.

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

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

1.2.7 Метод k -ближайших соседей

Метод k -ближайших соседей (KNN, K-nearest neighbors) относит объекты к классу, которому принадлежит большинство из k его ближайших соседей в многомерном пространстве признаков. Это один из простейших алгоритмов обучения классификационных моделей. Число k — это количество соседних объектов в пространстве признаков, которые сравниваются с классифицируемым объектом. Иными словами, если $k = 10$, то каждый объект сравнивается с 10-ю соседями. В процессе обучения алгоритм просто запоминает все векторы признаков и соответствующие им метки классов. При работе с реальными данными, т.е. наблюдениями, метки класса которых неизвестны, вычисляется расстояние между вектором нового наблюдения и ранее запомненными. Затем выбирается k ближайших к нему векторов, и новый объект относится к классу, которому принадлежит большинство из них.

К достоинствам алгоритма можно отнести:

- устойчивость к выбросам и аномальным значениям, поскольку вероятность попадания содержащих их записей в число k -ближайших соседей мала. Если же это произошло, то влияние на голосование (особенно взвешенное) также, скорее всего, будет незначительным, и, следовательно, малым будет и влияние на результаты классификации;
- программная реализация алгоритма относительно проста;
- результаты работы алгоритма легко поддаются интерпретации. Логика работы алгоритма понятна экспертам в различных областях.

К недостаткам алгоритма KNN можно отнести:

- метод не создает каких-либо моделей, обобщающих предыдущий опыт, а интерес могут представлять и сами правила классификации;
- при классификации объекта используются все доступные данные, поэтому метод KNN является достаточно затратным в вычислительном плане, особенно в случае больших объёмов данных;
- высокая трудоёмкость из-за необходимости вычисления расстояний до всех примеров, которая увеличивается квадратично с ростом числа обучающих примеров;
- повышенные требования к репрезентативности исходных данных.

1.3 Разведочный анализ данных

Цель разведочного анализа данных – выявить закономерности в данных. Для корректной работы большинства моделей желательна сильная зависимость выходных переменных от входных и отсутствие зависимости между входными переменными.

На рисунке 3 мы видели график попарного рассеяния точек. По форме «облаков точек» мы не заметили зависимостей, которые могли бы стать основой работы моделей. Однако перед дальнейшей работой есть необходимость проведения очистки от выбросов, выявленных ранее. Существуют следующие

методы выявления выбросов для признаков с нормальным распределением: метод трех сигм и метод межквартильных интервалов. Применив эти методы на нашем наборе данных было найдено 24 и 93 выброса соответственно.

Поскольку известно, что датасет был предварительно подготовлен и не имеет явного шума, следует применить метод 3-х сигм как более деликатный, чтобы не потерять возможно значимые данные. Значения, определенные как выбросы, удаляем. После удаления значений, определенных как выбросы, в наборе данных осталось 1000 строк и 13 признаков-переменных.

Помочь выявить связь между признаками может матрица корреляции, приведенная на рисунке 4.

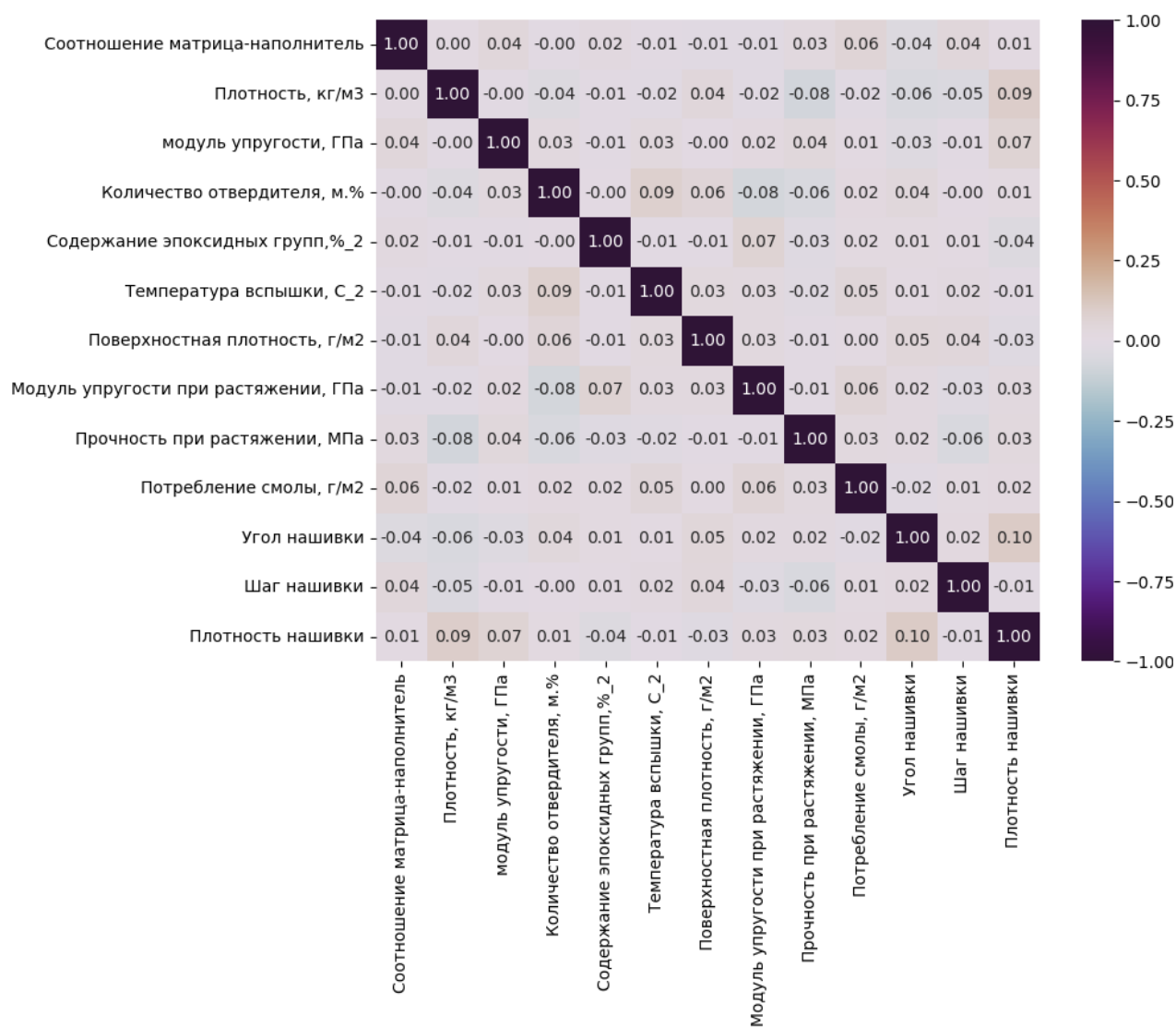


Рисунок 4 – Матрица корреляции признаков

По матрице корреляции мы видим, что все коэффициенты корреляции близки к нулю. Максимальная корреляция между плотностью нашивки и углом нашивки достигает 0.10. Корреляционные связи между переменными не наблюдаются.

2 Практическая часть

2.1 Предобработка данных

В данном разделе предполагается обучение алгоритма машинного обучения, который будет определять значения модуля упругости при растяжении и прочности при растяжении, а также написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель. Для каждого из целевых признаков строится отдельная модель, соответственно будут решены 3 отдельные задачи.

