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 Горбунова Анастасия Дмитриевна

Оглавление

[**Введение** 3](#_Toc122743515)

[1. Аналитическая часть 4](#_Toc122743516)

[1.1. Постановка задачи 4](#_Toc122743517)

[1.2. Описание используемых методов 11](#_Toc122743518)

[1.2.1. Простая линейная регрессия 15](#_Toc122743519)

[1.2.2. Регрессия по методу Лассо 16](#_Toc122743520)

[1.2.3. Гребневая регрессия 16](#_Toc122743521)

[1.2.4. Метод опорных векторов 17](#_Toc122743522)

[1.2.5. Метод k-ближайших соседей 18](#_Toc122743523)

[1.2.6. Деревья решений 19](#_Toc122743524)

[1.2.7. Случайный лес 20](#_Toc122743525)

[1.2.8. Градиентный бустинг 21](#_Toc122743526)

[1.2.9. Нейронная сеть 22](#_Toc122743527)

[1.3. Разведочный анализ данных 24](#_Toc122743528)

[2. Практическая часть 30](#_Toc122743529)

[2.1. Предобработка данных 30](#_Toc122743530)

[2.1.1. Обработка параметров для задачи прогнозирования модуля упругости при растяжении 31](#_Toc122743531)

[2.1.2. Обработка параметров для задачи прогнозирования прочности при растяжении 33](#_Toc122743532)

[2.1.3. Обработка параметров для задачи прогнозирования соотношения матрица-наполнитель 35](#_Toc122743533)

[2.2. Разработка и обучение модели 38](#_Toc122743534)

[2.2.1. Метрики качества моделей 38](#_Toc122743535)

[2.2.2. Разработка и обучение моделей для прогнозирования модуля упругости при растяжении 38](#_Toc122743536)

[2.2.3. Разработка и обучение моделей для прогнозирования прочности при растяжении 42](#_Toc122743537)

[2.3. Тестирование моделей 44](#_Toc122743538)

[2.3.1. Тестирование лучшей модели для прогноза модуля упругости при растяжении 44](#_Toc122743539)

[2.3.2. Тестирование лучшей модели для прогноза прочности при растяжении 46](#_Toc122743540)

[2.4. Нейросеть для рекомендации соотношения матрица-наполнитель 47](#_Toc122743541)

[2.5. Разработка приложения 55](#_Toc122743542)

[2.6. Создание удаленного репозитория и загрузка результатов работы на него. 55](#_Toc122743543)

[Заключение 56](#_Toc122743544)

[Библиографический список 57](#_Toc122743545)

[Приложение 58](#_Toc122743546)

# **Введение**

Композиционные материалы представляют собой многофазные системы, которые состоят из двух и более компонентов, сохраняющих индивидуальность (структуру и свойства) своего вещества в составе композита. Компонент, непрерывный в объеме композита, является матрицей или связующим элементом. Упрочняющие или армирующие элементы распределены в матрице в определенном порядке. Наполненные композиты содержат в матрице наполнители – дисперсные частицы неорганических и органических веществ, которые могут находиться в любой фазе. Разработка новых материалов имеет жизненно важное значение для решения насущных проблем в разных промышленных областях. Пространство гипотетических материалов, подлежащих рассмотрению, невероятно велико, и только небольшая часть возможных соединений может быть проверена экспериментально.

Уровень сложности разработки композитов очень высок высок из-за огромного количества компонентов и условий обработки, поскольку изменение одного фактора может иметь множество непредвиденных эффектов из-за их взаимосвязи. Традиционный подход к разработке новых материалов происходит в режиме «проб и ошибок», что приводит к проблемам низкой эффективности, высокой стоимости и неустойчивости полученных результатов. Между тем, многочисленные экспериментальные и вычислительные испытания накапливают огромные объемы многомерных, сложных и недостаточно изученных данных, которые могут скрывать важные правила «структура-свойства». Машинное обучение может помочь выявить подобные связи, а также предоставить возможность тестировать и оптимизировать несколько показателей одновременно, тем самым ускоряя процесс обнаружения и оптимизации конструирования материалов, спрогнозировав состав с желаемыми характеристиками и уникальными свойствами.

1. Аналитическая часть
   1. Постановка задачи

Предметом данного исследования является выбор состава и структуры композиционных материалов, объектом исследования - эффективность подхода к проектированию материалов на основе машинного обучения. Путем оптимального сочетания матриц, наполнителей, армирующих элементов и других компонентов, регулирования их концентрации и расположения в объеме изделия, а также выбора технологии обработки, можно в широких пределах управлять свойствами композиционных материалов. Однако сложность композиционных структур: механическая, термодинамическая, кинетическая совместимость компонентов, возникающие между ними физические, механические и химические связи в межфазном взаимодействии, могут легко привести к огромным пространствам конструирования, которые превышают вычислительный предел традиционных алгоритмов полнофакторного эксперимента, что подразумевает необходимость выведению принципиально новых подходов к решению задачи. В текущей работе будет выполнена реализация нескольких моделей машинного обучения, способных изучать и анализировать предложенные признаки для прогнозирования итогового результата. На входе имеется инженерный кейс с данными об эксплуатационных характеристиках компонентов композиционных материалов. На выходе необходимо:

1. Обучить алгоритм машинного обучения, который будет определять значение модуля упругости при растяжении
2. Обучить алгоритм машинного обучения, который будет определять значение прочности при растяжении
3. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель

В составе инженерного кейса были получены две выгрузки Excel:

1. X\_bp.xlsx
2. X\_nup.xlsx

Перед началом работы на основе данных выгрузок был сформирован датасет методом объединения INNER по индексу. После объединения были устранены строки для которых не нашлось индекса, а также отброшена неименнованная колонка. Дальнейшее исследование проводим с полученным датасетом.

