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 Васягин Игорь Евгеньевич

Москва, 2023

**Оглавление**

Введение ……………………………………………….…………………………… 3

1. Аналитическая часть …………………………………………………………..... 4
   1. Постановка задачи …………………………………………….………….. 4
   2. Входящие данные ………………………………………………………… 5
   3. Описание используемых методов ……………………………………….. 6
      1. Линейная регрессия ……………………………………………..… 6
      2. Дамми ………………………………………………………………..7
      3. Регрессия по методу Лассо ………………………….……………. 7
      4. Гребневая регрессия …………………………………………..…... 8
      5. Эластичная регрессия …………………………………………...… 8
      6. Метод k-ближайших соседей ……………………………………... 9
      7. Деревья решений ………………………………………………..…. 9
      8. Деревья решений ……………………………………………...…...10
      9. Градиентный бустинг …………………………………………….. 11
      10. Метод опорных векторов ………………….…………….. 11
      11. Нейронная сеть ………………….………… …………….. 12
   4. Разведочный анализ данных ……………..……………….…………….. 13
2. Практическая часть ……………………………………………………………. 17
   1. Предобработка данных
   2. Метрики качества моделей ………………….………………….………. 20
   3. Разработка и обучение моделей

«Модуль упругости при растяжении, ГПа» и

«Прочность при растяжении, МПа» …………………………………… 20

* + 1. Обработка параметров ………………………………..……….…. 20
    2. Разработка и обучение моделей …………………………………. 21
    3. Тестирование моделей ………………………………………...…. 22
    4. Ошибки ……………………………………………………………. 25
  1. Нейросеть для рекомендации

«Соотношение матрица-наполнитель» ………………….……….……. 26

* + 1. Обработка параметров …………………………….……..…….… 26
    2. Разработка и обучение модели ………………….……….………. 26
    3. Тестирование моделей ……………………...…………..………... 28
    4. Ошибки …………………………………………..…….……….…. 29

Заключение …………………………………………………………….………….. 30

Список используемых в работе ресурсов ………………………….……………. 31

**Введение**

Композиционный материал или композитный материал - многокомпонентный материал, изготовленный (человеком или природой) из двух или более компонентов с существенно различными физическими и/или химическими свойствами, которые, в сочетании, приводят к появлению нового материала с характеристиками, отличными от характеристик отдельных компонентов и не являющимися простой их суперпозицией. В составе композита принято выделять матрицу/матрицы и наполнитель/наполнители, последние выполняют функцию армирования (по аналогии с арматурой в таком композиционном строительном материале, как железобетон). В качестве наполнителей композитов, как правило, выступают углеродные или стеклянные волокна, а роль матрицы играет полимер. Сочетание разных компонентов позволяет улучшить характеристики материала и делает его одновременно лёгким и прочным. При этом отдельные компоненты остаются таковыми в структуре композитов, что отличает их от смесей и затвердевших растворов. Варьируя состав матрицы и наполнителя, их соотношение, ориентацию наполнителя, получают широкий спектр материалов с требуемым набором свойств. Многие композиты превосходят традиционные материалы и сплавы по своим механическим свойствам и в то же время они легче. Использование композитов обычно позволяет уменьшить массу конструкции при сохранении или улучшении её механических характеристик.

1. **Аналитическая часть**
   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).

Дано: датасет из 2-х файлов X\_bp.xlsx X\_nup.xlsx со свойствами композитов.

Требуется:

1. Объединение делать по индексу тип объединения INNER.
2. Провести разведочный анализ предложенных данных.
3. Провести предобработку данных (удаление шумов, нормализация и т.д.).
4. Обучить нескольких моделей для прогноза:

А) Модуля упругости при растяжении.

Б) Прочности при растяжении.

При построении модели необходимо 30% данных оставить на тестирование модели, на остальных произвести обучение моделей. При построении моделей провести поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10.

1. Написать нейронную сеть, которая будет рекомендовать соотношение:

А) Матрица-наполнитель.

1. Разработать приложение с графическим интерфейсом или интерфейсом командной строки, которое будет выдавать прогноз, полученный в задании 4 или 5 (один или два прогноза, на выбор учащегося).
2. Оценить точность модели на тренировочном и тестовом датасете.
3. Создать репозиторий в GitHub / GitLab и разместить там код исследования.
   1. **Входящие данные**

Датасет состоит из 13 признаков, 1023 строки.

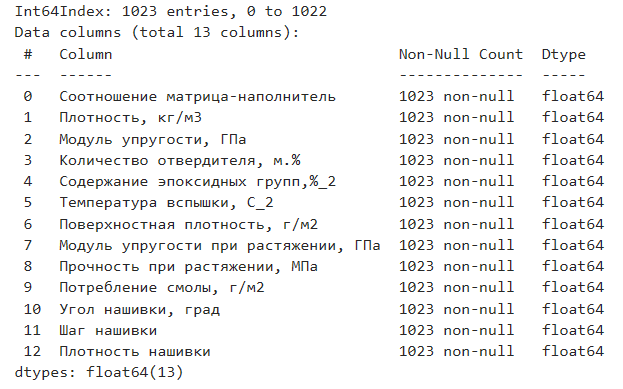


Рисунок 1 – Датасет

Исходя из поставленной задачи, распределим признаки на входные и выходные значения. Другими словами получаем модель для обучения с учителем.

|  |  |  |
| --- | --- | --- |
| Модель | Входные параметры | Выходные параметры, Прогноз |
| 1 | Соотношение матрица-наполнитель  Плотность, кг/м3  Модуль упругости, ГПа  Количество отвердителя, м.%  Содержание эпоксидных групп,%\_2  Температура вспышки, С\_2  Поверхностная плотность, г/м2  Потребление смолы, г/м2  Угол нашивки, град  Шаг нашивки  Плотность нашивки | Модуль упругости при растяжении, Гпа  Прочность при растяжении, МПа |
| 2 | 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 | Соотношение матрица-наполнитель |

Таблица 1 – Признаки модели

* 1. **Описание используемых методов**

Данная задача в рамках классификации категорий машинного обучения относится к машинному обучению с учителем и традиционно это задача регрессии.

Цель любого алгоритма обучения с учителем — определить функцию потерь и минимизировать её, соответственно, для получения наилучшего результата были применены следующие методы.

1. Linear regression - линейная регрессия
2. Dummy - Дамми
3. LASSO - Лассо
4. Ridge - гребневая регрессия
5. Elastic Net - Эластичная регрессия
6. K-Neighbors Regressor - к-ближайших соседей
7. Decision Tree Regressor - дерево решений
8. Random Forest - случайный лес
9. Gradient Boosting Regressor - градиентный бустинг
10. SVR - метод топорных векторов
11. Нейронная сеть
    * 1. **Линейная регрессия**

Линейная регрессия предполагает линейное отношение между свободной переменной и целевой переменной. При наличии нескольких входных переменных процедура называется множественной линейной регрессией. Алгоритм линейной регрессии используется, если метки непрерывны. Один из простейших способов выполнения вычислений основан на применении так называемого обычного метода наименьших квадратов.

Плюсы алгоритма: алгоритм прост в реализации и хорошо работает вне зависимости от размера набора данных. Не требует высоких вычислительных мощностей. За счет регуляризации можно снизить риск переобучения

Минусы алгоритма: возможны ситуации недообучения, когда модель машинного обучения не может правильно собрать данные. Обычно это происходит, когда функция гипотезы не может хорошо соответствовать данным. Поскольку линейная регрессия предполагает линейную связь между входными и выходными переменными, она не может должным образом соответствовать сложным наборам данных, из-за чего модель имеет низкую точность. Регрессия чувствительна к выбросам.

* + 1. **Дамми**

Фиктивная переменная качественная переменная, принимающая значения 0 и 1, включаемая в эконометрическую модель для учёта влияния качественных признаков и событий на объясняемую переменную. При этом фиктивные переменные позволяют учесть влияние не только качественных признаков, принимающих два значения, но и несколько возможных. В этом случае добавляются несколько фиктивных переменных. Фиктивная переменная может быть также индикатором принадлежности наблюдения к некоторой подвыборке. Последнее можно использовать для обнаружения структурных изменений.

* + 1. **Регрессия по методу Лассо**

Операция наименьшего абсолютного сокращения и выбора. Как следует из названия, LASSO использует метод «усадки», в котором определяются коэффициенты, которые уменьшаются по направлению к центральной точке в качестве среднего значения. Процедура лассо поощряет использование простых разреженных моделей (то есть моделей с меньшим количеством параметров). Этот конкретный тип регрессии хорошо подходит для моделей, демонстрирующих высокий уровень мультиколлинеарности, или когда нужно автоматизировать определенные части выбора модели, такие как выбор переменных/исключение параметров.

* + 1. Гребневая регрессия

Усовершенствование Линейной регрессии с повышенной устойчивостью к ошибкам, с ограничениями на коэффициенты регрессии для получения наиболее приближенного к реальности результата.

Плюсы алгоритма: результат гораздо проще интерпретировать.

Применяется метод для борьбы с переизбыточностью данных, когда независимые переменные коррелируют друг с другом.

* + 1. Эластичная регрессия

Включает в себя термины регуляризации как L-1, так и L-2. Это дает преимущества регрессии Лассо и Гребневой. Было установлено, что он обладает предсказательной способностью лучше, чем у Лассо, хотя все еще выполняет выбор функций. Поэтому получается лучшее из обоих методов, выполняя выбор функции Лассо с выбором группы объектов Ридж.

Лассо, Гребневая и Эластичная сеть - это модификации обычной линейной регрессии наименьших квадратов, которые используют дополнительные штрафы, чтобы сохранить значения коэффициента небольшими и упростить модель.

