

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»

Слушатель

Кислицин Никита Сергеевич

Москва, 2022

Содержание

1. Введение.....	3
2. Аналитическая часть.....	5
2.1. Постановка задачи	5
2.2. Описание используемых методов	7
2.3. Разведочный анализ данных.....	15
3. Практическая часть.....	22
3.1. Предобработка данных.....	22
3.2. Разработка и обучение модели	24
3.3. Тестирование модели	26
3.4. Разработка нейронной сети.....	31
3.5. Тестирование модели	36
4. Использование репозитория	37
5. Выводы по результатам работы	38
6. Библиографический список	39
Приложение А	41

1. Введение

Пояснительная записка подготовлена в рамках выпускной квалификационной работы по курсу «Data Science». При выполнении данной работы проводился анализ данных с использованием методов машинного обучения, реализованных в программной среде Python. Исследуемый набор данных содержал значения определенных параметров компонент композиционных материалов.

Многие материалы обладают определенным комплексом свойств. Например, бетон, обладая высокой жесткостью и хрупкостью, отлично работает на сжатие и не работает на растяжение. В этой связи бетонные конструкции давно используют в качестве фундаментов и различных опор. В свою очередь, металлы обладают пластичностью, высокой прочностью и достаточно хорошо работают на растяжение. Поэтому из материала, состоящего из металла и бетона («железобетон») изготавливают конструкции, имеющие достаточную жесткость и одновременно работающие на растяжение (перекрытия, балки, пролеты мостов и др.). Такие материалы, сочетающие в себе свойства, присущие нескольким материалам, обычно называются композиционными материалами.

Композиционные материалы (КМ) - это материалы, обладающие следующей совокупностью признаков:

- состоят из двух или более компонентов, различающихся по своему химическому составу и разделенных выраженной границей;
- имеют новые свойства, отличающиеся от свойств, составляющих эти материалы компонентов;
- неоднородны в микромасштабе и однородны в макромасштабе;
- свойства определяются каждым из компонентов, которые в связи с этим должны содержаться в материале в достаточно большом количестве (больше некоторого критического значения).

Компонент, непрерывный во всем объеме КМ, называется матрицей, а прерывистый, разъединенный в объеме композиции элемент (элементы) называется армирующим элементом (наполнителем). Матрица придает изделию

из КМ заданную форму и монолитность, обеспечивая передачу и перераспределение нагрузки по объему материала, защищает армирующие элементы от внешних воздействий. Армирующие или упрочняющие элементы распределены в матрице равномерно. Они, как правило, обладают высокой прочностью, твердостью, большим модулем упругости и по этим показателям значительно превосходят матрицу. Армирующие элементы вводят в композиционный материал с целью изменения его свойств (увеличения прочности, жесткости и пластичности; изменения плотности, электрических, теплофизических и других характеристик в различных направлениях и отдельных местах изделия). Выбор состава композиции - результат оптимизации объемного содержания матрицы и армирующих элементов.

С появлением композиционных материалов стал возможным селективный выбор свойств композитов, необходимых для каждой конкретной области их применения, и возникла потребность в проектировании таких материалов. Процесс создания композиционного материала включает следующие стадии: формирование проектных исходных данных; выбор состава композита и технологии его производства; оценка основных свойств созданного материала и сравнение их с заданием на проектирование. Для оценки основных свойств нового материала есть два пути: физические испытания образцов материалов, или прогнозирование характеристик. Суть прогнозирования заключается в симуляции представительного элемента объема композита, на основе данных о характеристиках входящих компонентов (матрицы и армирующих компонентов).

2. Аналитическая часть

2.1. Постановка задачи

Кейс основан на реальных производственных задачах Центра НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана).

Входные данные: начальные свойства компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.).

Необходимо спрогнозировать ряд конечных свойств получаемых композиционных материалов. Созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

Порядок выполнения выпускной работы:

- 1) Изучение теоретических основ и методов решения поставленной задачи;
- 2) Проведение разведочного анализа данных, в т.ч.:
 - а) Создание для каждой переменной гистограмм распределения и диаграмм «ящичков с усами», попарных графиков рассеяния точек;
 - б) Получение для каждой колонки датасета среднего и медианного значения;
 - в) Проведение анализа и исключения выбросов;
 - г) Проверка на наличие пропусков.
- 3) Проведение предобработки данных (удаление шумов, нормализация и т.д.);
- 4) Обучение нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении. При построении модели необходимо 30% данных оставить на тестирование модели, на остальных происходит обучение моделей. При построении моделей провести поиск гиперпараметров

модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10;

5) Написание нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель;

6) Разработка приложения с графическим интерфейсом или интерфейсом командной строки, которое будет выдавать прогноз, полученный в задании 4 или 5 (один или два прогноза, на выбор);

7) Оценка точности модели на тренировочном и тестовом датасете;

8) Создание репозитория в GitHub/GitLab и размещение в нем кода исследования. Оформление файла README.

Входные данные предоставлены в виде двух Excel-файлов.

Файл X_br.xlsx содержит таблицу с 1023 наборами измерений десяти свойств композитов:

- 1) Соотношение матрица-наполнитель;
- 2) Плотность, кг/м³;
- 3) Модуль упругости, ГПа;
- 4) Количество отвердителя, м;
- 5) Содержание эпоксидных групп, %₂;
- 6) Температура вспышки, С₂;
- 7) Поверхностная плотность, г/м²;
- 8) Модуль упругости при растяжении, ГПа;
- 9) Прочность при растяжении, МПа;
- 10) Потребление смолы, г/м².

Файл X_nir.xlsx содержит таблицу со 1040 наборами измерений трех свойств композитов:

- 1) Угол нашивки, град;
- 2) Шаг нашивки;
- 3) Плотность.

Согласно постановке задачи, эти таблицы необходимо было объединить по индексам в единый датасет, используя тип объединения INNER. Т.о., семнадцать

наборов измерений из файла X_pur.xlsx не были добавлены в единый датасет. Это составило 1,6% от общего числа данных в этом файле, что можно принять за несущественную потерю.

В итоге искомый рабочий датасет состоит из 1023 строк (наборов измерений параметров композитов) и 13 колонок (параметров композитов). Три параметра в зависимости от решаемой задачи будут становиться попеременно выходными прогнозируемыми переменными, а именно:

- 1) Модуль упругости при растяжении, ГПа;
- 2) Прочность при растяжении, МПа;
- 3) Соотношение матрица-наполнитель.

2.2. Описание используемых методов

Методы машинного обучения классифицируются согласно объему и типу контроля, которым они подвергаются во время обучения. Основная масса задач, решаемых при помощи методов МО относятся к трем видам: обучение с учителем, обучение без учителя и обучение с подкреплением. В данной работе решается типичная задача обучения с учителем - прогноз целевого числового значения переменной, имеющей связь с одной или несколькими независимыми переменными, называемыми также предикторами. Задачу подобного рода также называют регрессией.

При выполнении работы были использованы следующие алгоритмы машинного обучения:

- 1) линейная регрессия (linear regression),
- 2) случайный лес (random forest),
- 3) k ближайших соседей ((k-nearest neighbors),
- 4) градиентный бустинг (gradient boosting),
- 5) нейронная сеть (многослойный персептрон, MLP).

Линейная регрессия.

В модели линейной регрессии есть два вида переменных:

- вход или переменная предиктора (X) - переменная, которая помогает предсказать значение выходной переменной.
- выходная переменная (Y) - переменная, которую мы хотим предсказать.

Оценка Y выполняется через линейную функцию:

$$Y_e = \alpha + \beta X$$

где Y_e является оценочной или прогнозируемой стоимостью Y на основе нашего линейного уравнения.

Цель - найти статистически значимые значения параметра α , а также β , что минимизирует разницу между Y_a и Y_e .

Если мы сможем определить оптимальные значения этих двух параметров, то у нас будет линия наилучшего соответствия, которую мы можем использовать для прогнозирования значений Y , учитывая значения X .

Преимущества:

- Просто и легко понять
- Дешевые вычислительные затраты
- Основа для более сложных алгоритмов машинного обучения

Недостатки:

- Чувствителен к выбросам и шумам
- Не может использоваться, когда связь между зависимой и независимой переменной не является линейной.

Случайный лес.

Это тип контролируемого алгоритма машинного обучения, основанного на ансамблевом обучении, при котором объединяют различные типы алгоритмов или один и тот же алгоритм несколько раз, чтобы сформировать более мощную модель прогнозирования. Данный алгоритм включает несколько алгоритмов одного и того же типа, т.е. несколько решений *деревьев*, в результате чего

получается *лес деревьев*, отсюда и название “Случайный лес”. Алгоритм случайного леса может быть использован как для регрессионных, так и для классификационных задач.

Основные шаги, связанные с выполнением алгоритма случайного леса:

1. Выберите N случайных записей из набора данных.
2. Постройте дерево решений на основе этих N записей.
3. Выберите нужное количество деревьев в вашем алгоритме и повторите шаги 1 и 2.
4. В случае регрессионной задачи для новой записи каждое дерево в лесу предсказывает значение Y (выход). Конечное значение можно вычислить, взяв среднее значение всех значений, предсказанных всеми деревьями в лесу. Или, в случае проблемы классификации, каждое дерево в лесу предсказывает категорию, к которой принадлежит новая запись. Наконец, новый рекорд присваивается той категории, которая получает большинство голосов.

Преимущества:

- Не является предвзятым, поскольку существует несколько деревьев, и каждое дерево обучается на подмножестве данных. В принципе, алгоритм случайного леса опирается на силу “толпы”, поэтому общая предвзятость алгоритма уменьшается.
- Очень стабилен. Даже если новая точка данных введена в набор данных, общий алгоритм не сильно пострадает, так как новые данные могут повлиять на одно дерево, но ему очень трудно повлиять на все деревья.
- Хорошо работает, когда есть как категориальные, так и числовые признаки.
- Хорошо работает, когда данные имеют пропущенные значения или они не были хорошо.

Недостатки:

- Сложность. Этот метод требует гораздо больше вычислительных ресурсов из-за большого количества деревьев решений, соединенных вместе.

- Из-за своей сложности они требуют гораздо больше времени для обучения, чем другие сопоставимые алгоритмы.

Метод k-ближайших соседей

Еще один метод классификации, который адаптирован для регрессии - метод k-ближайших соседей (K-Nearest Neighbors). На интуитивном уровне суть метода проста: посмотри на соседей вокруг, какие из них преобладают, таковым ты и являешься.

В случае использования метода для регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны.

Для реализации метода необходима метрика расстояния между объектами. Используется, например, евклидово расстояние для количественных признаков или расстояние Хэмминга для категориальных.

Этот метод — пример непараметрической регрессии.

Преимущества:

- Простая реализация;
- Как правило, метод хорош для первого решения задачи, причем не только классификации или регрессии, но и, например, рекомендации;

Недостатки:

- Метод считается быстрым в сравнении, например, с ансамблевыми алгоритмами, но в реальных задачах, как правило, число соседей, используемых для классификации, будет большим (100-150), и в таком случае алгоритм будет работать не так быстро, как дерево решений;
- Если в наборе данных много признаков, то трудно подобрать подходящие веса и определить, какие признаки не важны для классификации/регрессии;
- Зависимость от выбранной метрики расстояния между примерами;
- Нет теоретических оснований выбора определенного числа соседей - только перебор (впрочем, чаще всего это верно для всех

гиперпараметров всех моделей). В случае малого числа соседей метод чувствителен к выбросам, то есть склонен переобучаться;

- Как правило, плохо работает, когда признаков много, из-за "проклятия размерности".

Градиентный бустинг

Градиентный бустинг - еще один представитель ансамблевых методов.

В отличие от случайного леса, где каждый базовый алгоритм строится независимо от остальных, бустинг воплощает идею последовательного построения линейной комбинации алгоритмов. Каждый следующий алгоритм старается уменьшить ошибку предыдущего.