Для удобства оценки данных, в приложении размещены отчеты в формате html:

1. pandas\_profiling\_report.html

Данный отчет сформирован с помощью инструментов библиотеки pandas\_profiling

1. sweetviz\_report.html

Данный отчет сформирован с помощью инструментов библиотеки sweetviz

1. jupyter\_notebook\_report.html

Данный отчет сформирован с помощью интерактивного блокнота jupyter notebook и предназначен для аттестации моих знаний и навыков в области работы с данными с помощью базовых инструментов дата-сайентиста

Перейдем к описанию полученных в ходе исследования значений. На основе данных выгрузок был сформирован датасет со следующими характеристиками:

Таблица 1- Статистика датасета

|  |  |
| --- | --- |
| Показатель | Количественное значение |
| Количество переменных | 13 |
| Количество наблюдений | 1023 |
| Отсутствующие значения | 0 |
| Повторяющиеся наблюдения | 0 |
| Количество категориальных переменных | 1 |
| Количество числовых переменных | 12 |

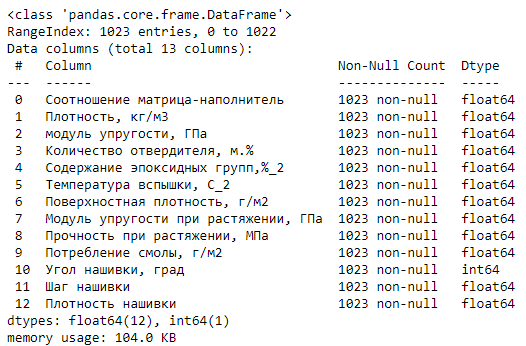
Наличие пропусков, дубликатов, поврежденных данных имеют сильное влияние на качество итоговых моделей. Поэтому данный этап подготовки очень важен для будущего результата. В текущем датасете отсутствуют проблемные данные, однако при обнаружении подобных значений, необходимо очистить, заменить либо исправить показатели. На полученном примере приемлемым решением было бы удаление показателей, т.к. при такой размерности датасета, искусственное пополнение выборки невозможно. Также на данном этапе возможно комбинирование данных из разных источников. Однако, в рамках полученного инженерного кейса не было указано, что можно выполнить данный этап. В ходе исследования, я нашла следующие источники данных по материалам с их эксплуатационными характеристиками:

Таблица 2 – Дополнительные источники данных

|  |  |  |
| --- | --- | --- |
| База данных | Категория материалов | Механические свойства |
| AFLOW | Сплавы, неорганические соединения | Упругие свойства |
| Materials Project (MP) | Неорганические соединения, нанопористые материалы | Упругие свойства |
| MATDAT | Стали, алюминиевые и титановые сплавы, наплавления и т.д. | Статические свойства, нелинейное поведение напряжение-деформация, циклическое стресс-деформационное поведение, усталостное поведение, стойкость материалов |
| MatWeb | Полимеры, металлы, керамика, полупроводники, волокна и т.д. | Эластичные свойства, прочность, твердость и т.д. |
| MatMatch | Металлы, композиты, керамика, полимеры, стеклоизделия и т.д. | Упругие свойства, прочность, твердость и т.д. |
| MakeItForm | Металлы, полимеры, керамика | Упругие свойства, прочность, твердость и т.д. |
| NIMS materials database  (MatNavi) | Полимеры, неорганические материалы, металлы | Упругие свойства, прочность, твердость и т.д. |

Для выбора входных и выходных факторов, рассмотрим сводную характеристику данных, используя рисунок 1

Рисунок 1 – Сводная характеристика данных



Входными параметрами для первой задачи (прогноз модуля упругости при растяжении) будут следующие значения:

* Соотношение матрица-наполнитель
* Плотность, кг/м3
* Модуль упругости, ГПа
* Количество отвердителя, м.%
* Содержание эпоксидных групп, %\_2
* Температура вспышки, С\_2
* Поверхностная плотность, г/м2
* Прочность при растяжении, МПа
* Потребление смолы, г/м2
* Угол нашивки, град
* Шаг нашивки
* Плотность нашивки

Выходным параметром будет: Модуль упругости при растяжении

Входными параметрами для второй задачи (прогноз прочности при растяжении) будут следующие значения:

* 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
* Потребление смолы, г/м2
* Угол нашивки, град
* Шаг нашивки
* Плотность нашивки

Выходным параметром будет: Прочность при растяжении

Входными параметрами для третьей задачи (рекомендация соотношения матрица-наполнитель) будут следующие значения:

* 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, Мпа
* Прочность при растяжении, МПа
* Потребление смолы, г/м2
* Угол нашивки, град
* Шаг нашивки
* Плотность нашивки

Выходным параметром будет: Соотношение матрица-наполнитель

Данное определение факторов не является окончательным и может быть изменено входе разведочного анализа данных

* 1. Описание используемых методов

Все выходные переменные имеют числовой тип данных и относятся к непрерывным случайным величинам. В проектах, где требуется присвоить выходному параметру определенного значения, используются регрессионные модели. Прогнозное регрессионное моделирование — это задача аппроксимации функции отображения от входных переменных до непрерывной выходной переменной. Данный тип относится к контролируемому виду обучения. Перед переходом к выбору наиболее оптимальных методов для решения поставленной задачи, рассмотрим в таблице 3 наиболее популярные методы машинного обучения при проектировании материалов