Лассо полезно для выбора функций, когда набор данных имеет функции с плохой предсказательной силой.

Регрессия гребня полезна для группового эффекта, при котором коллинеарные элементы могут быть выбраны вместе.

Эластичная регрессия сочетает в себе регрессию Лассо и Гребневую, что потенциально приводит к модели, которая является простой и прогнозирующей.

* + 1. **Метод k-ближайших соседей**

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

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

Плюсы: Алгоритм прост и легко реализуем, не чувствителен к выбросам, нет необходимости строить модель, настраивать несколько параметров или делать дополнительные допущения. Алгоритм универсален и его можно использовать для обоих типов задач: классификации и регрессии

Минусы: алгоритм работает значительно медленнее при увеличении объема выборки, предикторов или независимых переменных. Из аргумента выше следуют большие вычислительные затраты во время выполнения. Всегда нужно определять оптимальное значение k.

* + 1. **Деревья решений**

Метод, применяемый и для классификации и для регрессии. 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 производится 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.

Для задач регрессии применяется дисперсия вокруг среднего. Минимизируя дисперсию вокруг среднего, мы ищем признаки, разбивающие выборку таким образом, что значения целевого признака в каждом листе примерно равны.

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 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. Простота метода — одновременно преимущество и недостаток. Из-за этого применение дерева решений ограниченно. Алгоритм не подходит для решения задач с более сложными зависимостями.

* + 1. **Деревья решений**

Алгоритм создает несколько деревьев решений, которые объединяются для более точного прогноза.

Логика модели случайного леса заключается в том, что несколько отдельных деревьев решений работают гораздо лучше в группе, чем по отдельности. При использовании Random Forest для классификации каждое дерево дает классификацию или «голосование». Лес выбирает классификацию с большинством «голосов». При использовании Random Forest для регрессии лес выбирает среднее значение результатов всех деревьев.

Ключевым моментом здесь является тот факт, что корреляция между отдельными моделями, то есть между деревьями решений, составляющими более крупную модель случайного леса низка (или отсутствует). В то время как отдельные деревья решений могут давать ошибки, большая часть группы будет правильной, что приведет к общему результату в правильном направлении.

* + 1. **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.

Плюсы: алгоритм работает с любыми функциями потерь, предсказания в среднем лучше, чем у других алгоритмов, самостоятельно справляется с пропущенными данными.

Минусы: алгоритм крайне чувствителен к выбросам и при их наличии будет тратить огромное количество ресурсов на эти моменты. Однако стоит отметить, что использование  MAE вместо  MSE значительно снижает влияние выбросов на модель.

Модель будет склонна к переобучению при слишком большом количестве деревьев. Данная проблема присутствует в любом алгоритме, связанном с деревьями и исправляется правильной настройкой параметров.

* + 1. **Метод опорных векторов**

Метод опорных векторов - один из наиболее популярных методов машинного обучения. Он создает гиперплоскость или набор гиперплоскостей в многомерном пространстве, которые могут быть использованы для решения задач классификации и регрессии.

Основная идея метода заключается в построении гиперплоскости, разделяющей объекты выборки оптимальным способом. Интуитивно, хорошее разделение достигается за счет гиперплоскости, которая имеет самое большое расстояние до ближайшей точки обучающей выборки любого класса. Максимально близкие объекты разных классов определяют опорные вектора. Если в исходном пространстве объекты линейно неразделимы, то выполняется переход в пространство большей размерности.

Эффективность метода опорных векторов зависит от выбора ядра, параметров ядра и параметра для регуляризации.

Плюсы: хорошо работает с пространством признаков большого размера, хорошо работает с данными небольшого объема.

Минусы: неустойчивость к шуму, выбросы в обучающих данных становятся опорными объектами-нарушителями и напрямую влияют на построение разделяющей гиперплоскости.

* + 1. **Нейронная сеть**

Представляет собой вычислительные системы, основанные на биологических нейронных сетях, которые составляют мозг животных.

Нейронная сеть основана на наборе соединенных блоков или узлов, называемых искусственными нейронами, которые в общих чертах моделируют нейроны биологического мозга. Каждое соединение, подобно синапсам в биологическом мозге, может передавать сигнал другим нейронам.

Искусственный нейрон получает сигналы, затем обрабатывает их и может передавать сигналы нейронам, подключенным к нему. "Сигнал" при соединении представляет собой действительное число, а выходные данные каждого нейрона вычисляются с помощью некоторой нелинейной функции от суммы его входных данных. Соединения называются ребрами. Нейроны и ребра обычно имеют вес, который корректируется по мере обучения. Вес увеличивает или уменьшает силу сигнала при соединении. Нейроны могут иметь пороговое значение, такое, что сигнал отправляется, только если совокупный сигнал пересекает это пороговое значение.

Как правило, нейроны объединены в слои. Разные слои могут выполнять разные преобразования на своих входных данных. Сигналы передаются от первого слоя (входного слоя) к последнему слою (выходному слою), возможно, после многократного обхода слоев.

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. **Разведочный анализ данных**

Для дальнейшей передачи Датасета в процесс машинного обучения, необходимо провести анализ на выбросы, содержание различных ошибок и пропущенных значений.

|  |  |
| --- | --- |
| Уникальные значения | Содержание нулевых значений |

Рисунок 2 – Значения признаков

Нулевые значения в данных отсутствуют.

Из 13 признаков один является категориальным – «Угол нашивки, град».

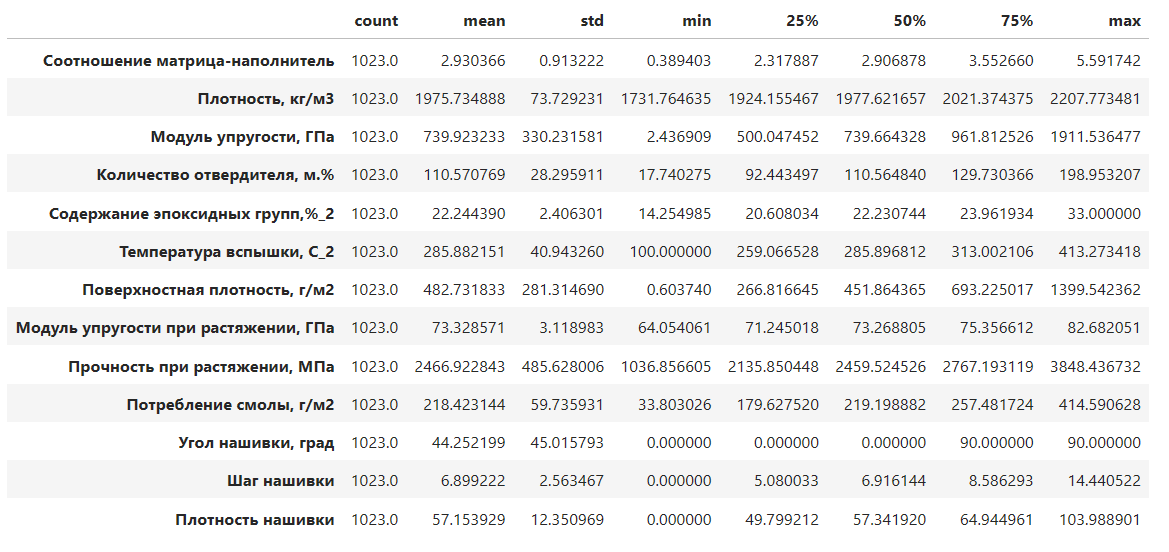


Рисунок 3 – Описание признаков

В качестве инструментов разведочного анализа используется: оценка статистических характеристик датасета; гистограммы распределения каждой из переменной; диаграммы boxplot (ящика с усами); попарные графики рассеивания точек; тепловая карта; описательная статистика для каждой переменной; анализ и полное исключение выбросов.



Рисунок 4 – Попарный график рассеивания данных

Графики попарного рассеивания позволяют нам увидеть возможную взаимосвязь между любыми двумя переменными соответствующего набора данных.

На попарных графиках распределения не видно корреляции между признаками. Единственная зависимость, которую можно отметить – это меньшая дисперсия значений Плотности нашивки при 90 градусов Угла нашивки, по сравнению со значением 0 градусов.

