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 Бутакова Наталья Валерьевна

Екатеринбург, 2023

Содержание

Введение

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

Композиционные материалы - это искусственно созданные материалы, состоящие из нескольких других, с четкой границей между ними. Композиты обладают теми свойствами, которые не наблюдаются у компонентов по отдельности. При этом композиты являются монолитным материалом, т.е. компоненты материала неотделимы друг от друга без разрушения конструкции в целом. Яркий пример композита - железобетон. Бетон прекрасно сопротивляется сжатию, но плохо растяжению. Стальная арматура внутри бетона компенсирует его неспособность сопротивляться сжатию, формируя тем самым новые, уникальные свойства. Современные композиты изготавливаются из других материалов: полимеры, керамика, стеклянные и углеродные волокна, но данный принцип сохраняется. У такого подхода есть и недостаток: даже если мы знаем характеристики исходных компонентов, определить характеристики композита, состоящего из этих компонентов, достаточно проблематично. Для решения этой проблемы 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 (связующего и армирующего компонента).

На входе имеются данные о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.). На выходе необходимо спрогнозировать ряд конечных свойств получаемых композиционных материалов. Кейс основан на реальных производственных задачах Центра НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана).

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

Для исследовательской работы были даны 2 файла: X_bp.xlsx (с данными о параметрах базальтопластика, состоящий из 1023 строк и 10 столбцов данных) и X_nup.xlsx (данными углепластика, состоящий из 1040 строк и 3 столбцов данных). Для разработки моделей по прогнозу модуля упругости при растяжении, прочности при растяжении и соотношения матрица-наполнитель нужно объединить 2 файла. Объединение по типу INNER, поэтому часть информации (17 строк таблицы X_nup.xlsx) не имеет соответствующих строк в таблице X_bp.xlsx и будет удалена. Также необходимо провести разведочный анализ данных, нарисовать гистограммы распределения каждой из переменной, диаграммы boxplot (ящик с усами), попарные графики рассеяния точек. Для каждой колонки получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков; сделать предобработку: удалить шумы и выбросы, сделать нормализацию и стандартизацию. Обучить несколько моделей для прогноза модуля упругости при растяжении и прочности при растяжении. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель. Разработать приложение с графическим интерфейсом, которое будет выдавать прогноз соотношения «матрица-наполнитель». Оценить точность модели на тренировочном и тестовом датасете. Создать репозиторий в GitHub и разместить код исследования. Оформить файл README.

Практическая часть работы будет реализована на языке Python. Далее по тексту упомянутые используемые метод будут относиться к библиотекам в Python.

1.2 Описание используемых методов

Данная задача в рамках классификации методов машинного обучения относится к машинному обучению с учителем, так как в предоставленном наборе данных имеются значения целевых параметров.

Так как перед нами стоит задача предсказания значений вещественной переменной — это задача регрессии.

В настоящее время разработано много методов регрессионного анализа. В данной работе были исследованы (и некоторые из них применены) следующие методы:

1) линейная регрессия (Linear regression);
2) полиномиальная регрессия (Polynomial regression);
3) лассо регрессия (Lasso regression);
4) эластичная сеть (Elastic Net);
5) обобщенная линейная модель с распределением Tweedie (GLM with a Tweedie distribution);
6) дерево решений (Decision Tree Regressor);
7) случайный лес (Random Forest);
8) K-ближайших соседей (KNeighbors Regressor);
9) градиентный бустинг (Gradient Boosting Regressor).

1.2.1 Линейная регрессия

Простая линейная регрессия имеет место, если рассматривается зависимость между одной входной и одной выходной переменными. Для этого

определяется уравнение регрессии (1) и строится соответствующая прямая, известная как линия регрессии.

$$y = ax + b \quad (1)$$

Коэффициенты a и b, называемые также параметрами модели, определяются таким образом, чтобы сумма квадратов отклонений точек, соответствующих реальным наблюдениям данных, от линии регрессии была бы минимальной. Коэффициенты обычно оцениваются методом наименьших квадратов.

Если ищется зависимость между несколькими входными и одной выходной переменными, то имеет место множественная линейная регрессия. Соответствующее уравнение имеет вид (2).

$$Y = b0 + b1 * x1 + b2 * x2 + \cdots + bn * xn, \quad (2)$$

где n - число входных переменных.

Очевидно, что в данном случае модель будет описываться не прямой, а гиперплоскостью. Коэффициенты уравнения множественной линейной регрессии подбираются так, чтобы минимизировать сумму квадратов отклонения реальных точек данных от этой гиперплоскости.

Линейная регрессия — первый тщательно изученный метод регрессионного анализа. Его главное достоинство — простота. Такую модель можно построить и рассчитать даже без мощных вычислительных средств. Простота является и главным недостатком этого метода. Тем не менее, именно с линейной регрессии целесообразно начать подбор подходящей модели.

1.2.2 Полиномиальная регрессия

Полиномиальная регрессия – это алгоритм машинного обучения, который используется для обучения линейной модели на нелинейных данных. Довольно часто данные намного сложнее, чем прямая линия, и в таких случаях обучение на основе алгоритма линейной регрессии не даст хороших результатов. Однако

можно использовать алгоритм полиномиальной регрессии, чтобы добавить производительности каждой функции, а затем обучить линейную модель на расширенном наборе функций. Этот подход поддерживает в целом высокую производительность линейных методов, позволяя им соответствовать гораздо более широкому диапазону данных.

Например, простую линейную регрессию можно расширить, построив полиномиальные признаки из коэффициентов. В случае стандартной линейной регрессии у нас может быть модель, которая выглядит следующим образом (для двумерных данных):

$$\hat{y}(w, x) = w_0 + w_1 x_1 + w_2 x_2 \tag{3}$$

Если мы хотим подогнать к данным параболоид вместо плоскости, мы можем объединить признаки в полиномы второго порядка, чтобы модель выглядела так:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2 \tag{4}$$

Мы по-прежнему видим линейную модель, но набор признаков теперь такой:

$$z = [x_1, x_2, x_1 x_2, x_1^2, x_2^2] \tag{5}$$

И мы можем представить модель в следующем виде:

$$\hat{y}(w, z) = w_0 + w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5 \tag{6}$$

Аналогичным образом можно работать с полиномами любых порядков.

Мы видим, что результирующая полиномиальная регрессия принадлежит к тому же классу линейных моделей, который мы рассматривали выше (т. Е. модель линейна по *w*) и может быть решена теми же методами.

1.2.3 Лассо регрессия

Метод регрессии лассо (LASSO, Least Absolute Shrinkage and Selection Operator) — это вариация линейной регрессии, специально адаптированная для данных, которые имеют сильную корреляцию признаков друг с другом.

Лассо-регрессия 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.

Регрессия лассо использует регуляризацию L1, то есть взвешивает ошибки по их абсолютному значению.

Регуляризация позволяет интерпретировать модели. Если коэффициент стал 0, значит данный входной признак не является значимым.

1.2.4 Эластичная сеть

Elastic-Net — это модель линейной регрессии с двумя регуляризаторами, L1 и L2. Эта комбинация позволяет изучать разреженную модель, в которой несколько весов не равны нулю, как у Лассо, при этом сохраняя свойства гребневой модели (Ridge). Модель Лассо и гребневая регрессия являются частными случаями Эластичной сети.

Эластичная сеть полезна, когда есть несколько признаков, которые коррелируют друг с другом. Лассо-регрессия, скорее всего, выберет один из них случайным образом, в то время как эластичная сеть, скорее всего, выберет оба.

1.2.5 Обобщенная линейная модель с распределением Tweedie

Обобщённые линейные модели (Generalized Linear Models) – универсальный метод построения регрессионных моделей, позволяющий учитывать взаимодействие между факторами, вид распределения зависимой переменной и предположения о характере регрессионной зависимости.

Обобщенные линейные модели (GLM) расширяют линейные модели двумя способами.

Во-первых, прогнозируемые значения $\hat{y}$ связаны с линейной комбинацией входных переменных X через функцию обратной связи h как:

$$\hat{y}(w, X) = h(Xw) \tag{7}$$

Во-вторых, квадратичная функции ошибки заменяется единичным отклонением распределения d в экспоненциальном семействе (точнее, моделью репродуктивной экспоненциальной дисперсии (EDM)).

$$\min_{w} \frac{1}{2n_{\text{samples}}} \sum_{i} d(y_i, \hat{y}_i) + \frac{\alpha}{2}||w||_2^2, \tag{8}$$

где α - штраф регуляризации L2.

Если указаны веса выборки, среднее значение становится средневзвешенным.

Ниже перечислены некоторые конкретные EDM и их единичное отклонение d (3я колонка):

Нормальное	$y \in (-\infty, \infty)$	$(y - \hat{y})^2$
Пуассон	$y \in [0, \infty)$	$2(y \log \frac{y}{\hat{y}} - y + \hat{y})$
Гамма	$y \in (0, \infty)$	$2(\log \frac{\hat{y}}{y} + \frac{y}{\hat{y}} - 1)$
Обратный гауссовский	$y \in (0, \infty)$	$\frac{(y-\hat{y})^2}{y\hat{y}^2}$

Все они являются экземплярами семейства Tweedie)

1.2.6 Дерево решений

Деревья решений (Decision Trees) - непараметрический метод, применяемый и для классификации, и для регрессии. Деревья решений используются в самых разных областях человеческой деятельности и представляют собой иерархические древовидные структуры, состоящие из правил вида «Если ..., то ...».

Решающие правила автоматически генерируются в процессе обучения на обучающем множестве путем обобщения обучающих примеров. Поэтому их называют индуктивными правилами, а сам процесс обучения — индукцией деревьев решений.

Дерево состоит из элементов двух типов: узлов (node) и листьев (leaf).

В узлах находятся решающие правила и производится проверка соответствия примеров этому правилу. В результате проверки множество примеров, попавших в узел, разбивается на два подмножества: удовлетворяющие правилу и не удовлетворяющие ему. Затем к каждому подмножеству вновь применяется правило и процедура рекурсивно повторяется пока не будет достигнуто некоторое условие остановки алгоритма. В последнем узле проверка и разбиение не производятся, и он объявляется листом.

В листе содержится не правило, а подмножество объектов, удовлетворяющих всем правилам ветви, которая заканчивается данным листом. Для классификации — это класс, ассоциируемый с узлом, а для регрессии — соответствующий листу интервал целевой переменной.