Таблица 3- Популярные методы машинного обучения при проектировании материалов

|  |  |  |
| --- | --- | --- |
| Метод машинного обучения | Характеристики | Примеры применения в проектировании материалов |
| Линейная регрессия  Полиномиальная регрессия | Моделируют линейную или полиномиальную связь между входными и выходными переменными | Прогноз модуля или прочности |
| Метод опорных векторов  Регрессия опорных векторов | Разделяют многомерное пространство данных с одной или несколькими гиперплоскостями | Прогноз прочности или твердости; структурная оптимизация топологии |
| Метод случайного леса | Построение нескольких деревьев решений для классификации или прогнозирования | Расчет модуля или прочности |
| Нейронные сети прямого  распространения  Многослойный персептрон | Соединение узлов (нейронов) с информацией, протекающее в одном направлении | Прогноз модуля, прочности и, предсказание гиперупругого или пластичного поведения |
| Сверточные нейронные сети | Захват функций на разных иерархических уровнях путем расчета сверток, работа с пиксельными или воксельными данными | Прогнозирование полей деформации или упругих свойств высококонтрастных композитов  Модуль однонаправленных композитов  Поля напряжений в консольных конструкциях  Предел текучести металлов аддитивного производства  Прогнозирование развития износа и трещин в поликристаллических сплавах  Предсказание пластичности кристаллов  Проектирование мозаичных композитов;  Дизайн растяжимого графенового киригами  Структурная оптимизация топологии |
| Рекуррентные нейронные сети  Сети с долговременной краткосрочной памятью  Управляемый рекуррентный нейрон | Соединение узлов (нейронов), формируя ориентированный граф с информацией об истории, хранящейся в скрытых состояниях, работа с последовательными данными | Прогнозирование закономерностей разрушения кристаллических тел  Предсказание пластического поведения гетерогенных материалов  Многомасштабное моделирование пористых сред |
| Генеративно-состязательная сеть | Обучение двух нейронных сетей-противников генерировать и различать показатели по отдельности, пока две сети не достигнут равновесия, генерирование новых данных в соответствии с распределением обучающего набора | Прогнозирование распределения модулей путем решения обратных задач упругости  Предсказание полей деформаций или напряжений в композитах  Структурная оптимизация топологии  Архитектурный дизайн материалов |
| Гауссова регрессия процесса  Байесовские методы машинного обучения | Рассмотрение параметров как случайные величины и вычисление распределения вероятностей этих переменных, количественная оценка неопределенность предсказаний модели | Прогноз модуля и прочности,  разработка сверхсжимаемых и восстанавливаемых метаматериалов |
| Активное обучение | Взаимодействует с пользователем на лету для маркировки новых данных, дополнение обучающих данных апостериорными экспериментами или симуляциями | Прогноз прочности |
| Генетические или эволюционные алгоритмы | Имитация эволюционных правил оптимизации целевой функции | Проекты прогнозирования твердости активных материалов проектирование модульных метаматериалов |
| Обучение с подкреплением | Максимизация совокупных награды с агентами, реагирующими на окружающую среду | Вывод законов тяги-разделения на основе микроструктуры |
| Графовая нейронная сеть | Обработка данных с неевклидовыми структурами. Включают прогнозирование ссылок, классификацию узлов и классификацию графов | Проектирование материалов с прогнозированием твердости |

* + 1. Простая 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 своей реализации, а также 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. Чувствительна к выбросам
  + 1. Регрессия по методу Лассо

LASSO - это 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, налагающая ограничения на коэффициенты регрессии для получения куда более приближенного к реальности результата. Вдобавок, этот результат гораздо проще интерпретировать. Применяется метод для борьбы с переизбыточностью данных, когда независимые переменные коррелируют друг с другом (мультиколлинеарность)

* + 1. Метод опорных векторов

Метод опорных векторов (support vector machine, SVM) — 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 выборке 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. Метод k-ближайших соседей

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

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

Плюсы:

* Алгоритм прост и легко реализуем
* Не чувствителен к выбросам
* Нет необходимости строить модель, настраивать несколько параметров или делать дополнительные допущения
* Алгоритм универсален. 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

Деревья решений (Decision Trees) - еще один непараметрический 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.

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 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. Случайный лес

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

Логика модели случайного леса заключается в том, что несколько некоррелированных моделей (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 (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.

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 Mean Absolute Error (MAE) вместо Mean Squared Error (MSE) значительно снижает влияние выбросов на вашу модель (выбор функции в параметре criterion)
* Ваша модель будет склонна к переобучению при слишком большом количестве деревьев. Данная проблема присутствует в любом алгоритме, связанном с деревьями и справляется правильной настройкой параметра n\_estimators
* Вычисления могут занять много времени. Поэтому, если у вас большой набор данных, всегда составляйте правильный размер выборки и не забывайте правильно настроить параметр min\_samples\_leaf
  + 1. Нейронная сеть

Нейронная сеть — это алгоритм машинного обучения, основанный на модели человеческого нейрона. Человеческий мозг состоит из миллионов нейронов. Он посылает и обрабатывает сигналы в виде электрических и химических сигналов. Эти нейроны связаны специальной структурой, известной как синапсы. Синапсы позволяют нейронам передавать сигналы. Из большого количества смоделированных нейронов формируются нейронные сети.

Искусственная нейронная сеть — это метод обработки информации. Это работает так же, как человеческий мозг обрабатывает информацию. Искусственная нейронная сеть включает в себя большое количество связанных блоков обработки, которые работают вместе для обработки информации. Они также генерируют значимые результаты от этого. У каждого нейрона есть определённое количество входов, куда поступают сигналы, которые суммируются с учётом значимости (веса) каждого входа.

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

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

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

- входной слой — его размер соответствует входным параметрам;

- скрытые слои — их количество и размерность определяем специалист;

- выходной слой — его размер соответствует выходным параметрам.

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

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.

* 1. Разведочный анализ данных

Когда данные достигают этого этапа, они не содержат ошибок и пропущенных значений и, следовательно, подходят для поиска закономерностей с использованием визуализации, а также для расширения пространства информационных признаков новыми данными. Значения, полученные далее, помогут нам понять, какой модели следуют данные.

