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 Синцова Виктория Викторовна

Москва
2023 г.

Оглавление

Введение

Тема выпускной квалификационной 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.

Цель данной работы прогнозирование конечных свойств новых композиционных материалов, используя данные о начальных свойствах компонентов композиционных материалов.

Актуальность темы заключается в том, что созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

Предмет исследования – методы используемые в Data Science для выявления закономерностей в наборах данных.

Объект исследования – свойства композитных материалов.

1. Аналитическая часть

1.1 Постановка задачи

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).

В рамках работы по прогнозированию конечных свойств композиционных материалов требуется выполнить следующее:

1. Изучить теоретические основы и методы решения поставленной задачи.

2. Выполнить разведочный анализ данных (отрисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек, для каждой переменной получить среднее и медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков).

3. Провести предобработку данных (удаление шумов, нормализацию и т.д).

4. Обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении.

5. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель.

6. Разработать приложение с графическим интерфейсом или интерфейсом командной строки, которое будет выдавать прогноз искомых величин.

7. Оценить точность модели на тренировочном и тестовом датасетах.

Входные данные предоставлены в виде двух Excel-файлов.

Файл X_bp.xlxs 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, С_2;
7) поверхностная плотность, г/м2;
8) модуль упругости при растяжении, Гпа;
9) прочность при растяжении, Мпа;
10) потребление смолы, г/м2.

Файл X_nup.xlxs 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_nup.xlxs 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). Характеристики параметров композитов представлены на рисунке 1.

Три параметра в зависимости от решаемой задачи будут становится попеременно выходными прогнозируемыми переменными, а именно:

1) модуль упругости при растяжении, Гпа;
2) прочность при растяжении, Мпа;
3) соотношение матрица-наполнитель.

При этом, будучи выходными, переменные исключаются из числа входных.

```
X_join.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 1 – Характеристики параметров композитов

1.2 Описание используемых методов

В данной работе требуется произвести регрессионный анализ данных.

Регрессионный анализ - это статистический аналитический метод, позволяющий вычислить предполагаемые отношения между зависимой переменной одной или несколькими независимыми переменными.

Создание регрессионной модели представляет собой итерационный процесс, направленный на поиск эффективных независимых переменных, чтобы объяснить зависимые переменные, которые мы пытаемся смоделировать или понять, запуская инструмент регрессии, чтобы определить, какие величины являются эффективными предсказателями. Затем пошаговое удаление и/или добавление переменных до тех пор, пока вы не найдете наилучшим образом подходящую регрессионную модель. Т.к. процесс создания модели часто исследовательский, он никогда не должен становиться простым «подгоном» данных. Процесс построения регрессионной модели должен учитывать теоретические аспекты, мнение экспертов в этой области и здравый смысл.

При анализе данных были использованы следующие методы:

1) линейная регрессия (Linear Regression);

2) гребневая регрессия (Ridge Regression);

3) регрессия по методу наименьших квадратов (Lasso Regression);

4) Градиентный бустинг (Gradient Boosting);

5) случайные лес (Random Forest Regression)

6) Регрессия дерева решений (Decision Tree Regression)

7) нейронные сети (Neural Network).

1.2.1 Линейная регрессия

Линейная регрессия (Linear regression) — это алгоритм машинного обучения, основанный на контролируемом обучении. Он выполняет задачу регрессии. Регрессионные модели представляют собой целевое значение прогноза, основанное на независимых переменных. В основном используется для выяснения взаимосвязи между переменными и прогнозирования.

Линейная регрессия сводится к нахождению уравнения вида:

$$Y = a + bx\ ;$$

Уравнение вида $Y = a + bx$ позволяет по заданным значениям фактора x иметь теоретические значения результативного признака, подставляя в него фактические значения фактора X.

Преимуществами линейной регрессии является скорость и простота получения модели, интерпретируемость модели, линейная модель является прозрачной и понятной для аналитика, по полученным коэффициентам регрессии можно судить о том, как тот или иной фактор влияет на результат, сделать на этой основе дополнительные полезные выводы, широкая применимость. К недостаткам можно отнести чувствителен к выбросам и шумам, а так же то, что линейная регрессия не может использоваться, когда связь между зависимой и независимой переменной не является линейной.

1.2.2 Гребневая регрессия

Ридж-регрессия или гребневая регрессия (англ. ridge regression) — это один из методов понижения размерности. Часто его применяют для борьбы с переизбыточностью данных, когда независимые переменные коррелируют друг с другом (т.е. имеет место мультиколлинеарность). Следствием этого является плохая обусловленность матрицы $X^T X$ и неустойчивость оценок коэффициентов регрессии. Оценки, например, могут иметь неправильный знак или значения, которые намного превосходят те, которые приемлемы из физических или практических соображений.

Применение гребневой регрессии нередко оправдывают тем, что это практический приём, с помощью которого при желании можно получить меньшее значение среднего квадрата ошибки.

Самым большим преимуществом гребневой регрессии является ее способность давать более низкую среднеквадратичную ошибку теста (MSE) по сравнению с регрессией наименьших квадратов, когда присутствует мультиколлинеарность.

Однако самым большим недостатком гребневой регрессии является ее неспособность выполнять выбор переменных, поскольку она включает все переменные-предикторы в окончательную модель. Поскольку некоторые предикторы будут сжаты очень близко к нулю, это может затруднить интерпретацию результатов модели.

1.2.3 Регрессия по методу наименьших квадратов

Lasso (Least absolute shrinkage and selection operator) - метод оценивания коэффициентов линейной регрессионной модели.

Метод заключается во введении ограничения на норму вектора коэффициентов модели, что приводит к обращению в 0 некоторых коэффициентов модели. Метод приводит к повышению устойчивости модели в случае большого числа обусловленности матрицы признаков X, позволяет

получить интерпретируемые модели - отбираются признаки, оказывающие наибольшее влияние на вектор ответов.