Чтобы построить алгоритм градиентного бустинга, нам необходимо выбрать базовый алгоритм и функцию потерь или ошибки (loss). Loss-функция – это мера, которая показывает насколько хорошо предсказание модели соответствует данным. Используя градиентный спуск и обновляя предсказания, основанные на скорости обучения (learning rate), ищем значения, на которых loss минимальна.

Бустинг, использующий деревья решений в качестве базовых алгоритмов, называется градиентным бустингом над решающими деревьями. Он отлично работает на выборках с «табличными», неоднородными данными и способен эффективно находить нелинейные зависимости в данных различной природы. На настоящий момент это один из самых эффективных алгоритмов машинного обучения. Благодаря этому он широко применяется во многих конкурсах и промышленных задачах. Он проигрывает только нейросетям на однородных данных (изображения, звук и т. д.).

Преимущества:

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

Недостатки:

- Алгоритм крайне чувствителен к выбросам и при их наличии будет тратить огромное количество ресурсов на эти моменты.
- Большие затраты времени на вычисления
- Необходимо грамотно подбирать гиперпараметры.

Нейронные сети

Нейронная сеть — это последовательность нейронов, соединенных между собой связями. Вычислительная единица нейронной сети — нейрон или персептрон.

У каждого нейрона есть определённое количество входов, куда поступают сигналы, которые суммируются с учётом значимости (веса) каждого входа.

Смещение — это дополнительный вход для нейрона, который всегда равен 1 и, следовательно, имеет собственный вес соединения.

Так же у нейрона есть функция активации, которая определяет выходное значение нейрона. Она используется для того, чтобы ввести нелинейность в нейронную сеть. Примеры активационных функций: relu, сигмоида.

У полносвязной нейросети выход каждого нейрона подается на вход всем нейронам следующего слоя. У нейросети имеется:

- входной слой - его размер соответствует входным параметрам;
- скрытые слои - их количество и размерность определяем специалист;
- выходной слой - его размер соответствует выходным параметрам.

Прямое распространение — это процесс передачи входных значений в нейронную сеть и получения выходных данных, которые называются прогнозируемым значением.

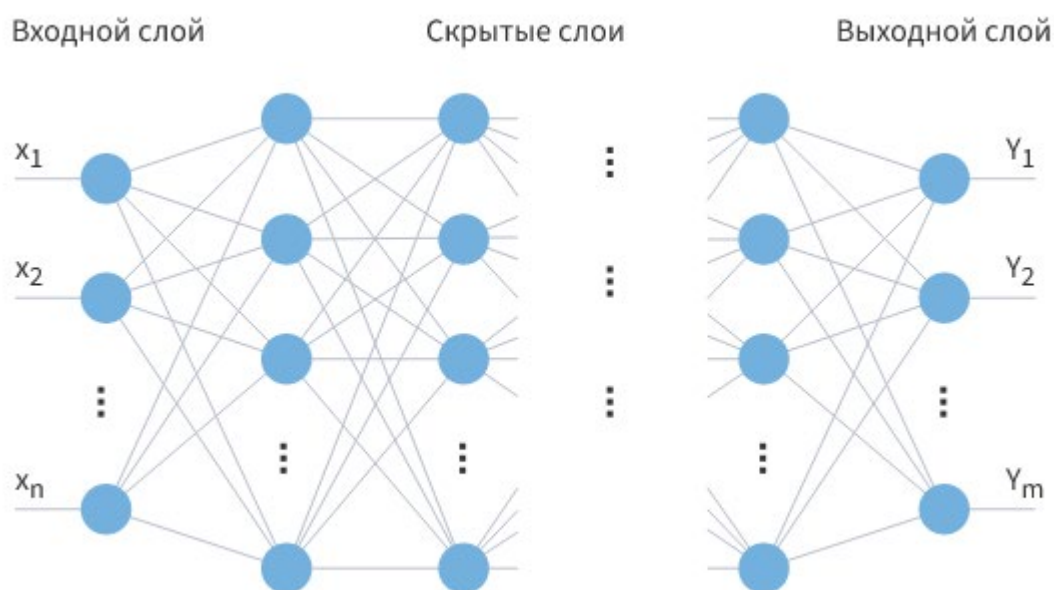


Рисунок 1 – Нейронная сеть

Прогнозируемое значение сравниваем с фактическим с помощью функции потерь. В методе обратного распространения ошибки градиенты (производные значений ошибок) вычисляются по значениям весов в направлении, обратном прямому распространению сигналов. Значение градиента вычитают из значения веса, чтобы уменьшить значение ошибки. Таким образом происходит процесс обучения. Обновляются веса каждого соединения, чтобы функция потерь минимизировалась.

Для обновления весов в модели используются различные оптимизаторы.

Количество эпох показывает, сколько раз выполнялся проход для всех примеров обучения.

Нейронные сети применяются для решения задач регрессии, классификации, распознавания образов и речи, компьютерного зрения и других. На настоящий момент это самый мощный, гибкий и широко применяемый инструмент в машинном обучении.

Многослойный персептрон — это класс искусственных нейронных сетей прямого распространения, состоящих как минимум из трех слоёв: входного, скрытого и выходного. За исключением входных, все нейроны используют нелинейную функцию активации. Необходимость в большом количестве

обучаемых слоёв отпадает, так как теоретически единственного скрытого слоя достаточно, чтобы перекодировать входное представление таким образом, чтобы получить линейную разделимость для выходного представления. Существует предположение, что, используя большее число слоёв, можно уменьшить число элементов в них, то есть суммарное число элементов в слоях будет меньше, чем если использовать один скрытый слой.

Персептроны часто применяются для решения контролируемых задач обучения: они тренируются по набору пар входных/выходных объектов и учатся моделировать корреляции (т. е. зависимости) между этими данными. Обучение включает в себя настройку параметров модели (весовых коэффициентов, смещений) для минимизации погрешности. Для корректировки этих параметров относительно погрешности используется алгоритм обратного распространения, а сама погрешность может быть вычислена различными способами, в том числе путем вычисления среднеквадратичного отклонения (RMSE).

Недостатки:

Выигрывая в скорости работы путем вытравливания алгоритмов, вы проигрываете в возможности их модификации. Это является реальной проблемой в машинном обучении, где алгоритмы самопроизвольно изменяются по мере обработки данных. Задача состоит в том, чтобы найти те части алгоритма, которые остаются стабильными даже при изменении параметров, например, операции с линейной алгеброй, которые в настоящее время обрабатываются GPU быстрее всего.

2.3. Разведочный анализ данных

Разведочный анализ данных (Exploratory Data Analysis, EDA) - это общий подход к исследованию наборов данных с помощью простой сводной статистики и графических визуализаций для более глубокого понимания данных. Он помогает в последующем более эффективно анализировать и моделировать данные.

Для получения статистики по набору данных использовались следующие команды библиотеки pandas для работы с данными:

- 1) `df.info()` - вывод информации о типах переменных;
- 2) `df.isnull().sum()` - вывод информации о количестве пропусков;
- 3) `df.duplicated().sum()` - количество полностью совпадающих строк;
- 4) `df.shape` - информация о количестве наблюдений и количестве переменных;
- 5) `df.nunique()` - количество уникальных значений по каждой переменной;
- 6) `df.columns.tolist()` - использовалась для построения цикла подсчета повторяющихся значений в каждой переменной;
- 7) `df.describe()` - вывод статистик по количественным переменным:
 - `count` - количество значений;
 - `mean` - среднее арифметическое значение;
 - `std` – среднее квадратическое (стандартное) отклонение;
 - `min` - минимальное значение;
 - `max` - максимальное значение;
 - `25%` - верхнее значение 1-го квартиля;
 - `50%` - верхнее значение 2-го квартиля (медиана);
 - `75%` - верхнее значение 3-го квартиля.

Сбор статистики показал, что в наборе данных все параметры имеют количественные значения (вещественные числа), параметров с качественными значениями нет. В наборе данных отсутствуют пропуски (нулевые значения) и строки-дубликаты. Также в нем нет бесполезных для анализа данных, т.е. таких

параметров, у которых уникальных значений столько же, сколько и наблюдений, а также параметров только с одним уникальным значением (параметры-константы).

Таблица 1 – Уникальные значения и повторы данных

Параметр	Исходный файл	Уникальные значения, шт.	Списки одинаковых значений (число показывает кол-во повторов)
Угол нашивки, град	X_nup	2	520, 503
Плотность нашивки	X_nup	988	10, 10, 10, 9
Шаг нашивки	X_nup	989	8, 8, 8, 8, 7
Соотношение матрица-наполнитель	X_br	1014	4, 2, 2, 2, 2
Температура вспышки, C_2	X_br	1003	12, 10
Содержание эпоксидных групп,%_2	X_br	1004	12, 9
Количество отвердителя, м.%	X_br	1005	12, 8
Потребление смолы, г/м2	X_br	1004	14, 5, 4
Поверхностная плотность, г/м2	X_br	1004	9, 5, 5, 4
Модуль упругости при растяжении, Гпа	X_br	1004	9, 5, 5, 4
Прочность при растяжении, Мпа	X_br	1004	9, 5, 5, 4
Плотность, кг/м3	X_br	1013	6
модуль упругости, Гпа	X_br	1020	4

В наборе данных явно выделяется параметр *Угол нашивки, град*, имеющий только два значения – 0 и 90 градусов.

Кроме того, практически полное совпадения количества повторяющихся значений по отдельным параметрам (см. Таблицу 1) позволяет сделать предположение о возможной корреляции свойств материалов по этим параметрам:

- *Плотность нашивки <-> Шаг нашивки;*
- *Содержание эпоксидных групп,%_2 <-> Количество отвердителя, м.% <-> Температура вспышки, C_2;*
- *Поверхностная плотность, г/м2 <-> Модуль упругости при растяжении, Гпа <-> Прочность при растяжении, Мпа.*

Таблица 2 – Описательная статистика

Параметр	count	std	min	mean	max	25%	50%	75%
Соотношение матрица-наполнитель	1023	0.91	0.39	2.93	5.59	2.32	2.91	3.55
Плотность, кг/м3	1023	73.73	1731.8	1975.7	2207.8	1924.2	1977.6	2021.4
модуль упругости, ГПа	1023	330.23	2.44	739.92	1911.5	500.05	739.66	961.81
Количество отвердителя, м.%	1023	28.3	17.74	110.57	198.95	92.44	110.56	129.73
Содержание эпоксидных групп,%_2	1023	2.41	14.25	22.24	33	20.61	22.23	23.96
Температура вспышки, С 2	1023	40.94	100	285.88	413.27	259.07	285.9	313
Поверхностная плотность, г/м2	1023	281.31	0.6	482.73	1399.5	266.82	451.86	693.23
Модуль упругости при растяжении, ГПа	1023	3.12	64.05	73.33	82.68	71.25	73.27	75.36
Прочность при растяжении, МПа	1023	485.63	1036.9	2466.9	3848.4	2135.9	2459.5	2767.2
Потребление смолы, г/м2	1023	59.74	33.8	218.42	414.59	179.63	219.2	257.48
Угол нашивки, град	1023	45.02	0	44.25	90	0	0	90
Шаг нашивки	1023	2.56	0.1	6.9	14.44	5.08	6.92	8.59
Плотность нашивки	1023	12.35	0.1	57.15	103.99	49.8	57.34	64.94

В данных описательной статистики (см. Таблицу 2) обращает на себя внимание большой разброс значений у параметров *модуль упругости, ГПа* и *Поверхностная плотность, г/м2*. Средние значения и медианы (50%) различаются незначительно у большинства параметров. Исключение опять же у параметра *Поверхностная плотность, г/м2* - у него среднее значение существенно больше медианы, т.е. по этому параметру распределение данных имеет перекося влево.