Рассмотрим подробнее статистические значения по каждой переменной датасета:

Таблица 4 – Описательная статистика категориальной переменной «Угол нашивки, град»

|  |  |  |
| --- | --- | --- |
| Принимаемое значение | Количество в выборке | Процентное количество в выборке |
| 0 | 520 | 50.8 |
| 90 | 503 | 49.2 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | MIN | MAX | Q1 | Q3 | MEDIAN | RANGE | Interquartile range (IQR) | 95-th  percentile | 5-th percentile |
| Соотношение матрица-наполнитель | 0.3894026052 | 5.591741599 | 2.317886644 | 3.552659897 | 2.90687765 | 5.202338994 | 1.234773253 | 4.466327342 | 1.497694829 |
| Плотность, кг/м3 | 1731.764635 | 2207.773481 | 1924.155467 | 2021.374375 | 1977.621657 | 476.0088455 | 97.2189077 | 2099.687888 | 1854.093559 |
| Модуль упругости, ГПа | 2.436908754 | 1911.536477 | 500.0474518 | 961.8125265 | 739.6643277 | 1909.099568 | 461.7650747 | 1274.138406 | 1274.138406 |
| Количество отвердителя, м.% | 17.74027456 | 198.9532072 | 92.44349745 | 129.7303657 | 110.5648399 | 181.2129326 | 37.28686826 | 155.3054223 | 60.54599576 |
| Содержание эпоксидных групп, %\_2 | 14.2549855 | 33 | 20.60803427 | 23.96193376 | 22.23074376 | 18.7450145 | 3.353899497 | 26.1717864 | 18.38447957 |
| Температура вспышки, С\_2 | 100 | 413.2734182 | 259.0665278 | 313.0021055 | 285.8968123 | 313.2734182 | 53.93557769 | 352.3482186 | 220.4278438 |
| Поверхностная плотность, г/м2 | 0.6037399252 | 1399.542362 | 266.8166453 | 693.2250171 | 451.8643652 | 1398.938622 | 426.4083718 | 980.5062869 | 62.97211982 |
| Модуль упругости при растяжении, ГПа | 64.05406056 | 82.68205104 | 71.24501837 | 75.35661204 | 73.26880459 | 18.62799048 | 4.111593672 | 78.87213912 | 68.37767913 |
| Прочность при растяжении, МПа | 1036.856605 | 3848.436732 | 2135.850448 | 2767.193119 | 2459.524526 | 2811.580127 | 631.342671 | 3307.048697 | 1676.377372 |
| Потребление смолы, г/м2 | 33.80302553 | 414.5906284 | 179.6275202 | 257.4817241 | 219.1988822 | 380.7876028 | 77.85420383 | 313.6899267 | 120 |
| Шаг нашивки | 0 | 14.44052188 | 5.080032843 | 8.58629272 | 6.916143856 | 14.44052188 | 3.506259877 | 11.35618372 | 2.789696371 |
| Плотность нашивки | 0 | 103.9889013 | 49.79921216 | 64.94496114 | 57.34191985 | 103.9889013 | 15.14574898 | 77.12087746 | 36.0430183 |

Таблица 5 – Квантильная статистика числовых переменных

Таблица 6- Описательная статистика числовых переменных

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Standard  deviation (STD) | Coefficient of variation (CV) | Kurtosis | Mean | Median  Absolute  Deviation (MAD) | Skewness | Sum | Variance |
| Соотношение матрица-наполнитель | 0.9132222362 | 0.3116410397 | -0.1993339274 | 2.930365773 | 0.6295085376 | 0.08227115527 | 2997.764186 | 0.8339748527 |
| Плотность, кг/м3 | 73.72923055 | 0.03731737036 | 0.06362876195 | 1975.734888 | 48.48881833 | 0.05862923479 | 2021176.791 | 5435.999438 |
| Модуль упругости, ГПа | 330.2315806 | 0.4463051921 | -0.2063308896 | 739.9232328 | 227.6633966 | 0.1201735567 | 756941.4671 | 109052.8968 |
| Количество отвердителя, м.% | 28.29591129 | 0.2559077018 | 0.1457629197 | 110.5707686 | 18.87729992 | -0.0861967668 | 113113.8963 | 800.6585957 |
| Содержание эпоксидных групп, %\_2 | 2.406301292 | 0.1081756497 | 0.04263435238 | 22.24438955 | 1.680714274 | 0.05425796923 | 22756.01051 | 5.790285906 |
| Температура вспышки, С\_2 | 40.94325995 | 0.1432172654 | 0.1879241692 | 285.8821514 | 27.06826393 | -0.0565263971 | 292457.4408 | 1676.350536 |
| Поверхностная плотность, г/м2 | 281.3146902 | 0.5827556233 | -0.4391026268 | 482.731833 | 212.3290462 | 0.3954659538 | 493834.6652 | 79137.95494 |
| Модуль упругости при растяжении, ГПа | 3.118982889 | 0.04253434693 | -0.1613885538 | 73.32857125 | 2.054448804 | 0.1272831807 | 75015.12839 | 9.728054265 |
| Прочность при растяжении, МПа | 485.6280063 | 0.1968557743 | 0.02214620265 | 2466.922843 | 314.9080881 | 0.0785646041 | 2523662.068 | 235834.5605 |
| Потребление смолы, г/м2 | 59.73593087 | 0.2734871858 | -0.1241727109 | 218.4231437 | 39.27376454 | 0.006294281818 | 223446.876 | 3568.381437 |
| Шаг нашивки | 2.563467073 | 0.3715588575 | -0.01280982443 | 6.899222078 | 1.758723666 | 0.1019973888 | 7057.904185 | 6.571363434 |
| Плотность нашивки | 12.3509688 | 0.2161000813 | 0.8606961026 | 57.15392943 | 7.588061413 | -0.2295348796 | 58468.46981 | 152.5464303 |

Визуализируем распределение переменных с помощью гистограмм.

Одним из самых удобных способов определить тип распределения вероятностей является визуализация данных с помощью гистограмм. По оси X отображаются значения в наборе данных, а по оси Y — частота каждого значения. В зависимости от значений в наборе данных гистограмма может принимать различные формы. Распространенной формой является кривая в форме колокола, известная как «нормальное распределение». В нормальном распределении точки с такой же вероятностью могут появиться как по одну сторону от среднего значения, так и по другую. Согласно рисунку 2, переменная «Поверхностная плотность, г/м2» имеет положительное ассиметричное распределение, прочие переменные имеют нормальное распределение.