Ход решения каждой из задач и построения оптимальной модели будет следующим:

- разделить данные на тренировочную и тестовую выборки. В задании указано, что на тестирование оставить 30% данных;
- выполнить препроцессинг (подготовку исходных данных);
- выбрать базовую модель для определения нижней границы качества предсказания. Используется базовая модель, возвращающая среднее значение целевого признака. Лучшая модель по своим характеристикам должна быть лучше базовой;
- взять несколько моделей с гиперпараметрами по умолчанию, и используя перекрестную проверку, проверить их метрики на тренировочной выборке;
- подобрать для этих моделей гиперпараметры с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10;
- сравнить метрики моделей после подбора гиперпараметров и выбрать лучшую;
- получить предсказания лучшей и базовой моделей на тестовой выборке, сделать выводы;
- сравнить качество работы лучшей модели на тренировочной и тестовой выборке.

Цель препроцессинга, или предварительной обработки данных – обеспечить корректную работу моделей.

Его необходимо выполнять после разделения на тренировочную и тестовую выборку, таким образом как будто заранее параметры тестовой выборки неизвестны (минимум, максимум, математическое ожидание, стандартное отклонение).

Препроцессинг для категориальных и количественных признаков выполняется по-разному. Однако категориальный признак «Угол нашивки, град» принимает только два значения 0 и 90 и поэтому был преобразован в 0 и 1 соответственно.

Вещественных количественных признаков у нас большинство. Проблема вещественных признаков в том, что их значения лежат в разных диапазонах, в разных масштабах. Это видно в таблице 2. Необходимо провести одно из двух возможных преобразований:

- нормализацию, т.е. приведение в диапазон от 0 до 1 с помощью `MinMaxScaler`;
- стандартизацию, т.е. приведение к матожиданию 0, стандартному отклонению 1 с помощью `StandardScaler`.

Удобно реализовать предварительную обработку с помощью `ColumnTransformer`, а потом сохранить и загрузить этот объект аналогично объекту модели. Выходные переменные никаким образом не изменяются.

Для обеспечения статистической устойчивости метрик модели используется перекрестная проверка или кросс-валидация. Чтобы ее реализовать, выборка разбивается необходимое количество раз на тестовую и валидационную. Модель обучается на тестовой выборке, а затем выполняется расчет метрик качества на валидационной. В качестве результата мы получаем средние метрики качества для всех валидационных выборок. Перекрестную проверку реализует функция `cross_validate` из `sklearn`.

Поиск гиперпараметров по сетке реализует класс `GridSearchCV` из `sklearn`. Он получает модель и набор гиперпараметров, поочередно передает их в модель,

выполняет обучение и определяет лучшие комбинации гиперпараметров. Перекрестная проверка уже встроена в этот класс.

Существует множество различных метрик качества, применимых для регрессии. В этой работе используются:

- R^2 или коэффициент детерминации измеряет долю дисперсии, объясненную моделью, в общей дисперсии целевой переменной. Если он близок к единице, то модель хорошо объясняет данные, если же он близок к нулю, то прогнозы сопоставимы по качеству с константным предсказанием;

- RMSE (Root Mean Squared Error) или корень из средней квадратичной ошибки принимает значения в тех же единицах, что и целевая переменная. Метрика использует возведение в квадрат, поэтому хорошо обнаруживает грубые ошибки, но сильно чувствительна к выбросам;

- MAE (Mean Absolute Error) - средняя абсолютная ошибка так же принимает значения в тех же единицах, что и целевая переменная;

- MAPE (Mean Absolute Percentage Error) или средняя абсолютная процентная ошибка — безразмерный показатель, представляющий собой взвешенную версию MAE;

- max error или максимальная ошибка данной модели в единицах измерения целевой переменной.

RMSE, MAE, MAPE и max error принимают положительные значения. Но отображать я их буду со знаком «-». Так корректно отработает выделение цветом лучших моделей — эти метрики надо минимизировать.

R^2 в норме принимает положительные значения. Эту метрику надо максимизировать. Отрицательные значения коэффициента детерминации означают плохую объясняющую способность модели.

2.1.1 Для прогнозирования модуля упругости при растяжении

Признаки датасета были разделены на входные и выходные, а строки – на тренировочное и тестовое множество. Размерности полученных наборов данных показаны на рисунке 4. Описательная статистика входных признаков до и после

предобработки показана на рисунке 5. Описательная статистика выходного признака показана на рисунке 6.

```
x1_train: (700, 11) y1_train: (700, 1)
x1_test: (300, 11) y1_test: (300, 1)
```

Рисунок 4 – Размерности тренировочного и тестового множеств после разбиения для 1-й задачи

# Описательная статистика входных данных до предобработки show_statistics(x1_train_raw)											
	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м. %	Содержание эпоксидных групп, %_2	Температура вспышки, C_2	Поверхностная плотность, г/м2	Потребление смолы, г/м2	Угол нашивки	Шаг нашивки	Плотность нашивки
min	0.547391	1784.482245	2.436909	33.624187	15.695894	173.973907	1.668002	41.048278	0.000000	0.037639	20.571633
max	5.591742	2192.738783	1649.415706	192.851702	28.907470	403.652861	1288.691844	386.903431	1.000000	14.033215	92.963492
mean	2.943860	1972.286516	738.627618	112.119243	22.179055	286.449560	481.805877	216.838475	0.495714	6.880379	57.403269
std	0.902194	73.148332	326.130594	28.056458	2.335087	40.645101	278.253589	58.108052	0.500339	2.590968	12.036623
# Описательная статистика входных данных после предобработки show_statistics(pd.DataFrame(x1_train, columns=(x1_continuous + x_categorical)))											
	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м. %	Содержание эпоксидных групп, %_2	Температура вспышки, C_2	Поверхностная плотность, г/м2	Потребление смолы, г/м2	Шаг нашивки	Плотность нашивки	Угол нашивки
min	-2.658166	-2.569280	-2.258964	-2.799754	-2.778397	-2.769241	-1.726774	-3.027393	-2.642886	-3.062152	0.000000
max	2.937033	3.015925	2.794707	2.879558	2.883502	2.885639	2.901896	2.928795	2.762655	2.956448	1.000000
mean	0.000000	0.000000	-0.000000	-0.000000	-0.000000	0.000000	0.000000	-0.000000	0.000000	-0.000000	0.495714
std	1.000715	1.000715	1.000715	1.000715	1.000715	1.000715	1.000715	1.000715	1.000715	1.000715	0.500339

Рисунок 5 – Описательная статистика входных признаков до и после предобработки для 1-й задачи

```
min    64.054061
max    82.237600
mean   73.398761
std     3.128575
```

Рисунок 6 – Описательная статистика выходного признака для 1-й задачи

2.1.2 Для прогнозирования прочности при растяжении

Вышеописанные манипуляции были произведены для соответствующей задачи. Результаты отображены на рисунках 7, 8.