При формировании правила для разбиения в очередном узле дерева необходимо выбрать атрибут, по которому это будет сделано. Для регрессии критерием является дисперсия от среднего значения.

Огромное преимущество деревьев решений в том, что они легко интерпретируемы, понятны человеку. Они могут использоваться для извлечения правил на естественном языке. Еще преимущества — высокая точность работы, нетребовательность к подготовке данных.

Недостаток деревьев решений - склонность переобучаться. Переобучение в случае дерева решений - это ситуация, когда происходит точное распознавание примеров, участвующих в обучении, и полная несостоятельность на новых данных. В худшем случае, дерево будет большой глубины и сложной структуры,

а в каждом листе будет только один объект. Для решения этой проблемы используют разные критерии остановки алгоритма.

1.2.7 Случайный лес

Случайный лес (RandomForest) — представитель ансамблевых методов.

Если точность дерева решений оказалось недостаточной, мы можем множество моделей собрать в коллектив. Формула итогового решателя (9) — это усреднение предсказаний отдельных деревьев.

$$a(x) = \frac{1}{N}\sum_{i=1}^{N} b_i(x) \tag{9}$$

где

N – количество деревьев;

i – счетчик для деревьев;

b – решающее дерево;

x – сгенерированная нами на основе данных выборка.

Для определения входных данных каждому дереву используется метод случайных подпространств. Базовые алгоритмы обучаются на различных подмножествах признаков, которые выделяются случайным образом.

Преимущества случайного леса:

– высокая точность предсказания;

– редко переобучается;

– практически не чувствителен к выбросам в данных;

– одинаково хорошо обрабатывает как непрерывные, так и дискретные признаки, данные с большим числом признаков;

– высокая параллелизуемость и масштабируемость.

Из недостатков можно отметить, что его построение занимает больше времени. Так же теряется интерпретируемость.

1.2.8 Метод К-случайных соседей

Метод К-ближайших соседей (k Nearest Neighbors) – это метод классификации, который адаптирован для регрессии. 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, эвклидово расстояние для количественных признаков или расстояние Хэмминга для категориальных.

Этот метод — пример непараметрической регрессии.

1.2.9 Градиентный бустинг

Градиентный бустинг (GradientBoosting) — 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 ошибки (loss). Loss-функция – это мера, которая показывает насколько хорошо предсказание модели соответствуют данным. Используя градиентный спуск и обновляя предсказания, основанные на скорости обучения (learning rate), ищем значения, на которых функция ошибки 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.).

Из недостатков алгоритма можно отметить только затраты времени на вычисления и необходимость грамотного подбора гиперпараметров.

1.2.10 Нейронная сеть

Нейронная сеть — это последовательность нейронов, соединенных между собой связями. Структура нейронной сети пришла в мир программирования из биологии. 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, гиперболический тангенс.

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 определяет 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.

Прогнозируемое значение сравниваем с фактическим с помощью функции потери. 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. Веса каждого соединения обновляются таким образом, чтобы минимизировать значение функции потерь.

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.

1.3 Разведочный анализ данных

Прежде всего необходимо изучить датасет и выявить его основные характеристики.

В итоговом (объединенном из двух файлов) датасете имеется 1023 объекта с 13ю признаками, 3 из которых будут выступать в качестве целевой переменной (входных данных). На рисунке 1 можно видеть заголовки и первые 5 строк датасета.

Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
1.857143	2030.0	738.736842	30.00	22.267857	100.000000	210.0	70.0	3000.0	220.0	0	4.0	57.0
1.857143	2030.0	738.736842	50.00	23.750000	284.615385	210.0	70.0	3000.0	220.0	0	4.0	60.0
1.857143	2030.0	738.736842	49.90	33.000000	284.615385	210.0	70.0	3000.0	220.0	0	4.0	70.0
1.857143	2030.0	738.736842	129.00	21.250000	300.000000	210.0	70.0	3000.0	220.0	0	5.0	47.0
2.771331	2030.0	753.000000	111.86	22.267857	284.615385	210.0	70.0	3000.0	220.0	0	5.0	57.0

Рисунок 1 Заголовки и первые 5 строк датасета

В первой части работы мы будем прогнозировать Модуль упругости при растяжении и Прочность при растяжении, за входные данные будем брать остальные 11 признаков. Для прогнозирования каждой целевой переменной будет подбираться своя модель.

Во второй части работы займемся прогнозированием Соотношения матрица-наполнитель с помощью нейронных сетей. За входные признаки будут взяты остальные 12 характеристик композитных материалов из датасета.

Было установлено, что все характеристики являются числовыми (12 признаков с вещественными числами и один с целыми), пропусков в данных нет (см. рисунок 2).

```
#Смотрим информацию о датасете
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1023 entries, 0 to 1022
Data columns (total 13 columns):
 #   Column                             Non-Null Count  Dtype  
---  ------                             --------------  -----  
 0   Соотношение матрица-наполнитель    1023 non-null   float64
 1   Плотность, кг/м3                   1023 non-null   float64
 2   модуль упругости, ГПа              1023 non-null   float64
 3   Количество отвердителя, м.%        1023 non-null   float64
 4   Содержание эпоксидных групп,%_2    1023 non-null   float64
 5   Температура вспышки, С_2           1023 non-null   float64
 6   Поверхностная плотность, г/м2      1023 non-null   float64
 7   Модуль упругости при растяжении, ГПа  1023 non-null   float64
 8   Прочность при растяжении, МПа      1023 non-null   float64
 9   Потребление смолы, г/м2            1023 non-null   float64
 10  Угол нашивки, град                 1023 non-null   int64  
 11  Шаг нашивки                        1023 non-null   float64
 12  Плотность нашивки                  1023 non-null   float64
dtypes: float64(12), int64(1)
memory usage: 111.9 KB
```

Рисунок 2 Информация о датасете

Кроме того, было установлено, что в основном объекты имеют различные значения признаков, за исключением Угла нашивки. Это хорошо видно на рисунке 3. Однако, учитывая физический смысл величины, попробуем оставить этот признак в неизмененном виде.

```
#Смотрим количество уникальных значений в каждом столбце
df.nunique()
```

```
Соотношение матрица-наполнитель         1014
Плотность, кг/м3                        1013
модуль упругости, ГПа                   1020
Количество отвердителя, м.%             1005
Содержание эпоксидных групп,%_2         1004
Температура вспышки, С_2                1003
Поверхностная плотность, г/м2           1004
Модуль упругости при растяжении, ГПа    1004
Прочность при растяжении, МПа           1004
Потребление смолы, г/м2                 1003
Угол нашивки, град                         2
Шаг нашивки                              989
Плотность нашивки                        988
dtype: int64
```

Рисунок 3 Количество уникальных значений в каждом столбце

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

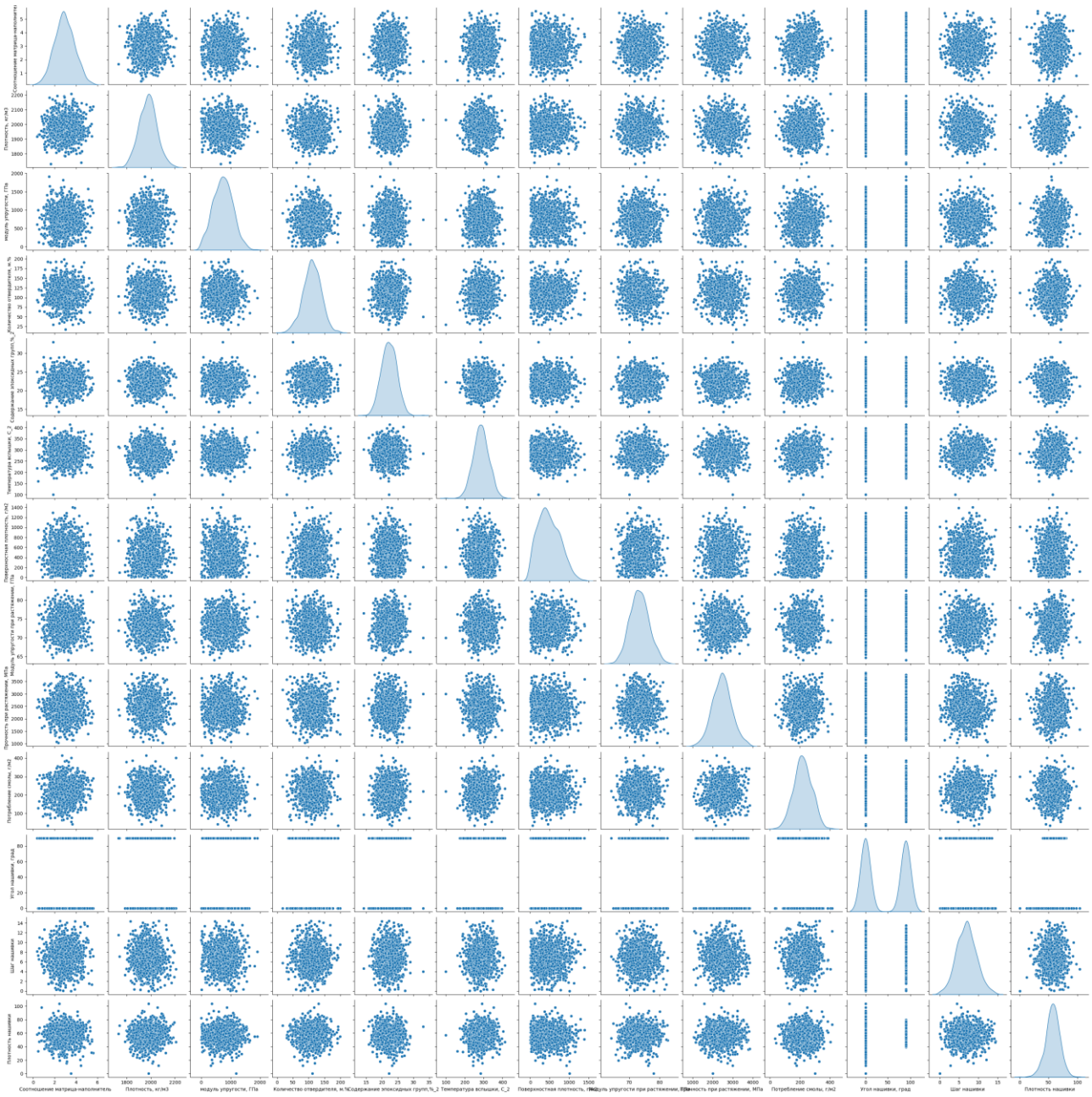

Рисунок 4 График попарного рассеяния точек

На рисунке 4 мы видим график попарного рассеяния точек. По форме «облаков точек» не видно каких-либо закономерностей, что означает отсутствие линейных и других зависимостей, похожих на какую-либо известную функцию, между парами признаков (простыми словами, «облака точек» не стремятся к прямой, гиперболе, экспоненте и т.п.). Кроме того, очевидно наличие выбросов (об этом говорят достаточно удаленные точки от общего «облака точек»).