Основными достоинствами лассо являются большое измерение. Лассо работает в тех случаях, когда количество людей меньше, чем количество переменных, если, однако, небольшое количество этих переменных оказывает влияние на наблюдения (предположение экономичности). Это свойство неверно в случае классической линейной регрессии с сопутствующим риском, который увеличивается по мере увеличения размерности пространства переменных, даже если предположение о экономии подтверждается.

Разреженный выбор: лассо используется для выбора ограниченного подмножества переменных (в зависимости от параметра). Такой ограниченный выбор часто позволяет лучше интерпретировать модель.

Последовательность выбора: когда истинный вектор решения полый, то есть только подмножество переменных используется для прогноза, при правильных условиях лассо сможет выбрать эти переменные из интереса раньше всех других переменных.

С другой стороны, были продемонстрированы определенные пределы лассо. Сильная корреляция: если переменные сильно коррелированы друг с другом и важны для прогноза, лассо будет отдавать предпочтение одному в ущерб другим. Другой случай, когда корреляции проблематичны, это когда интересующие переменные коррелируют с другими переменными. В этом случае последовательность выбора лассо больше не гарантируется.

Очень большая размерность: когда, в частности, размерность слишком велика (очень большая по сравнению с) или истинный вектор недостаточно полый (слишком много интересующих переменных), лассо не сможет найти все эти интересующие переменные.

1.2.4 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 соответствуют 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.).

К преимуществам можно отнести то, что алгоритм работает с любыми функциями потерь. Предсказания в среднем лучше, чем у других алгоритмов. Алгоритм может самостоятельно справляется с пропущенными данными.

При этом алгоритм крайне чувствителен к выбросам и при их наличии будет тратить огромное количество ресурсов на эти моменты. Имеет большие затраты времени на вычисления и необходимо грамотно подбирать гиперпараметры.

1.2.5 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.

Алгоритм 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.

Так же 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, а также когда данные имеют пропущенные значения или они не были хорошо.

К минусам алгоритма можно отнести 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.

1.2.6 Дерево решений

Регрессия дерева решений строит регрессионную модель в виде древовидной структуры. Структура дерева представляет собой «листья» и «ветки». На рёбрах («ветках») дерева решения записаны признаки, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах — признаки, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе. Каждый лист представляет собой значение целевой переменной, изменённой в ходе движения от корня по рёбрам дерева до листа. Каждый внутренний узел сопоставляется с одной из входных переменных. Алгоритм вычисляет информационный прирост для каждой характеристики и выбирает ту, которая дает наивысшее значение.

Деревья принятия решений имеют следующие преимущества:

- они эффективны с точки зрения вычисления и использования памяти во время обучения и прогнозирования;

- они могут представлять границы нелинейного принятия решений;

- они выполняют выбор признаков и классификацию и являются устойчивыми при наличии шумовых признаков.

Недостатки алгоритма следующие:

- склонность к переобучению. Завершенная модель дерева решений может быть чрезмерно сложной и содержать ненужную структуру;

- плохие результаты на небольших наборах данных.

Эта модель регрессии состоит из совокупности деревьев принятия решений. Каждое дерево в регрессионном лесу решений выводит распределение по Гауссу в виде прогноза. По совокупностям деревьев выполняется агрегирование с целью найти распределение по Гауссу, ближайшее к объединенному распределению для всех деревьев модели.

1.2.7 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 (рисунок 2). 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.

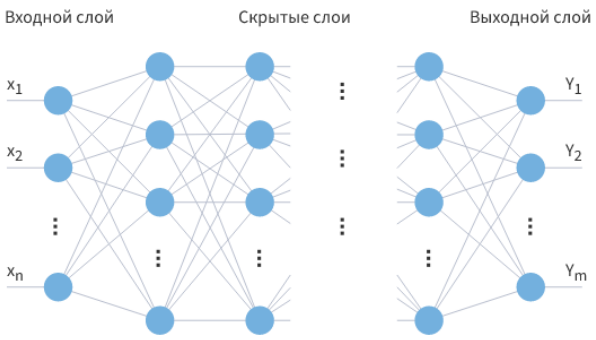


Рисунок 2 – Нейронная сеть

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. 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 использует 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).

Нейронные сети эффективны при моделировании сложных нелинейных отношений ввиду многослойности, гибки (не нужно беспокоиться о структуре данных в нейронных сетях), их производительность растет с увеличением тренировочных данных. Но возможна сложная архитектура и модель в целом. Требуется тщательная настройка гиперпараметров и скорости обучения, а для достижения высокой производительности нейронным сетям необходимо огромное количество данных, и в результате, как правило, нейросети уступают другим.

1.3 Метрики оценки качества прогнозирования

Для оценки точности и качества работы выбранных моделей прогнозирования и нейронных сетей применяются следующие метрики:

- MAE (средняя абсолютная ошибка) - определяет среднее абсолютное расстояние между прогнозируемыми и целевыми значениями – то, насколько число в прогнозе разошлось с реальным числом. Данную ошибку удобно трактовать – погрешность измеряется в тех же единицах, что и значения целевой переменной.

$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|$, где n – количество элементов, y - реальное целевое значение, $\hat{y}$ - предсказанной целевое значение.

- MSE (средняя квадратичная ошибка) - определяет среднеквадратичную ошибку между прогнозируемыми и целевыми значениями. Настроена на отражение влияния именно больших ошибок на качество модели. Менее удобна для понимания ввиду измерения в квадратных единицах. Данная метрика обычно применяется для сравнения моделей между собой.

$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, где n – количество элементов, y - реальное целевое значение, $\hat{y}$ - предсказанной целевое значение.

- R2 (коэффициент детерминации) - показывает, какую долю разнообразия данных модель смогла объяснить. Метрика просто интерпретируема: модель, для которой R2 больше 0,5, является удовлетворительной. Если R2 больше 0,8, то модель рассматривается как очень хорошая. Значения, меньшие 0,5, говорят о том, что модель некачественна.

$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\bar{y}_i - y_i)^2}$, n – количество элементов, y - реальное целевое значение, $\hat{y}$ - предсказанной целевое значение.

2. Практическая часть

2.1 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.

В ходе решения поставленной задачи применим следующие инструменты разведочного анализа:

– гистограммы распределения каждой из переменной;

– диаграммы ящика с усами;

– попарные графики рассеяния точек;

– описательная статистика для каждой переменной;

– анализ и исключение выбросов;

– проверка наличия пропусков.

Проверка наличия пропусков показала, что впропусков в датасете нет (рисунок 3). Это может косвенно свидетельствовать о том, что датасет уже обрабатывался.