Рисунок 2- Гистограммы распределения переменных

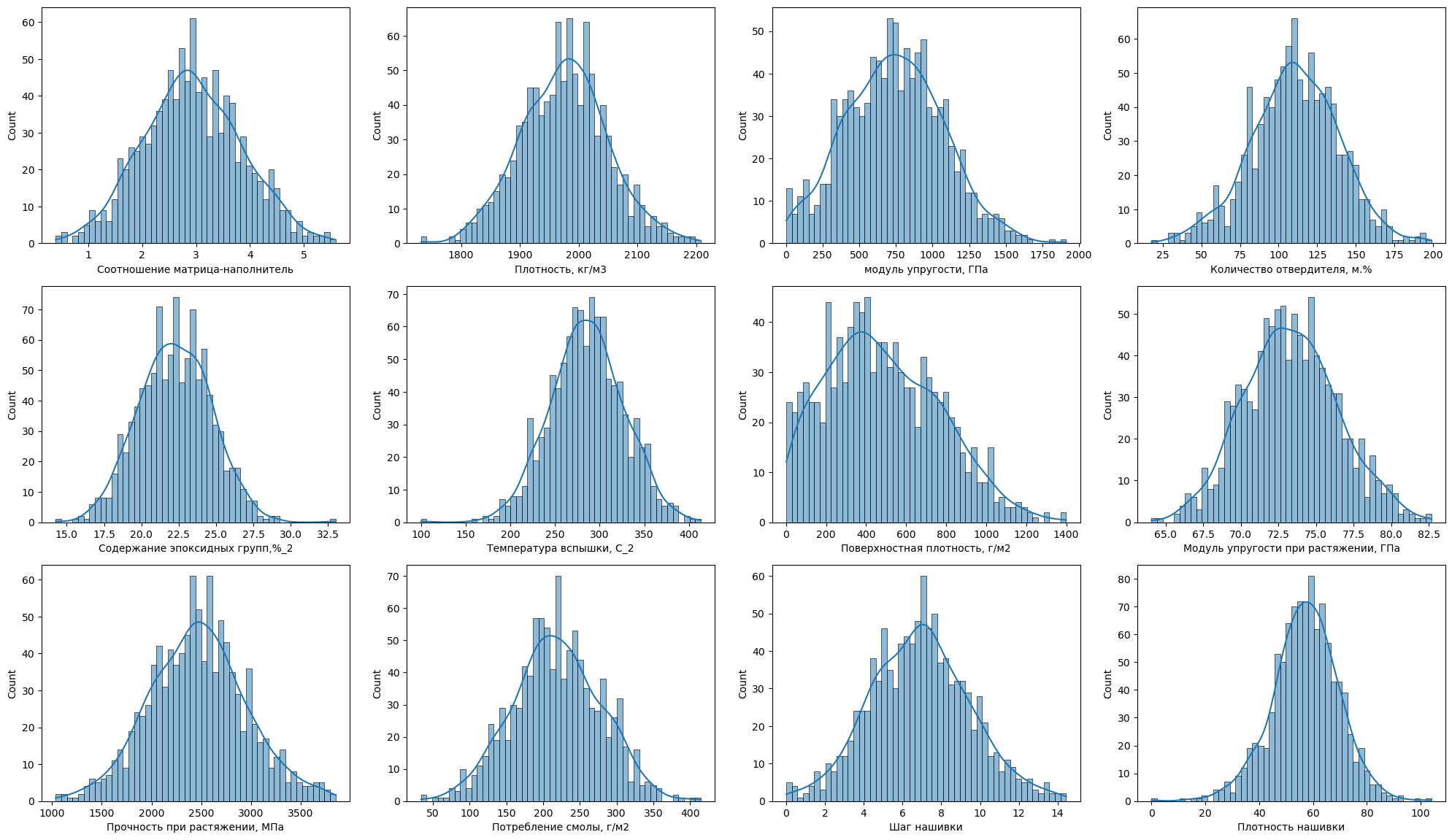
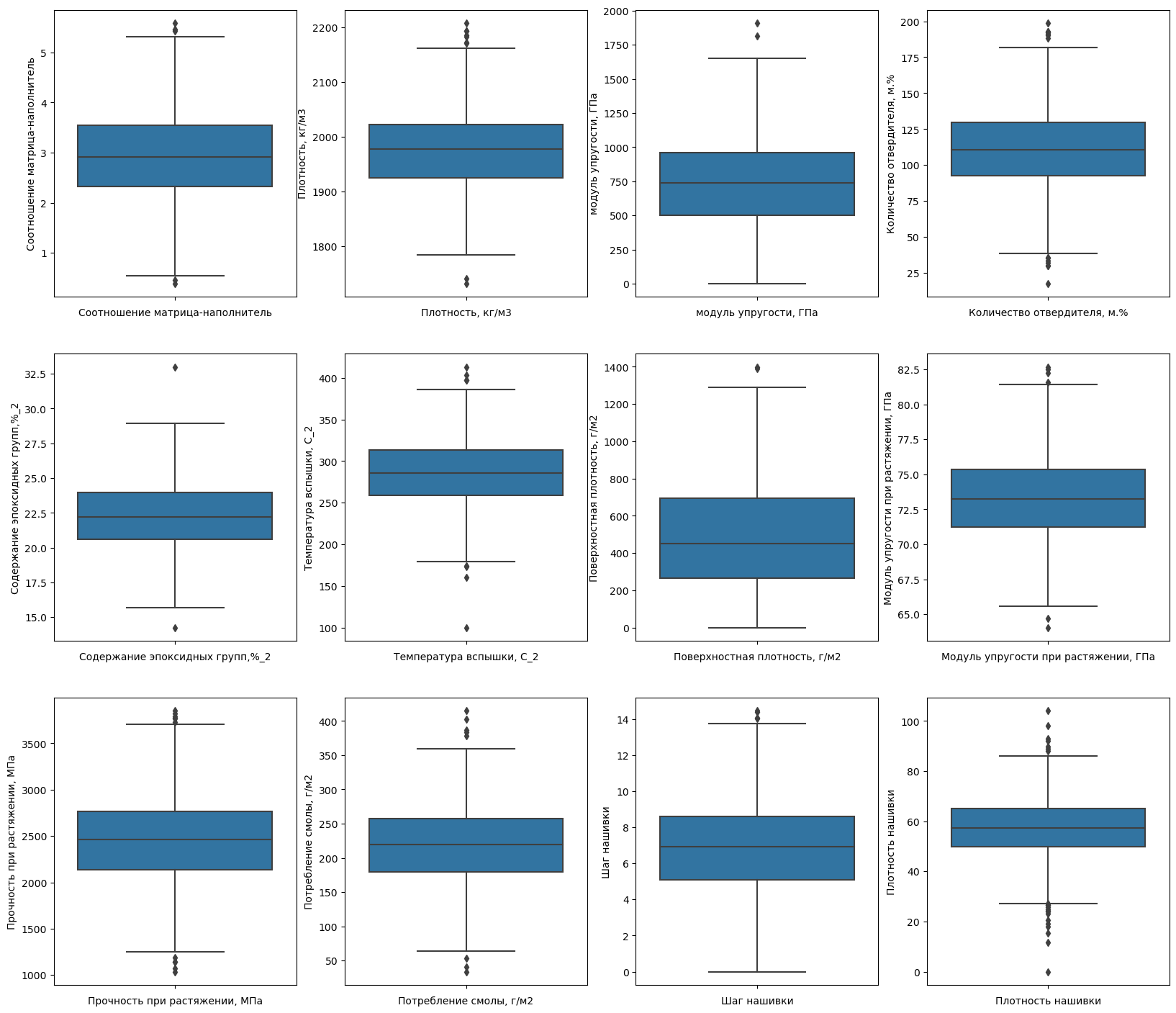