# Описательная статистика входных данных до предобработки											
show_statistics(x2_train_raw)											
	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м. %	Содержание эпоксидных групп, %_2	Температура вспышки, C_2	Поверхностная плотность, г/м2	Потребление смолы, г/м2	Угол нашивки	Шаг нашивки	Плотность нашивки
min	0.547391	1784.482245	2.436909	33.624187	15.695894	173.973907	1.668002	41.048278	0.000000	0.037639	20.571633
max	5.591742	2192.738783	1649.415706	192.851702	28.907470	403.652861	1288.691844	386.903431	1.000000	14.033215	92.963492
mean	2.943860	1972.286516	738.627618	112.119243	22.179055	286.449560	481.805877	216.838475	0.495714	6.880379	57.403269
std	0.902194	73.148332	326.130594	28.056458	2.335087	40.645101	278.253589	58.108052	0.500339	2.590968	12.036623
# Описательная статистика входных данных после предобработки											
show_statistics(pd.DataFrame(x2_train, columns=(x2_continuous + x_categorical)))											
	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м. %	Содержание эпоксидных групп, %_2	Температура вспышки, C_2	Поверхностная плотность, г/м2	Потребление смолы, г/м2	Шаг нашивки	Плотность нашивки	Угол нашивки
min	-2.567145	-2.753587	-2.225848	-2.727895	-2.679251	-2.825478	-1.713713	-2.972700	-2.802971	-3.198418	0.000000
max	2.893896	2.918573	2.752658	3.128279	2.626492	3.029659	2.942989	2.723003	2.846822	3.161091	1.000000
mean	0.027286	-0.144307	-0.000482	0.159048	-0.075626	0.041830	0.023519	-0.077706	-0.040665	0.037182	0.495714
std	0.976720	1.016295	0.985831	1.031879	0.937766	1.036153	1.008775	0.956950	1.045933	1.057398	0.500339

Рисунок 7 – Описательная статистика входных признаков до и после предобработки для 2-й задачи

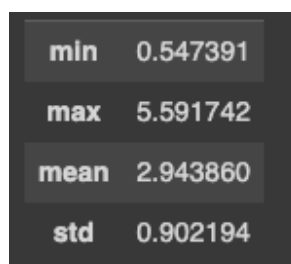
min	1071.123751
max	3848.436732
mean	2469.109198
std	493.531741

Рисунок 8 – Описательная статистика выходного признака для 2-й задачи

2.1.3 Для прогнозирования соотношения матрица-наполнитель

# Описательная статистика входных данных до предобработки											
show_statistics(x3_train_raw)											
	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м. %	Содержание эпоксидных групп, %_2	Температура вспышки, C_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки	Шаг нашивки
min	1784.482245	2.436909	33.624187	15.695894	173.973907	1.668002	64.054061	1071.123751	41.048278	0.000000	0.037639
max	2192.738783	1649.415706	192.851702	28.907470	403.652861	1288.691844	82.237600	3848.436732	386.903431	1.000000	14.033215
mean	1972.286516	738.627618	112.119243	22.179055	286.449560	481.805877	73.398761	2469.109198	216.838475	0.495714	6.880379
std	73.148332	326.130594	28.056458	2.335087	40.645101	278.253589	3.128575	493.531741	58.108052	0.500339	2.590968
# Описательная статистика входных данных после предобработки											
show_statistics(pd.DataFrame(x3_train, columns=(x3_continuous + x_categorical)))											
	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м. %	Содержание эпоксидных групп, %_2	Температура вспышки, C_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Шаг нашивки	Плотность нашивки
min	-2.753587	-2.225848	-2.727895	-2.679251	-2.825478	-1.713713	-2.955472	-2.979732	-2.972700	-2.802971	-3.198418
max	2.918573	2.752658	3.128279	2.626492	3.029659	2.942989	2.965684	3.000494	2.723003	2.846822	3.161091
mean	-0.144307	-0.000482	0.159048	-0.075626	0.041830	0.023519	0.087468	0.030468	-0.077706	-0.040665	0.037182
std	1.016295	0.985831	1.031879	0.937766	1.036153	1.008775	1.018767	1.062693	0.956950	1.045933	1.057398

Рисунок 9 – Описательная статистика входных признаков до и после предобработки для 3-й задачи



min	0.547391
max	5.591742
mean	2.943860
std	0.902194

Рисунок 9 – Описательная статистика выходного признака для 3-й задачи

2.2 Разработка и обучение моделей

Для подбора лучшей модели в определении параметров модуля упругости при растяжении были взяты следующие модели:

- LinearRegression — линейная регрессия;
- Ridge — гребневая регрессия;
- Lasso — лассо-регрессия;
- SVR — метод опорных векторов;
- KNeighborsRegressor — метод ближайших соседей;
- DecisionTreeRegressor — деревья решений;
- RandomForestRegressor — случайный лес.

В качестве базовой модели взят DummyRegressor, возвращающий среднее значение целевого признака.

Метрики работы выбранных моделей с гиперпараметрами по умолчанию, полученные с помощью перекрестной проверки на тестовом множестве, приведены на рисунке 10.

Ни одна из выбранных моделей не оказалась подходящей для имеющегося набора данных.

Коэффициент детерминации R^2 близок к 0 для линейных моделей и метода опорных векторов. Значит, они не лучше базовой модели. И остальные метрики у них примерно совпадают с базовой моделью.

Хуже линейных моделей с гиперпараметрами по умолчанию отработали метод ближайших соседей и деревья решений.

Случайный лес отработал лучше, чем одно дерево решений, но хуже, чем линейные модели.

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.019376	-3.126837	-2.510495	-0.034288	-7.798195
LinearRegression	-0.018532	-3.123936	-2.502366	-0.034179	-8.098392
Ridge	-0.018463	-3.123834	-2.502325	-0.034178	-8.097452
Lasso	-0.019376	-3.126837	-2.510495	-0.034288	-7.798195
SVR	-0.041456	-3.157875	-2.499637	-0.034118	-8.357012
KNeighborsRegressor	-0.238674	-3.443022	-2.725185	-0.037216	-8.823389
DecisionTreeRegressor	-1.034156	-4.403633	-3.589790	-0.048987	-11.822403
RandomForestRegressor	-0.075323	-3.208305	-2.555245	-0.034924	-8.380008

Рисунок 10 – Результаты работы моделей с гиперпараметрами по умолчанию

После выполнения подбора гиперпараметров по сетке с перекрестной проверкой, получили метрики, приведенные на рисунке 11.

Можно сделать вывод, что подбирая гиперпараметры, можно значительно улучшить предсказание выбранной модели.

Ridge(alpha=80, positive=True, solver='lbfgs')	-0.016604	-3.121555	-2.494491	-0.034068	-7.851250
Lasso(alpha=0.05)	-0.012094	-3.114368	-2.500839	-0.034157	-7.965382
SVR(C=0.01, kernel='linear')	-0.017814	-3.123659	-2.500515	-0.034147	-8.061850
KNeighborsRegressor(n_neighbors=29)	-0.036593	-3.147992	-2.512539	-0.034342	-8.157406
DecisionTreeRegressor(max_depth=2, max_features=2, random_state=42)	-0.018267	-3.125442	-2.460189	-0.034017	-8.154902
RandomForestRegressor(bootstrap=False, criterion='absolute_error', max_depth=5, max_features=1, n_estimators=50, random_state=42)	-0.017821	-3.124964	-2.497013	-0.034103	-8.146335

Рисунок 11 – Результаты моделей после подбора гиперпараметров

Все модели крайне плохо описывают исходные данные – положительного значения R2 добиться не удалось. Самая лучшая модель дает коэффициент детерминации близкий к нулю, что соответствует базовой модели. Линейные модели совпадают с базовой моделью. Их характеристики улучшились, но не значительно. Метод опорных векторов в процессе подбора гиперпараметров лучшим ядром выбрал линейное и отработал аналогично линейным моделям. Метод ближайших соседей увеличением количества соседей радикально улучшил качество работы. Но его лучшие результаты все равно по-прежнему

отстают от линейных моделей. Деревья решений при кропотливом подборе параметров превзошли результат линейной модели. Но они не являются объясняющей зависимостью моделью. Собирая деревья в ансамбли, можно улучшать характеристики. Но подбор параметров для леса затруднен тем, что это затратный по времени процесс. В качестве лучшей модели выбирается лассо-регрессия. На рисунке 12 приведена визуализация работы лучшей модели на тестовом множестве.