Выбросы – это такие объекты из датасета, которые отклоняются от общего набора данных.

Помочь выявить связь между признаками может тепловая карта матрицы корреляции, приведенная на рисунке 5. На пересечении признаков указаны значения коэффициентов корреляции для данной пары признаков. Чем ближе коэффициент корреляции к 0, тем более слабая зависимость между признаков, и тем более темным цветом закрашена соответствующая клетка.

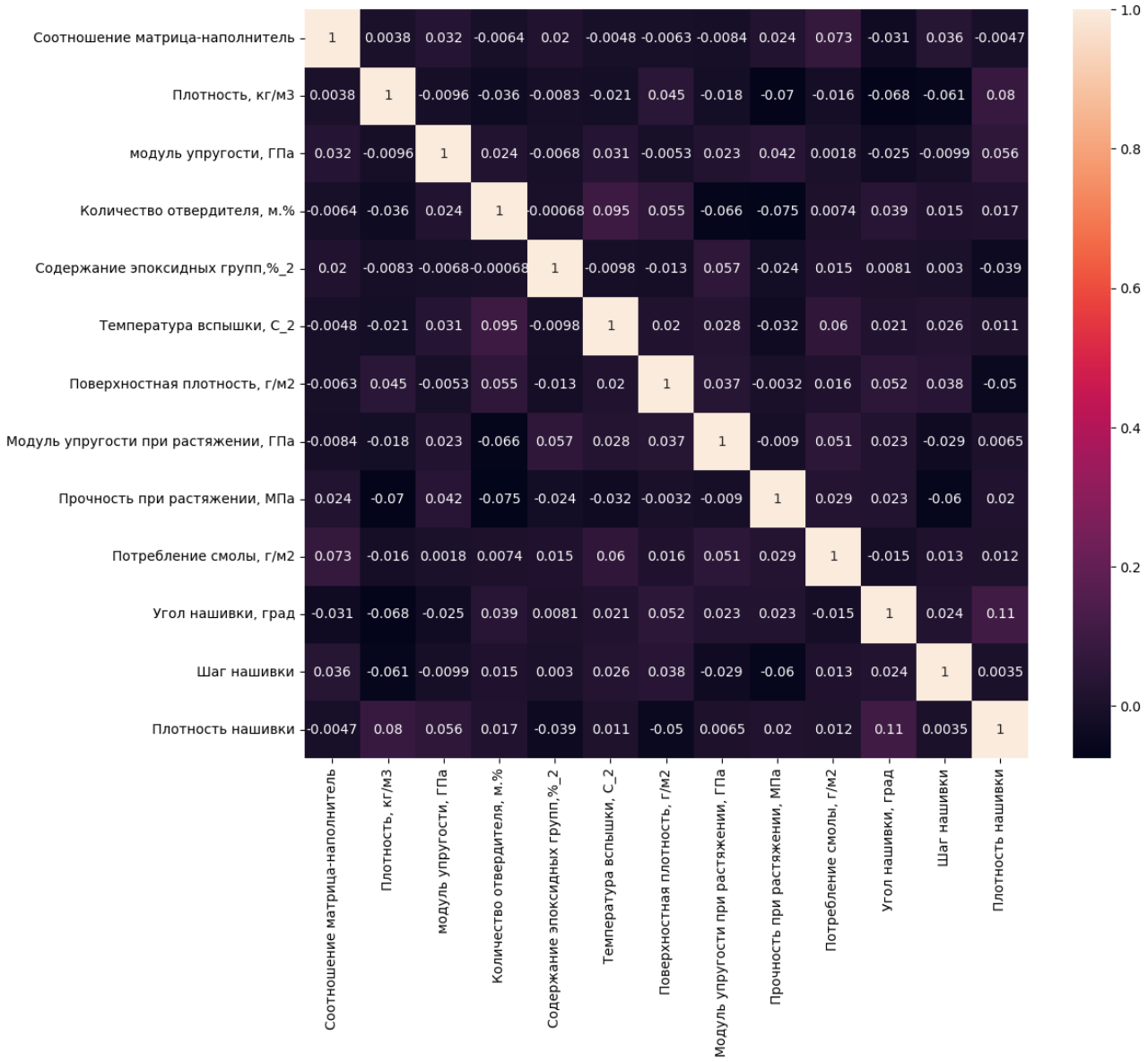

Рисунок 5 Тепловая карта матрицы корреляции

Коэффициент корреляции в математической статистике – это показатель, характеризующий силу статистической связи между двумя или несколькими случайными величинами.

Значения коэффициента корреляции всегда расположены в диапазоне от -1 до 1 и интерпретируются следующим образом:

- если коэффициент корреляции близок к 1, то между переменными наблюдается положительная корреляция. Иными словами, отмечается высокая степень связи между переменными. Если значения одной переменной будут возрастать, то вторая переменная будет увеличиваться;

- если коэффициент корреляции близок к -1, это означает, что между переменными имеет место сильная отрицательная корреляция. Иными словами, так же отмечается высокая степень связи между переменными. Но если значения одной переменной будут возрастать, то вторая переменная будет уменьшаться;

- промежуточные значения, близкие к 0, указывают на слабую корреляцию между переменными и, соответственно, низкую зависимость. Иными словами, поведение одной переменной не будет совсем (или почти совсем) влиять на поведение другой.

Очевидно, что если корреляция между переменными высокая, то, зная поведение входной переменной, проще предсказать поведение выходной, и полученное предсказание будет точнее (говорят, что входная переменная хорошо «объясняет» выходную).

Простой коэффициент корреляции (а здесь мы говорили именно о нем) Пирсона описывает только степень линейной связи и применим к непрерывным величинам.

По нашей матрице корреляции мы видим, что все коэффициенты корреляции близки к нулю, что означает отсутствие линейной зависимости между признаками. Это сразу говорит о том, что найти хорошо работающую модель для решения данной задачи будет сложно.

2. Практическая часть

2.1 Предобработка данных

Цель предобработки данных — обеспечить корректную работу моделей. Обычно этот процесс включает:

- выделение числовых и категориальных признаков (тип признака может влиять на другие шаги предобработки данных);

- удаление дубликатов;

- «борьбу» с пропусками в данных (возможны различные стратегии, удаляющие объекты пропусками или, наоборот, заполняющие пропуски в данных);

- удаление выбросов;

- масштабирование или нормализацию данных.

Итак, после знакомства с датасетом и разведочного анализа данных нам уже известно, что в итоговом датасете имеется 1023 объекта с 13ю признаками, 3 из которых будут выступать в качестве целевой переменной (входных данных), все признаки имеют очень слабую корреляцию между собой, и в датасете явно присутствуют выбросы (объекты, существенно отличающиеся от общей массы объектов датасета). Соответственно необходимо будет произвести удаление выбросов, т.к. они мешают обучению моделей.

Так же мы установили, что все характеристики являются числовыми (12 признаков с вещественными числами и один с целыми), пропусков в данных нет.

Далее проверяем наличие дубликатов и пропусков в данных. Их в нашем датасете не обнаружено, поэтому удаление дубликатов и «борьбу» с пропусками производить не нужно.

Посмотрим описательную статистику данных. Результаты приведены на рисунке 6, где:

count - количество значений;

mean - среднее значение;

std - стандартное отклонение;

min – минимум;

25% - верхнее значение первого квартиля;

50% - медиана;

75% - верхнее значение третьего квартиля;

max – максимум;

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660	5.591742
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375	2207.773481
модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	500.047452	739.664328	961.812526	1911.536477
Количество отвердителя, м.%	1023.0	110.570769	28.295911	17.740275	92.443497	110.564840	129.730366	198.953207
Содержание эпоксидных групп,%_2	1023.0	22.244390	2.406301	14.254985	20.608034	22.230744	23.961934	33.000000
Температура вспышки, С_2	1023.0	285.882151	40.943260	100.000000	259.066528	285.896812	313.002106	413.273418
Поверхностная плотность, г/м2	1023.0	482.731833	281.314690	0.603740	266.816645	451.864365	693.225017	1399.542362
Модуль упругости при растяжении, ГПа	1023.0	73.328571	3.118983	64.054061	71.245018	73.268805	75.356612	82.682051
Прочность при растяжении, МПа	1023.0	2466.922843	485.628006	1036.856605	2135.850448	2459.524526	2767.193119	3848.436732
Потребление смолы, г/м2	1023.0	218.423144	59.735931	33.803026	179.627520	219.198882	257.481724	414.590628
Угол нашивки, град	1023.0	44.252199	45.015793	0.000000	0.000000	0.000000	90.000000	90.000000
Шаг нашивки	1023.0	6.899222	2.563467	0.000000	5.080033	6.916144	8.586293	14.440522
Плотность нашивки	1023.0	57.153929	12.350969	0.000000	49.799212	57.341920	64.944961	103.988901

Рисунок 6 Описательная статистика данных

Значения признаков имеют различный масштаб, поэтому потребуется нормализация данных (приведение всех данных к некоторому заданному диапазону, обычно [0..1] или [-1..1]), т.к. имеем дело с задачей регрессии. Многие регрессионные модели достаточно чувствительны к масштабу данных. Таким образом, нормализация данных позволит сделать все признаки равными по влиянию на результат работы модели.

Так же видим, что все признаки принимают только неотрицательные значения.

Построим гистограммы распределения и диаграммы "ящика с усами" для каждого признака. Они представлены на рисунках 7 и 8.