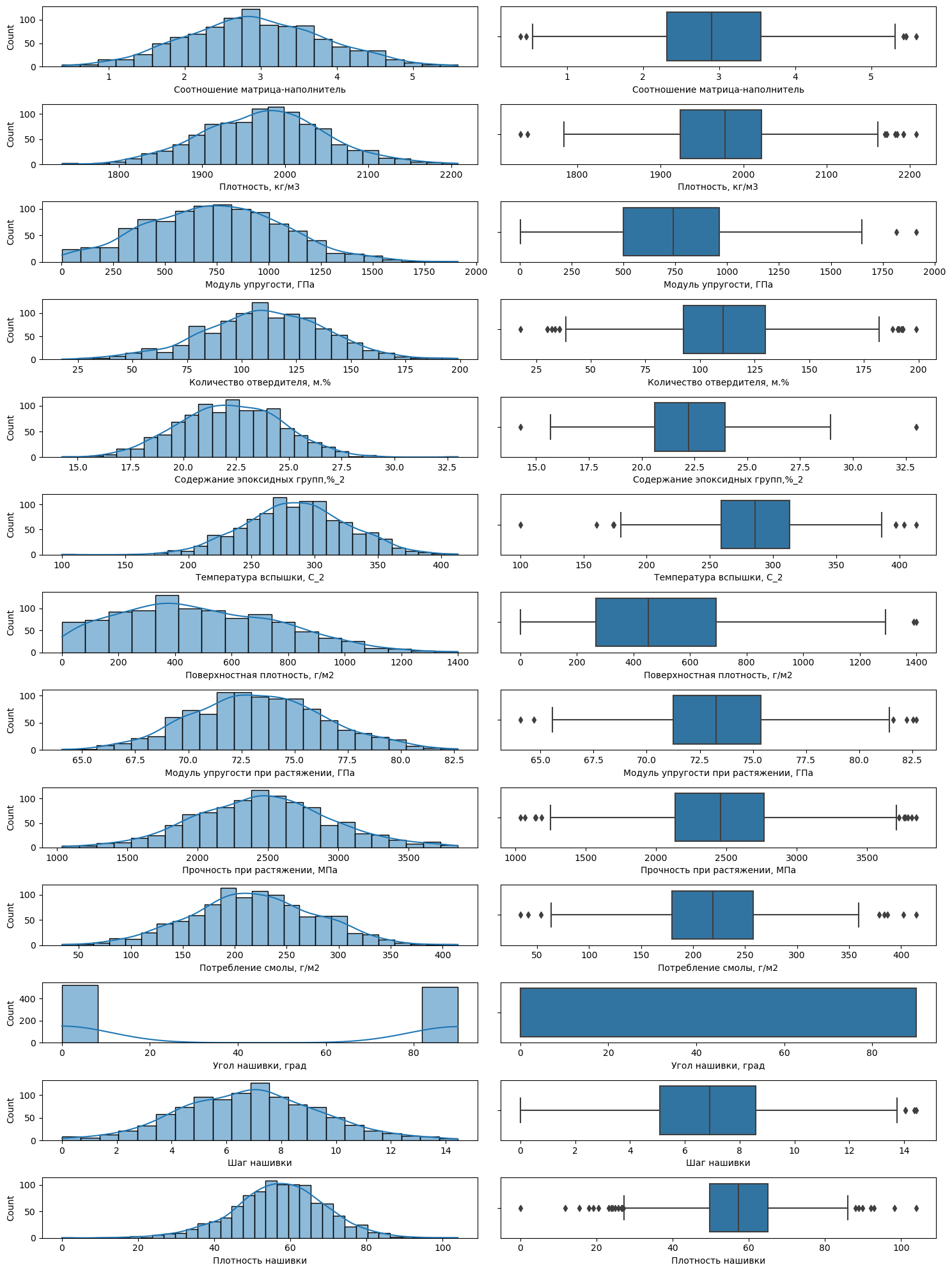


Рисунок 5 – График распределения, «Ящик с усами»

Box-plot показывает наличие выбросов в данных. График слева – нормальное распределение по Гауссу.

Вынесем «Ящик с усами» и кривые на отдельные графики.

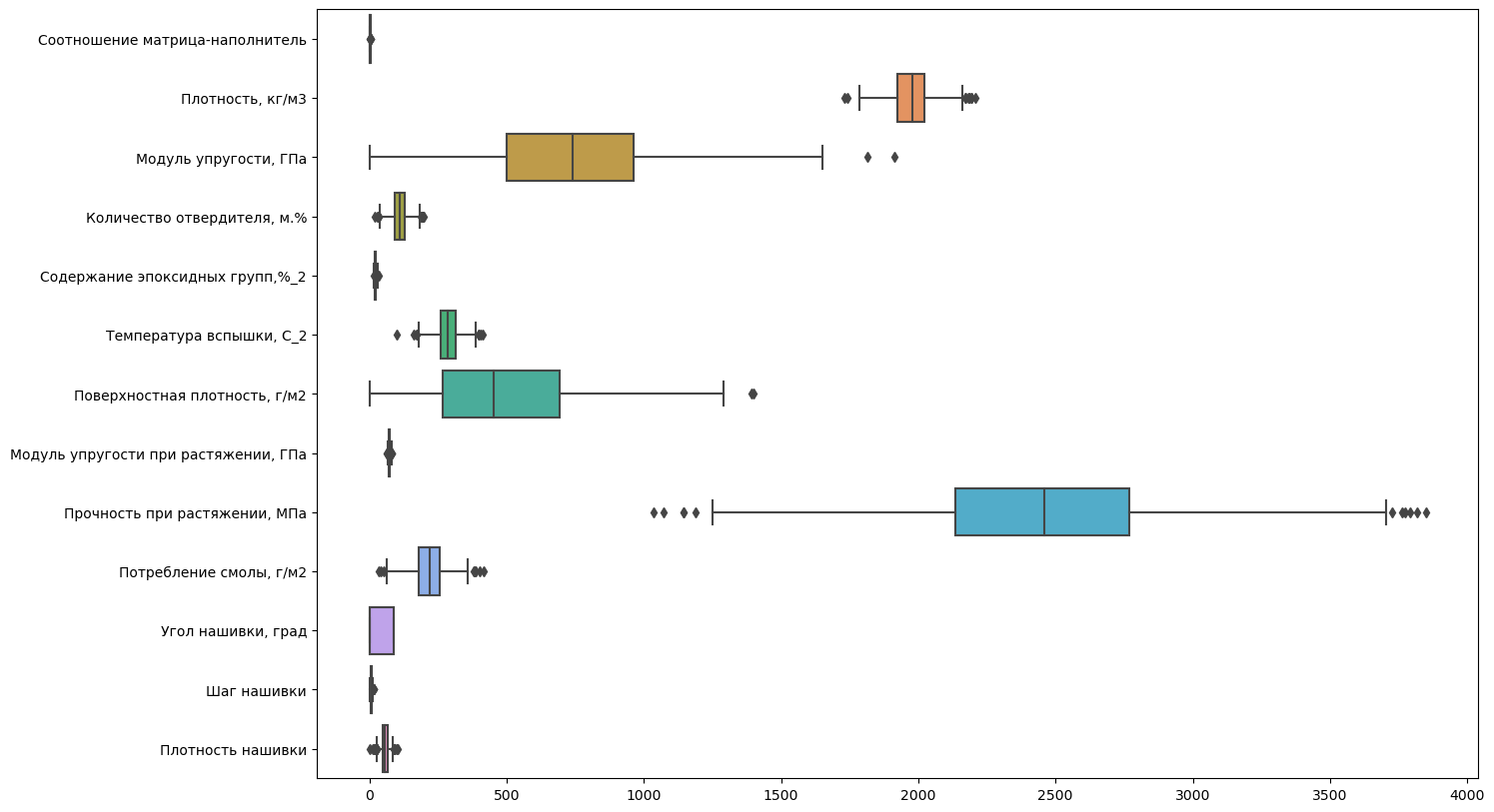


Рисунок 6 – «Ящик с усами»

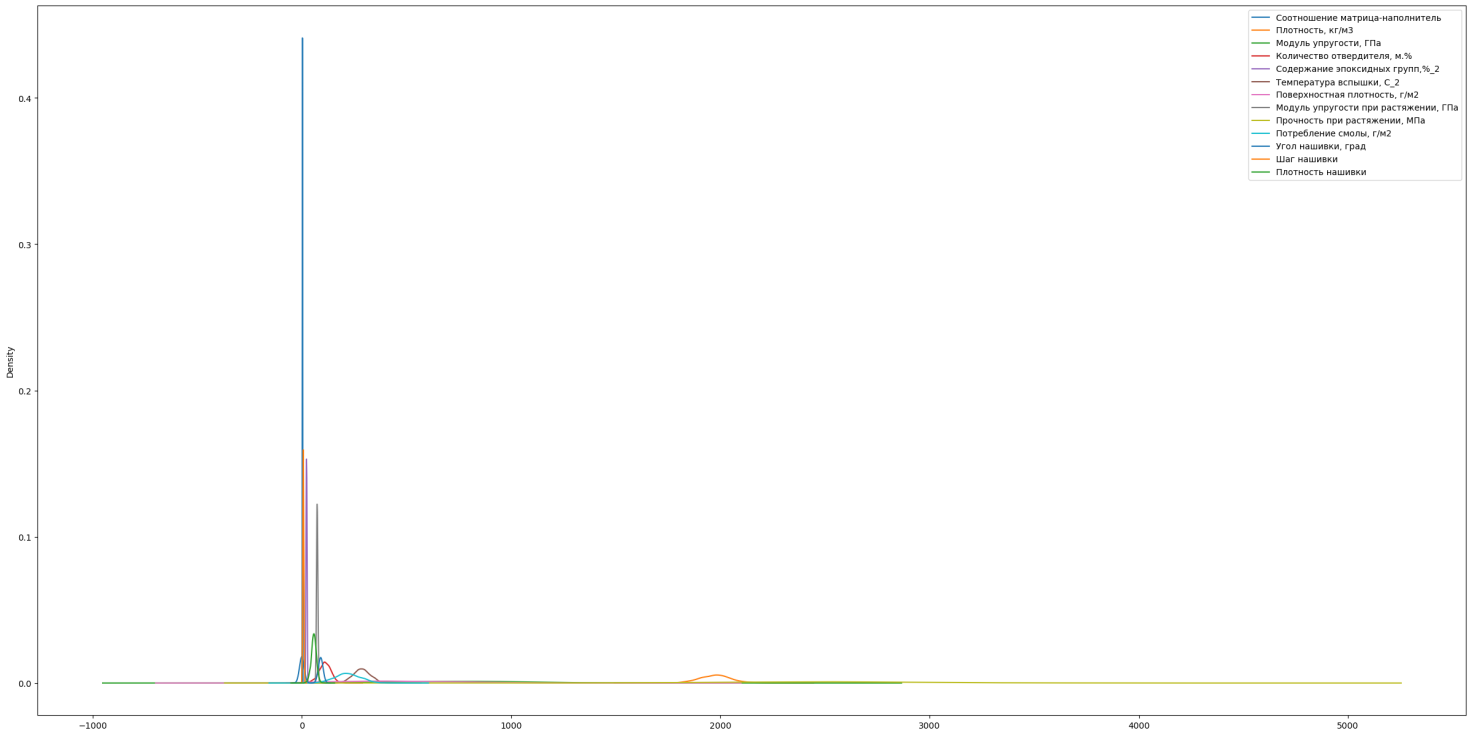


Рисунок 7 – Нормальное распределение

На графиках четко видны выбросы и значительный разброс значений по признакам.