Рисунок 3- Диаграммы «Ящик с усами»

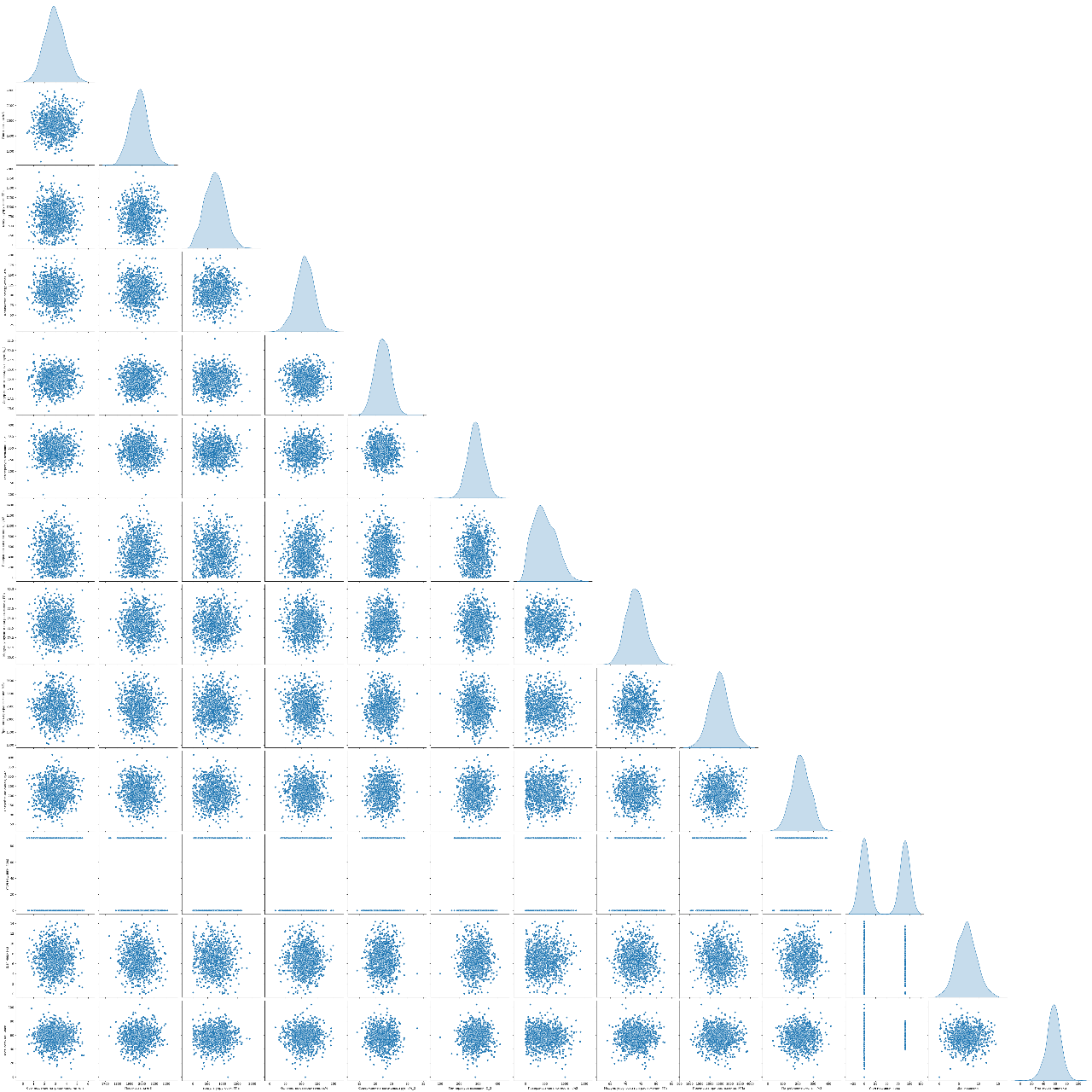


На рисунке 3 находятся диаграммы блочного типа, которые отображают сведения о переменных по пяти значениям:

1. Минимальное значение
2. Максимальное значение
3. Медиану
4. Нижний квартиль
5. Верхний квартиль

Диапазон значений от нижнего к верхнему квартилю называется межквартильным диапазоном. Средние 50% значений попадают в межквартильный диапазон. Усы идут от каждого квартиля к минимуму или максимуму. Данные, находящиеся в крайних частях, проверяются на выбросы. Как правило, выбросы попадают дальше указанного расстояния от первого и третьего квартилей.