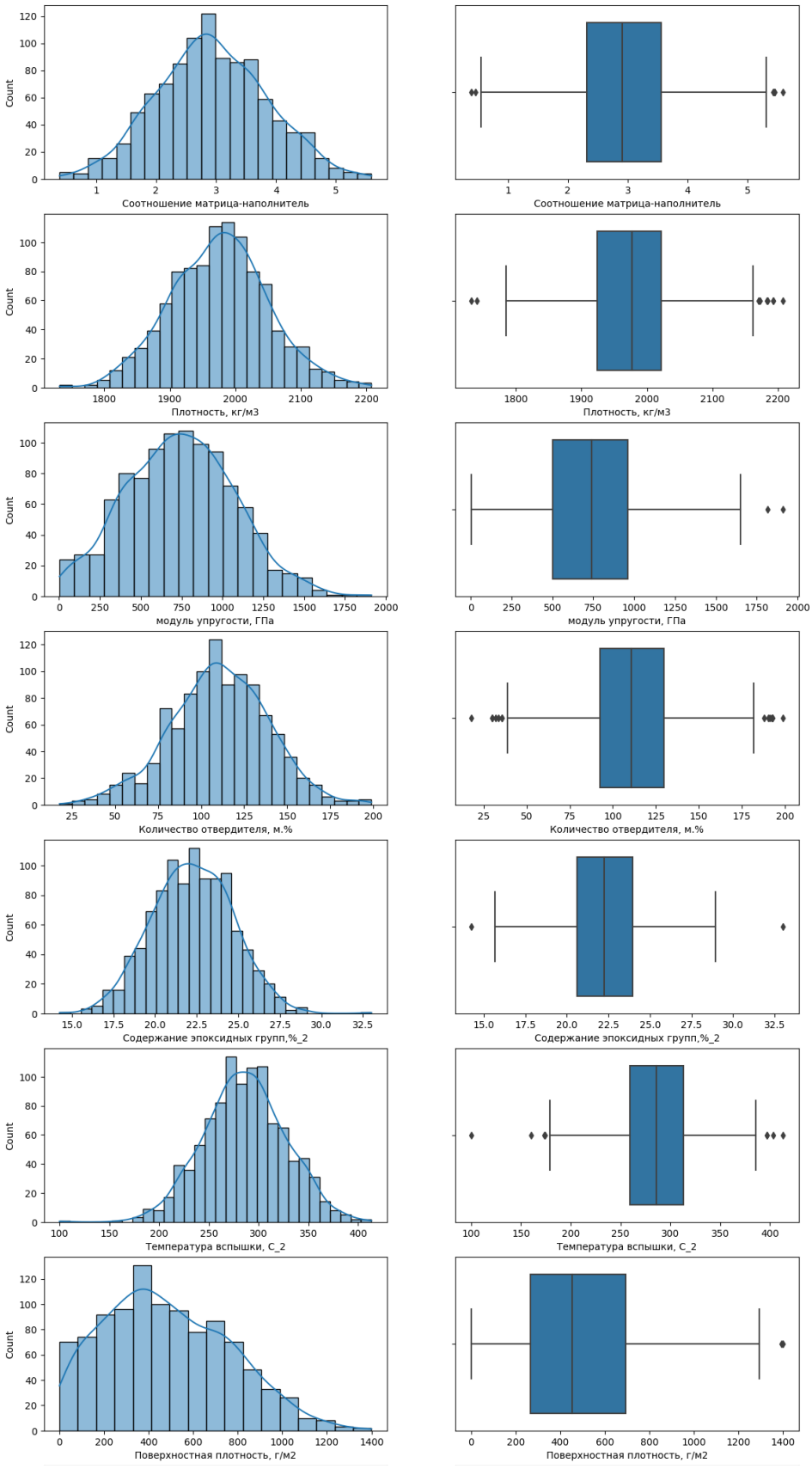

Рисунок 7 Гистограммы распределения и диаграммы "ящика с усами" для первых семи признаков

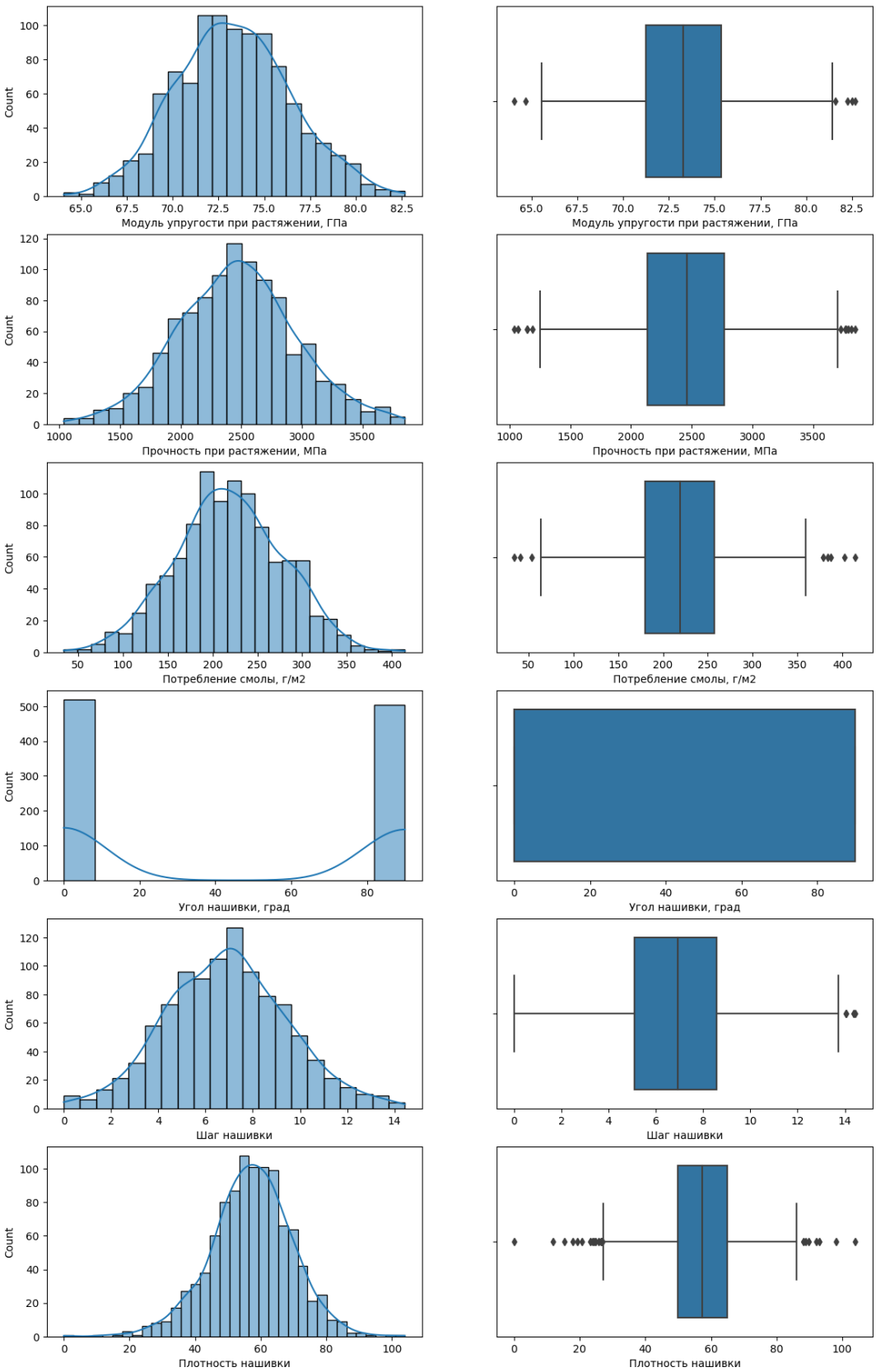

Рисунок 8 Гистограммы распределения и диаграммы "ящика с усами" для остальных шести признаков

Диаграмма «ящика с усами» (box plot) - график, использующийся в описательной статистике, компактно изображающий одномерное распределение вероятностей. Такой вид диаграммы в удобной форме показывает медиану, нижний и верхний квартили, минимальное и максимальное значение выборки и выбросы. Расстояния между различными частями ящика позволяют определить степень разброса (дисперсии) и асимметрии данных, и выявить выбросы.

Все признаки имеют распределение, достаточно близкое к нормальному (некоторые со смещением), и имеют некоторое количество выбросов. Исключение составляет признак "Угол нашивки", но это объясняется тем, что он имеет всего 2 значения в исследуемом датасете.

В данной работе датасет был полностью нормализован с помощью метода MinMaxScaler библиотеки sklearn. Этот метод преобразует каждый признак по отдельности, так, чтобы его значения находились в диапазоне от 0 до 1. На рисунке 9 можно увидеть описательную статистику данных после нормализации.

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	0.488427	0.175541	0.0	0.370696	0.483912	0.608045	1.0
Плотность, кг/м3	1023.0	0.512533	0.154890	0.0	0.404175	0.516497	0.608413	1.0
модуль упругости, ГПа	1023.0	0.386301	0.172978	0.0	0.260652	0.386165	0.502528	1.0
Количество отвердителя, м.%	1023.0	0.512273	0.156147	0.0	0.412240	0.512240	0.618003	1.0
Содержание эпоксидных групп,%_2	1023.0	0.426215	0.128370	0.0	0.338919	0.425487	0.517842	1.0
Температура вспышки, С_2	1023.0	0.593354	0.130695	0.0	0.507756	0.593401	0.679924	1.0
Поверхностная плотность, г/м2	1023.0	0.344638	0.201092	0.0	0.190296	0.322574	0.495105	1.0
Модуль упругости при растяжении, ГПа	1023.0	0.497880	0.167435	0.0	0.386030	0.494672	0.606751	1.0
Прочность при растяжении, МПа	1023.0	0.508634	0.172724	0.0	0.390881	0.506003	0.615432	1.0
Потребление смолы, г/м2	1023.0	0.484838	0.156875	0.0	0.382955	0.486875	0.587411	1.0
Угол нашивки, град	1023.0	0.491691	0.500175	0.0	0.000000	0.000000	1.000000	1.0
Шаг нашивки	1023.0	0.477768	0.177519	0.0	0.351790	0.478940	0.594597	1.0
Плотность нашивки	1023.0	0.549616	0.118772	0.0	0.478890	0.551423	0.624537	1.0

Рисунок 9 Описательная статистика после нормализации

Видно, что минимальные значения по всем признакам теперь 0, максимальные – 1. Построим еще раз гистограммы распределения и сразу посмотрим диаграммы "ящика с усами" для каждого признака после нормализации. Они представлены на рисунках 10 и 11.