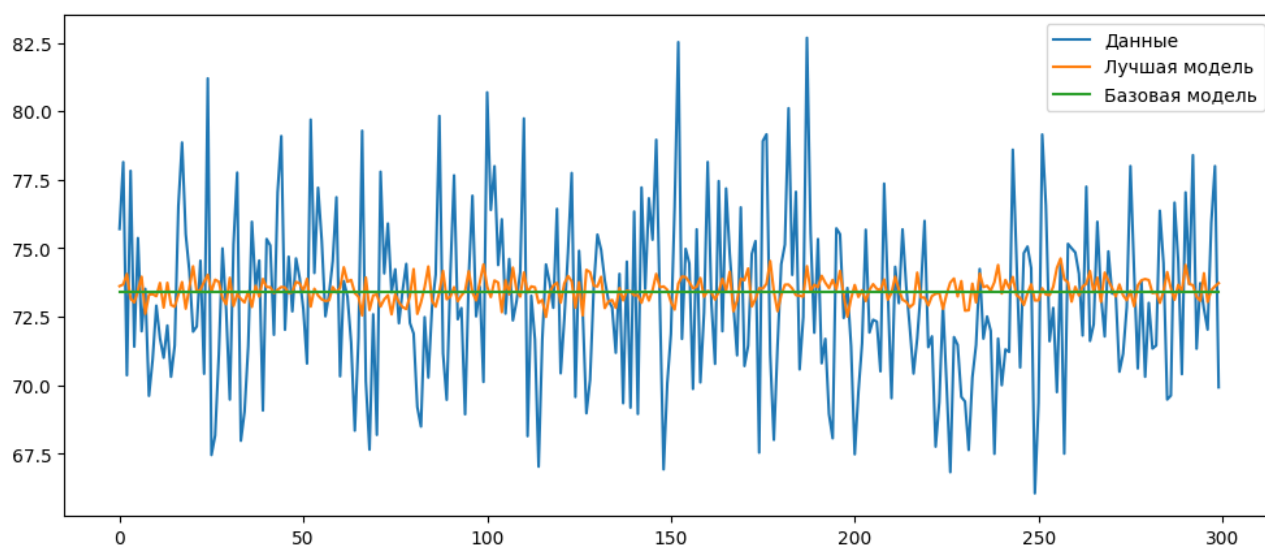


Рисунок 12 – Визуализация работы модели

Сложно визуализировать регрессию в многомерном пространстве. Но даже на таком графике мы видим, насколько не соответствует лучшая модель исходным данным и насколько она неудачна.

	R2	RMSE	MAE	MAPE	max_error
Базовая модель	-0.007651	-3.082670	-2.479138	-0.033976	-9.283290
Лучшая модель (lasso)	-0.021771	-3.104193	-2.512508	-0.034472	-8.816359

Рисунок 13 – Метрики работы лучшей модели на тестовом множестве

В ходе исследования работы моделей для параметров прочности при растяжении были получены схожие результаты (представлены на рисунках 14–16).

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.022944	-493.539876	-391.010975	-0.171456	-1281.791709
LinearRegression	-0.014804	-491.329446	-391.262712	-0.171170	-1305.947015
Ridge	-0.014749	-491.316685	-391.248131	-0.171164	-1305.883895
Lasso	-0.013580	-491.039500	-390.926249	-0.171039	-1304.848543
SVR	-0.021077	-493.116843	-390.543237	-0.170382	-1279.655107
DecisionTreeRegressor	-1.097452	-700.282012	-561.980002	-0.244517	-1784.349498
GradientBoostingRegressor	-0.033926	-496.051787	-398.082126	-0.172868	-1274.138037

Рисунок 14 – Результаты работы моделей с гиперпараметрами по умолчанию

Ни одна из выбранных моделей не соответствует данным. R2 близок к 0 для линейных моделей и метода опорных векторов. Значит, они не лучше базовой модели. И остальные метрики у них примерно совпадают с базовой моделью. Гораздо хуже линейных моделей с гиперпараметрами по умолчанию отработали деревья решений. Градиентный бустинг с параметрами по умолчанию отработал лучше дерева. Он тоже соответствует базовой модели.

	R2	RMSE	MAE	MAPE	max_error
Ridge(alpha=710, solver='sparse_cg')	-0.010171	-490.370725	-389.300830	-0.170584	-1291.112095
Lasso(alpha=20)	-0.010149	-490.357498	-389.214674	-0.170515	-1293.747766
SVR(C=0.02, kernel='llinear')	-0.021227	-493.150121	-390.526504	-0.170386	-1279.607766
DecisionTreeRegressor(max_depth=1, max_features=6, random_state=42, splitter='random')	-0.019887	-492.777398	-390.140019	-0.171163	-1281.728096
GradientBoostingRegressor(max_depth=2, max_features=1, random_state=42)	-0.033926	-496.051787	-398.082126	-0.172868	-1274.138037

Рисунок 15 – Результаты моделей после подбора гиперпараметров

Подбор гиперпараметров не помог получить модель, превосходящую базовую. Все модели плохо описывают исходные данные. Не удалось добиться коэффициента детерминации, большего нуля. Линейные модели после подбора немного улучшили характеристики. Метод опорных векторов отработал аналогично линейным моделям. Деревья решений после подбора параметров улучшили неудачный результат с параметрами по умолчанию. Но лучший результат дает градиентный бустинг. Значения ошибок примерно такие же, как у дерева решений. Но коэффициент детерминации немного больше, что показывает чуть лучшую объясняющую способность модели.

Поэтому в качестве лучшей модели выбран градиентный бустинг. На рисунке 16 приведена визуализация работы лучшей модели на тестовом множестве.

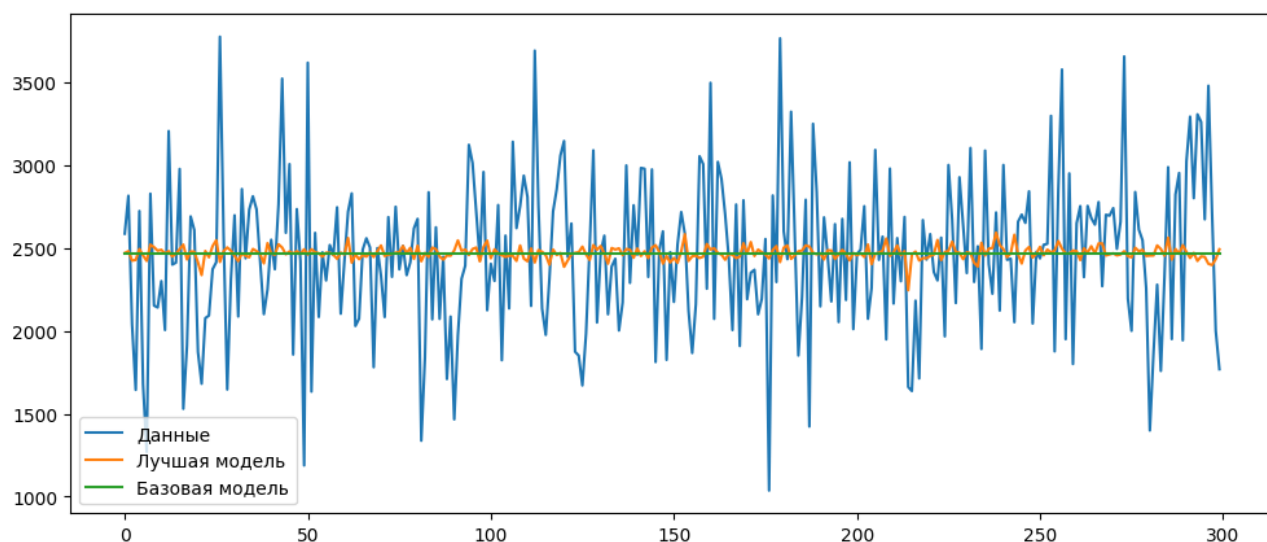


Рисунок 16 – Визуализация работы модели

Визуализируя результаты градиентного бустинга с выбранными параметрами, мы видим насколько они плохи и далеки от исходных данных. Но результаты выглядят более «естественно», чем те, что получены деревом решений для модуля упругости при растяжении.