```
[ ]  X_join.isnull().sum()
```

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

Рисунок 3 – Количество пропусков в датасете

Результаты проверки датасета на количество уникальных значений в каждом столбце показаны на рисунке 4.

```
X_join.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 4 - Количество уникальных значений в столбцах датасета

Общее количество строк в датасете 1023, при этом количество уникальных значений варьируется от 988 до 1014 (столбец "Угол нашивки, град" не рассматриваем). Т.к. данные являются результатом экспериментальных измерений, такое количество уникальных значений кажется странным.

Рассмотрим данные в столбцах более подробно. Для этого рассмотрим графики данных в каждом столбце (на рисунке 5 приведены некоторые из них).

Из графиков видно, что начальные значения в данных либо являются ошибками, либо искусственные (рисунок 6). Чтобы они не влияли на моделирование необходимо их удалить (40 строк).

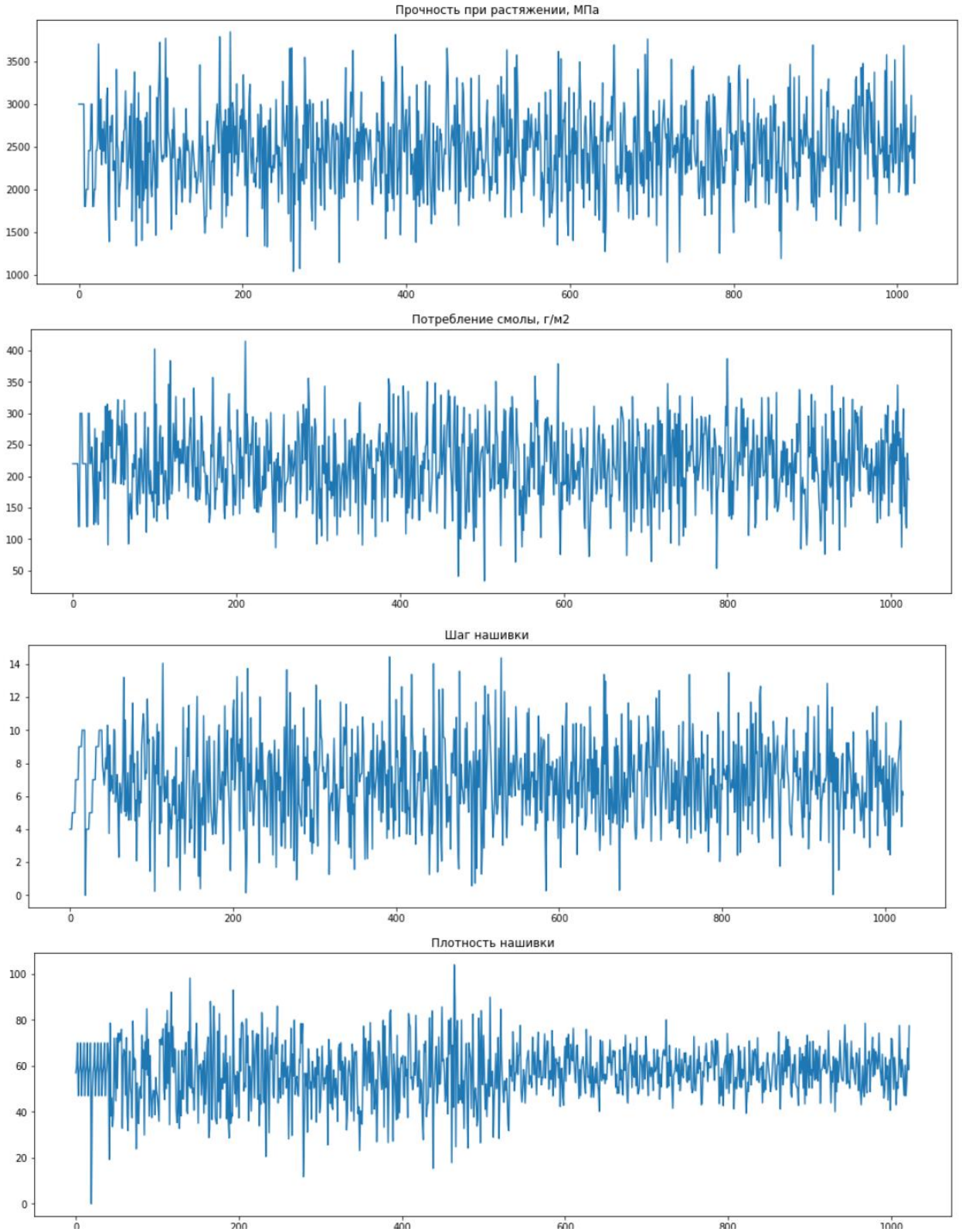


Рисунок 5 – Графики данных

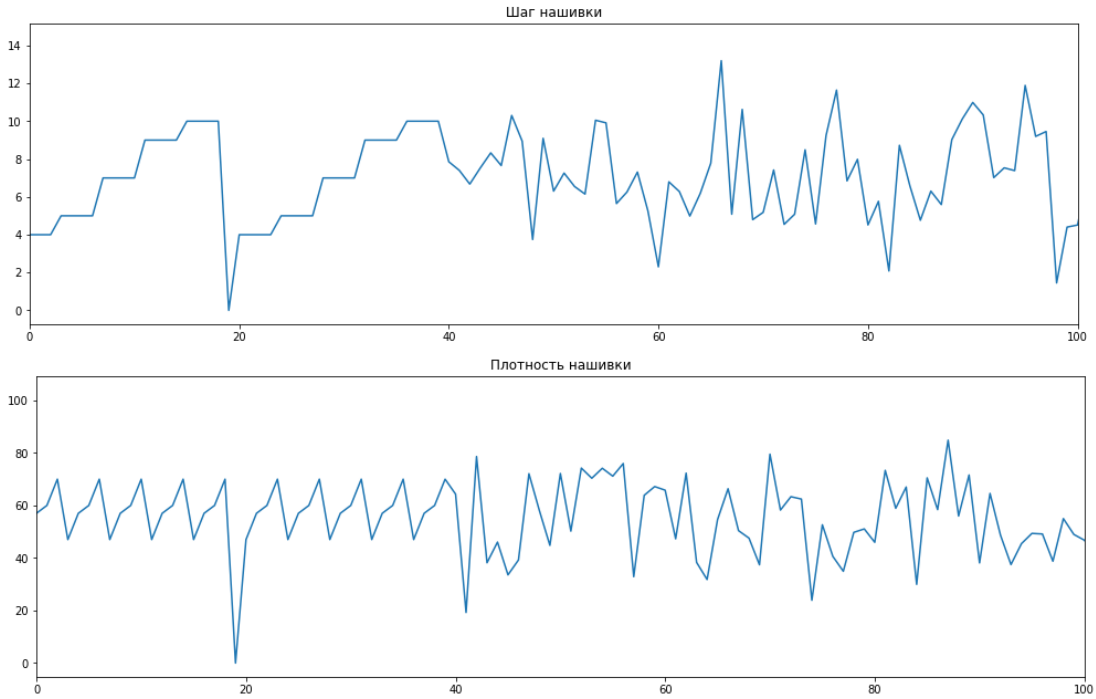
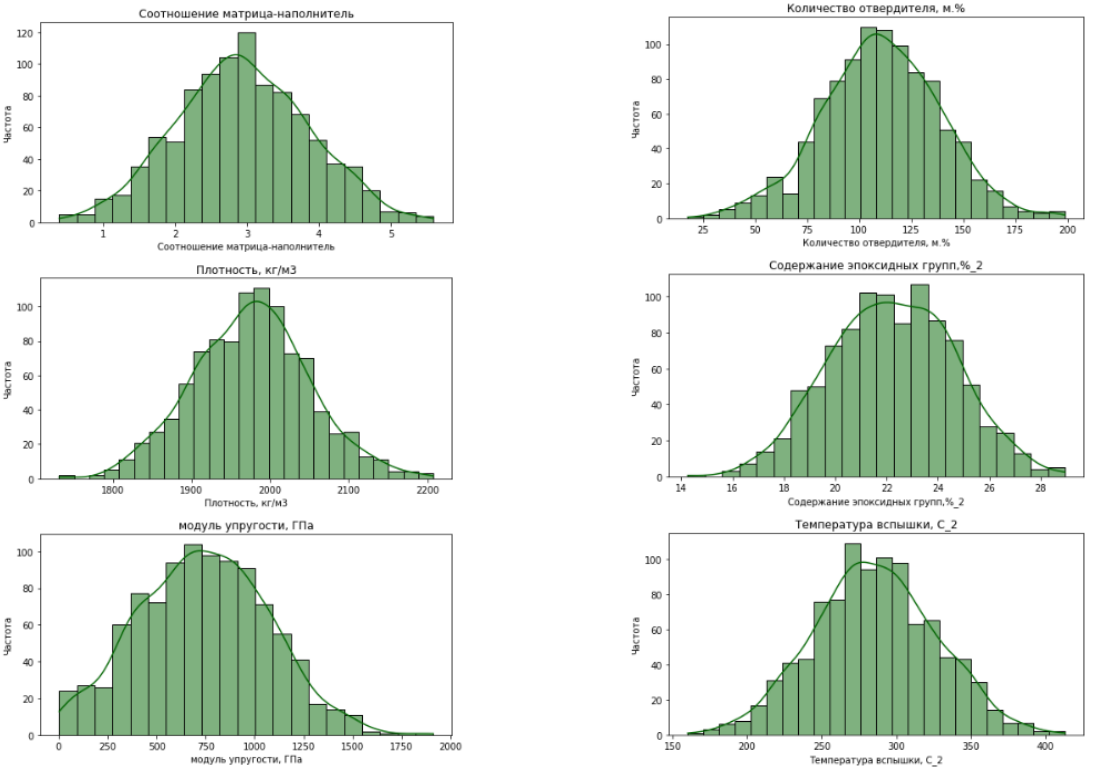


Рисунок 6 – Графики данных (первые 100 значений)

Гистограммы распределения данных приведены на рисунке 7. 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.