Рисунок 4- Попарные графики рассеяния



Графики попарного рассеяния на рисунке 4 позволяют нам увидеть взаимосвязь между любыми двумя переменными соответствующего набора данных.

1. Практическая часть
   1. Предобработка данных

Для обеспечения корректной работы моделей необходимо провести предобработку данных. При этом значения выходных параметров предобработкой перед подачей в модель на этом этапе не затрагиваются. Нам необходимо обработать категориальную переменную «Угол нашивки, град». Для этого можно использовать кодировщики OneHotEncoder() или OrdinalEncoder(). LabelEncoder() не подходит, т.к. в документации указано, что данный метод не используют на входных переменных. Я выбрала OrdinalEncoder() по причине того, что не хотела менять размерность своих датафреймов. Также у нас есть всего два значения для данной переменной (0 и 90), а также слабая коррелируемость параметров, соответственно можно обойтись и OrdinalEncoder(). Числовые переменные необходимо привести к одному диапазону значений, т.к. дисбаланс между значениями признаков может вызвать неустойчивость работы модели, ухудшить результаты обучения и замедлить процесс моделирования. Для нормализации числовых данных используются следующие методы:

− MinMaxScaler() – масштабирование до заданного диапазона. Не использую, т.к. наши данные имеют нормальное распределение

− RobustScaler () – аналогичен методы MinMax, но он использует межквартильный диапазон (а не минимальные и максимальные значения, используемые в масштабере MinMax). Чаще используется для данных, устойчивых к выбросам

− StandartScaler() - используется для изменения размера распределения значений так, чтобы среднее значение наблюдаемых значений было равно 0, а стандартное отклонение – 1. Используя стандартизацию, мы центрируем столбцы признаков по среднему значению 0 со стандартным отклонением 1, 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, в отличие от масштабирования минимум-максимум, которое масштабирует данные до ограниченного диапазона значений.

Графики распределения для каждого признака до нормализации предоставлены на рисунках 2, 3. Приступим к предобработке данных перед подачей в модели.

Для начала необходимо удалить выбросы. Для числовых признаков с нормальным распределением я использовала метод межквартильного диапазона. Формула поиска выбросов по межквартильному диапазону:

IQR = Q3 − Q1, (1)

где Q3 – третий квартиль (или 75 процентиль);

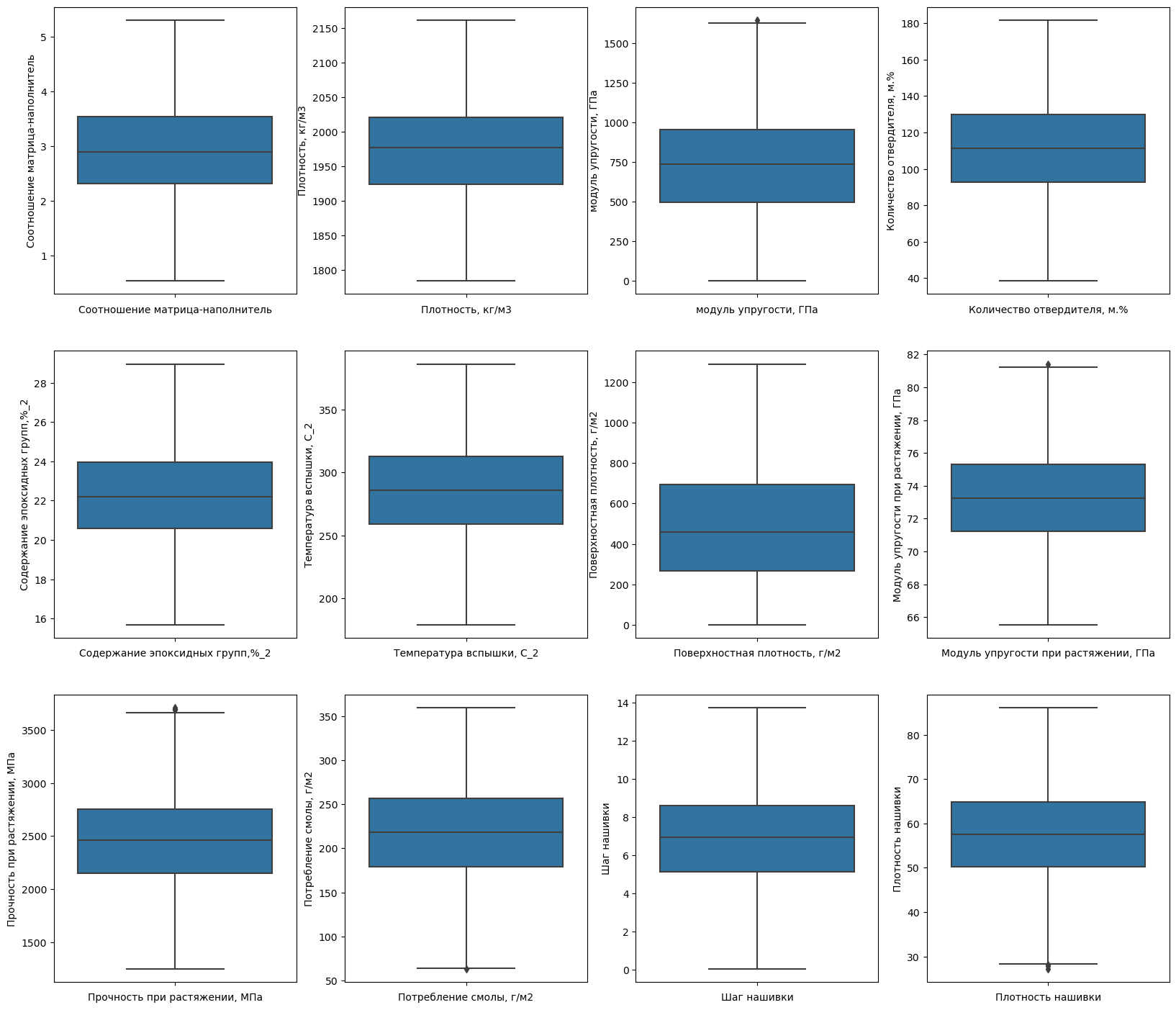
Q1 – первый квартиль (или 25 процентиль);

Уравнения для расчета низких или высоких выбросов с помощью диапазона IQR:

* + Высокий выброс-это значение, превышающее Q3 + (1,5 x IQR)
  + Низкий выброс-это значение, находящееся ниже границы Q1 − (1,5 x IQR)

Графики распределения признаков после удаления выбросов:

Рисунок 5 - Диаграммы «Ящик с усами»



* + 1. Обработка параметров для задачи прогнозирования модуля упругости при растяжении

Для выполнения задачи прогнозирования модуля упругости при растяжении разделим датафрейм на два по входным и выходному признаку.