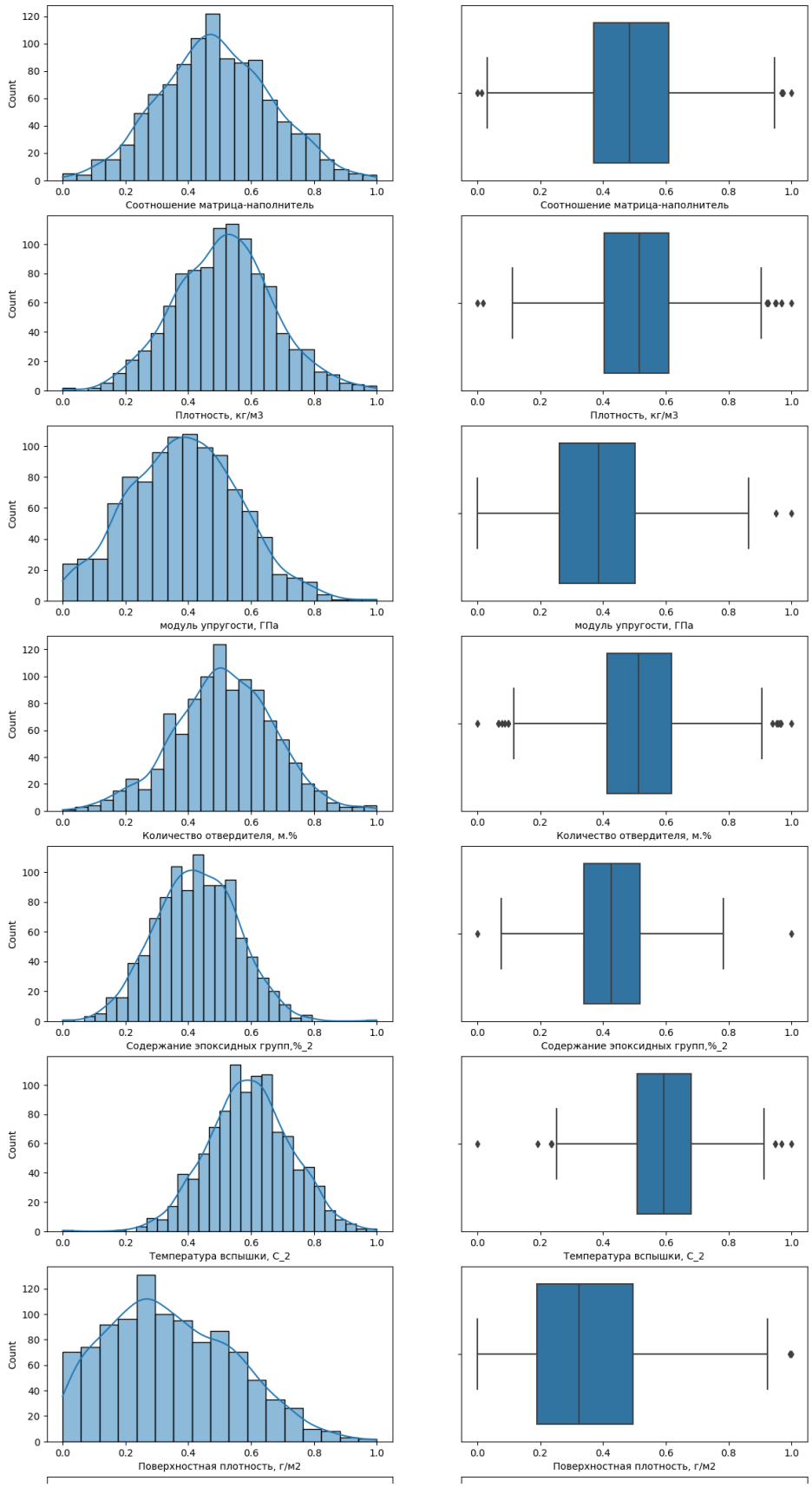

Рисунок 10 Гистограммы распределения и диаграммы "ящика с усами" для первых семи признаков после нормализации

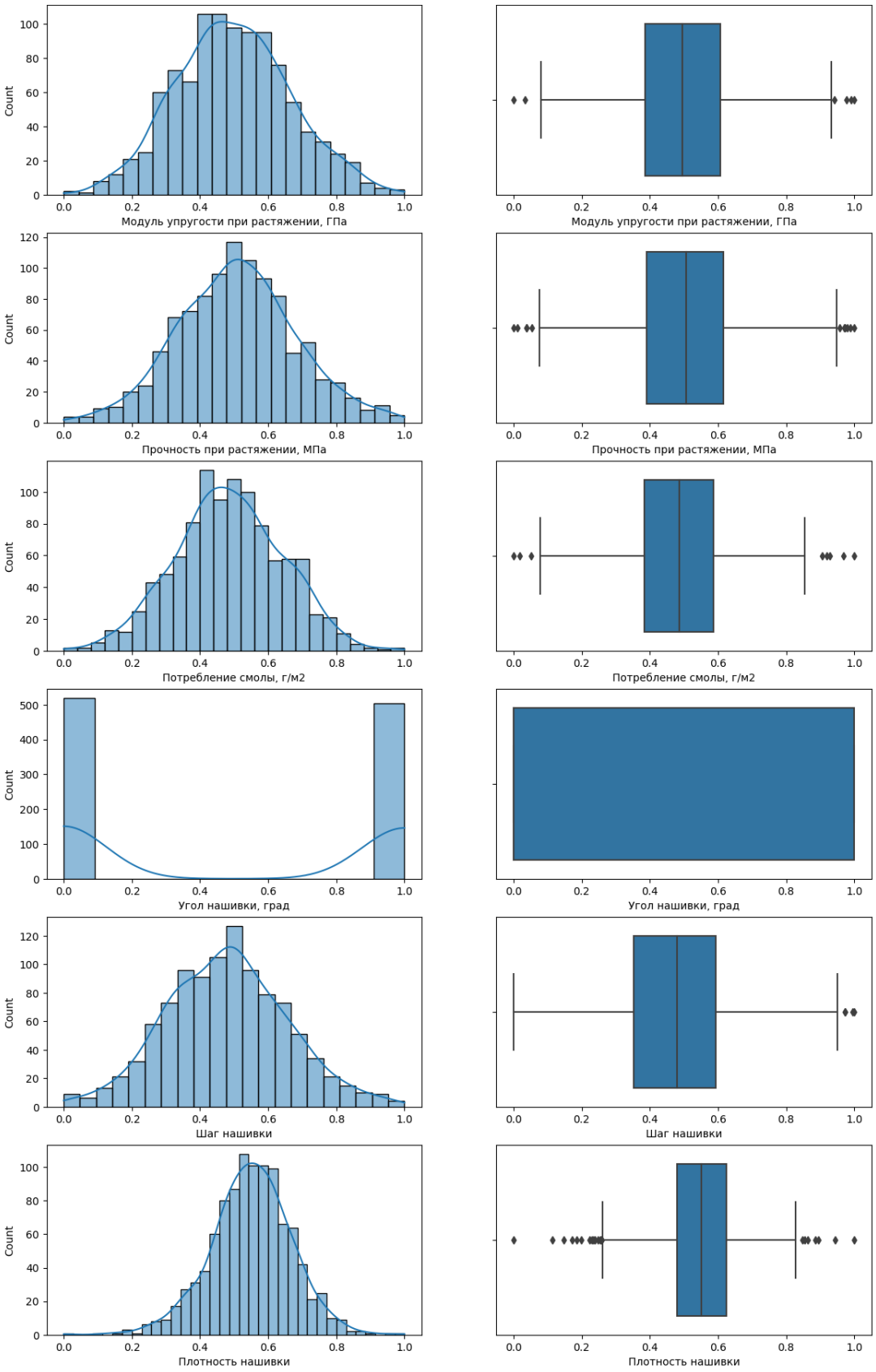

Рисунок 11 Гистограммы распределения и диаграммы "ящика с усами" для остальных шести признаков

Здесь так же видно, что значения всех признаков теперь находятся в диапазоне от 0 до 1. Общий вид гистограмм распределения при этом не изменился, как и ожидалось.

Теперь перейдем к работе с выбросами.

Существует 2 основных метода «борьбы» с выбросами. Если распределение признака близко к нормальному (визуально), то применяют стандартный метод на основе среднеквадратичного отклонения (или метод трех сигм). Если же распределение скошенное, то эффективнее применять метод на основе межквартильного размаха (или межквартильных расстояний).

Квартиль – это то значение, которые делит упорядоченные данные на части, кратные одной четверти, или 25%. Так, 1-й квартиль (Q1) – это значение, ниже которого находится 25% выборки данных. 2-й квартиль (Q2) делит совокупность данных пополам (то есть медиана), и 3-й квартиль (Q3) отделяет 25% наибольших значений. Межквартильный размах или межквартильный интервал (IQR) – это разница между 3-м и 1-м квартилями. Диаграмма «ящика с усами» как раз и демонстрирует межквартильный размах и наличие выбросов. Это хорошо видно на рисунке 12.

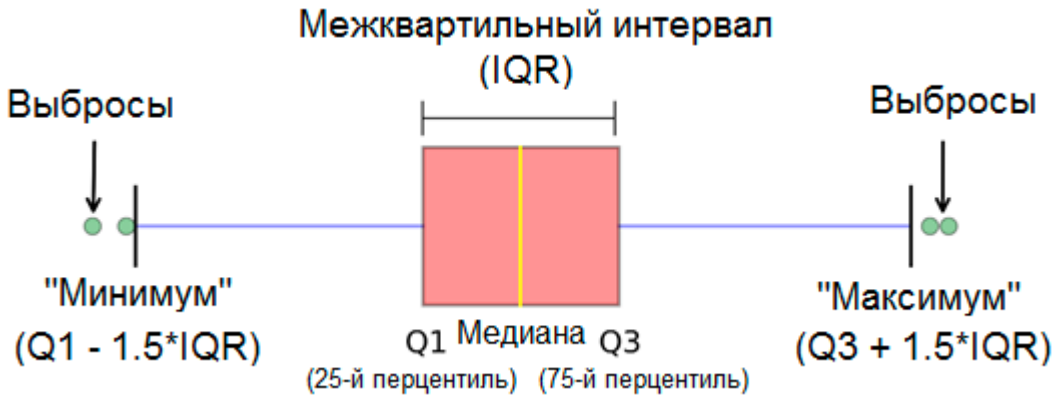

Рисунок 12 Структура "ящика с усами"

Так как гистограммах распределений, представленных ранее на рисунках 10 и 11, для некоторых признаков наблюдаем ассиметричное распределение, используем более подходящий в таких случаях метод межквартильного размаха.