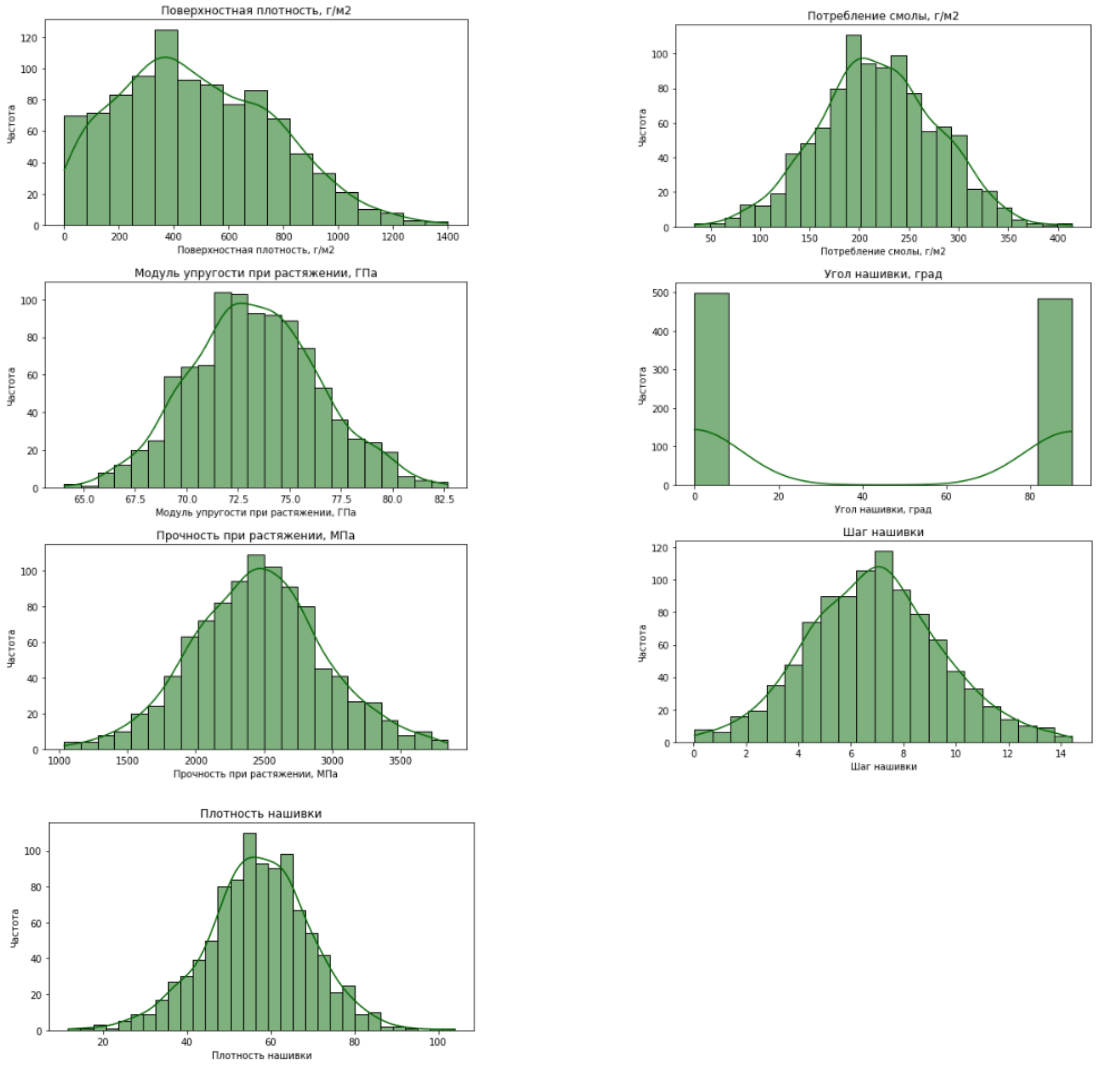


Рисунок 7 - Гистограммы распределения

Для определения наличия выбросов в датасете удобно использовать диаграммы «ящик с усами» (рисунок 8). Можно сделать вывод, что выбросы в датасете присутствуют и в дальнейшем их нужно исключить из рассмотрения. Для этого найденные выбросы (метод межквартильных расстояний) были заменены на пустые значения, после чего строки с пустыми значениями были удалены из датасета. После удаления выбросов осталось 890 строк со значениями.

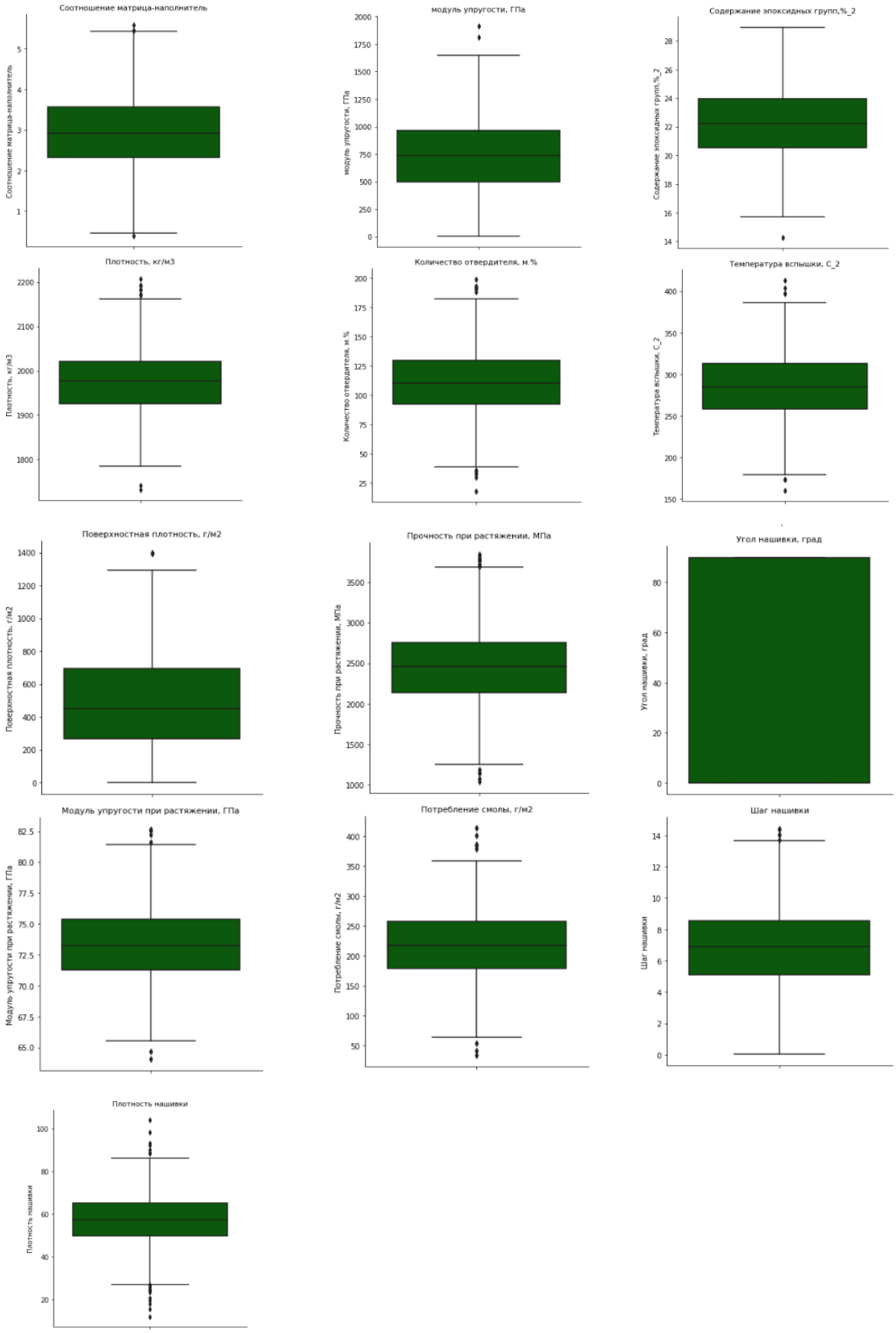


Рисунок 8 - диаграммы «ящик с усами»

Описательная статистика окончательного датасета приставлена на рисунке 9.