Гистограммы распределения данных приведены на Рисунке 1. Они показывают, что данные всех параметров, кроме *Поверхностной плотности* имеют распределение, близкое к нормальному. Параметр *Поверхностная плотность* имеет распределение со смещением влево, что говорит о преобладании данных с меньшим показателем поверхностной плотности.

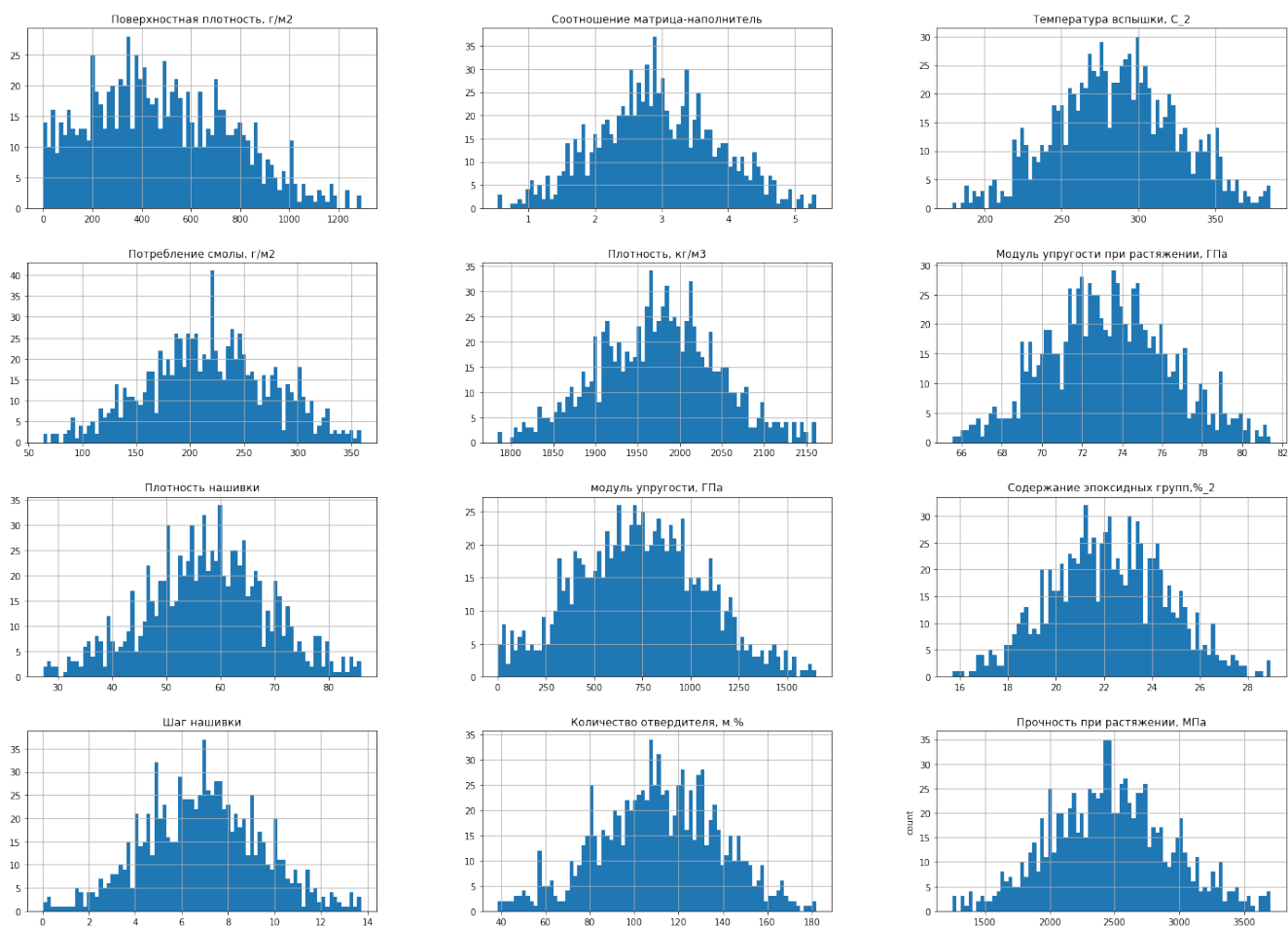


Рисунок 2 - Гистограммы распределения

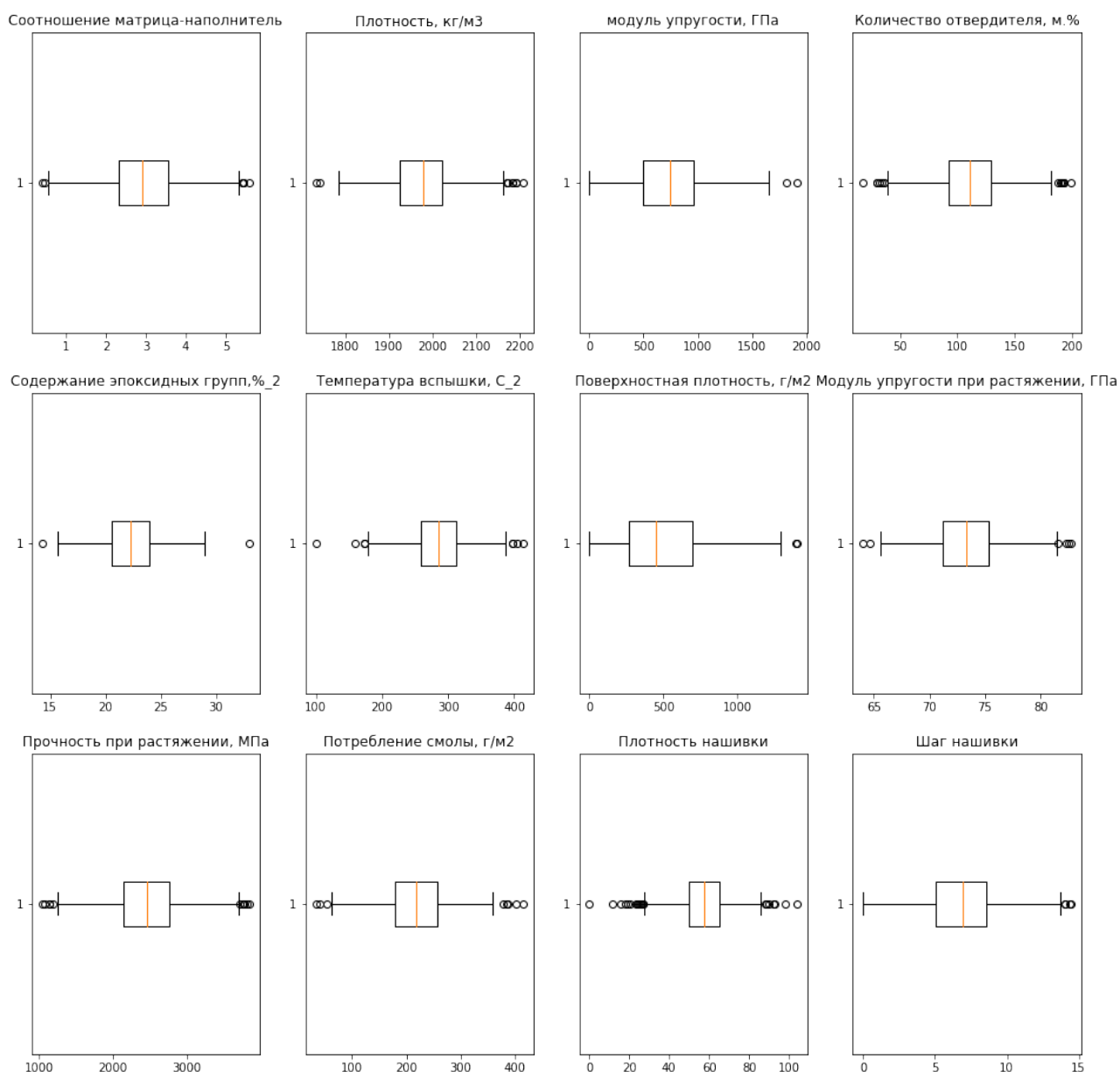


Рисунок 3 – Диаграммы размаха

Диаграммы размаха, или "ящики с усами" приведены на Рисунке 2. Из них видно, что все параметры имеют выбросы, причем большинство из них с обеих сторон - как в наименьших, так и в наибольших значениях этих параметров.

Для параметра *Угол нашивки*, имеющего только два значения, гистограмма и диаграмма размаха не выводились в виду их малой информативности.

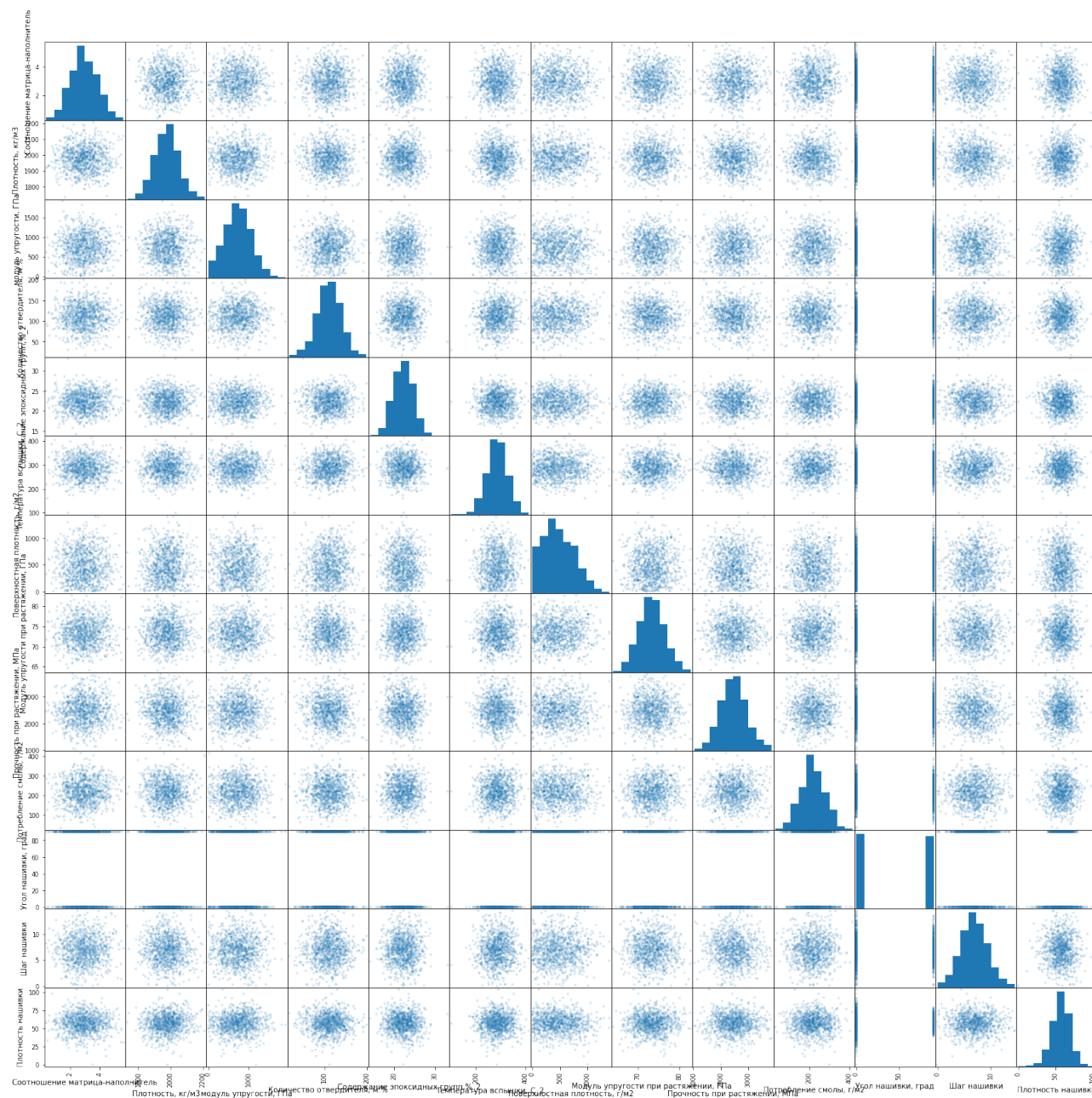


Рисунок 4 – Попарные графики рассеяния

Попарные графики рассеяния показывают отсутствие явно выраженной линейной зависимости между параметрами.