Суть метода межквартильного размаха заключается в том, что выбросами «назначаются» данные, которые более чем в 1,5 межквартильных диапазонах (IQR) ниже первого квартиля или выше третьего квартиля.

На рисунке 13 видно какое количество выбросов определили для каждого признака.

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

Рисунок 13 Выбросы

Выбросов получилось не много, около 10% выборки, поэтому удаляем их. После удаления в датасете осталось 936 объектов. Посмотрим диаграммы «ящиков с усами» после удаления выбросов, результаты представлены на рисунке 14.

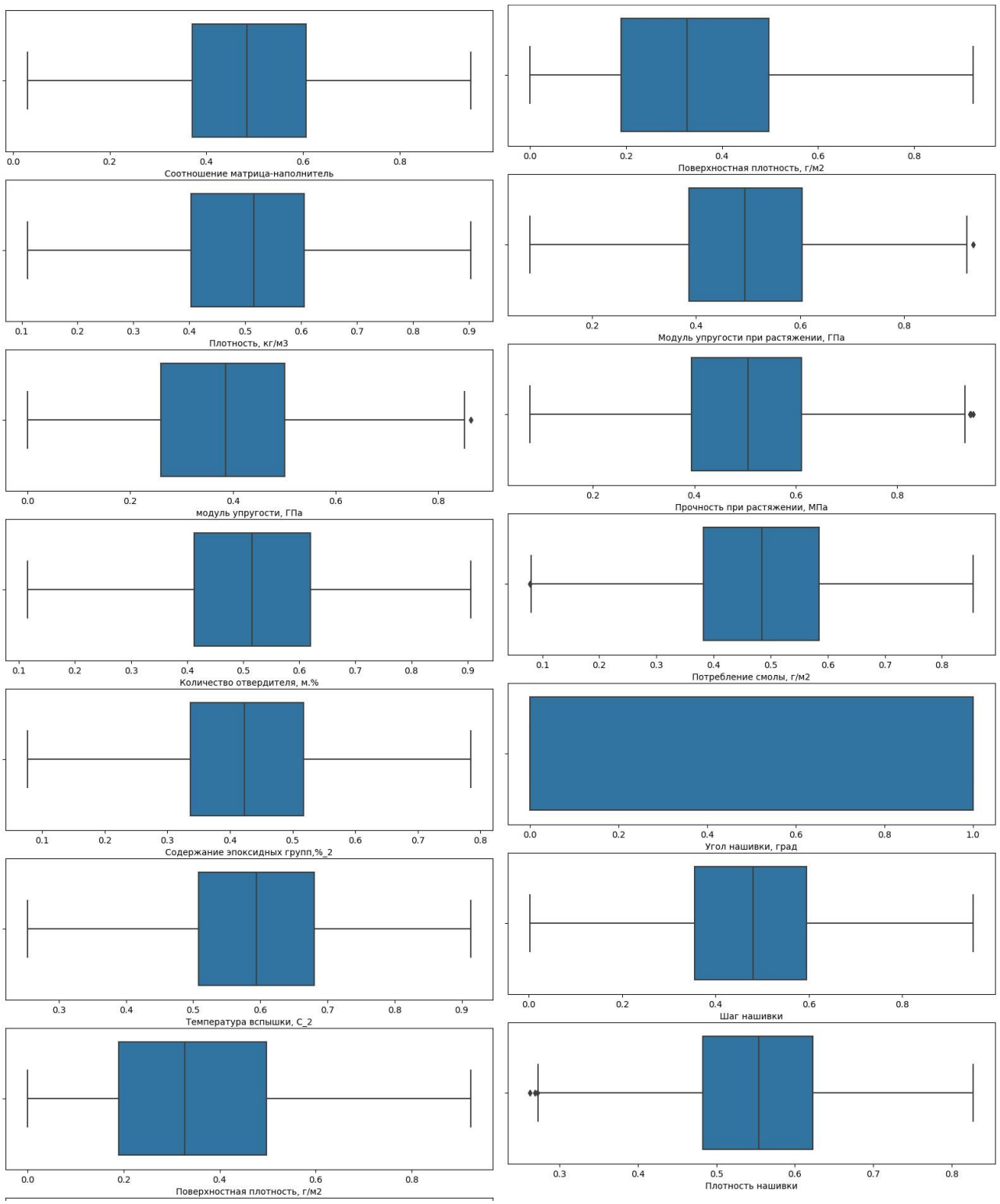

Рисунок 14 Диаграммы "ящика с усами" после удаления выбросов

Видим, что выбросов стало существенно меньше, а для некоторых признаков их вообще не стало. Дальнейшую чистку от выбросов я посчитала

нецелесообразной, так как датасет не большой и потеря еще части данных может негативно сказаться на обучении моделей.

Для задачи прогнозирования Соотношения матрица-наполнитель с помощью нейронных сетей я попробовала брать данные и очищенные от выбросов, и данные без удаления выбросов.

И финальный этап подготовки данных – это разделение датасета на тренировочную и тестовую выборку. В соответствии с аттестационным заданием датасет был разделен в соотношении 70% и 30% соответственно.

2.2 Разработка, обучение и тестирование моделей

В данной работе я буду рассматривать несколько моделей для каждой задачи и для сравнения моделей между собой, а также для оценки качества работы каждой модели необходимо определить метрики.

Существует множество различных метрик качества, применимых для задач регрессии. В этой работе я использую:

– R2 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;

– MAE (Mean Absolute Error) - средняя абсолютная ошибка, принимает значениях в тех же единицах, что и целевая переменная, по ней можно понять абсолютное значение, на которое модель ошибается;

– MSE (Mean Squared Error) - средняя квадратичная ошибка, её нельзя никак интерпретировать. Её можно только сравнить со среднеквадратичной ошибкой другой модели. Т.е. это только способ сравнить 2 модели между собой.

Так же я буду смотреть точность, рассчитанную на основе MAE, и вычисляемую по формуле (10):

Точность (%) = (1-MAE/mean)*100, (10)

где mean – среднее значение тестовой выборки.

2.2.1 Разработка, обучение и тестирование моделей для прогнозирования Модуля упругости при растяжении.

Целевой признак – Модуль упругости при растяжении.

На вход моделям будем подавать 11 признаков:

- 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.

Изначально я решила использовать следующие модели:

- Линейная регрессия (метод LinearRegression библиотеки sklearn) – дала очень слабые результаты;

- Линейная регрессия с полиномиальными параметрами (дополнительное использование метода PolynomialFeatures библиотеки sklearn) - пробовала со 2, 3, 4 и 6 степенью, однако этот вариант дал еще более слабые результаты и с увеличением степени многочлена ошибка росла, коэффициент детерминации удалялся от 1, а точность падала;

- Метод К-ближайших соседей (метод KNeighborsRegressor библиотеки sklearn) - показал существенно лучший результат по коэффициенту

детерминации, чем линейная регрессия, остальные используемые метрики дали практически те же результаты;

- Случайный лес (метод RandomForestRegressor библиотеки sklearn) – не дал хороших результатов;

- Градиентный бустинг (метод GradientBoostingRegressor библиотеки sklearn) – показал достаточно хорошие на общем фоне результаты, практически с такими же значениями метрик как и метод К-ближайших соседей;

- Дерево решений (метод DecisionTreeRegressor библиотеки sklearn) – результат оказался слабым.

Далее я решила использовать библиотеку lazypredict для дальнейшего выбора моделей. Метод LazyRegressor этой библиотеки перебирает различные регрессоры и выдает таблицу с метриками. Однако все методы были показаны с отрицательным коэффициентом детерминации. По результатам работы этой процедуры я все-таки выбрала еще два метода для исследования, из тех, что были показаны с лучшими результатами:

- Лассо регрессия (метод Lasso библиотеки sklearn) – действительно показала неплохой результат в сравнении с остальными, коэффициент детерминации по крайней мере очень близкий к 0;

- Эластичная сеть (метод ElasticNet библиотеки sklearn) – показала аналогичные результаты.

Перебор перечисленных выше моделей показал, что на данном наборе данных действительно хороших моделей не подобрать. Однако можно попытаться подобрать гиперпараметры моделям так, чтобы коэффициент детерминации получал хотя бы положительные значения.

При первичном рассмотрении моделей я подбирала гиперпараметры вручную, пробовала разные варианты. Ниже, в таблице 1, приведены лучшие результаты по рассмотренным моделям.

Таблица 1 - Сравнение моделей для прогнозирования Модуля упругости при растяжении

Название модели	R2	MSE	MAE	Точность, %
Линейная регрессия	-0,019	0.027	0.137	72.31
Полиномиальная регрессия 2го порядка	-0,238	0.033	0.148	69.944
Полиномиальная регрессия 3го порядка	-2.314	0.089	0.227	53.961
Полиномиальная регрессия 4го порядка	-18.089	0.515	0.525	-6.374
Полиномиальная регрессия 6го порядка	-16.479	0.471	0.496	-0.452
Случайный лес	-0.048	0.028	0.137	72.242
Метод К-ближайших соседей	-0.001	0.027	0.136	72.43
Градиентный бустинг	-0.001	0.027	0.135	72.559
Дерево решений	-0,131	0,03	0,141	41,404
Лассо	-0,001	0,027	0.135	72,561
Эластичная сеть	-0,001	0,027	0.135	72,561

Для четырёх лучших моделей выполняем автоматический подбор гиперпараметров. В соответствии с аттестационным заданием проведен поиск по сетке с перекрестной проверкой, количество блоков равно 10. Для этого использован метод GridSearchCV библиотеки sklearn. Лучшие результаты (с самыми оптимальными из перебранных гиперпараметрами) представлены в таблице 2.

Таблица 2 Сравнение лучших моделей для прогнозирования Модуля упругости при растяжении

Название модели с указанием параметров	R2	MSE	MAE	Точность, %
Метод К-ближайших соседей KNeighborsRegressor (n_neighbors=106)	0.001	0.027	0.136	72.424
Градиентный бустинг GradientBoostingRegressor (n_estimators=30, learning_rate=0.0001)	-0.001	0.027	0.135	72.559
Лассо Lasso(alpha=0.1)	-0,001	0,027	0.135	72,561
Эластичная сеть ElasticNet(alpha=0.1)	-0,001	0,027	0.135	72,561

Методы указаны со значимыми параметрами, значения которых отличаются от значений по умолчанию.

В результате удалось получить модель с положительным коэффициентом детерминации 0,001. Этот результат дал метод К-ближайших соседей. Однако этот лучший результат нельзя назвать хорошим, коэффициент детерминации очень близок к 0, средняя абсолютная ошибка в 0.136 достаточна высока для значений от 0 до 1. Точность в 72.424% очень низкая.