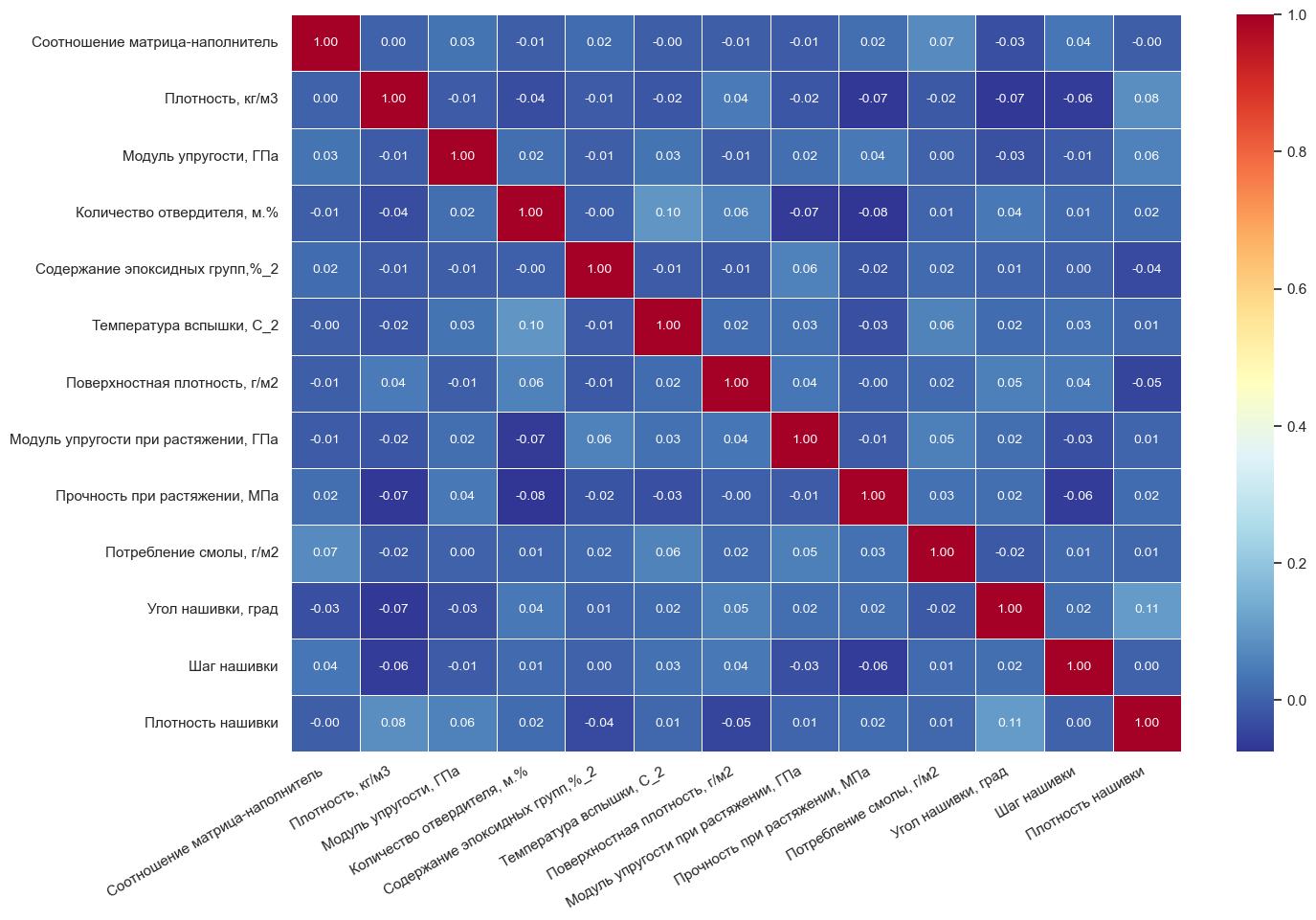


Рисунок 8 – Тепловая карта

Тепловая карта показывает отсутствие корреляции между признаками и целевыми переменными. Значения близки к нулю, связи между переменными не наблюдаются.

1. **Практическая часть**
   1. **Предобработка данных**

В качестве подготовки к дальнейшей обработке данных используем метод межквартильного размаха IQR для очистки от выбросов, а также стандартизируем данные методом StandartScaler.



Рисунок 9 – Описание датасета после стандартизации

Используя стандартизацию, мы центрируем столбцы признаков по среднему значению 0, чтобы столбцы признаков принимали форму нормального распределения, что упрощает изучение весов. Кроме того, стандартизация сохраняет полезную информацию о выбросах и делает алгоритм менее чувствительным к ним.

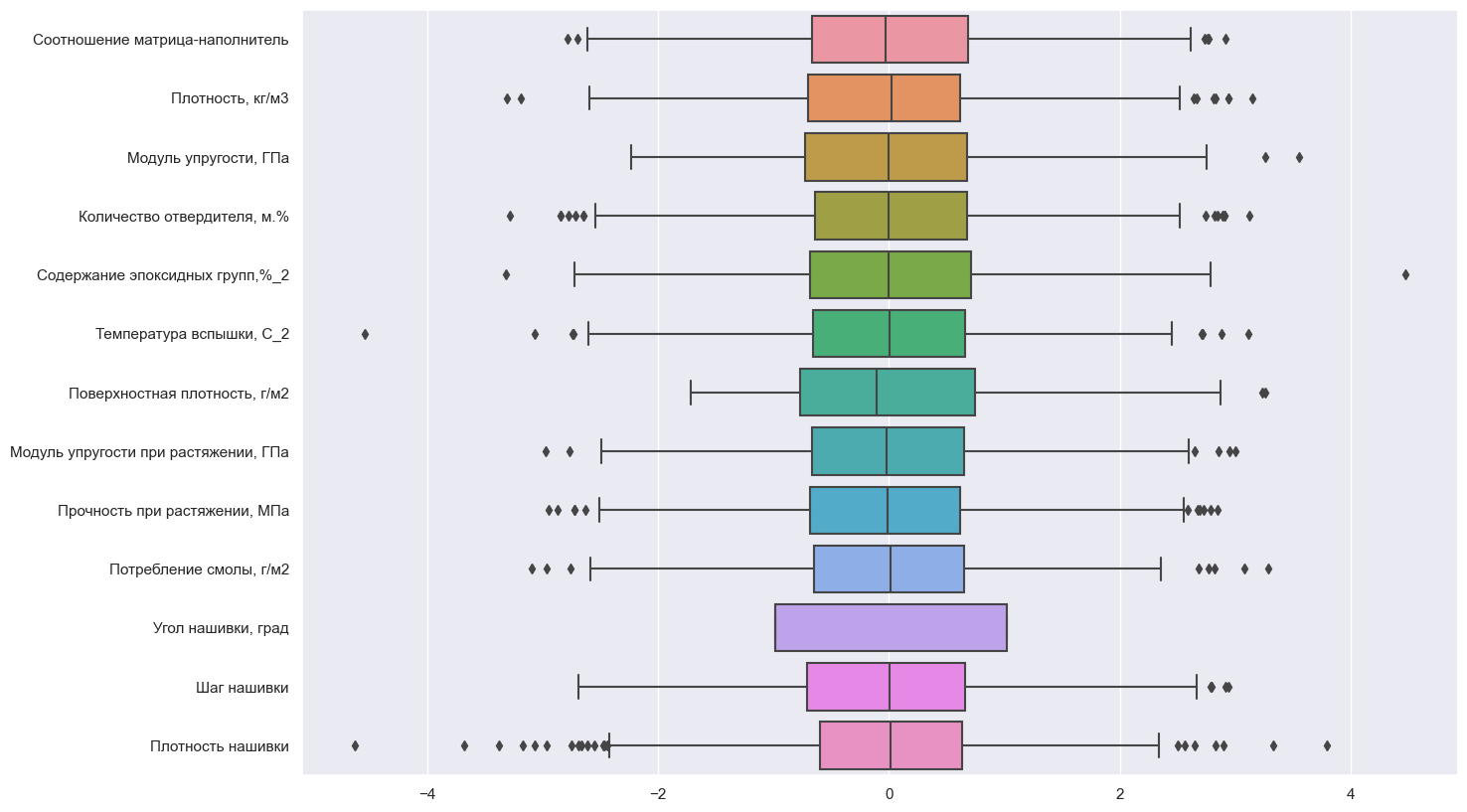


Рисунок 10 - «Ящик с усами» после стандартизации, до удаления выбросов

Далее удаляем выбросы методом межквартильного размаха.