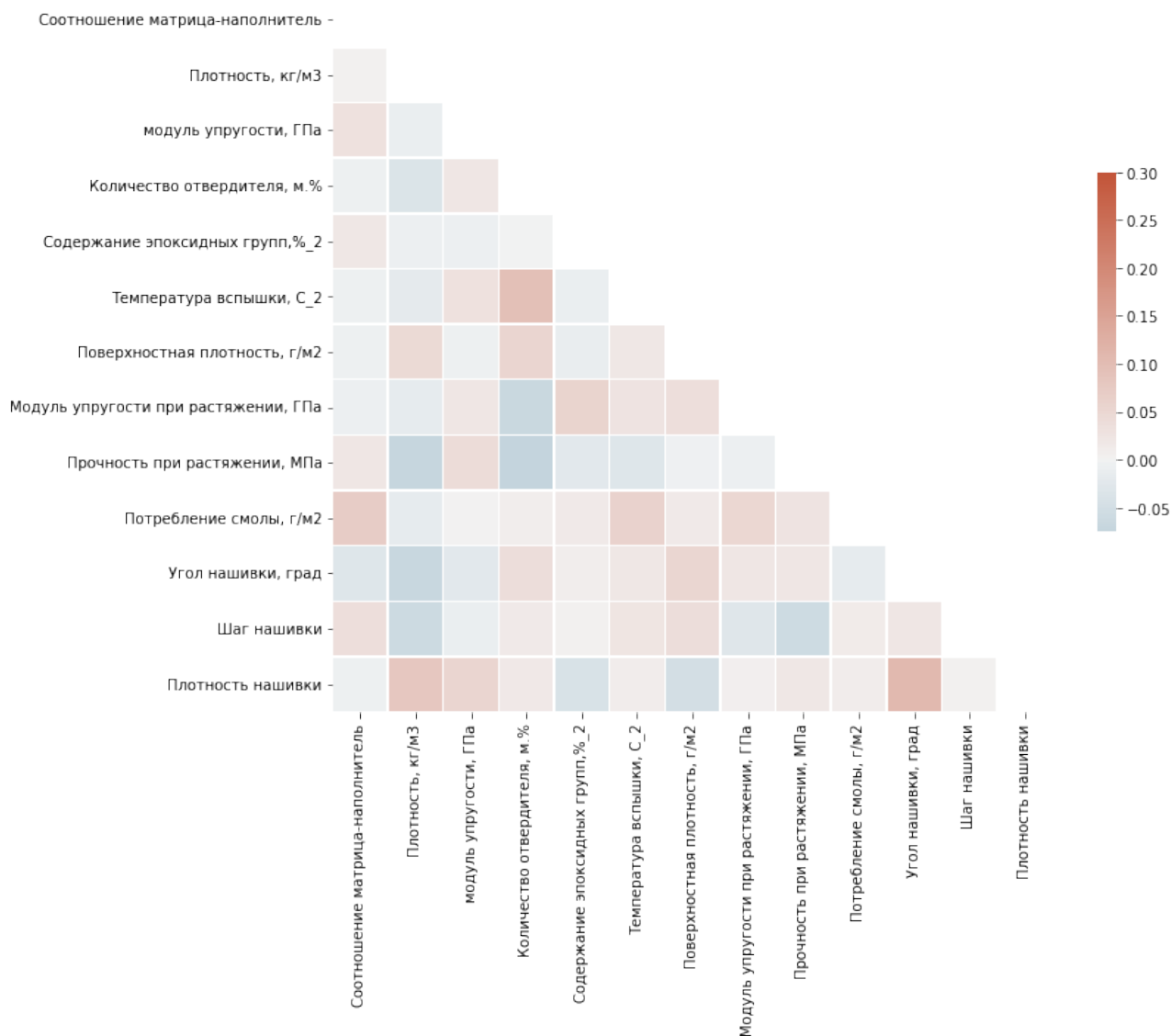


Рисунок 5 – Тепловая карта

По тепловой карте также видно, что зависимости между параметрами набора данных выражены очень слабо. Наибольшие положительные коэффициенты корреляции зафиксированы между *Плотностью нашивки* и *Углом нашивки* (равен +0,11) и между *Температура вспышки* и *Количество отвердителя* (равен +0,10). А наибольший отрицательный коэффициент корреляции - между *Прочностью при растяжении* и *Количеством отвердителя* (равен -0,08).

На этом разведочный анализ данных был завершен.

3. Практическая часть

3.1. Предобработка данных

Удаление выбросов

В ходе проведения разведочного анализа с помощью диаграмм размаха было визуально установлено наличие в рабочем наборе данных выбросов. Для их нахождения был использован метод межквартильных интервалов (InterQuartile Range, IQR).

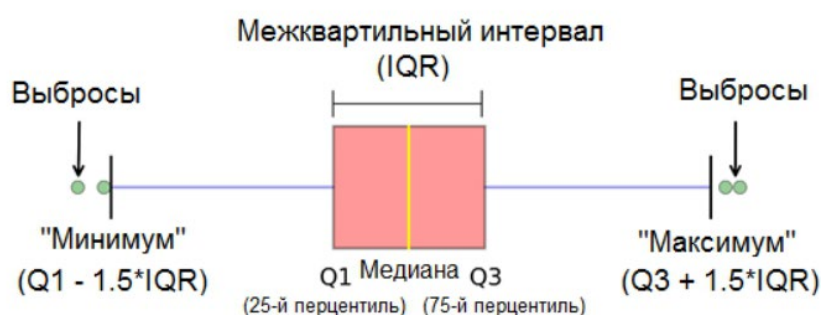


Рис. 6 Расчет выбросов по методу IQR

С его помощью было установлено, что в рабочем наборе данных имеется 93 выброса, 9% от общего количества данных.


	<code>df.isnull().sum()</code>	
	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	

Рис. 7 - Количество выбросов в каждом параметре

Все выбросы были помечены как NaN («не число») и удалены с помощью функции библиотеки pandas `dropna()`.

Нормализация

Также при выполнении разведочного анализа данных было замечено, что значения данных изменяются в очень больших диапазонах и также у разных параметров отличаются на порядки. Это может приводить к некорректной работе моделей машинного обучения – большой дисбаланс между значениями признаков может ухудшать результаты обучения и замедлять сам процесс моделирования. Поэтому данные были нормализованы с использованием метода MinMaxScaler из библиотеки Sklearn. Т.к. в нашем наборе данных нет отрицательных значений, то этот метод отмасштабировал все данные от 0 до 1.

В таблице 3 приведена описательная статистика после нормализации.

Таблица 3 – Описательная статистика после нормализации

Параметр	count	mean	std	min	0.25	0.50	0.75	max
Соотношение матрица-наполнитель	936	0.50	0.19	0	0.37	0.49	0.63	1
Плотность, кг/м3	936	0.50	0.19	0	0.37	0.51	0.63	1
модуль упругости, ГПа	936	0.45	0.20	0	0.30	0.45	0.58	1
Количество отвердителя, м. %	936	0.50	0.19	0	0.38	0.51	0.64	1
Содержание эпоксидных групп, %_2	936	0.49	0.18	0	0.37	0.49	0.62	1
Температура вспышки, С_2	936	0.52	0.19	0	0.39	0.52	0.65	1
Поверхностная плотность, г/м2	936	0.37	0.22	0	0.21	0.35	0.54	1
Модуль упругости при растяжении, ГПа	936	0.49	0.19	0	0.36	0.49	0.62	1
Прочность при растяжении, МПа	936	0.50	0.19	0	0.37	0.49	0.61	1
Потребление смолы, г/м2	936	0.52	0.20	0	0.39	0.52	0.65	1
Угол нашивки, град	936	0.51	0.50	0	0.00	1.00	1.00	1
Шаг нашивки	936	0.50	0.18	0	0.37	0.50	0.62	1
Плотность нашивки	936	0.51	0.19	0	0.39	0.52	0.64	1

Нормализация изменяет только диапазон величин, в пределах которого лежат данные, и не меняет форму распределения внутри этого диапазона. Это наглядно подтверждают гистограммы, сделанные после нормализации – их форма не изменилась.

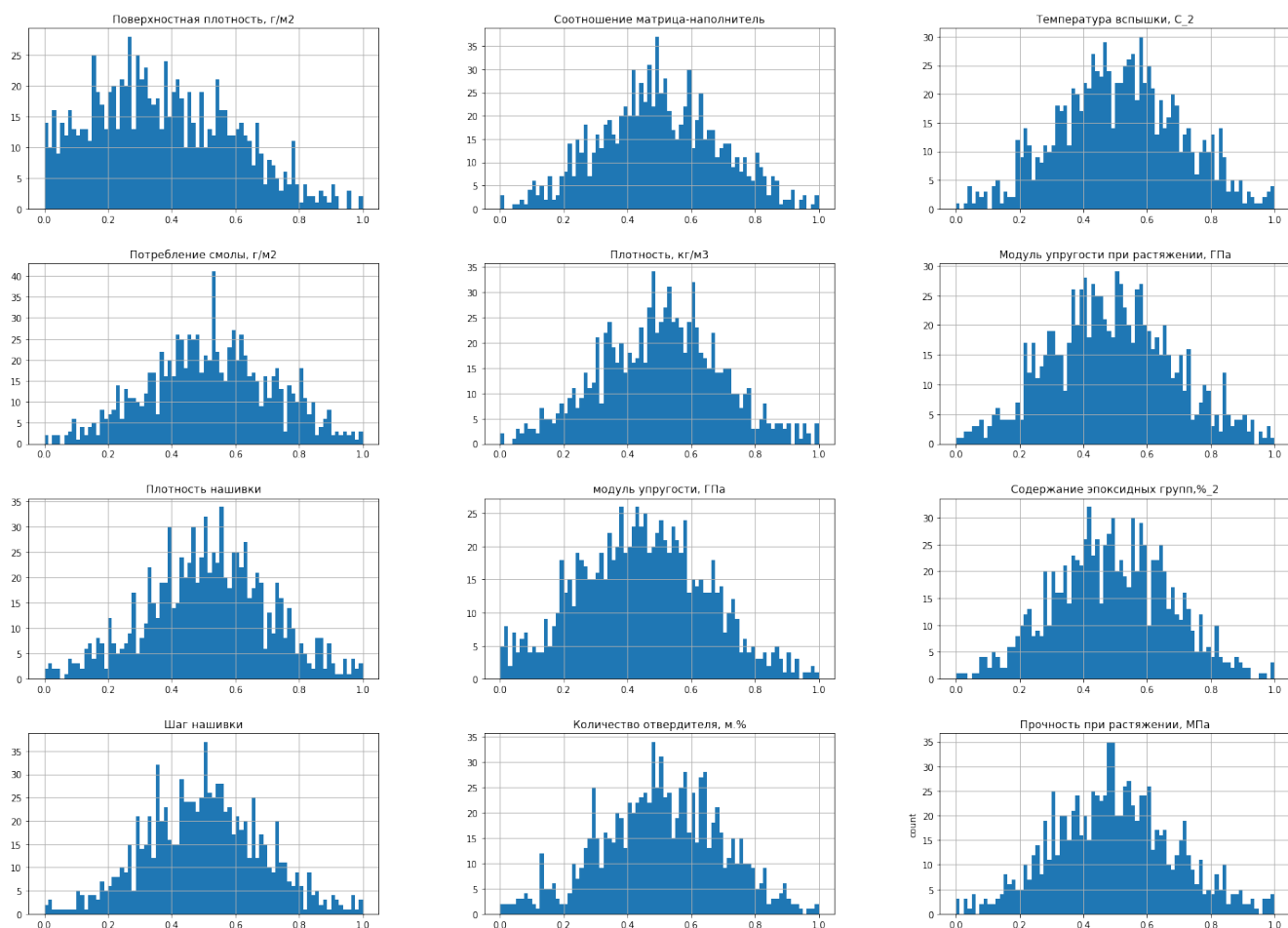


Рисунок 7 - Гистограммы распределения после нормализации

3.2. Разработка и обучение модели

Данный раздел выпускной работы согласно постановке задачи выполнялся для двух параметров нормализованного набора данных по очереди:

1. *Модуль упругости при растяжении, ГПа*
2. *Прочность при растяжении, Мпа*

Т.е. рабочий набор данных в каждом случае разделялся на две части. В одну часть (X – прогнозирующая переменная, или предиктор) отбирались все значения набора данных, кроме значений того параметра, которые будет

предсказываться, а в другую (y – целевая переменная) - значения только предсказываемого параметра. Затем обе части были разделены обучающую (train) и тестовую (test) выборки в соотношении 70% на 30%:

Таблица 4 - Размер тестовой и обучающей выборок

Размерность обучающей выборки для Модуля упругости при растяжении: 655
Размерность тестовой выборки для Модуля упругости при растяжении: 281
Размер обучающей выборки для Прочности при растяжении: 655
Размер тестовой выборки для Прочности при растяжении: 281

Для построения прогноза значений целевых параметров применялись алгоритмы, краткое описание которых приведено в разделе 2.2 выпускной работы.