На рисунках 15-16 представлена визуализация работы моделей на тестовой выборке.

Рисунок 15 Сравнительные графики тестовых и прогнозируемых значений

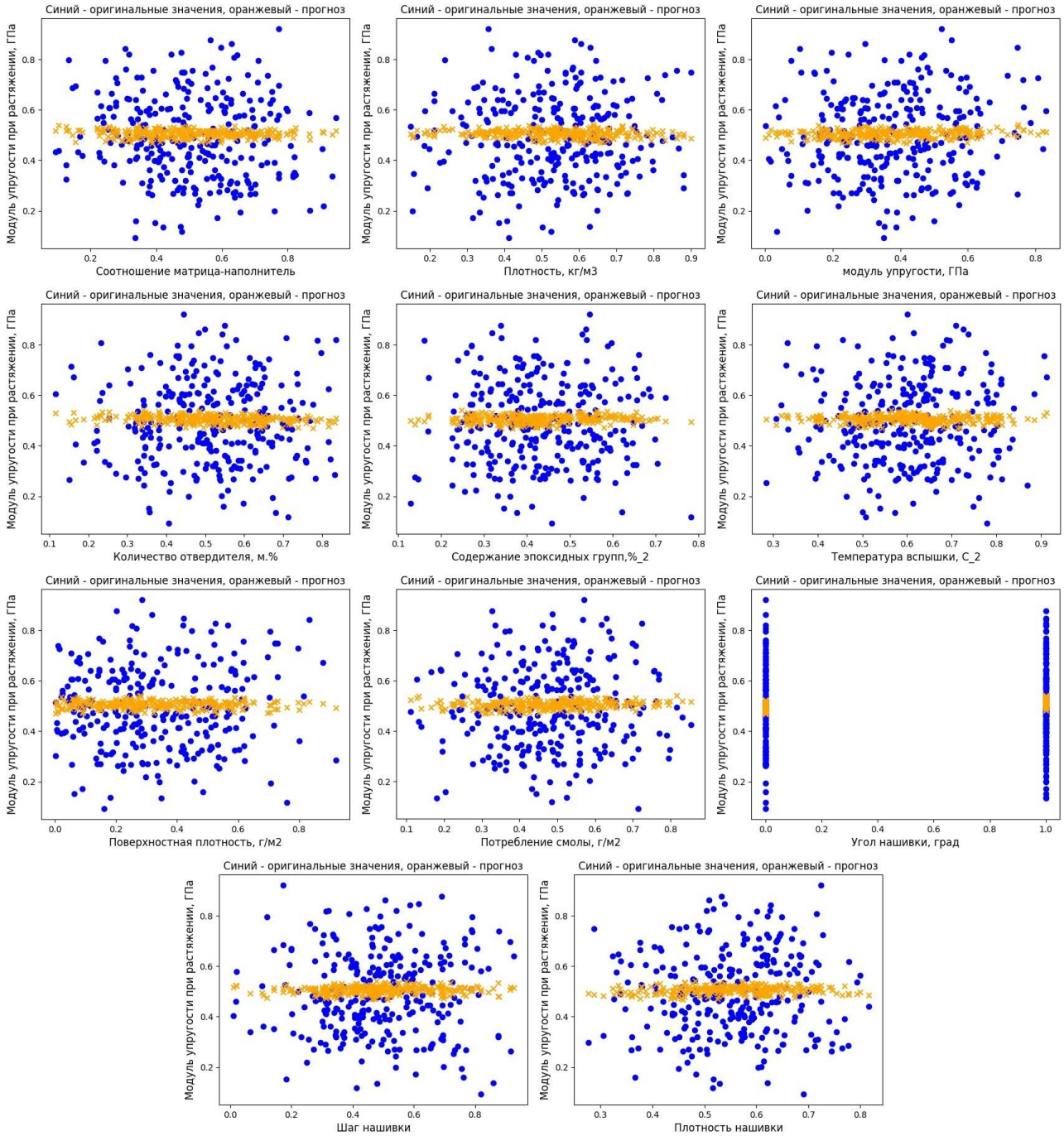

Рисунок 16 Точечные графики тестовых и прогнозируемых значений целевого признака от каждого входного признака

По графикам видно, что модель очень слабая и выдает целевые значения только из средней части всего диапазона значений выборки. Видим, насколько не соответствует лучшая модель исходным данным и насколько она неудачна.

Результат исследования отрицательный. Не удалось получить модели, которая могла бы оказать помощь в принятии решений специалисту предметной области.

2.2.2 Разработка, обучение и тестирование моделей для прогнозирования Прочности при растяжении

Целевой признак – Прочность при растяжении.

На вход моделям будем подавать 11 признаков:

- 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.

На этот раз я решила сразу использовать библиотеку lazypredict для дальнейшего выбора моделей. Метод LazyRegressor этой библиотеки перебирает различные регрессоры и выдает таблицу с метриками. Однако все методы были показаны с отрицательным коэффициентом детерминации, очевидно хороших моделей не выявлено, поэтому не будем полностью полагаться на её результаты.

Я решила исследовать следующие модели:

- Линейная регрессия (метод LinearRegression библиотеки sklearn) – дала слабые результаты;

- Линейная регрессия с полиномиальными параметрами (дополнительное использование метода PolynomialFeatures библиотеки sklearn) - пробовала со 2, 3 и 4 степенью, однако этот вариант дал еще более слабые результаты и с увеличением степени многочлена ошибка росла, коэффициент детерминации удалялся от 1, а точность падала;

- Метод К-ближайших соседей (метод KNeighborsRegressor библиотеки sklearn) – показал существенно лучший результат по коэффициенту детерминации, чем линейная регрессия, остальные используемые метрики дали практически те же результаты, уже при ручном подборе гиперпараметров удалось получить положительный коэффициент детерминации;

- Градиентный бустинг (метод GradientBoostingRegressor библиотеки sklearn) – показал слабые результаты;

- Обобщенная линейная модель с распределением Tweedie (метод TweedieRegressor библиотеки sklearn) – получили практически одинаковые значения метрик для всех вариантов распределения, и такие же как у Градиентного спуска;

Перебор перечисленных выше моделей показал, что на данном наборе данных действительно хороших моделей не подобрать. Однако можно попытаться подобрать гиперпараметры моделям так, чтобы приблизить коэффициент детерминации к единице.

При первичном рассмотрении моделей я подбирала гиперпараметры вручную, пробовала разные варианты. Ниже, в таблице 3, приведены лучшие результаты по рассмотренным моделям.

Таблица 3 Сравнение моделей для прогнозирования Прочности при растяжении

Название модели	R2	MSE	MAE	Точность, %
Линейная регрессия	-0,008	0,029	0,136	73,686
Полиномиальная регрессия 2го порядка	-0.062	0.029	0.137	72.339

Полиномиальная регрессия 3го порядка	-1.859	0.077	0.216	56.299
Полиномиальная регрессия 4го порядка	-26.868	0.751	0.552	-11.795
Метод К-ближайших соседей	0.002	0.028	0.136	73.806
Градиентный бустинг	-0.006	0.029	0.136	73.73
Обобщенная линейная модель с нормальным распределением	-0.006	0.029	0.136	73.735
Обобщенная линейная модель с распределением Пуассона	-0.006	0.029	0.136	73.732
Обобщенная линейная модель с составным распределением Гамма-Пуассона	-0.006	0.029	0.136	73.734
Обобщенная линейная модель с Гамма-распределением	-0.005	0.029	0.136	73.735
Обобщенная линейная модель с обратным распределением Гаусса	-0.005	0.029	0.136	73.74

Для трёх лучших моделей выполняем автоматический подбор гиперпараметров. В соответствии с аттестационным заданием проведен поиск по сетке с перекрестной проверкой, количество блоков равно 10. Для этого использован метод GridSearchCV библиотеки sklearn. Лучшие результаты (с самыми оптимальными из перебранных гиперпараметрами) представлены в таблице 2.

Таблица 4 Сравнение лучших моделей для прогнозирования Модуля упругости при растяжении

Название модели с указанием параметров	R2	MSE	MAE	Точность, %
Метод К-ближайших соседей KNeighborsRegressor (n_neighbors=145)	0.006	0.028	0.135	73.86
Градиентный бустинг GradientBoostingRegressor (learning_rate=0.01, n_estimators=3)	-0.006	0.029	0.136	73.73
Обобщенная линейная модель TweedieRegressor(alpha=100, max_iter=10, power=1, verbose=1)	-0.006	0.029	0.136	73.729

Методы указаны со значимыми параметрами, значения которых отличаются от значений по умолчанию.

Лучший результат опять показала модель с методом К-ближайших соседей, коэффициент детерминации 0,006. Однако этот лучший результат нельзя назвать хорошим, коэффициент детерминации по-прежнему очень близок к 0, средняя абсолютная ошибка в 0.135 достаточна высока для значений от 0 до 1. Точность в 73.86% очень низкая.

На рисунках 17-18 представлена визуализация работы моделей на тестовой выборке.