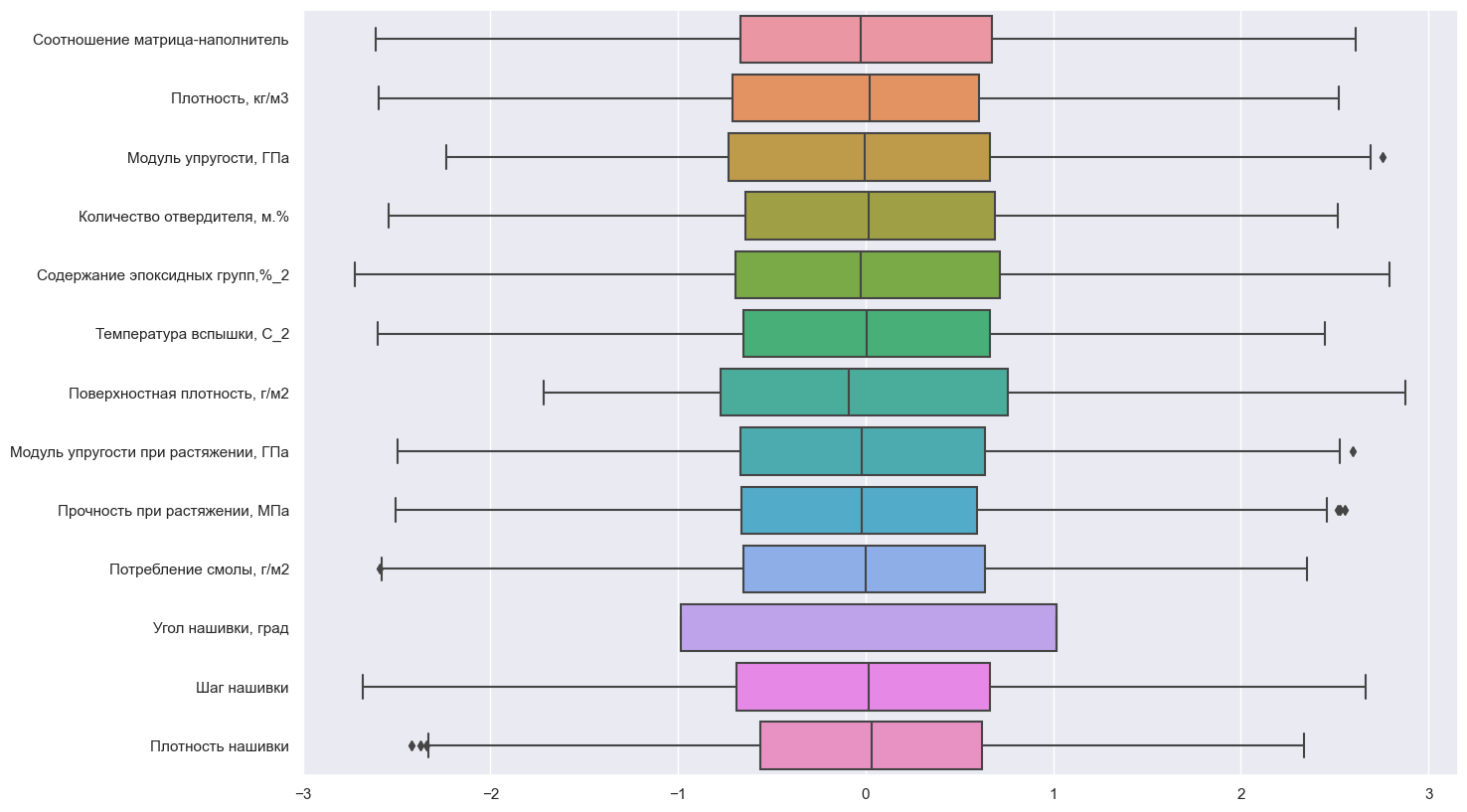


Рисунок 11 - «Ящик с усами» после стандартизации и удаления выбросов

Результат очистки от выбросов виден на графике.

Нормализуем данные методом Normalizer().

Смотрим обновленный очищенный Датасет:



Рисунок 12 – Очищенный и нормализованный датасет

На этом подготовка Датасета к дальнейшему обучению закончена.

Сделана копия Датасета для последующей обработки нейросетью.

* 1. Метрики качества моделей

Список и описание метрик качества моделей регрессии:

R2: коэффициент детерминации. Показывает, насколько условная дисперсия модели отличается от дисперсии реальных значений Y. Если этот коэффициент близок к 1, то условная дисперсия модели достаточно мала и весьма вероятно, что модель неплохо описывает данные. Если же коэффициент R-квадрат сильно меньше, например, меньше 0.5, то, с большой долей уверенности модель не отражает реальное положение вещей.

MSE: метрика, которая сообщает нам среднеквадратичную разницу между прогнозируемыми значениями и фактическими значениями в наборе данных. Чем ниже MSE, тем лучше модель соответствует набору данных.

RMSE (Root Mean Squared Error): метрика, которая сообщает нам квадратный корень из средней квадратичной разницы между прогнозируемыми значениями и фактическими значениями в наборе данных. Чем ниже RMSE, тем лучше модель соответствует набору данных.

MAE (Mean Absolute Error): средняя абсолютная ошибка. Рассчитывается как среднее абсолютных разностей между наблюдаемым и предсказанным значениями.

MAPE (Mean Absolute Percentage Error): средняя абсолютная процентная ошибка - безразмерный показатель, представляющий собой взвешенную версию.

* 1. Разработка и обучение моделей «Модуль упругости при растяжении, ГПа» и «Прочность при растяжении, МПа»
     1. **Обработка параметров**

Согласно заданию делим Датасет на тестовые и тренировочные значения методом train\_test\_split в соответствии 70:30.

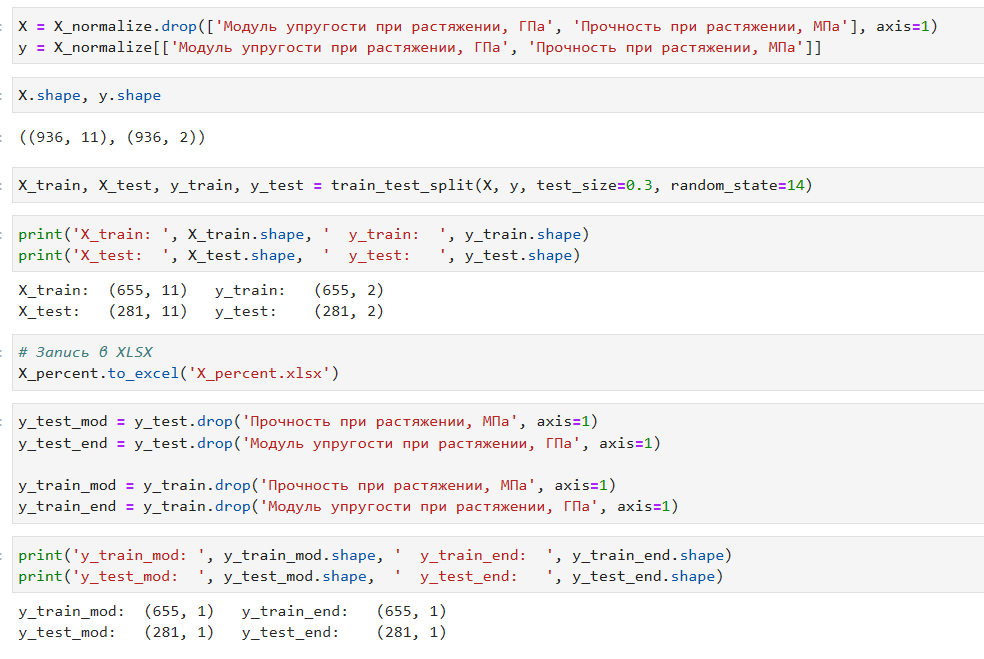


Рисунок 13 – Деление датасета на тренировочные и тестовые выборки

* + 1. **Разработка и обучение моделей**

В качестве поиска лучших параметров для используемых регрессоров, применил метод GridSearchCV(). Для старта перечислил все основные гиперпараметры.



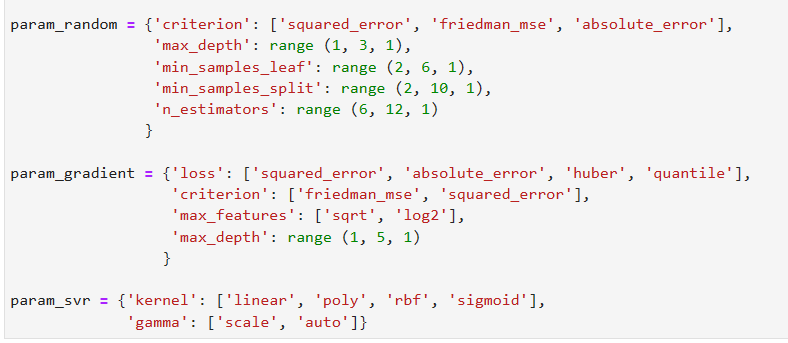


Рисунок 14 – Настройки регрессоров

Итогом запуска цикла по поиску гиперпараметров стало составление списка для двух признаков.