На первом этапе обучение модели выполнялось с параметрами по умолчанию. Затем производился поиск лучших значений гиперпараметров для каждой модели с использованием класса GridSearchCV из библиотеки Sckit-Learn. GridSearchCV выполняет поиск лучших параметров по сетке всех возможных их значений (grid search) и применяет перекрестную проверку (cross-validation) для выбора наилучших значений. Количество разрезов кросс-валидации, которые необходимо сделать, было взято равным 10, как и указано в постановке задачи.

Т.к. GridSearchCV ищет наилучшие значения гиперпараметров путем обычного перебора (т.е. создает модель для каждой возможной комбинации параметров), то время его работы значительно возрастает с увеличением количества передаваемых для поиска гиперпараметров. Поэтому для алгоритма Random Forest вместо GridSearchCV использовался класс RandomizedSearchCV. Его отличие в том, что перебирается определенное количество случайных комбинаций случайного значения для каждого гиперпараметра.

3.3. Тестирование модели

Для оценки результатов прогноза, которые сделали модели, использующие выбранные в п.3.2 алгоритмы в каждом случае были определены наиболее типичные для задач регрессии метрики:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}|$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

Where,

\hat{y} – predicted value of y

\bar{y} – mean value of y

Рис 8 - Метрики в задачах регрессии

1) MAE (Mean Absolute Error), средняя абсолютная ошибка - показывает среднее абсолютное отклонение предсказанных значений от реальных. Чем выше её значение, тем модель хуже. У идеальной модели MAE=0.

2) MSE (Mean Squared Error), среднеквадратичная ошибка - показывает средний квадрат отклонений предсказанных значений от реальных. Чем выше значение MSE, тем модель хуже. У идеальной модели MSE=0. MSE больше учитывает сильные отклонения и поэтому более чувствителен к выбросам.

3) R^2 (R-squared, coefficient of determination) - коэффициент детерминации, характеризует степень сходства реальных значений и предсказанных. Если модель всегда предсказывает точно, метрика равна 1. Для тривиальной модели - 0. Значение метрики может быть отрицательно, если модель предсказывает хуже, чем тривиальная. Полученные значения этих метрик приведены в Таблице 5.

Таблица 5 – Метрики качества обучения выбранных моделей

Модуль упругости при растяжении			
Model	MSE	MAE	R ²
RandomForestRegressor_best	0.03472	0.15124	-0.00263
KNeighborsRegressor_best	0.03472	0.15123	-0.00280
LinearRegression_default	0.03481	0.15221	-0.00522
LinearRegression_best	0.03481	0.15221	-0.00522
DummyRegression_mean	0.03483	0.15120	-0.00593
MLPRegressor_best	0.03483	0.15121	-0.00599
GradientBooster_best	0.03510	0.15179	-0.01367
RandomForestRegressor_default	0.03558	0.15344	-0.02753
MLPRegressor_default	0.03643	0.15627	-0.05194
GradientBooster_default	0.03780	0.15633	-0.09154
KNeighborsRegressor_default	0.04169	0.16648	-0.20410
Прочность при растяжении			
Model	MSE	MAE	R ²
KNeighborsRegressor_best	0.03481	0.14658	-0.02799
DummyRegression_mean	0.03486	0.14675	-0.02967
RandomForestRegressor_best	0.03495	0.14725	-0.03239
LinearRegression_default	0.03545	0.14790	-0.04700
LinearRegression_best	0.03545	0.14790	-0.04700
MLPRegressor_best	0.03562	0.14855	-0.05194
GradientBooster_best	0.03567	0.15002	-0.05363
RandomForestRegressor_default	0.03599	0.15050	-0.06305
GradientBooster_default	0.03789	0.15570	-0.11903
MLPRegressor_default	0.03789	0.15570	-0.11903
KNeighborsRegressor_default	0.04143	0.15725	-0.22358

У всех моделей коэффициент детерминации имеет отрицательные значения. Т.о., модели не дают прогнозов, которые были бы лучше простого расчета среднего значения. Также зафиксированы большие значения ошибок (MAE и MSE), что также свидетельствуют о низком качестве моделей.