Рисунок 17 Сравнительные графики тестовых и прогнозируемых значений

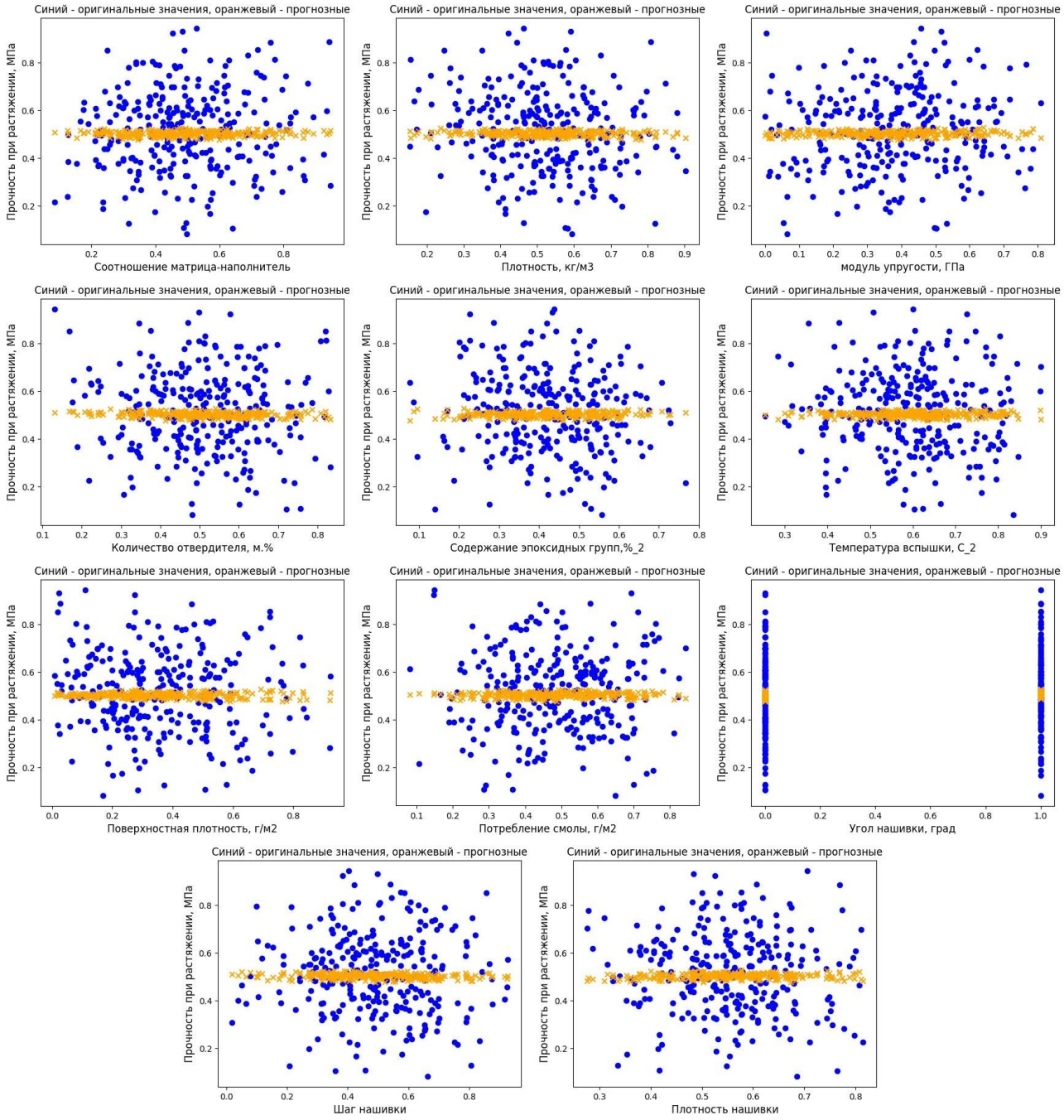

Рисунок 18 Точечные графики тестовых и прогнозируемых значений целевого признака от каждого входного признака

По графикам видно, что модель очень слабая и выдает целевые значения только из средней части всего диапазона значений выборки. Видим, насколько не соответствует лучшая модель исходным данным и насколько она неудачна.

Результат исследования отрицательный. Не удалось получить модели, которая могла бы оказать помощь в принятии решений специалисту предметной области.

2.2.3 Разработка и обучение нейронной сети для прогнозирования Соотношения матрица-наполнитель.

Целевой признак – Соотношение матрица-наполнитель.

На вход моделям будем подавать 11 признаков:

- 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;
- Потребление смолы, г/м2;
- Угол нашивки, град;
- Шаг нашивки;
- Плотность нашивки.

Нейронную сеть строю с помощью класса Sequential библиотеки keras.

Первоначально я использовала для нейросети датасет с той же предобработкой, что и для предыдущих двух задач, т.е. нормализованный и с удаленными выбросами.

Пробовала 3 немного отличающиеся архитектуры и лучшее, что удалось получить это нейронная сеть со следующими значениями метрик:

- Средняя квадратичная ошибка MSE=0.03;
- Средняя абсолютная ошибка MAE=0.143;

- Коэффициент детерминации R2=-0.001;

- Точность модели (%) 71.475.

Показатели показывают, что модель слабая. Это же подтверждает и график, представленный на рисунке 19.

Рисунок 19 Сравнительные графики тестовых и прогнозируемых значений нейросетью

Данная нейронная сеть имеет следующие параметры:

– входной слой является полносвязным, с 8 нейронами, функцией активации является гиперболический тангенс;

– один скрытый полносвязный слой с 8 нейронами, функцией аквтивации является гиперболический тангенс;

– один Dropout слой;

– выходной полносвязный слой с 1 нейроном и линейной функцией активации;

– оптимизатор: стохастический градиентный спуск (SGD);

– loss-функция: среднеквадратичная ошибка (mean_squared_error).

Нейросеть обучается за 140 эпох. Количество эпох пробовала больше, но видно, что после 140й эпохи ошибка начинает колебаться и перестает уменьшаться.

Так как результат получен неудовлетворительный, я решила попробовать изменить подход к предобработке данных:

- не делать удаление выбросов вообще, возможно для нейросети это даст положительный эффект;

- делать нормализацию только входных параметров. В таком случае нам не нужна будет денормализация для получения прогноза в естественной шкале, что упростит использование полученной модели в приложении.

После измененной предобработки данных попробовал 2 немного отличающиеся архитектуры. Удалось достичь положительного коэффициента корреляции, однако метрики все равно показывают, что модель не состоятельна.

Данная нейронная сеть имеет следующие параметры:

– входной слой

– первый полносвязный слой с 8 нейронами и функцией активации ReLu;

– второй полносвязный слой с 16 нейронами и функцией активации ReLu;

– один Dropout слой;

– выходной полносвязный слой с 1 нейроном и линейной функцией активации;

– оптимизатор: стохастический градиентный спуск (SGD);

– loss-функция: среднеквадратичная ошибка (mean_squared_error).

На рисунке 20 можно видеть архитектуру нейронной сети.

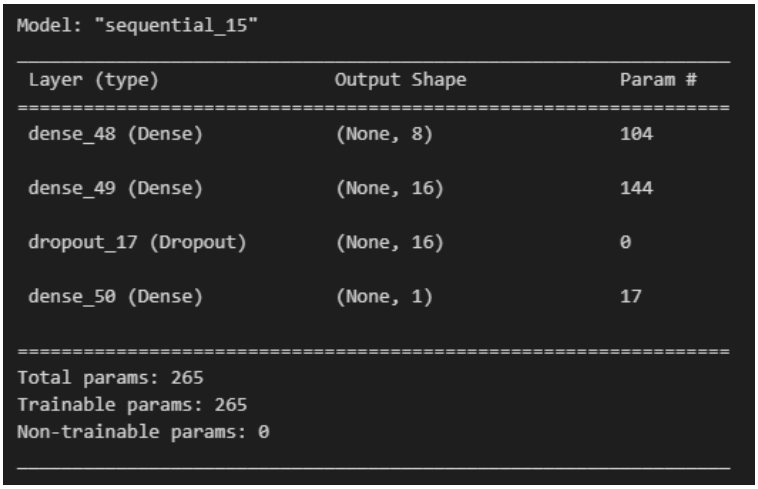

Рисунок 20 Архитектура нейронной сети

Нейросеть обучилась за 100 эпох. Количество эпох пробовала больше, но видно, что после 100й эпохи ошибка начинает колебаться и перестает уменьшаться.

Данная нейронная сеть выдает следующие значениями метрик на тестовых данных:

- Средняя квадратичная ошибка MSE=0.859;
- Средняя абсолютная ошибка MAE=0.743;
- Коэффициент детерминации R2=-0.002;
- Точность модели (%) 74.68.

Метрики показывают, что модель слабая. Это же подтверждает и график, представленный на рисунке 21.

Рисунок 21 Сравнительные графики тестовых и прогнозируемых значений нейросетью

Изменением стратегии предобработки данных не удалось достичь значимых результатов.

2.3. Создание удаленного репозитория и загрузка результатов работы на него

Для данного исследования был создан удаленный репозиторий на GitHub, который находится по адресу https://github.com/NatalyButakova/Data-Science-Graduate-Project. На него были загружены результаты работы: 2 исследовательских notebook (Butakova_BKP_part1.ipynb и Butakova_BKP_part2.ipynb), пояснительная записка (Дипломная работа Бутаковой Натальи.docx).

Заключение

В этой работе я не смогла решить поставленную задачу - не получила моделей, которые бы описывали закономерности предметной области. Я проделала ряд исследований, которые в моей компетенции как начинающего

дата-сайентиста, применила большую часть знаний, полученных в ходе прохождения курса.

Нехватка времени, отведенного на данную работу, не позволила мне создать приложение и устранить возможные причины неудачи, перечисленные ниже.

Возможные причины неудачи:

- Нечеткая постановка задачи, отсутствие дополнительной информации о зависимости признаков с точки зрения физики процесса. Незначимые признаки являются для модели шумом, и мешают найти зависимость целевых от значимых входных признаков;

- Исследование предварительно обработанных данных. Возможно, на "сырых", не предобработанных данных можно было бы получить более качественные модели, воспользовавшись другими методами очистки и подготовки;

- Неверная предобработка данных, выполненная мной. Можно было перебрать различные методы и подходы к удалению выбросов, а также к нормализации данных, возможно провести еще какую-то обработку;

- Мой недостаток знаний и опыта. Нейросети являются самым современным подходам к решению такого рода задач. Они способны находить скрытые и нелинейные зависимости в данных. Но выбор оптимальной архитектуры нейросети является неочевидной задачей.

Дальнейшие возможные пути решения этой задачи могли бы быть:

- углубиться в изучение нейросетей, попробовать различные архитектуры более осознанно, подобрать более оптимальные параметры обучения и т.д.;

- изменить методы предобработки данных;

- лучше изучить опробованные модели с целью более точного подбора гиперпараметров;

- проконсультироваться у экспертов в предметной области.

Библиографический список

1) Документация по языку программирования python: – Режим доступа: https://docs.python.org/3.8/index.html.

2) Документация по библиотеке numpy: – Режим доступа: https://numpy.org/doc/1.22/user/index.html#user.

3) Документация по библиотеке pandas: – Режим доступа: https://pandas.pydata.org/docs/user_guide/index.html#user-guide.

4) Документация по библиотеке matplotlib: – Режим доступа: https://matplotlib.org/stable/users/index.html.

5) Документация по библиотеке seaborn: – Режим доступа: https://seaborn.pydata.org/tutorial.html.

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

7) Документация по библиотеке keras: – Режим доступа: https://keras.io/api/.

8) Loginom Вики. Алгоритмы: – Режим доступа: https://wiki.loginom.ru/algorithms.html.