Метрики работы лучшей модели на тестовом множестве и сравнение с базовой отражены на рисунке 17. Несмотря на то, что градиентный бустинг по показателям немного отстает от базовой модели, результат исследования отрицательный.

	R2	RMSE	MAE	MAPE	max_error
Базовая модель	-0.000928	-464.631542	-363.886617	-0.162616	-1432.252593
Лучшая модель (градиентный бустинг)	-0.006114	-465.833555	-364.992369	-0.162767	-1399.342924

Рисунок 17 – Метрики работы лучшей модели на тестовом множестве

2.3 Создание нейронной сети, рекомендующей соотношение матрица-наполнитель

В соответствии с заданием для соотношения матрица-наполнитель необходимо построить нейросеть. Строится нейронная сеть с помощью класса `MLPRegressor` следующей архитектуры:

- слоев: 8;
- нейронов на каждом слое: 24;
- активационная функция: `relu`;
- оптимизатор: `adam`;
- пропорция разбиения данных на тестовые и валидационные: 30%;
- ранняя остановка, если метрики на валидационной выборке не улучшаются;
- количество итераций: 5000.

Нейросеть обучилась за 1,12 сек и 33 итерации. График обучения приведен на рисунке 18.

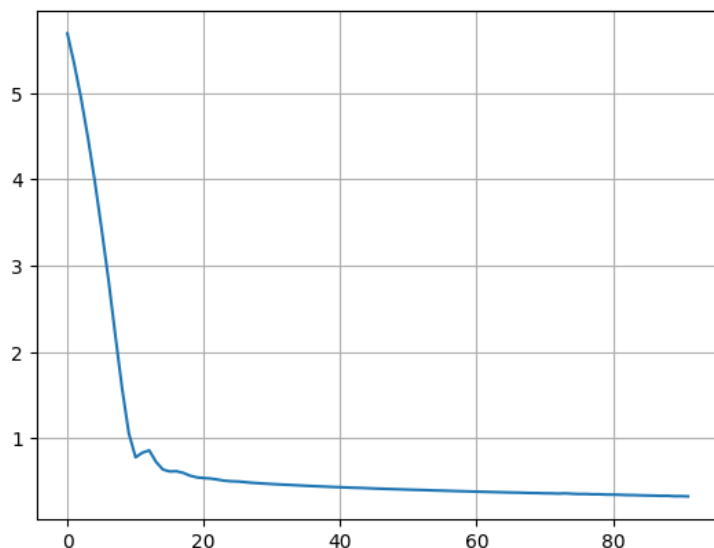


Рисунок 18 – График обучения MLPRegressor

Визуализация результатов, полученных нейросетью, приведена на рисунке 19. Видно, что нейросеть пыталась подстроиться под исходные данные, но хорошо не получилось.

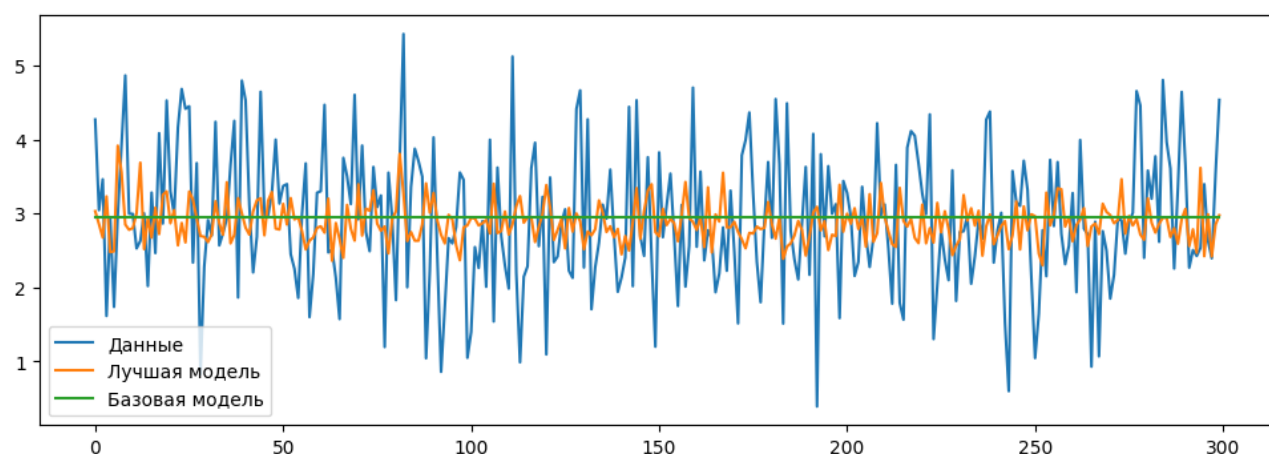


Рисунок 19 – Визуализация работы модели

Метрики работы нейросети MLPRegressor на тестовом множестве и сравнение с базовой моделью отражены на рисунке 20. Несмотря на красивый график с рисунка 19, метрики говорят об отсутствии результата, который можно внедрить. Ошибка нейросети составляет порядка 30 процентов, а ее значения ошибок хуже, чем у базовой модели.

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.000744	-0.924041	-0.739327	-0.340221	-2.554458
MLPRegressor	-0.093429	-0.965885	-0.780564	-0.346753	-2.699168

Рисунок 20 – Метрики работы нейросети MLPRegressor на тестовом множестве

Также дополнительно строится нейронная сеть с помощью класса `keras.Sequential` со следующими параметрами:

- входной слой для 12 признаков;
- выходной слой для 1 признака;
- скрытых слоев: 8;
- нейронов на каждом скрытом слое: 24;
- активационная функция скрытых слоев: `relu`;
- оптимизатор: `Adam`;
- loss-функция: `MeanAbsolutePercentageError`.

Архитектура нейросети приведена на рисунке 21.

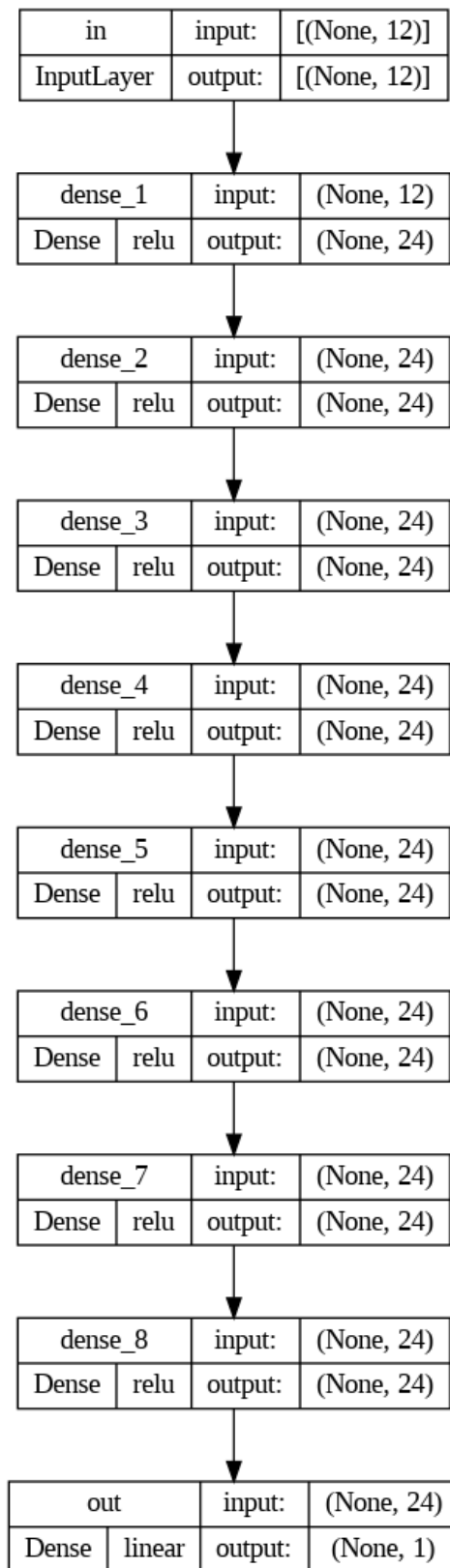


Рисунок 21 – Архитектура нейросети на основе TensorFlow

Запуск обучения нейросети происходит со следующими параметрами:

- пропорция разбиения данных на тестовые и валидационные: 30%;
- количество эпох: 50.