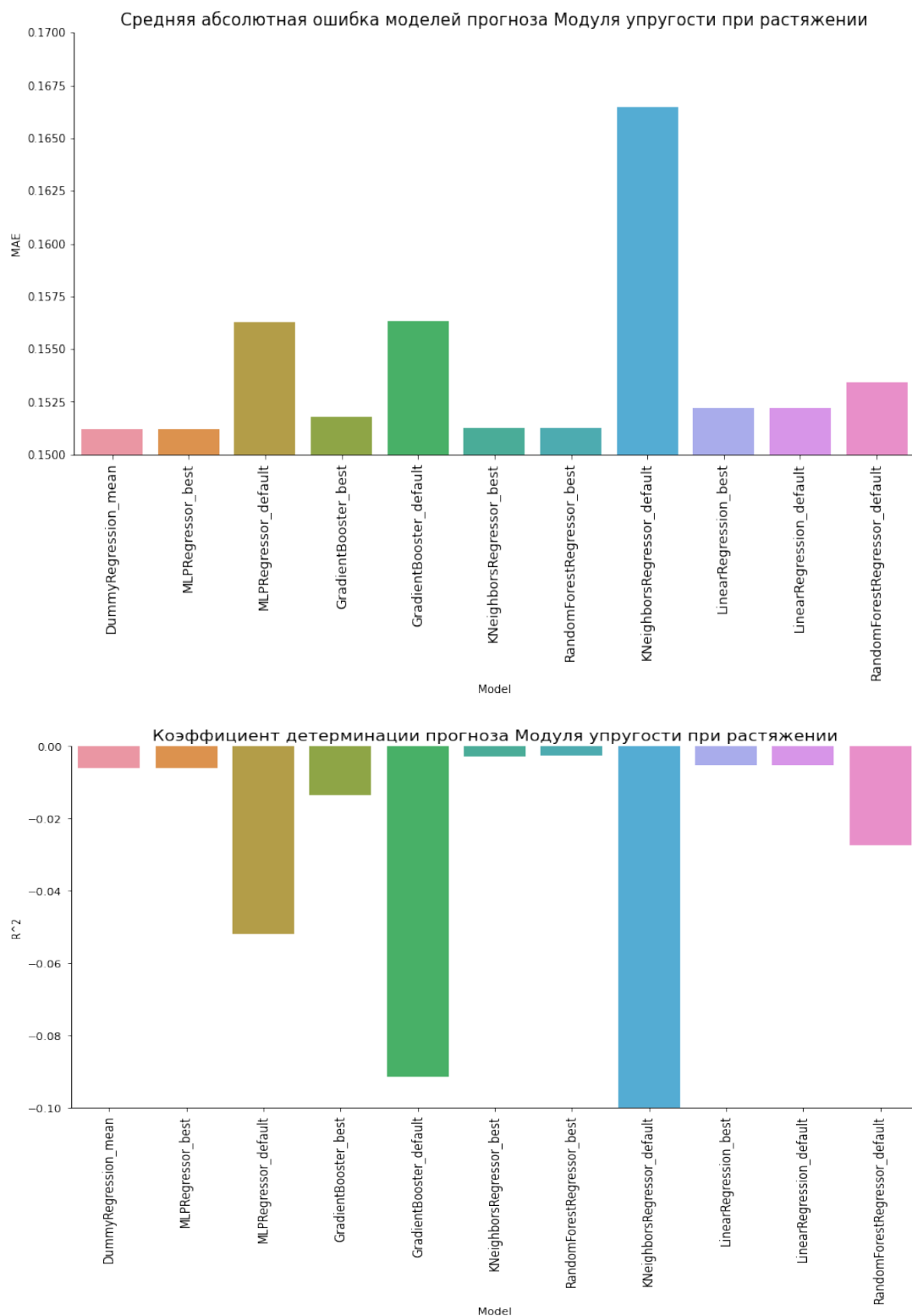


Рисунок 9 – Метрики прогноза Модуля упругости при растяжении

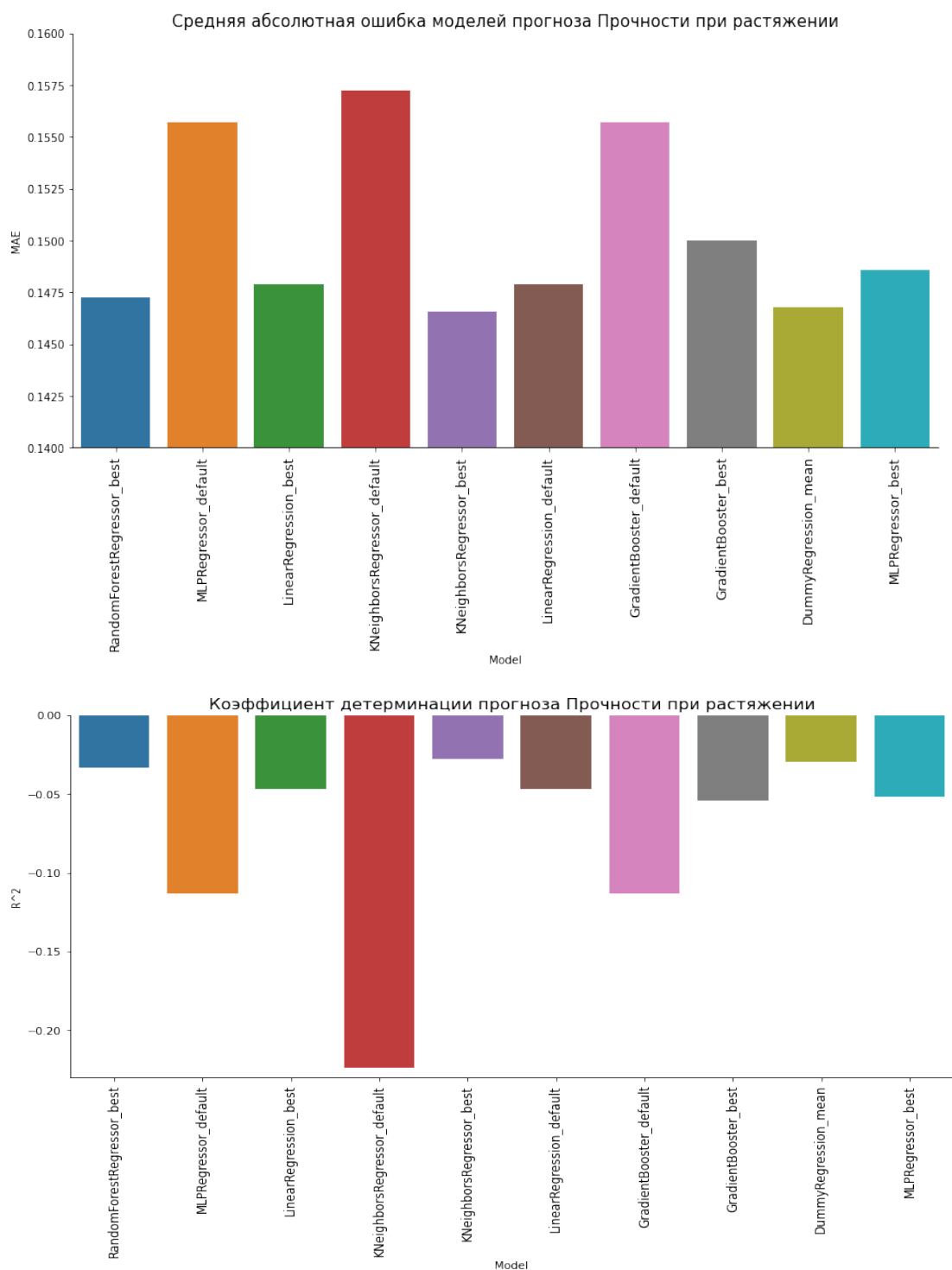


Рисунок 10 – Метрики прогноза Прочности при растяжении

Настройка гиперпараметров с помощью GridSearchCV и RandomizedSearchCV не повлияло на метрики линейной регрессии, что было ожидаемо, и улучшило результат работы остальных моделей (случайный лес, k-ближайших соседей, градиентный бустинг и многослойный персептрон).

Наилучшие показатели после настройки гиперпараметров выдали следующие модели:

Случайный лес для Модуля упругости при растяжении с параметрами
{'bootstrap': 'True', 'criterion': 'poisson', 'max_depth': 1,
'min_samples_leaf': 3, 'min_samples_split': 25, 'n_estimators': 710}

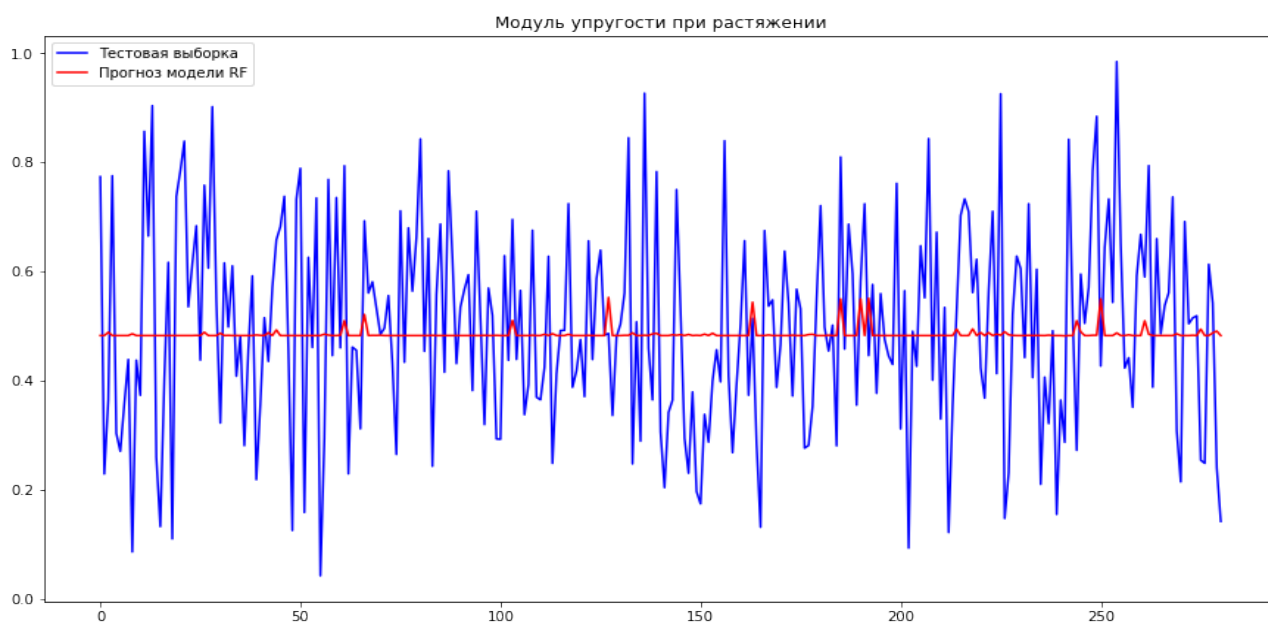


Рисунок 11 - Отображение для случайного леса

К-ближайших соседей для Прочности при растяжении с параметрами:
{'algorithm': 'auto', 'n_neighbors': 243, 'weights': 'uniform'}

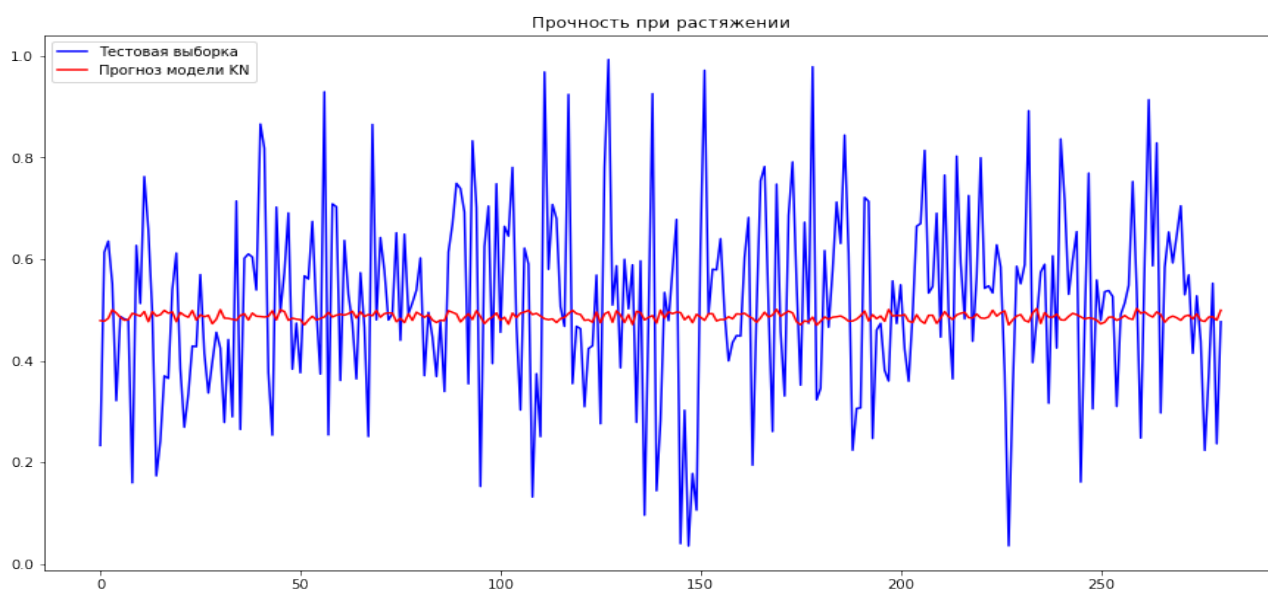


Рисунок 12 - Отображение для К-ближайших соседей.

3.4. Разработка нейронной сети

Для прогнозирования Соотношения матрица-наполнитель с помощью нейронной сети использовалась библиотека Keras фреймворка TensorFlow. Были смоделированы три варианта

- сеть со стандартными параметрами;
- сеть с ранней остановкой обучения;
- сеть с Dropout.

Количество скрытых слоев для первых двух вариантов было установлено равным варианту, полученному после настройки гиперпараметров многослойный персептрона в разделе 3.3.

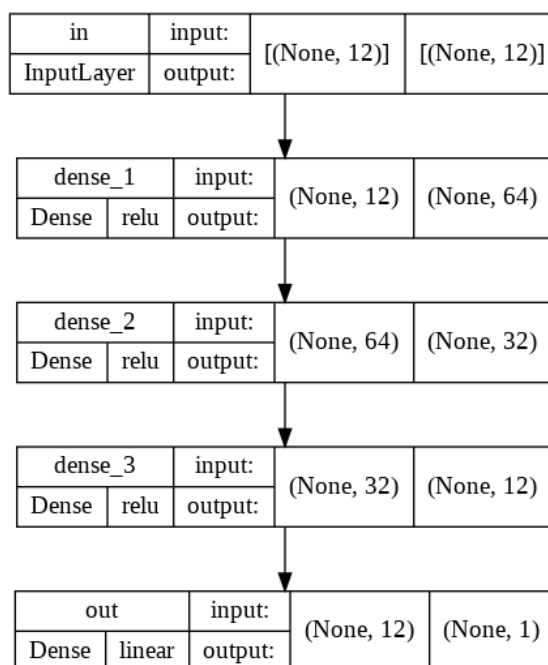


Рис 13 – Архитектура сети (стандарт и ранняя остановка)

График обучения нейросети со стандартными параметрами приведен ниже.

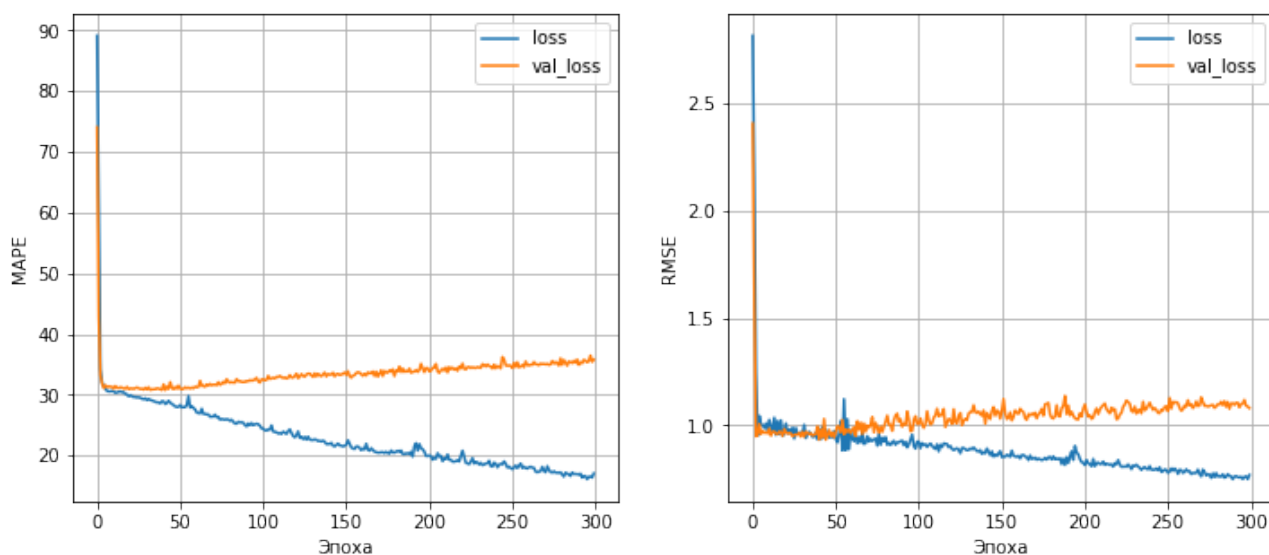


Рис 14 – Архитектура сети (стандарт и ранняя остановка)

Из него видно, что, начиная примерно с 10-й эпохи сеть начала переобучаться. Значение loss на тестовых выборках продолжило уменьшаться, а на валидационной стало расти.