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	890.0	2.928925	0.896982	0.547391	2.320191	2.908811	3.551339	5.314144
Плотность, кг/м3	890.0	1974.340221	70.869622	1784.482245	1923.887189	1977.603973	2020.082671	2161.565216
модуль упругости, ГПа	890.0	735.880602	327.653047	2.436909	498.275517	733.016158	960.465724	1649.415706
Количество отвердителя, м.%	890.0	111.010058	26.973789	38.668500	92.497018	110.573604	130.404874	181.828448
Содержание эпоксидных групп,%_2	890.0	22.187011	2.418853	15.695894	20.521955	22.146953	23.966198	28.955094
Температура вспышки, С_2	890.0	285.664458	39.879163	179.374391	258.386295	285.853960	313.040444	386.067992
Поверхностная плотность, г/м2	890.0	482.105364	280.915278	0.603740	265.027350	452.891920	696.248199	1291.340115
Модуль упругости при растяжении, ГПа	890.0	73.302274	3.039914	65.793845	71.241213	73.184259	75.322715	81.417126
Прочность при растяжении, МПа	890.0	2463.507094	457.353044	1250.392802	2150.188224	2456.394188	2751.499231	3660.450210
Потребление смолы, г/м2	890.0	218.010506	57.575275	64.524180	179.719792	216.779521	256.995883	359.052220
Угол нашивки, град	890.0	45.910112	45.016093	0.000000	0.000000	90.000000	90.000000	90.000000
Шаг нашивки	890.0	6.909465	2.506636	0.145034	5.167202	6.922196	8.568608	13.653987
Плотность нашивки	890.0	57.428007	11.288299	28.237746	50.211993	57.546947	64.798593	86.012427

Рисунок 9 – Описательная статистика данных

Для визуализации коэффициентов корреляции и определения того, между какими переменными установлена более тесная взаимосвязь, была построена тепловая карта коэффициентов корреляции (рисунок 10) и попарные графики рассеяния точек (рисунок 11).

Рисунок 10 - тепловая карта коэффициентов корреляции

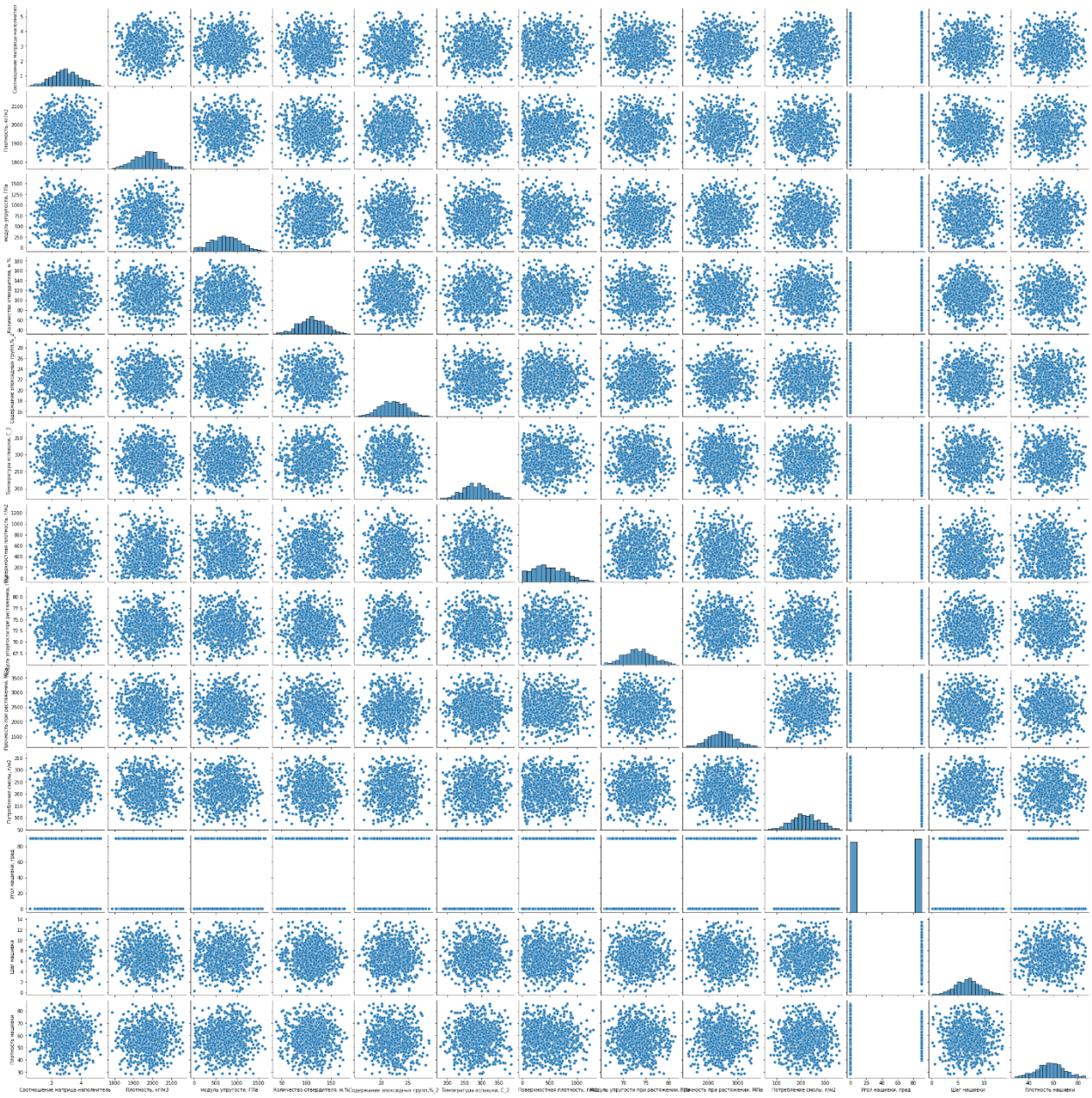


Рисунок 11 - попарные графики рассеяния точек

Разведочный анализ данных показал, что линейной связи между любыми переменными нет, корреляция равна 0 это наглядно видно на рисунках 10 и 11. При этом все параметры (за исключением параметра «Угол нашивки») имеют нормальное распределение, что может свидетельствовать о случайной генерации чисел.

2.2 Предобработка данных

При выполнении разведочного анализа данных было замечено, что значения данных изменяются в очень больших диапазонах и также у разных параметров отличаются на порядки (рисунок 12).

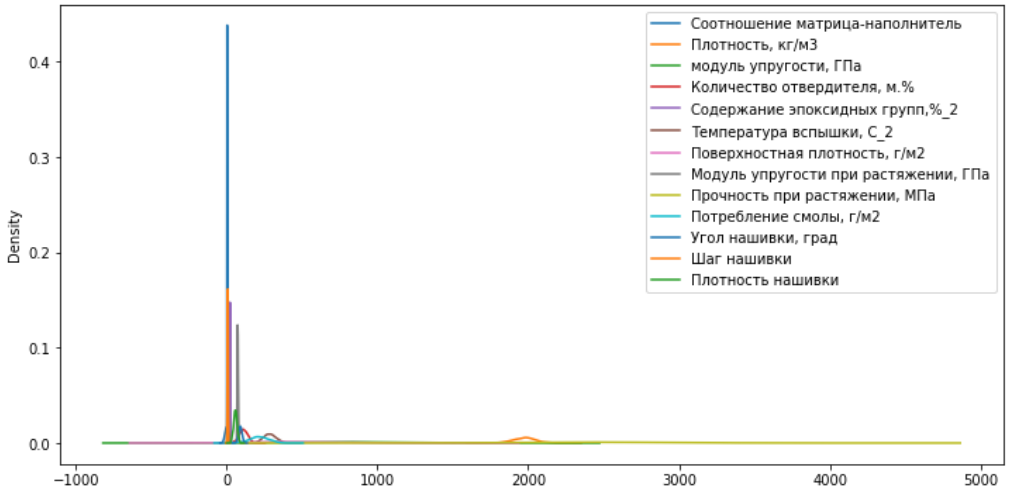


Рисунок 12 - Разброс данных до нормализации

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 (рисунок 13).