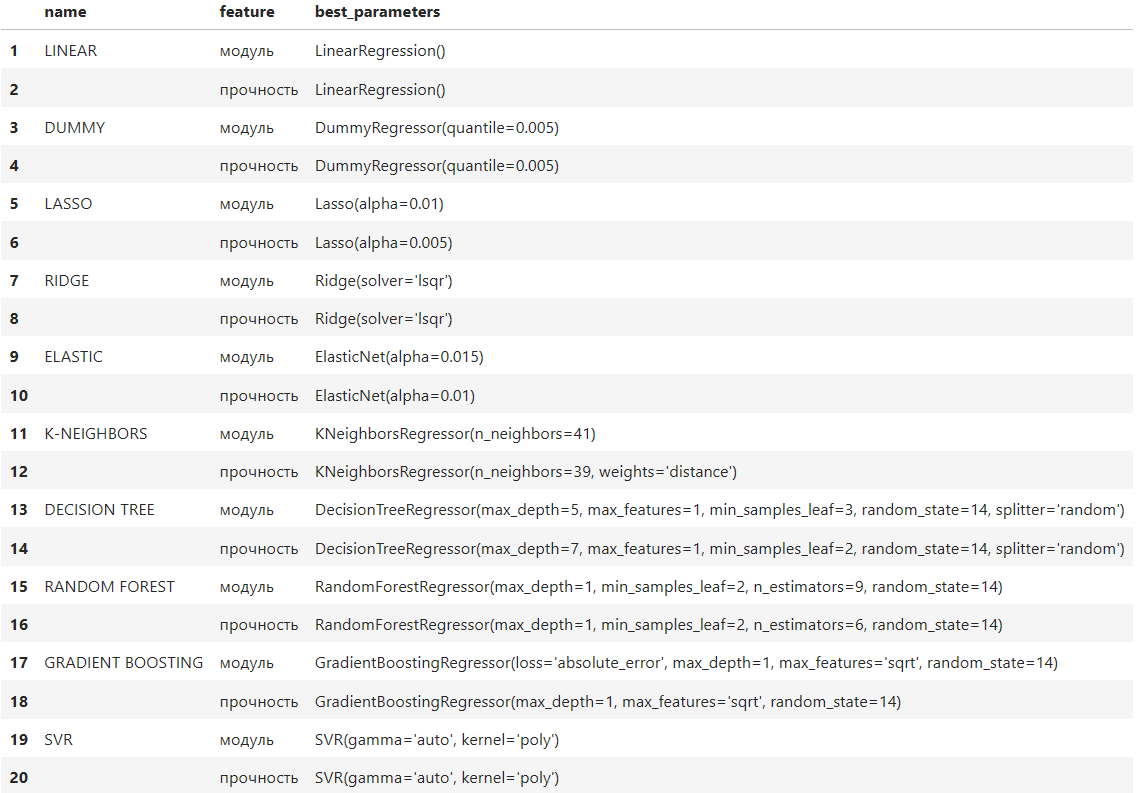
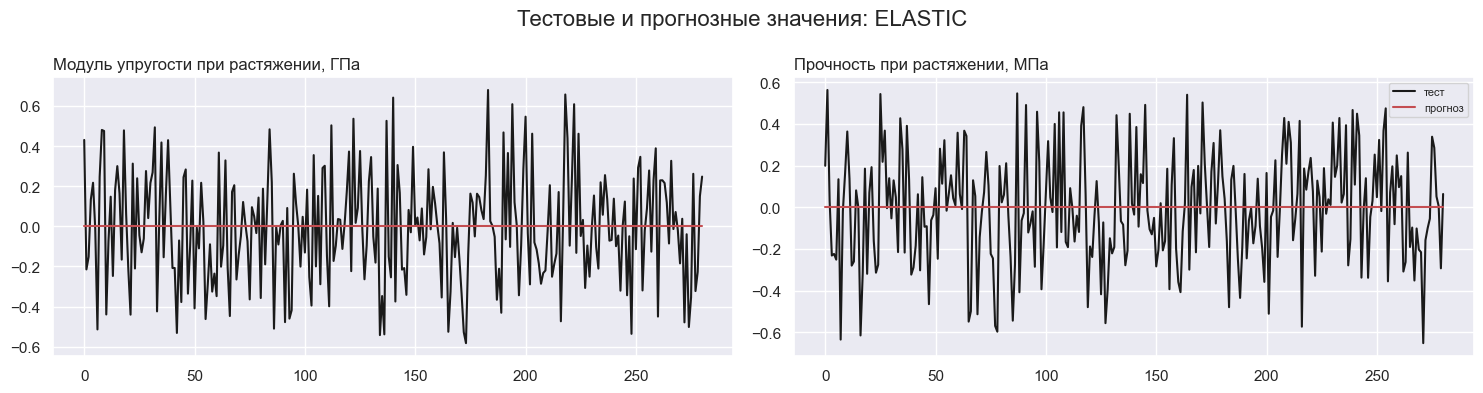
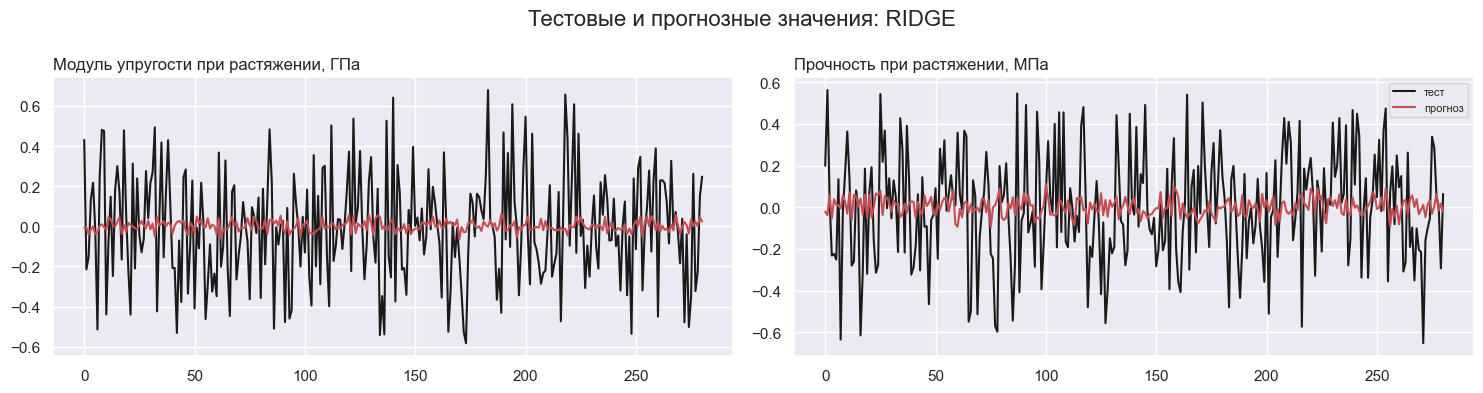
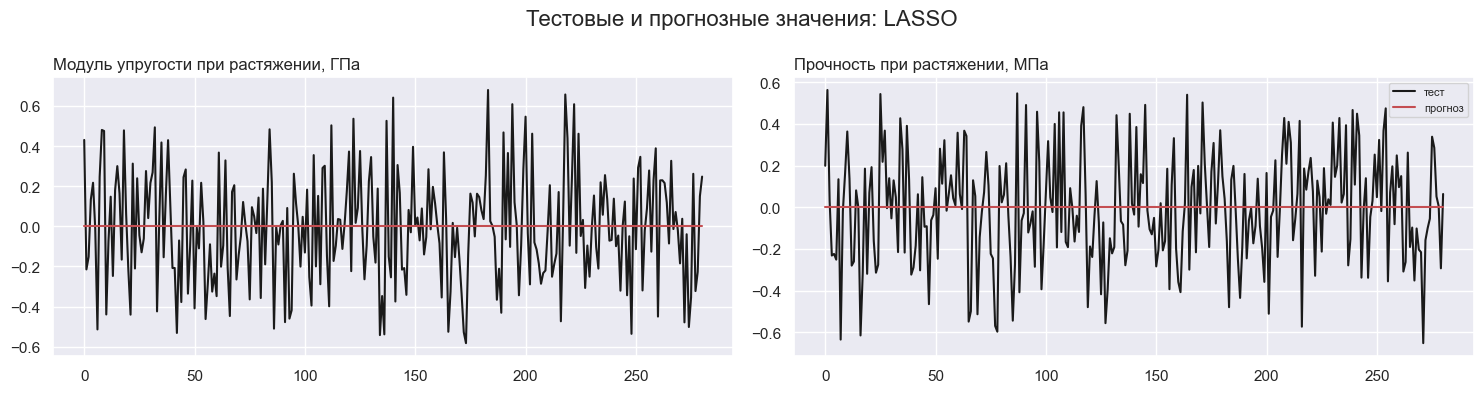
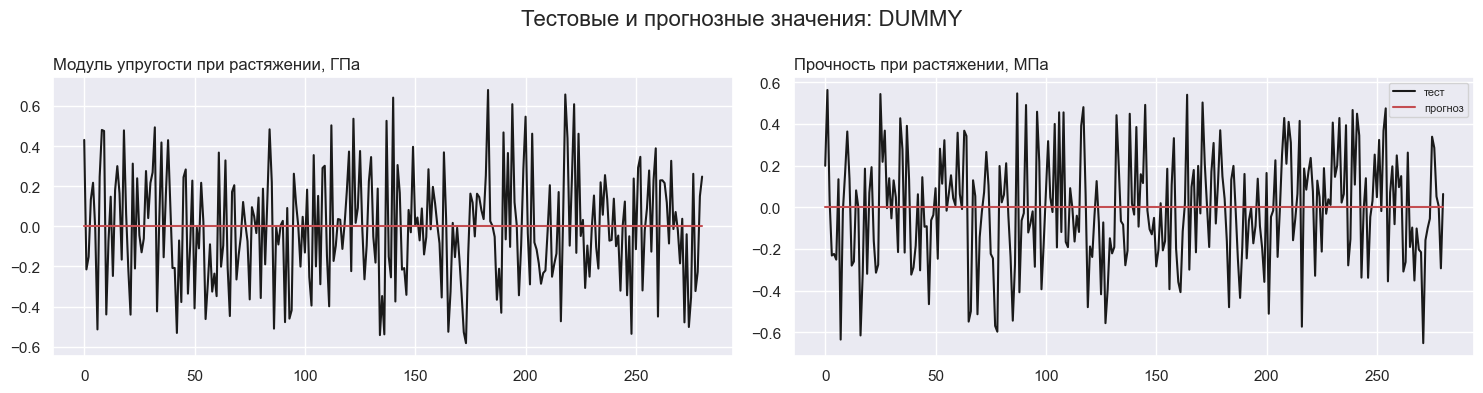
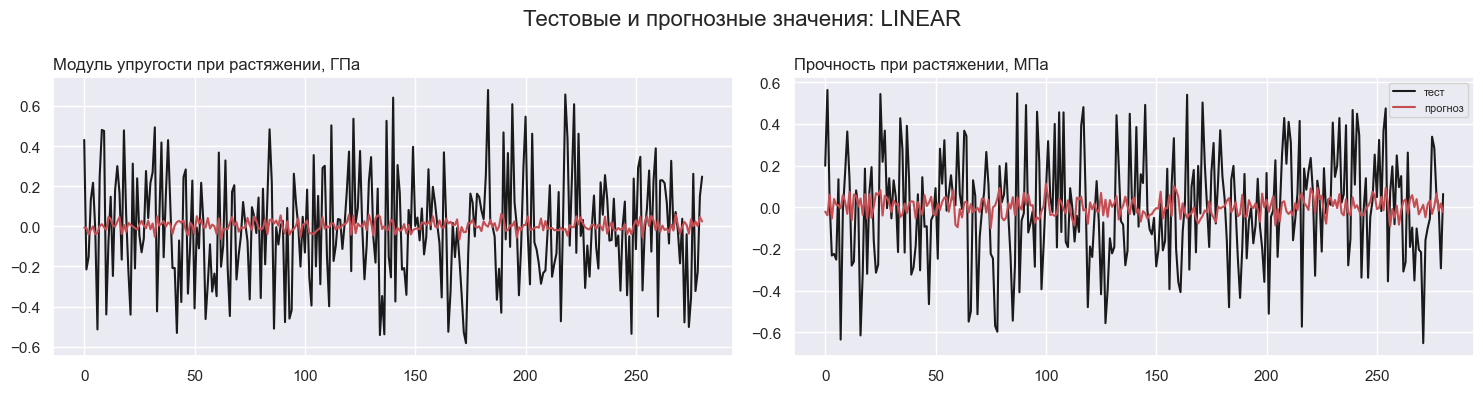