Одним из способов борьбы с переобучением может быть ранняя остановка обучения. Для этого в Tensorflow используются обратные вызовы (callbacks). Для сети с той же архитектурой было запущено обучение с ранней остановкой. График обучения приведен на рисунке ниже.

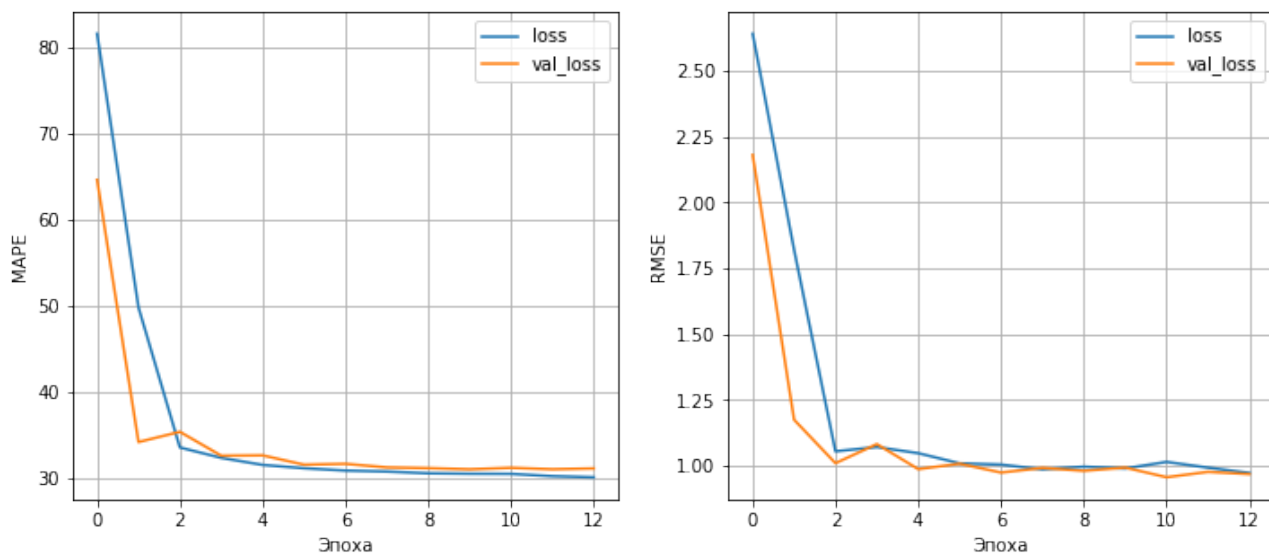


Рис 15 – Архитектура сети (стандарт и ранняя остановка)

Из этого графика можно сделать вывод, что ранняя остановка решает проблему переобучения и повышает точность модели на новых данных.

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 выключает 5% случайных нейронов на каждом слое.

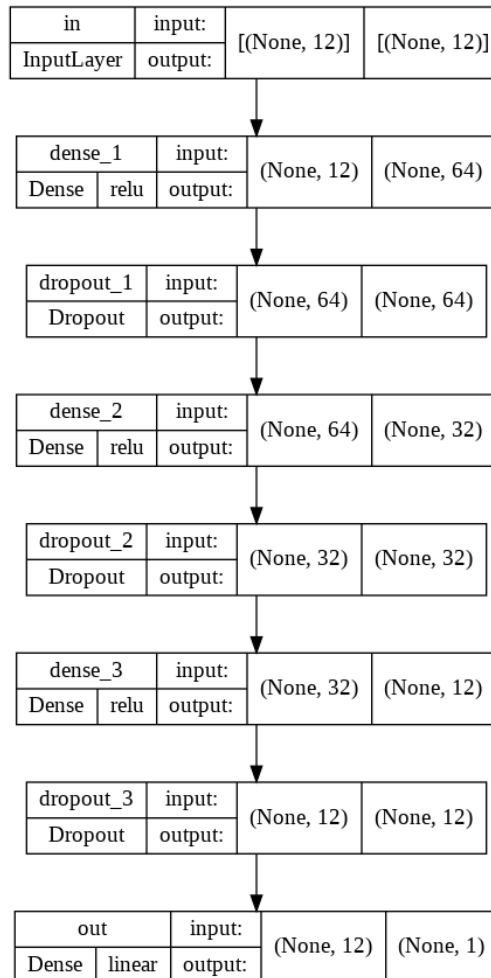


Рис 16 – Архитектура сети (стандарт и ранняя остановка)

График обучения приведен на рисунке ниже.

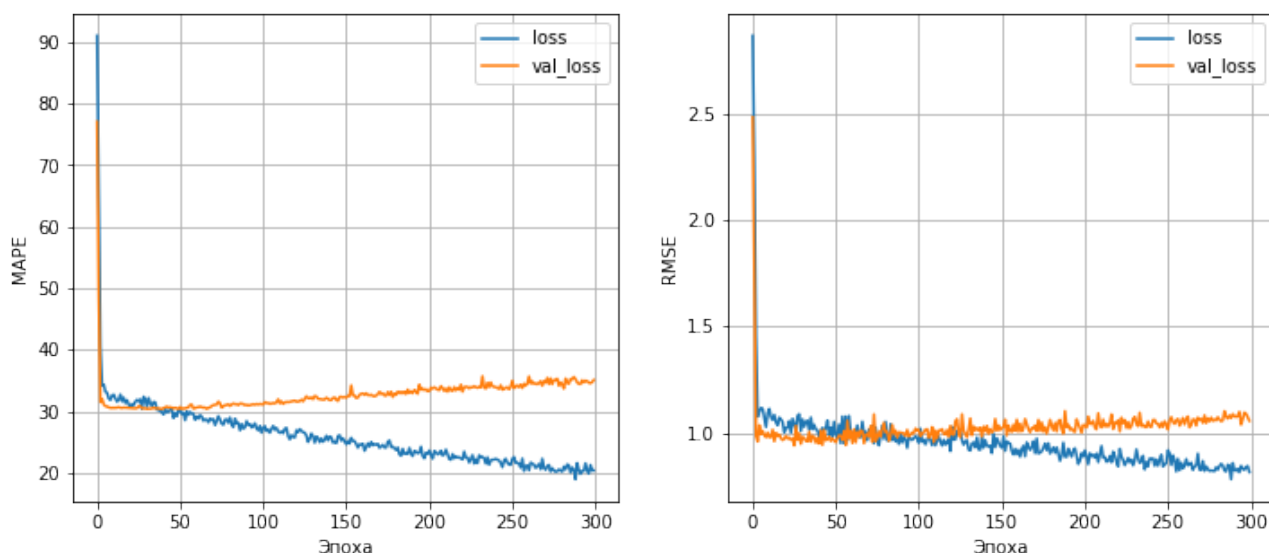


Рис 17 – Архитектура сети (стандарт и ранняя остановка)

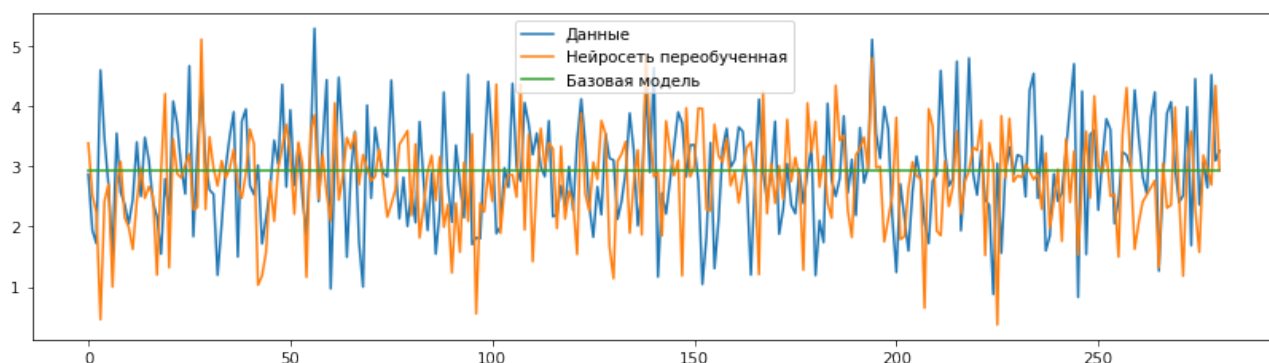
Видно, что Dropout-слои справились с переобучением лучше, чем сеть без них (первый вариант). Но при увеличении количества проходов значение loss на валидационных выборках снова стало расти.

Метрики, полученные при всех трех вариантах

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.001569	-0.873433	-0.694175	-0.298186	-2.359781
Нейросеть переобученная	-0.736772	-1.150166	-0.912612	-0.367097	-4.147318
Нейросеть с ранней остановкой	-0.085876	-0.909451	-0.736130	-0.282649	-2.381804
Нейросеть dropout	-0.609724	-1.107299	-0.895750	-0.343997	-3.888687

Рис 18 – Сравнение метрик смоделированных нейронных сетей

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



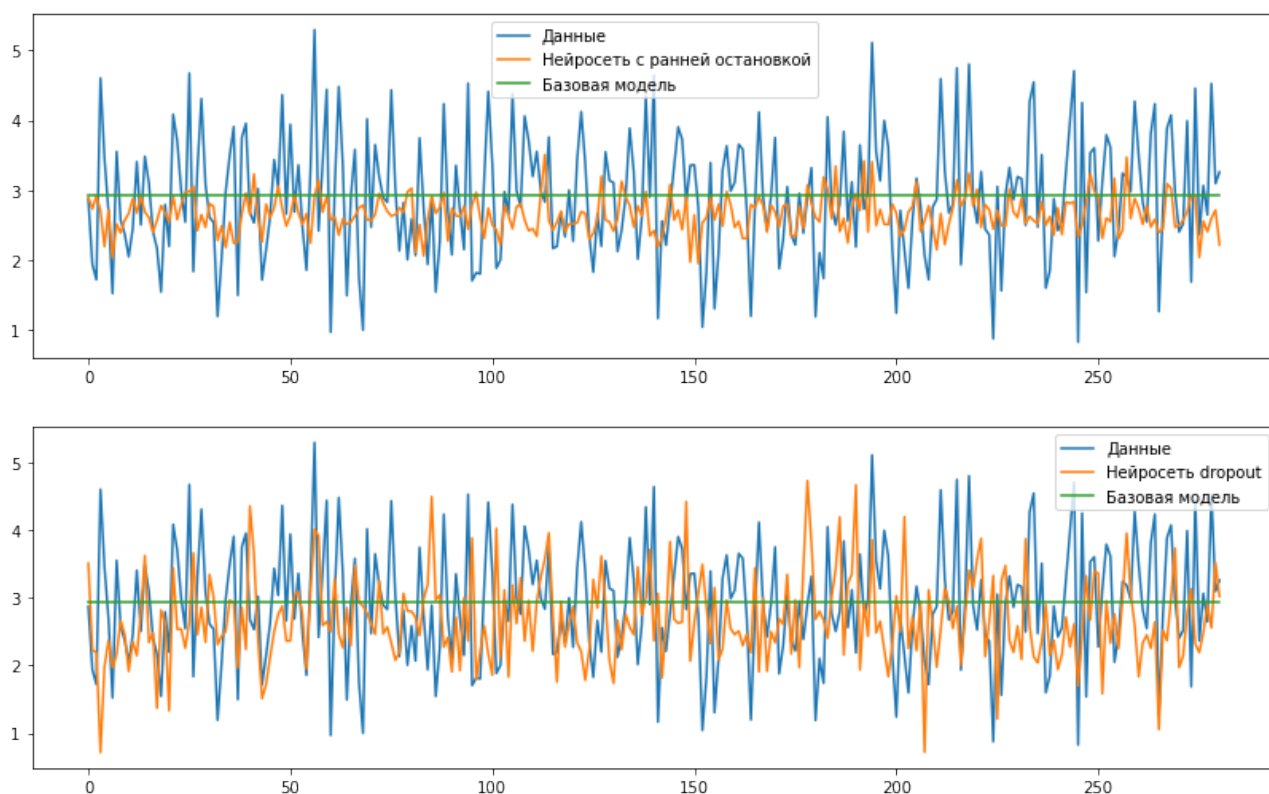


Рисунок 19 – Сравнение прогнозов трех вариантов нейронных сетей

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.001569	-0.873433	-0.694175	-0.298186	-2.359781
Нейросеть переобученная	-0.539060	-1.082722	-0.863116	-0.338348	-3.268139
Нейросеть с ранней остановкой	-0.111431	-0.920090	-0.738757	-0.277341	-2.416448
Нейросеть dropout	-0.652519	-1.121921	-0.883273	-0.345372	-3.127986

Рисунок 20 – Метрики нейронных сетей

Из сводной таблицы метрик и графиков на рисунке видно, что лучшая обобщающая способность и меньшие значения ошибок на тестовом множестве оказались у нейронной сети, обученной с ранней остановкой. Но и она предсказывает гораздо хуже базовой модели (среднего значения). Ошибка по любой из нейронных сетей больше, чем у рассмотренных выше моделей библиотеки Scikit-learn. Лучший прогноз по-прежнему дает метод К-ближайших соседей.