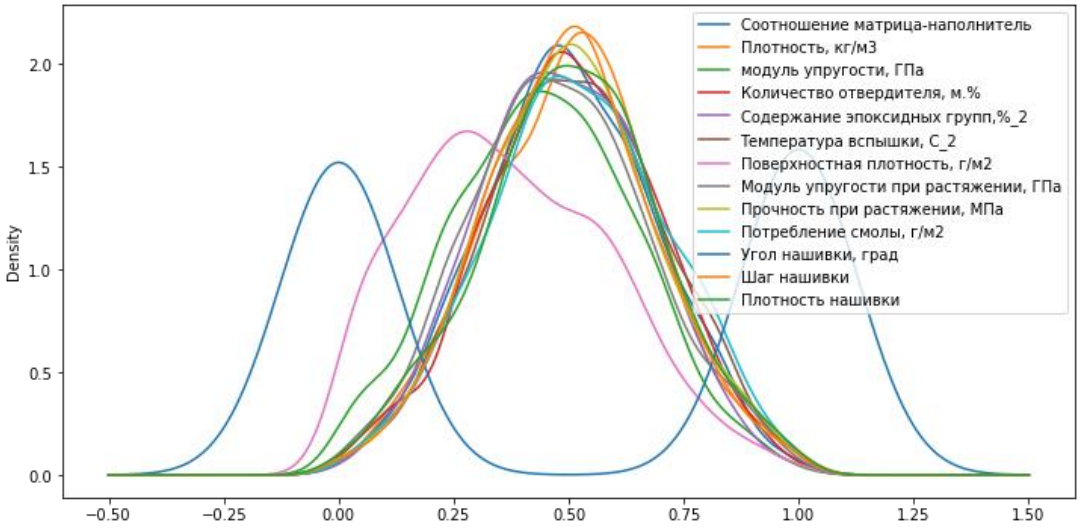


Рисунок 13 - Разброс данных после нормализации

На рисунке14 приведена описательная статистика после нормализации.

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	890.0	0.499613	0.188175	0.0	0.371909	0.495394	0.630187	1.0
Плотность, кг/м3	890.0	0.503491	0.187942	0.0	0.369693	0.512147	0.624797	1.0
модуль упругости, ГПа	890.0	0.445327	0.198942	0.0	0.301059	0.443588	0.581689	1.0
Количество отвердителя, м.%	890.0	0.505320	0.188417	0.0	0.376003	0.502271	0.640796	1.0
Содержание эпоксидных групп,%_2	890.0	0.489556	0.182428	0.0	0.363978	0.486535	0.623741	1.0
Температура вспышки, С_2	890.0	0.514240	0.192939	0.0	0.382266	0.515157	0.646687	1.0
Поверхностная плотность, г/м2	890.0	0.373044	0.217640	0.0	0.204863	0.350411	0.538952	1.0
Модуль упругости при растяжении, ГПа	890.0	0.480592	0.194576	0.0	0.348670	0.473039	0.609915	1.0
Прочность при растяжении, МПа	890.0	0.503355	0.189769	0.0	0.373350	0.500404	0.622851	1.0
Потребление смолы, г/м2	890.0	0.521126	0.195483	0.0	0.391119	0.516947	0.653492	1.0
Угол нашивки, град	890.0	0.510112	0.500179	0.0	0.000000	1.000000	1.000000	1.0
Шаг нашивки	890.0	0.500737	0.185554	0.0	0.371766	0.501679	0.623555	1.0
Плотность нашивки	890.0	0.505243	0.195385	0.0	0.380344	0.507302	0.632818	1.0

Рисунок 14 – Описательная статистика после нормализации

2.3 Разбиение и предобработка данных

Поставленная задача предполагает прогнозирование трех переменных: «Модуль упругости при растяжении», «Прочность при растяжении» и «Соотношение матрица-наполнитель». Разделяем исходный датасет на «X» (feature-переменные) и «y» (target-переменные). К target-переменные относятся : «Модуль упругости при растяжении, Гпа», «Прочность при растяжении, Мпа», «Соотношение матрица-наполнитель».

Полученные данные разделем на обучающие и тестовые выборки методом train_test_split, размер тестовой выборки принят равным 30% от общей, путем указания значения random_state зафиксирован набор данных в выборках для создания единых условий для разных моделей прогнозирования.

2.4 Разработка и обучение модели

Для решения задачи предсказания модуля упругости при растяжении и прочности при растяжении были использованы следующие методы:

- линейная регрессия (Linear Regression);
- гребневая регрессия (Ridge Regression);
- регрессия по методу наименьших квадратов (Lasso Regression);
- градиентный бустинг (Gradient Boosting);
- случайные лес (Random Forest Regression);
- регрессия дерева решений (Decision Tree Regression).

В работе с каждой моделью придерживались следующего алгоритма:

1. вызвать модель регрессии, передать параметры;

2. вызвать метод GridSearchCV – это инструмент для автоматического подбора гиперпараметров для моделей машинного обучения путем поиска по сетке с перекрестной проверкой, т.е. он создает модель для каждой возможной комбинации параметров. Передаем в GridSearchCV модель регрессии и возможные параметры. Количество блоков по условию задачи равняется 10. Выводим результат лучшего параметра;

3. обучаем модель с учетом лучших параметров;

4. считаем ошибки модели, вносим в сводную таблицу по всем моделям.

В качестве оценки работы моделей были использованы следующие метрики: MSE (среднеквадратичная ошибка), R2 (Коэффициент детерминации), MAE (средняя абсолютная ошибка).

2.5 Тестирование модели

Результаты оценок каждой модели для предсказания модуля упругости при растяжении и прочности при растяжении представлены на рисунках 15 и 16.