Рисунок 15 – Список лучших параметров для регрессоров

* + 1. **Тестирование моделей**

Тестируем обе модели по лучшим параметрам. Выводим на печать.







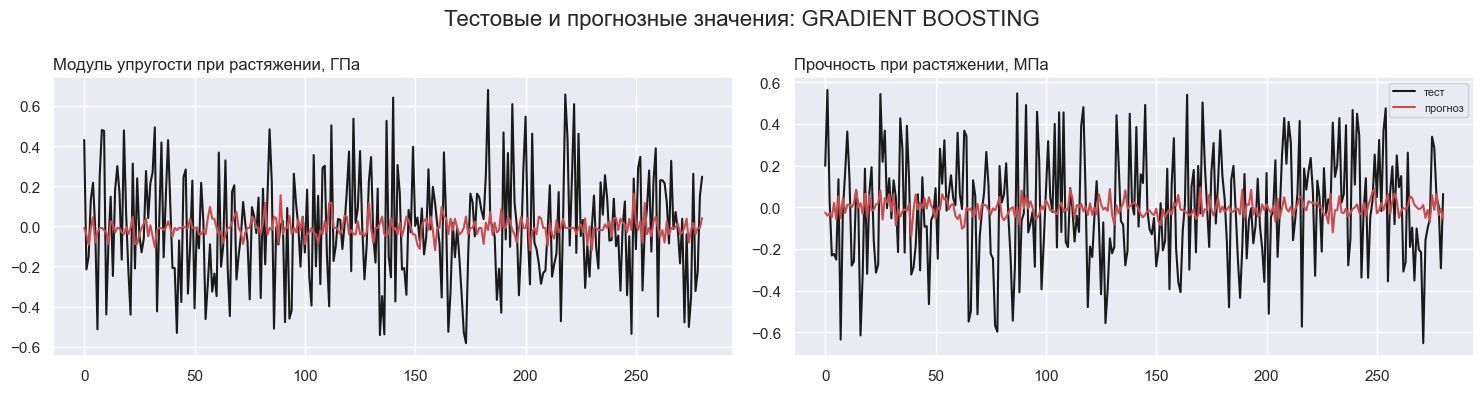




Рисунок 16 – Графики соответствия прогноза тестовой выборки

* + 1. **Ошибки**

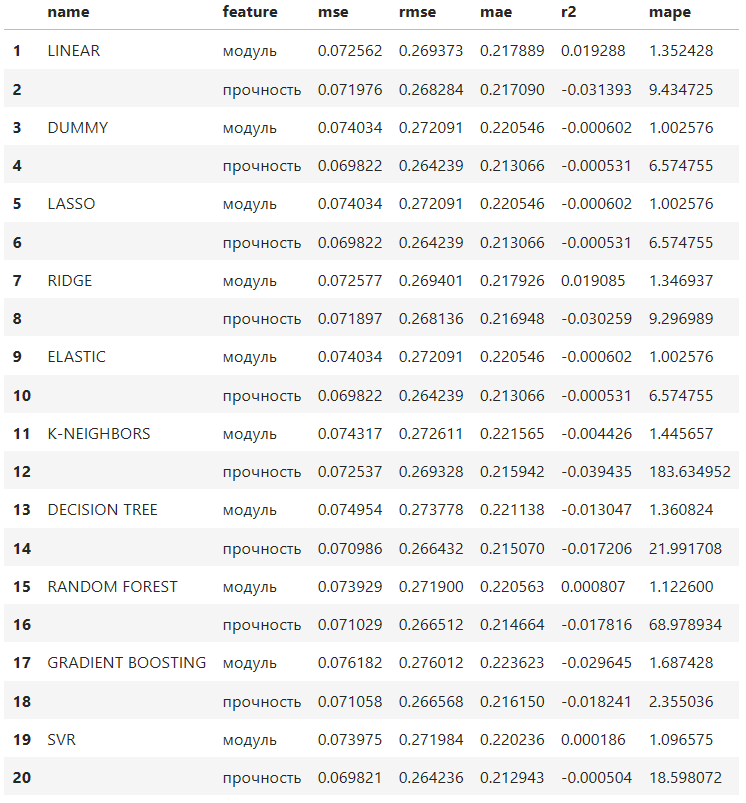


Рисунок 17 – Ошибки моделей

Визуально ни одна модель не отработала хорошо. Список полученных ошибок говорит о том же.

* 1. Нейросеть для рекомендации «Соотношение матрица-наполнитель»
     1. **Обработка параметров**

За основу взят подготовленный заранее подготовленный Датасет (п. 2.1.1).

Делим Датасет на тестовые и тренировочные значения методом train\_test\_split в соответствии 70:30.

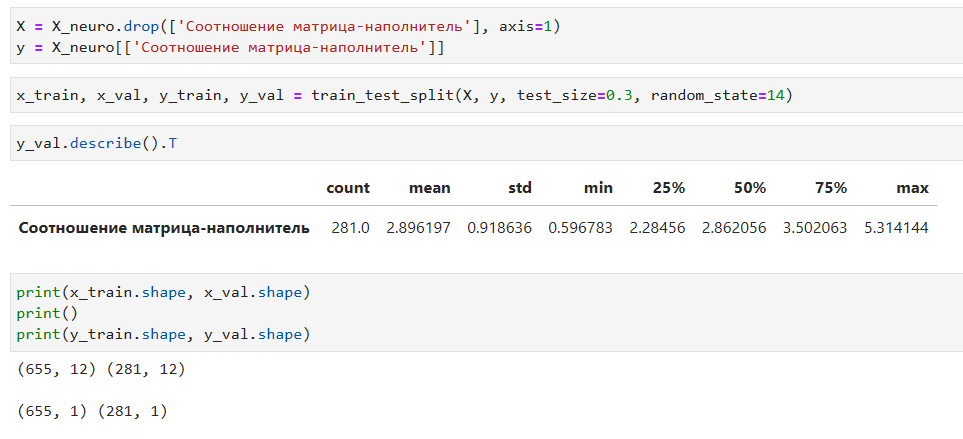


Рисунок 18 – Деление датасета на тестовую и валидационную выборки

* + 1. **Разработка и обучение модели**

Для поиска лучших параметров настройки нейросети был взят инструмент Tuner из библиотеки Keras Tensor Flow.

Настраиваем Tuner на поиск параметров. В том числе, по оптимальному количеству слоев модели.