3.5. Тестирование модели

Согласно заданию, необходимо было сравнить ошибку модели на тренировочной и тестирующей части выборки. Это было сделано для лучшей модели К-ближайших соседей и для нейронной сети с ранней остановкой.

Таблица 6 - Сравнение ошибок модели для модуля упругости при растяжении на тренировочном и тестовом датасете

	R2	RMSE	MAE	max_error
Случайный лес				
Модуль упругости при растяжении, тренировочный	0.01238	-0.19223	-0.15483	-0.51702
Модуль упругости при растяжении, тестовый	-0.00263	-0.18633	-0.15124	-0.49743
К-ближайших соседей				
Прочность при растяжении, тренировочный	0.00650	-0.18943	-0.15111	-0.50857
Прочность при растяжении, тестовый	-0.02799	-0.18656	-0.14658	-0.49695
Сеть с ранней остановкой				
Соотношение матрица-наполнитель, тренировочный	-0.18256	-0.98054	-0.79410	-3.03017
Соотношение матрица-наполнитель, тестовый	-0.11143	-0.92009	-0.73876	-2.41645

Случайный лес и Метод К-ближайших соседей показали положительный коэффициент детерминации на тренировочной выборке. Возможно модели чему-то все-таки научилось. Но даже на тренировочном датасете они не нашли закономерностей во входных данных.

У нейронной сети показатели для тестовой выборки сильнее отличаются от показателей тренировочной. Т.е. с переобучением она справилась. Но требуется более тщательное и грамотное построение архитектуры нейронной сети, чтобы получить лучший результат.

В целом все модели работают не точнее среднего, и бесполезны для применения в реальных условиях. Задачу прогнозирования параметров решить не удалось.

4. Использование репозитория

Для данного исследования был создан удаленный репозиторий на GitHub, который находится по адресу:

<https://github.com/nikitkiss>

5. Выводы по результатам работы

В ходе выполнения квалификационной работы пройден практически весь Dataflow pipeline, рассмотрена обширная часть операций и задач, которые приходится выполнять специалисту по работе с данными.

Этот поток операций и задач включает:

- изучение теоретических методов анализа данных и машинного обучения;
- изучение основ предметной области, в которой решается задача;
- проведение разведочного анализа данных статистическими методами;
- DataMining — извлечение признаков из датасета и их анализ;
- разделение имеющихся, в нашем случае размеченных, данных на обучающую, валидационную, тестовую выборки;
- выполнение предобработки (препроцессинга) данных для обеспечения корректной работы моделей;
- построение аналитического решения. Это включает выбор алгоритма решения и модели, сравнение различных моделей, подбор гиперпараметров модели;
- визуализация модели и оценка качества аналитического решения;
- сохранение моделей;
- разработка и тестирование приложения для поддержки принятия решений специалистом предметной области, которое использовало бы найденную модель;

Дальнейшие возможные пути решения этой задачи могли бы быть:

- углубиться в изучение нейросетей, попробовать различные архитектуры, параметры обучения и т.д.;
- провести отбор признаков разными методами. Испробовать методы уменьшения размерности, например метод главных компонент;
- после уменьшения размерности градиентный бустинг может улучшить свои результаты. Так же есть большой простор для подбора гиперпараметров для этого метода;

Библиографический список

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. Документация по языку программирования python: – Режим доступа: <https://docs.python.org/3.8/index.html>.
5. Документация по библиотеке numpy: – Режим доступа: <https://numpy.org/doc/1.22/user/index.html#user>.
6. Документация по библиотеке pandas: – Режим доступа: https://pandas.pydata.org/docs/user_guide/index.html#user-guide.
7. Документация по библиотеке matplotlib: – Режим доступа: <https://matplotlib.org/stable/users/index.html>.
8. Документация по библиотеке seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>.
9. Документация по библиотеке sklearn: – Режим доступа: https://scikit-learn.org/stable/user_guide.html.
10. Документация по библиотеке keras: – Режим доступа: <https://keras.io/api/>.
11. Руководство по быстрому старту в flask: – Режим доступа: <https://flask-russian-docs.readthedocs.io/ru/latest/quickstart.html>.
12. Loginom Вики. Алгоритмы: – Режим доступа: <https://wiki.loginom.ru/algorithms.html>.
13. Andre Ye. 5 алгоритмов регрессии в машинном обучении, о которых вам следует знать: – Режим доступа: <https://habr.com/ru/company/vk/blog/513842/>.

14. 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>.
15. 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/>.
16. Yury Kashnitsky. Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес: – Режим доступа: <https://habr.com/ru/company/ods/blog/324402/>.
17. 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>.
18. Alex Maszański. Решаем задачи машинного обучения с помощью алгоритма градиентного бустинга: – Режим доступа: <https://proglib.io/p/reshaem-zadachi-mashinnogo-obucheniya-s-pomoshchyu-algoritma-gradientnogo-bustinga-2021-11-25>

Приложение А. Скриншоты веб-приложения

Реализованы следующие функции:

- выбор целевой переменной для предсказания (модуль упругости при растяжении и прочности при растяжении или соотношение матрица-наполнитель);
- ввод входных параметров;
- проверка введенных параметров;
- загрузка сохраненной модели, получение и отображение прогноза выходных параметров.

При проверке введенных параметров считаем, что значения не могут быть пустыми, должны быть вещественными, не могут содержать некорректных символов и должны соответствовать допустимому диапазону.

Выпускная квалификационная работа по курсу «Data Science»

Прогнозирование конечных свойств новых материалов (композиционных материалов)

[Прогнозирование модуля упругости при растяжении и прочности при растяжении](#)

[Прогнозирование соотношения матрица-наполнитель](#)

Рисунок 21 — Начальное окно, выбор целевых переменных

Прогнозирование модуля упругости при растяжении и прочности при растяжении

Соотношение матрица-наполнитель (0..6)

Плотность, кг/м3 (1700...2300)

Модуль упругости, ГПа (2...2000)

Количество отвердителя, м.% (17...200)

Содержание эпоксидных групп, %_2 (14...34)

Температура вспышки, C_2 (100...414)

Поверхностная плотность, г/м2 (0.6...1400)

Потребление смолы, г/м2 (33...414)

Угол нашивки, град (0 или 90)

Шаг нашивки (0...15)

Плотность нашивки (0...104)

Рисунок 22 - Ввод входных параметров для прогнозирования модуля упругости при растяжении и прочности при растяжении

Прогнозирование модуля упругости при растяжении и прочности при растяжении

Содержание эпоксидных групп, % 2 - некорректное значение "а"
Количество отвердителя, м. % - значение вне корректного диапазона
Некоторые значения отсутствуют!

Соотношение матрица-наполнитель (0..6)3

Плотность, кг/м3 (1700...2300)1800

Модуль упругости, ГПа (2...2000)900

Количество отвердителя, м. % (17...200)0

Содержание эпоксидных групп, % 2 (14...34)а

Температура вспышки, С_2 (100...414)

Поверхностная плотность, г/м2 (0.6...1400)

Потребление смолы, г/м2 (33...414)

Угол нашивки, град (0 или 90)

Шаг нашивки (0...15)

Плотность нашивки (0...104)

Отправить

Рисунок 23 - Проверка входных параметров для прогнозирования модуля упругости при растяжении и прочности при растяжении

Прогнозирование модуля упругости при растяжении и прочности при растяжении

Соотношение матрица-наполнитель (0..6)4.02912621359223

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 вспышки, С_2 (100...414)284.615384615384

Поверхностная плотность, г/м2 (0.6...1400)470.0

Потребление смолы, г/м2 (33...414)220.0

Угол нашивки, град (0 или 90)90.0

Шаг нашивки (0...15)4.0

Плотность нашивки (0...104)60.0

Отправить

Входные переменные:

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м. %	Содержание эпоксидных групп, % 2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0	4.029126	1880.0	622.0	111.86	22.267857	284.615385	470.0	220.0	90.0	4.0	60.0

Результат модели:

Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа
72.81891497929365	2523.9223070281537

Рисунок 24 - Результат работы модели для модуля упругости при растяжении и прочности при растяжении

Прогнозирование соотношения матрица-наполнитель

Плотность, кг/м3 (1700...2300)

Модуль упругости, ГПа (2...2000)

Количество отвердителя, м.% (17...200)

Содержание эпоксидных групп,%_2 (14...34)

Температура вспышки, С_2 (100...414)

Поверхностная плотность, г/м2 (0.6...1400)

Модуль упругости при растяжении, ГПа (64...83)

Прочность при растяжении, МПа (1036...3849)

Потребление смолы, г/м2 (33...414)

Угол нашивки, град (0 или 90)

Шаг нашивки (0...15)

Плотность нашивки (0...104)

Отправить

Рисунок 25 - Ввод входных параметров для прогнозирования соотношения матрица-наполнитель

Прогнозирование соотношения матрица-наполнитель

Поверхностная плотность, г/м2 - некорректное значение "50066"
Температура вспышки, С_2 - значение вне корректного диапазона
Некоторые значения отсутствуют!

Плотность, кг/м3 (1700...2300)

Модуль упругости, ГПа (2...2000)

Количество отвердителя, м.% (17...200)

Содержание эпоксидных групп,%_2 (14...34)

Температура вспышки, С_2 (100...414)

Поверхностная плотность, г/м2 (0.6...1400)

Модуль упругости при растяжении, ГПа (64...83)

Прочность при растяжении, МПа (1036...3849)

Потребление смолы, г/м2 (33...414)

Угол нашивки, град (0 или 90)

Шаг нашивки (0...15)

Плотность нашивки (0...104)

Отправить

Рисунок 26 - Проверка входных параметров для прогнозирования соотношения матрица-наполнитель

Прогнозирование соотношения матрица-наполнитель

Плотность, кг/м3 (1700...2300)

Модуль упругости, ГПа (2...2000)

Количество отвердителя, м.% (17...200)

Содержание эпоксидных групп,%_2 (14...34)

Температура вспышки, С_2 (100...414)

Поверхностная плотность, г/м2 (0.6...1400)

Модуль упругости при растяжении, ГПа (64...83)

Прочность при растяжении, МПа (1036...3849)

Потребление смолы, г/м2 (33...414)

Угол нашивки, град (0 или 90)

Шаг нашивки (0...15)

Плотность нашивки (0...104)

Отправить

Входные переменные:

	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	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
2.5154960585858928

Рисунок 27 - Результат работы модели для соотношения матрица-наполнитель