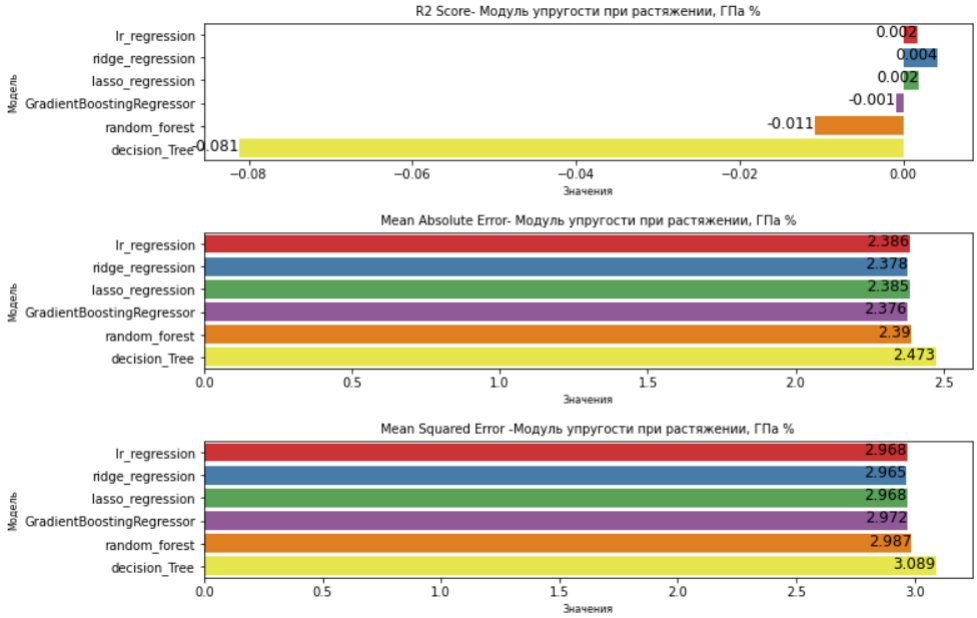


Рисунок 15 – Ошибки модели предсказания модуля упругости

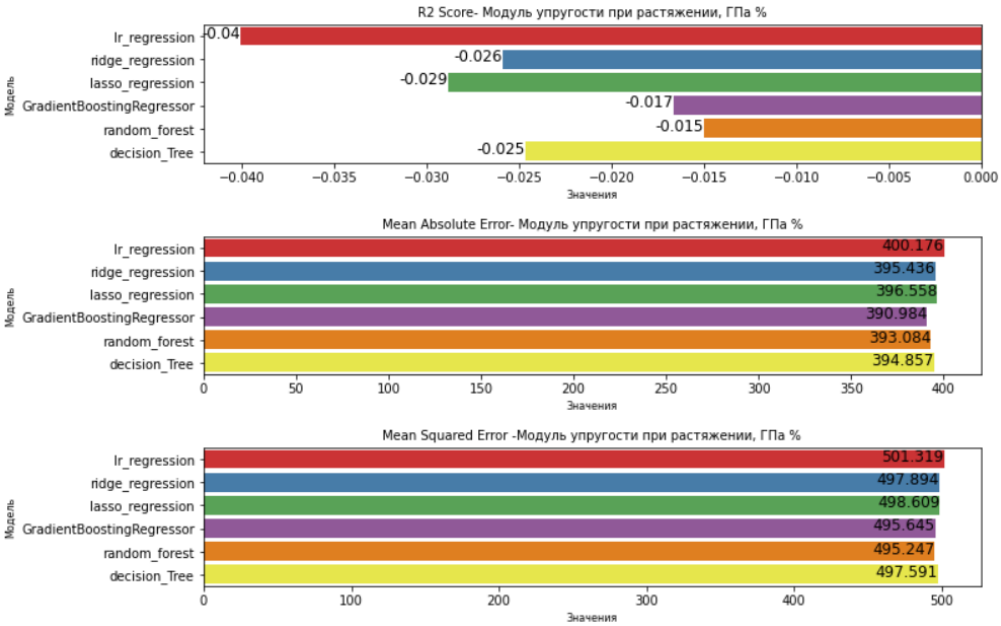


Рисунок 16 - Ошибки модели предсказания прочности

Результаты построения и обучения моделей, к сожалению, не дали значительного положительного результата. Все модели крайне плохо описывают исходные данные - не удалось добиться положительного сильно отличающегося от 0 значения R2.

Наилучший коэффициент детерминации в предсказании модуля упругости при растяжении получилась у модели гребневая регрессия (чуть больше 0), в предсказании прочности при растяжении у модели «случайный лес» (отрицательный).

На рисунке 17 приведена визуализация работы лучшей модели на тестовом множестве для предсказания модуля упругости при растяжении, на рисунке 18 – для предсказания прочности при растяжении.

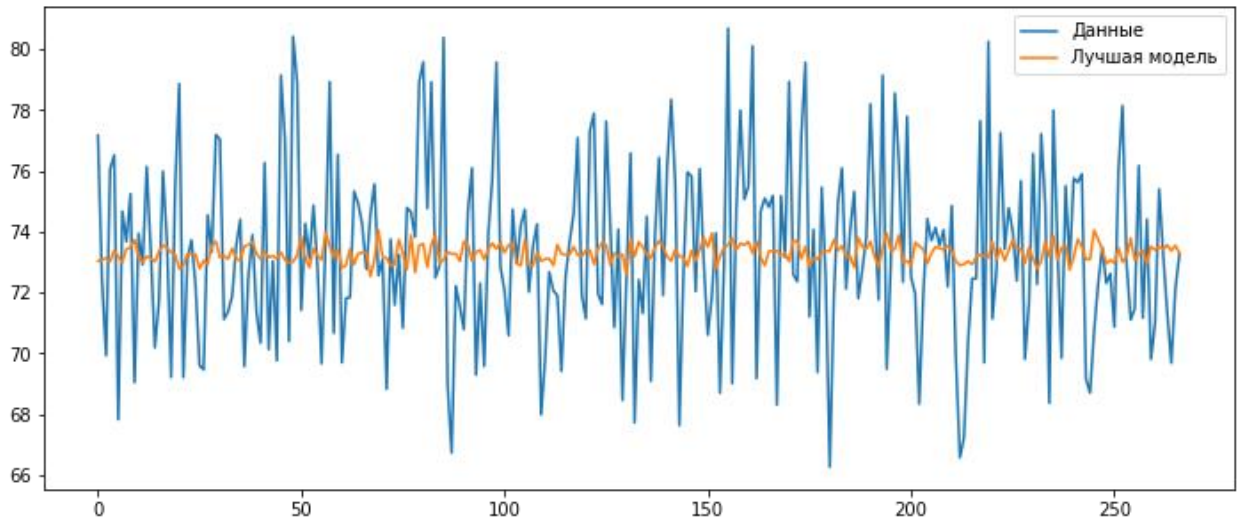


Рисунок 17 - Визуализация работы модели предсказания модуля упругости при растяжении

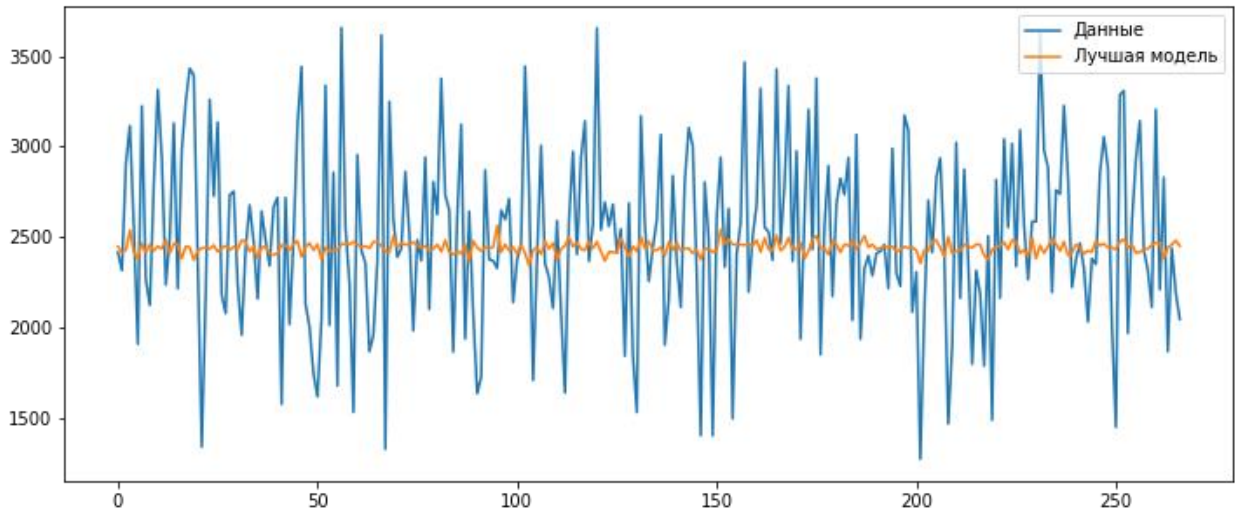


Рисунок 18 - Визуализация работы модели предсказания прочности при растяжении

2.6 Разработка нейронной сети для прогнозирования соотношения матрица-наполнитель

По заданию для соотношения матрица-наполнитель необходимо построить нейросеть. Нейронная сеть создавалась с помощью Sequential - модель в библиотеке Keras, позволяющая создать нейронную сеть прямого распространения путем последовательного добавления слоев.

Было протестировано несколько нейросетей с разными параметрами. Менялось количество скрытых слоев, количество нейронов в скрытом слое, активационная функция скрытых слоев. Также менялось количество эпох обучения. Среди всех нейросетей была выбрана сеть с наилучшими показателями ошибок (MSE и R2). На рисунке 19 представлена архитектура наилучшей нейросети.