Рисунок 19 – Настройка Тюнера

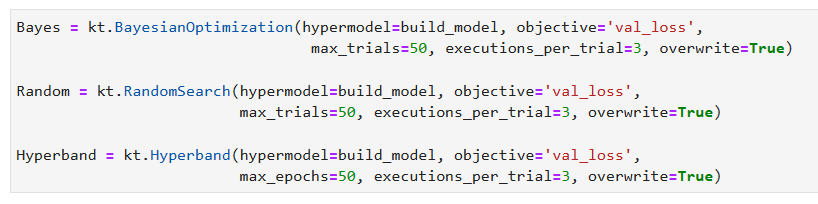
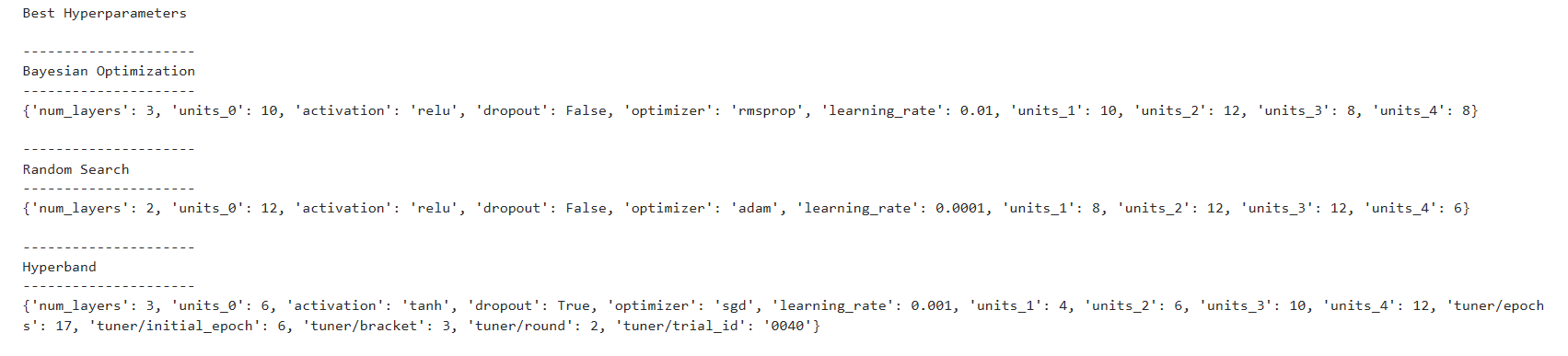
После определения области поиска подготавливаем классы тюнера для запуска. Выбираем BayesianOptimization, RandomSearch и Hyperband. 

Рисунок 20 – Используемые классы тюнеров

Запускаем Tuner и получаем лучшие параметры.



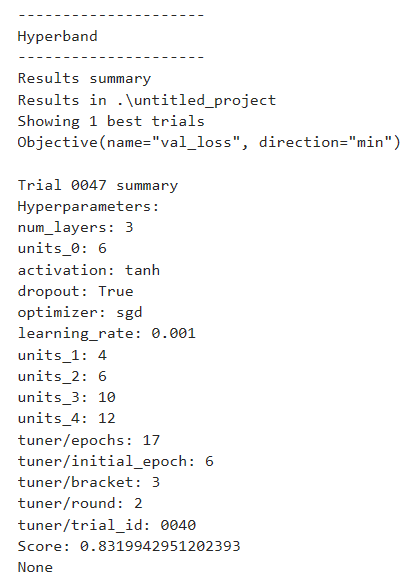
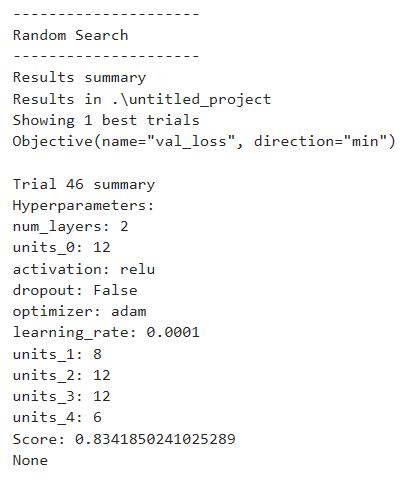
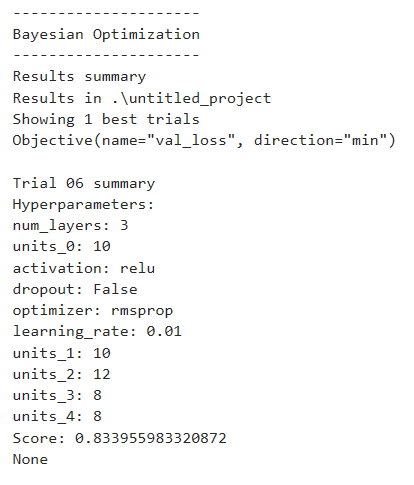


Рисунок 21 –Лучшие параметры для моделей

* + 1. **Тестирование моделей**

Тестируем модели по лучшим параметрам. Выводим на печать.

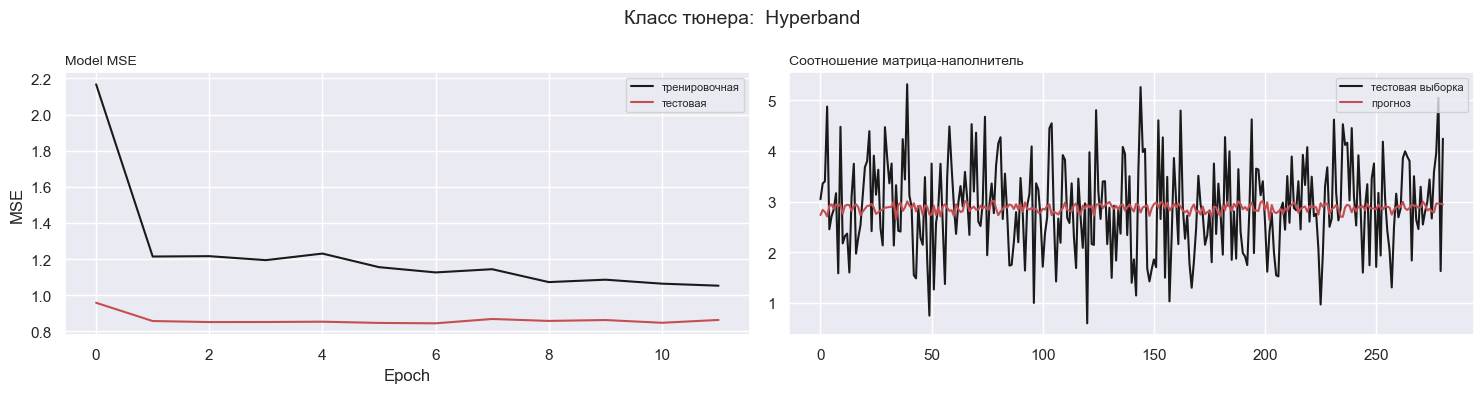
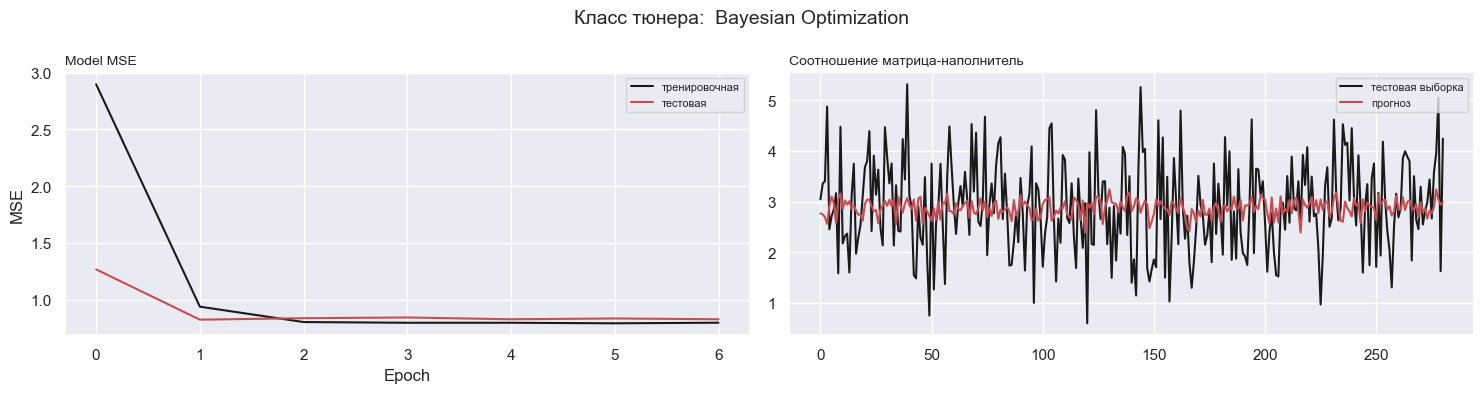


Рисунок 22 – График ошибок по эпохам и предсказание на тестовых данных

* + 1. **Ошибки**

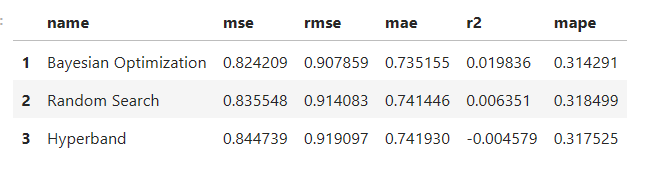


Рисунок 23 – Ошибки моделей нейросети

Как и в случае с регрессиями, визуально, ни одна модель нейросети не отработала хорошо. Этот вывод подтверждает список полученных ошибок.

**Заключение**

В ходе выполнения выпускной работы были выполнены все задачи предложенного пайплайна.

Проведенная исследовательская работа позволяет сделать некоторые основные выводы по теме:

1. Распределение данных соответствует нормальному.
2. Корреляция между парами признаков стремится к нулю.
3. Использованные при разработке моделей методы не позволили получить достоверные прогнозы.
4. Применённые модели регрессии и нейросеть не показали эффективности в прогнозировании свойств композитов.

**Список используемых в работе ресурсов**

1. Документация по библиотеке Keras:

<https://keras.io/api/>

1. Документация по библиотеке Matplotlib: <https://matplotlib.org/stable/plot_types/index>
2. Документация по библиотеке Numpy: <https://numpy.org/doc/1.22/user/index.html#user>
3. Документация по библиотеке Pandas: <https://pandas.pydata.org/docs/user_guide/index.html#user-guide>
4. Документация по библиотеке Scikit-learn:

<https://scikit-learn.org/stable/user_guide.html>

1. Документация по библиотеке Seaborn: <https://seaborn.pydata.org/tutorial.html>
2. Документация по библиотеке Tensorflow:

<https://www.tensorflow.org/?hl=ru>

1. Документация по языку программирования Python: <https://docs.python.org/3.9/>