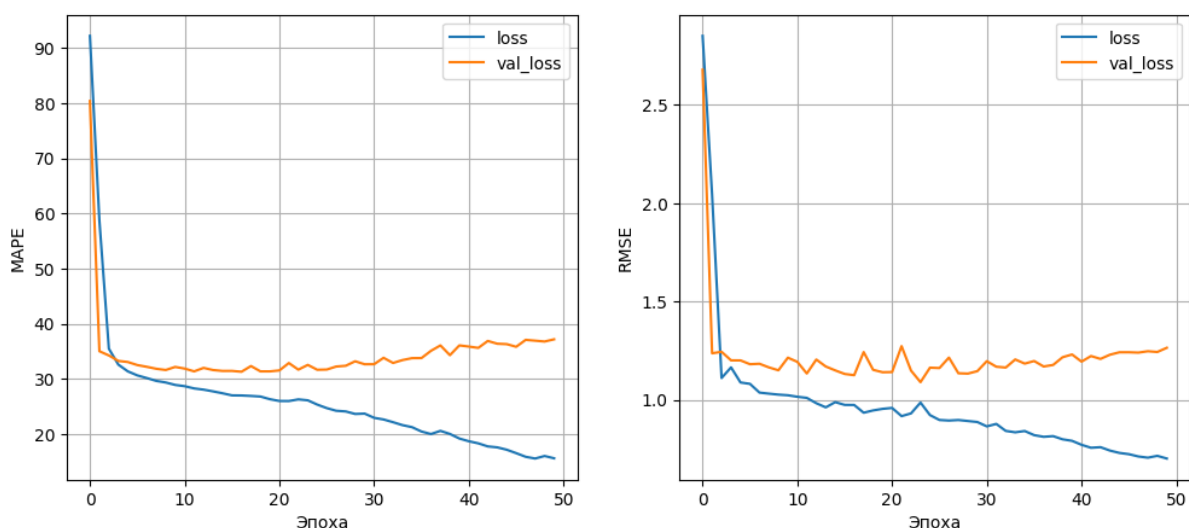


Рисунок 22 – График обучения

Видно, что примерно до 7-й эпохи обучение шло хорошо, а потом сеть начала переобучаться. Значение `loss` на тестовых выборках продолжило уменьшаться, а на валидационной начало расти.

Одним из способов борьбы с переобучением может быть ранняя остановка обучения, если `val_loss` начинает расти. Для этого в `tensorflow` используются `callbacks`. Попробую взять нейросеть с той же архитектурой и запустить обучение с ранней остановкой. График обучения приведен на рисунке 23. Очевидно, что решение проблемы переобучения повышает точность модели на новых данных.

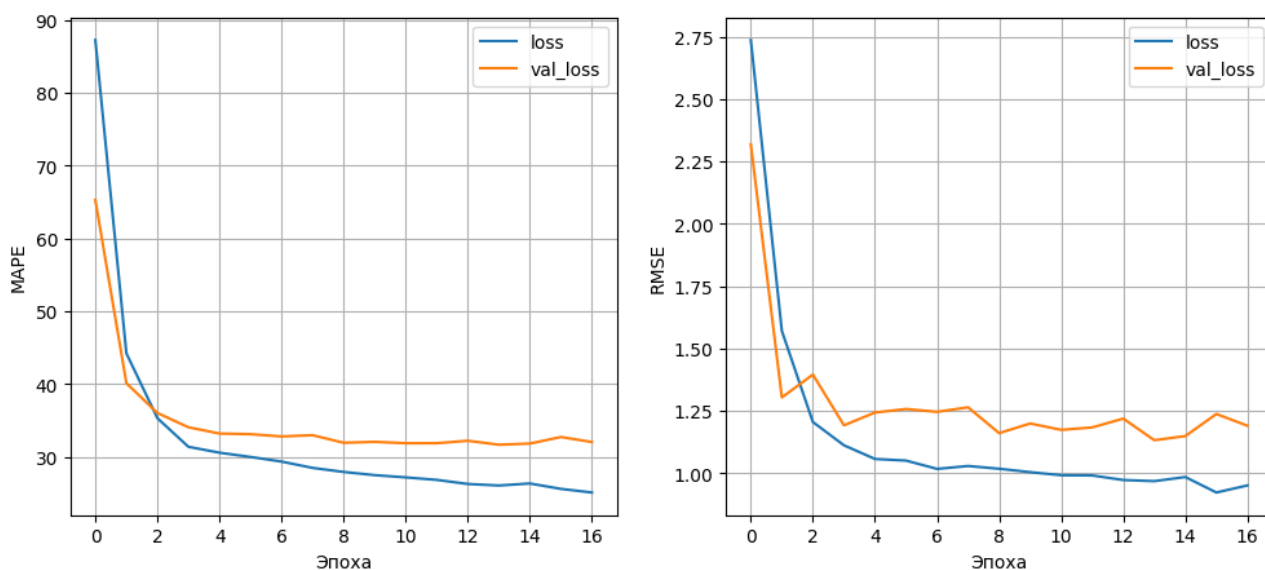


Рисунок 23 – График обучения с ранней остановкой

Еще одним методом борьбы с переобучением является добавление Dropout-слоев. Построим модель аналогичной архитектуры, только после каждого скрытого слоя добавим слой Dropout с параметром 0.05. Такой слой выключит 5% случайных нейронов на каждом слое.

График обучения приведен на рисунке 24. Видно, что Dropout-слои справились с переобучением.