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_24 (Dense)            (None, 50)                550

 dropout_19 (Dropout)        (None, 50)                0

 dense_25 (Dense)            (None, 128)               6528

 dropout_20 (Dropout)        (None, 128)               0

 dense_26 (Dense)            (None, 64)                8256

 dropout_21 (Dropout)        (None, 64)                0

 dense_27 (Dense)            (None, 1)                 65

=================================================================
Total params: 15,399
Trainable params: 15,399
Non-trainable params: 0
_________________________________________________________________
```

Рисунок 19 – Архитектура нейронной сети

Архитектура нейронной сети может быть описана следующим образом.

Модель состоит из двух скрытых уровней. Первый содержит 128 нейрона, второй – 64 нейрона. Снижение числа нейронов на каждом уровне сжимает информацию, которую сеть обработала на предыдущих уровнях. Во входном слое 10 признаков. Выходной слой с 1 нейроном (т.е. для одного признака), так как на выходе выводится одно значение для введенных данных. Активационная функция скрытых слоев – softplus. Для оптимизации, как наиболее распространенный и дающий лучшие результаты, был применен

метод Adam (adaptive moment estimation). Для оценки качества модели применена loss-функция: MeanSquaredError (MSE). Для борьбы с переобучением добавлены Dropout-слои с параметром 0.12. Такие слои выключат 12% случайных нейронов на каждом слое.

Обучение нейросети происходило со следующими параметрами: пропорция разбиения данных на тестовые и валидационные: 20%; количество эпох - 100.

По результатам обучения строим график, на котором две кривых – отображение среднеквадратической ошибки модели на тестовых (голубая линия) и валидационных данных (красная линия) относительно числа итераций. На рисунке 20 видим, что линии идут рядом, ошибка постепенно снижается и выходит на плато, где остается приблизительно на одном уровне до конца обучения.

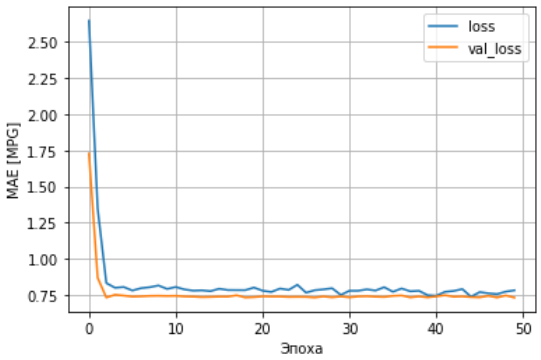


Рисунок 20 – Визуализация ошибки модели

Визуализация результатов работы нейросети отображена на рисунке 21.

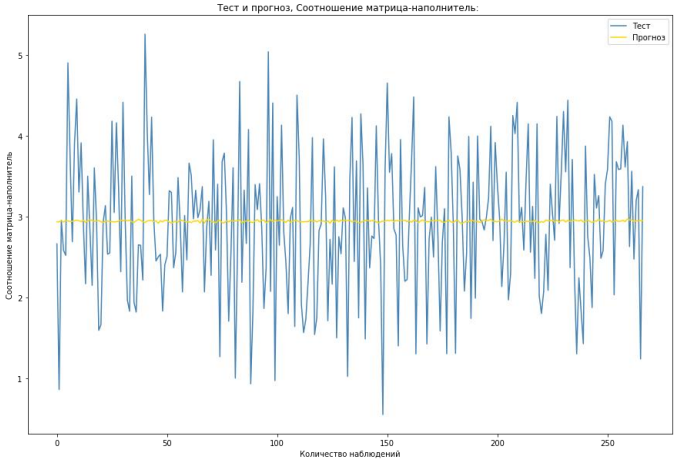


Рисунок 21 - Визуализация результата работы модели на тестовых данных

Ошибки модели, следующие: MSE= 0,783, R2 = -0,0007. Стоит отметить, что метрика R2 так же, как и во всех ранее построенных моделях и нейронных сетях показала отрицательный результат, что опять-таки может свидетельствовать об отсутствии взаимосвязей и необходимости более глубокой обработки исходных данных.

2.7 Разработка приложения

В процессе выполнения ВКР было разработано Flask приложение для прогноза модуля упругости при растяжении и прочности при растяжении. Интерфейс приложения показан на рисунке 22.

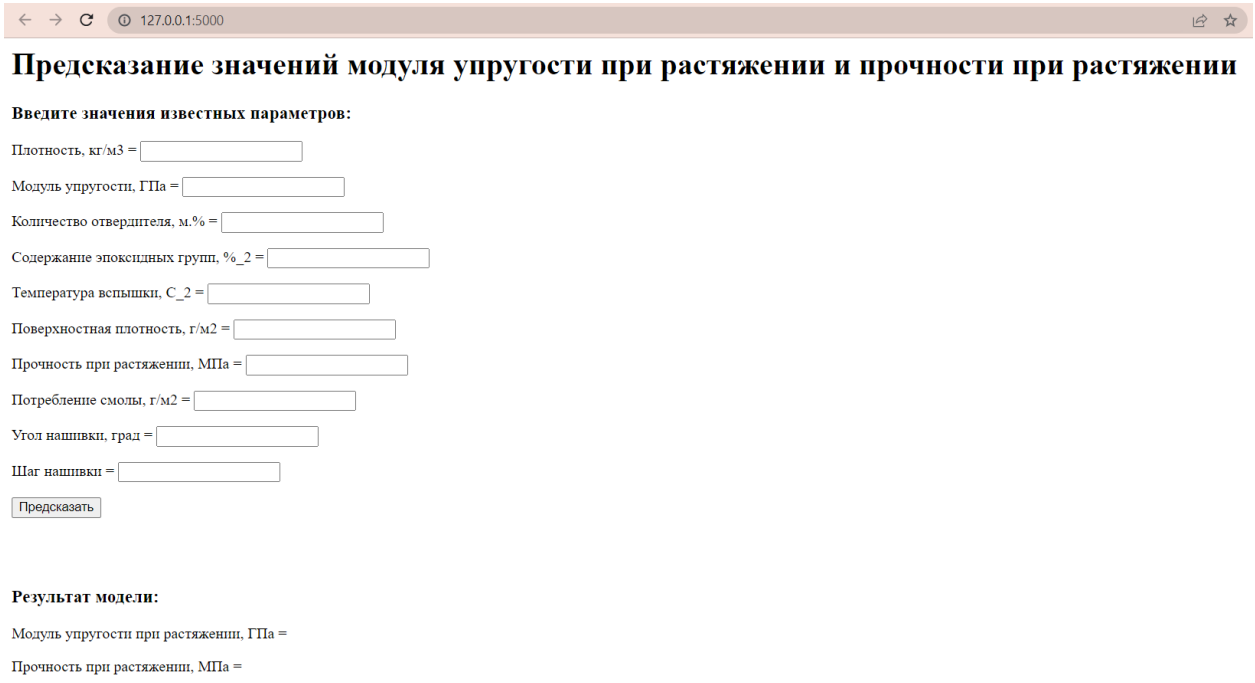


Рисунок 22 – Интерфейс веб-приложения

2.8 Создание репозитория

По итогам работы все материалы, включающие исследование в формате jupyter notebook, пояснительная записка, презентация, Flask приложение были размещены в репозитории на GitHub.

Заключение

В ходе решения задачи прогнозирования конечных свойств новых материалов были изучены основные теоретические и практические методы машинного обучения. Проведен предварительный анализ данных и их предобработка. Изучены основные алгоритмы машинного обучения и проведен сравнительный анализ полученных результатов. В моделях были настроены гиперпараметры.

После выполнения исследования разработано веб-приложение, данные загружены в репозиторий.

В ходе выполнения ВКР не удалось разработать модель, которая предсказывала бы значения с приемлемой точностью. Модель нейронной сети так же не показала приемлемого результата.

Библиографический список

1) Силен Дэви, Мейсман Арно, Али Мохамед. Основы Data Science и Big Data. Python и наука о данных. – СПб.: Питер, 2017. – 336 с.: ил.

2) ГрасД. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2021. - 416 с.: ил.

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

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

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

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

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

8) Документация по библиотеке sklearn: – Режим доступа: https://scikitlearn.org/stable/user_guide.html.

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

10) Руководство по быстрому старту в flask: – Режим доступа: https://flaskrussian-docs.readthedocs.io/ru/latest/quickstart.html.

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