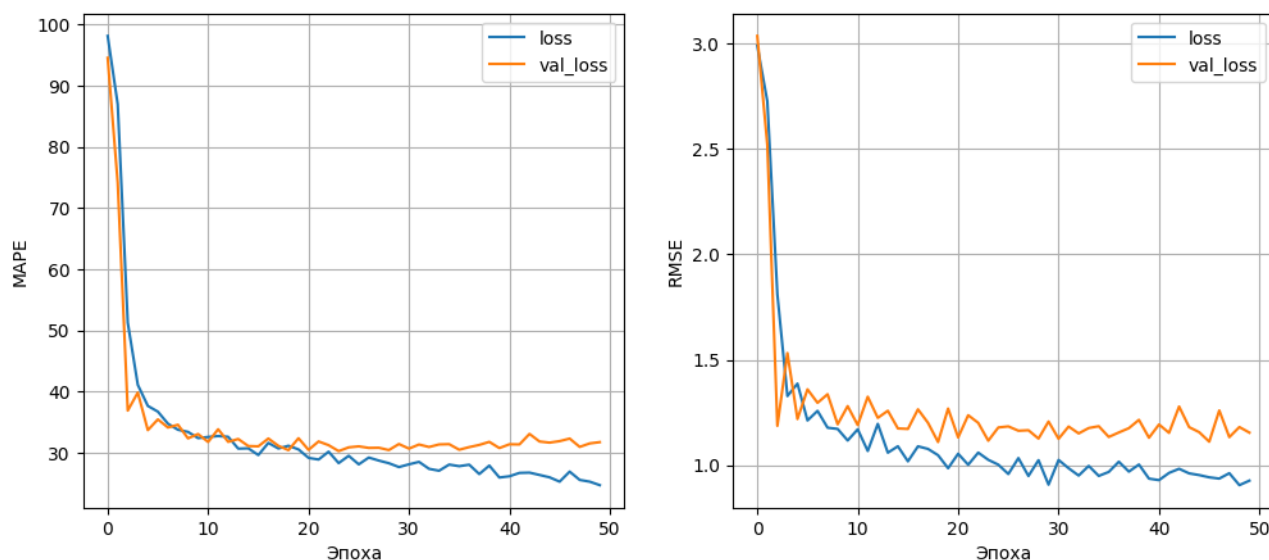
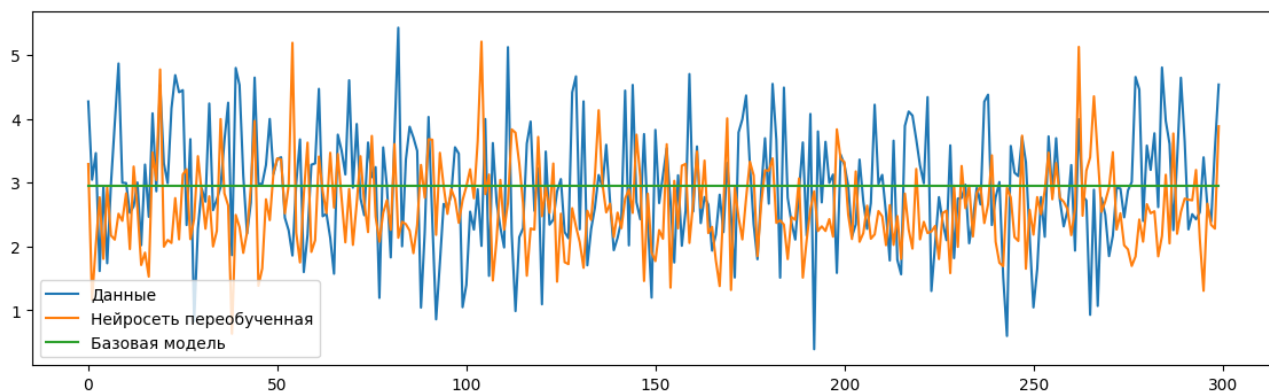


Рисунок 24 – График обучения с дропаут слоем

Использование ранней остановки сокращает время на обучение модели, а использование Dropout увеличивает. Но уменьшается риск, что мы остановились слишком рано.

Визуализация результатов работы нейросетей отображена на рисунке 25, а их метрики – на рисунке 26.



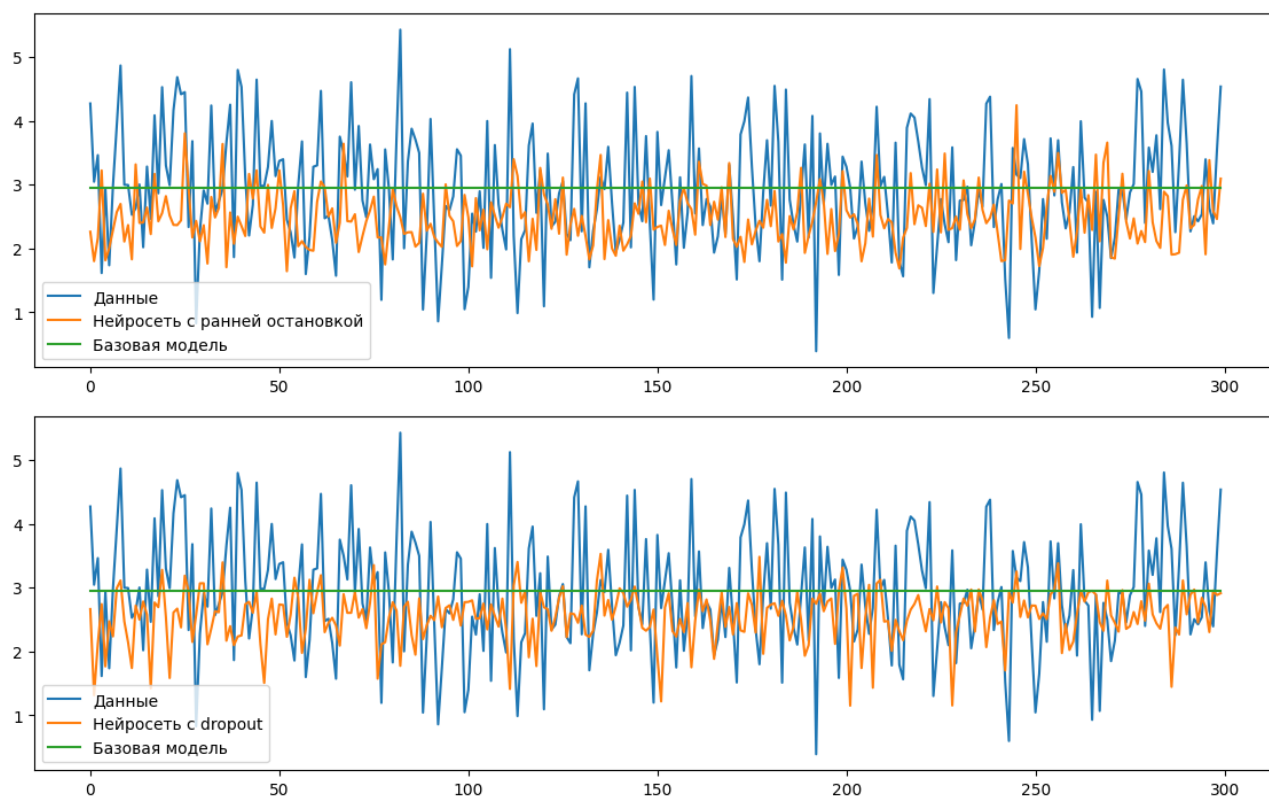


Рисунок 25 – Визуализация работы нейронных сетей

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.000744	-0.924041	-0.739327	-0.340221	-2.554458
Нейросеть переобученная	-0.702315	-1.205174	-0.971796	-0.396919	-3.332204
Нейросеть с ранней остановкой	-0.357113	-1.076064	-0.853269	-0.330199	-2.936997
Нейросеть dropout	-0.377819	-1.084241	-0.847411	-0.342221	-3.708873

Рисунок 26 – Метрики работы нейросетей

Визуализация результатов показывает, что нейросеть из библиотеки tensorflow старалась подстроиться к данным. Выглядят результаты «похоже», но метрики разочаровывают. Лучшая обобщающая способность и меньшие значения ошибок на тестовом множестве оказались у нейросети, обученной с ранней остановкой, но и она предсказывает гораздо хуже базовой модели.

2.4 Оценка точности работы моделей

Согласно заданию, необходимо сравнить ошибку каждой модели на тренировочной и тестирующей части выборки.

Модель для предсказания модуля упругости при растяжении – Lasso(alpha=0.05). Сравнение ее ошибок показано на рисунке 27.

	R2	RMSE	MAE	MAPE	max_error
Модуль упругости, тренировочный	0.029799	-3.079407	-2.464290	-0.033661	-8.937810
Модуль упругости, тестовый	-0.021771	-3.104193	-2.512508	-0.034472	-8.816359

Рисунок 27 – Сравнение ошибок модели для модуля упругости при растяжении на тренировочном и тестовом датасете

Лассо регрессия имеет ошибку на тренировочном датасете меньше, чем на тестовом, потому что чему-то все-таки научилось. Но даже на тренировочном датасете оно не нашло закономерности во входных данных. Задачу решить не удалось.

Модель для предсказания прочности при растяжении – GradientBoostingRegressor (max_depth=1, max_features=1, n_estimators=50). Сравнение ее ошибок показано на рисунке 28.

	R2	RMSE	MAE	MAPE	max_error
Прочность при растяжении, тренировочный	0.056663	-479.002875	-379.538641	-0.166241	-1369.735087
Прочность при растяжении, тестовый	-0.006114	-465.833555	-364.992369	-0.162767	-1399.342924

Рисунок 28 – Сравнение ошибок модели для модуля упругости при растяжении на тренировочном и тестовом датасете

Градиентный бустинг показал положительный, хоть и близкий к 0 коэффициент детерминации. Ошибка на тестовом множестве незначительно больше, чем на тренировочном. Значит, модель нашла следы зависимости, а не выучила данные. Но задача не решена.

Если прочность при растяжении лежит в диапазоне [1071.12-3848.44], то наша модель дает предсказание с точностью ± 1399.34 . Она работает не точнее среднего и бесполезна для применения в реальных условиях.

Модель для предсказания соотношения матрица-наполнитель — нейросеть из tensorflow, обученная с ранней остановкой. Сравнение ее ошибок показано на рисунке 29.

	R2	RMSE	MAE	MAPE	max_error
Соотношение матрица-наполнитель, тренировочный	-0.261320	-1.012517	-0.773151	-0.266820	-3.698861
Соотношение матрица-наполнитель, тестовый	-0.357113	-1.076064	-0.853269	-0.330199	-2.936997

Рисунок 29 – Сравнение ошибок модели для соотношения матрица-наполнитель на тренировочном и тестовом датасете.

У нейросети все показатели для тестовой выборки отличаются в худшую сторону от показателей тренировочной кроме максимальной ошибки. Это может говорить о том, что она не нашла закономерностей, а стала учить данные из тестовой выборки. Возможно, требуется более тщательное и грамотное построение архитектуры нейронной сети, чтобы получить лучший результат. Но сейчас задача далека от решения.

Если соотношение матрица-наполнитель лежит в диапазоне [0.39-5.46], то наша модель может предсказать с точностью ± 2.93 . Она работает не точнее среднего, и бесполезна для применения в реальных условиях.

Заключение

Данная исследовательская работа позволяет сделать некоторые основные выводы по теме. Распределение полученных данных в объединённом датасете близко к нормальному, но коэффициенты корреляции между парами признаков стремятся к нулю. 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 – лассо-регрессия, для прочности при растяжении, МПа – градиентный бустинг.

Был сделан вывод, что невозможно определить из имеющихся 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.

Необходимы дополнительные вводные данные, получение новых результирующих признаков в результате математических преобразований, релевантных доменной области, консультации экспертов предметной области, новые исследования, работа эффективной команды, состоящей из различных учёных.

В целом прогнозирование конечных свойств композитных материалов без изучения материаловедения, погружения в вопрос экспериментального анализа характеристик композитных материалов не демонстрирует сколько-нибудь удовлетворительных результатов. Проработка моделей и построение прогнозов требует внедрения в процесс производных от имеющихся показателей для выявления иного уровня взаимосвязей. Отсюда, также учитывая отсутствие корреляции между признаками, делаем вывод, что текущим набором алгоритмов задача не решается, возможно, решается трудно или не решается совсем.

В настоящее время для ускоренного открытия новых материалов необходима активная работа по пониманию эффективности (теоретической или численной) различных подходов к обратному проектированию для применения машинного обучения на композитах. Хотя сегодняшнее обратное проектирование в настоящее время является сложной задачей в этой области, считается, что нейросети сыграют многообещающую роль в решении этой инженерной задачи в будущем.

Библиографический список

1. Алгоритмы // Loginom. [2023]. [Электронный ресурс] : – Режим доступа: <https://wiki.loginom.ru/algorithms.html> (дата обращения: 04.03.2023).
2. Борис Цейтлин @btseytlin : Нормально разбираемся в Нормальном распределении [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/articles/730936/> (дата обновления: 22.04.2023, дата обращения: 22.04.2023)
3. В. В. Воронина, А. В. Михеев, Н. Г. Ярушкина, К. В. Святков : Теория и практика машинного обучения : учебное пособие / Ульяновск : УлГТУ, 2017. [Электронный ресурс] : – Режим доступа: <http://lib.ulstu.ru/venec/disk/2017/191.pdf> (дата обращения: 16.03.2023)
4. Виталий Радченко @vradchenko : Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес. [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/companies/ods/articles/324402/> (дата обновления: 27.03.2017, дата обращения: 08.04.2023)
5. Дарья Суслова (@darsus) : Подготовка данных в Data Science-проекте: рецепты для молодых хозяек. [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/articles/470650/> (дата обновления: 08.10.2019, дата обращения: 08.04.2023)
6. Документация по языку программирования Python: – Режим доступа: <https://docs.python.org/3.8/index.html>. (дата обращения: 15.03.2023).
7. Документация по библиотеке Numpy: – Режим доступа: <https://numpy.org/doc/1.22/user/index.html#user>. (дата обращения: 16.03.2023).
8. Документация по библиотеке Pandas: – Режим доступа: https://pandas.pydata.org/docs/user_guide/index.html#user-guide. (дата обращения: 16.03.2023).
9. Документация по библиотеке Matplotlib: – Режим доступа: <https://matplotlib.org/stable/users/index.html>. (дата обращения: 15.03.2023)

10. Документация по библиотеке Keras: – Режим доступа: <https://keras.io/api/>. (дата обращения: 17.03.2023).
11. Документация по библиотеке Scikit-learn: – Режим доступа: https://scikit-learn.org/stable/user_guide.html. (дата обращения: 15.03.2023).
12. Документация по библиотеке Seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>. (дата обращения: 15.03.2023).
13. Документация по библиотеке Tensorflow: – Режим доступа: <https://www.tensorflow.org/overview> (дата обращения: 18.03.2023).
14. Композитный материал // Википедия. [2023]. [Электронный ресурс] : – Режим доступа: <https://ru.wikipedia.org/?curid=11119&oldid=128527945> (дата обновления: 15.02.2023, дата обращения: 04.03.2023).
15. Олег Седухин @boygenius : Теория вероятностей в машинном обучении. Часть 1: модель регрессии. [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/companies/ods/articles/324402/> (дата обновления: 31.01.2023, дата обращения: 18.03.2023)
16. Станислав Калинин @befuddle : Сверточная сеть на python. Часть 3. Применение модели. [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/companies/ods/articles/344888/> (дата обновления: 25.12.2017, дата обращения: 14.04.2023)
17. Рубаков Сергей Валерьевич : Современные методы анализа данных // Управление наукой и наукометрия. 2008. №7. [Электронный ресурс] : – Режим доступа: <https://cyberleninka.ru/article/n/sovremennyye-metody-analiza-dannyh> (дата обращения: 16.03.2023).
18. Экспресс-анализ данных на Python. @DmitriyB_33 [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/articles/729292/> (дата обновления: 15.04.2023, дата обращения: 16.04.2023)
19. Chen, C., & Gu, G. (2019). Machine learning for composite materials. MRS Communications, 9(2), 556-566. doi:10.1557/mrc.2019.32 [Электронный ресурс] : – Режим доступа: <https://www.cambridge.org/core/journals/mrs->

communications/article/machine-%20learning-for-composite-materials/F54F60A
C0048291BA47E0B671733ED15 (дата обращения: 23.04.2023)