Показатели входных переменных до обработки:

Рисунок 6 – Статистические показатели входных переменных до нормализации



Для нормализации данных, к числовым параметрам был применен метод StandardScaler(), для категориального параметра был использован кодировщик OrdinalEncoder (), чтобы параметр принял значения 0 и 1.

Показатели входных переменных после обработки:

Рисунок 7 – Статистические показатели входных переменных после нормализации



Рисунок 8 – Статистические показатели выходной переменной

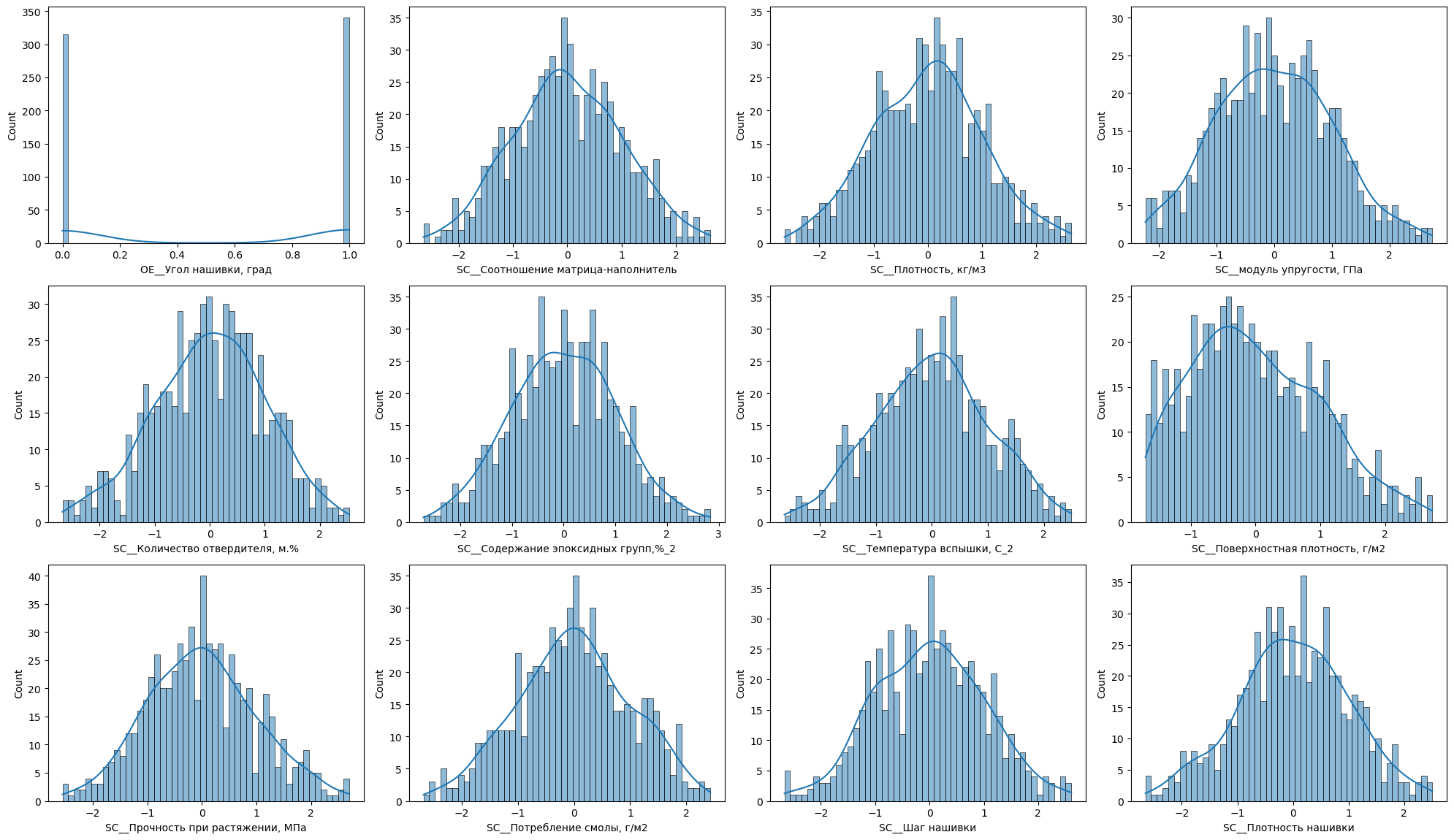


После этого выполняем разделение на тестовые и тренировочные значения методом train\_test\_split. Согласно требованиям инженерного кейса, тестовая 30% данных оставлено на тестирование модели

Рисунок 9 – Размерность тестовых датафреймов и тренировочных датафреймов



Рисунок 10 – Гистограммы распределения переменных после нормализации



* + 1. Обработка параметров для задачи прогнозирования прочности при растяжении

Для выполнения задачи прогнозирования модуля упругости при растяжении разделим датафрейм на два по входным и выходному признаку.

Показатели входных переменных до обработки:

Рисунок 11 – Статистические показатели входных переменных до нормализации



Для нормализации данных, к числовым параметрам был применен метод StandardScaler(), для категориального параметра был использован кодировщик OrdinalEncoder (), чтобы параметр принял значения 0 и 1.

Показатели входных переменных после обработки:

Рисунок 12 – Статистические показатели входных переменных после нормализации



Рисунок 13 – Статистические показатели выходной переменной

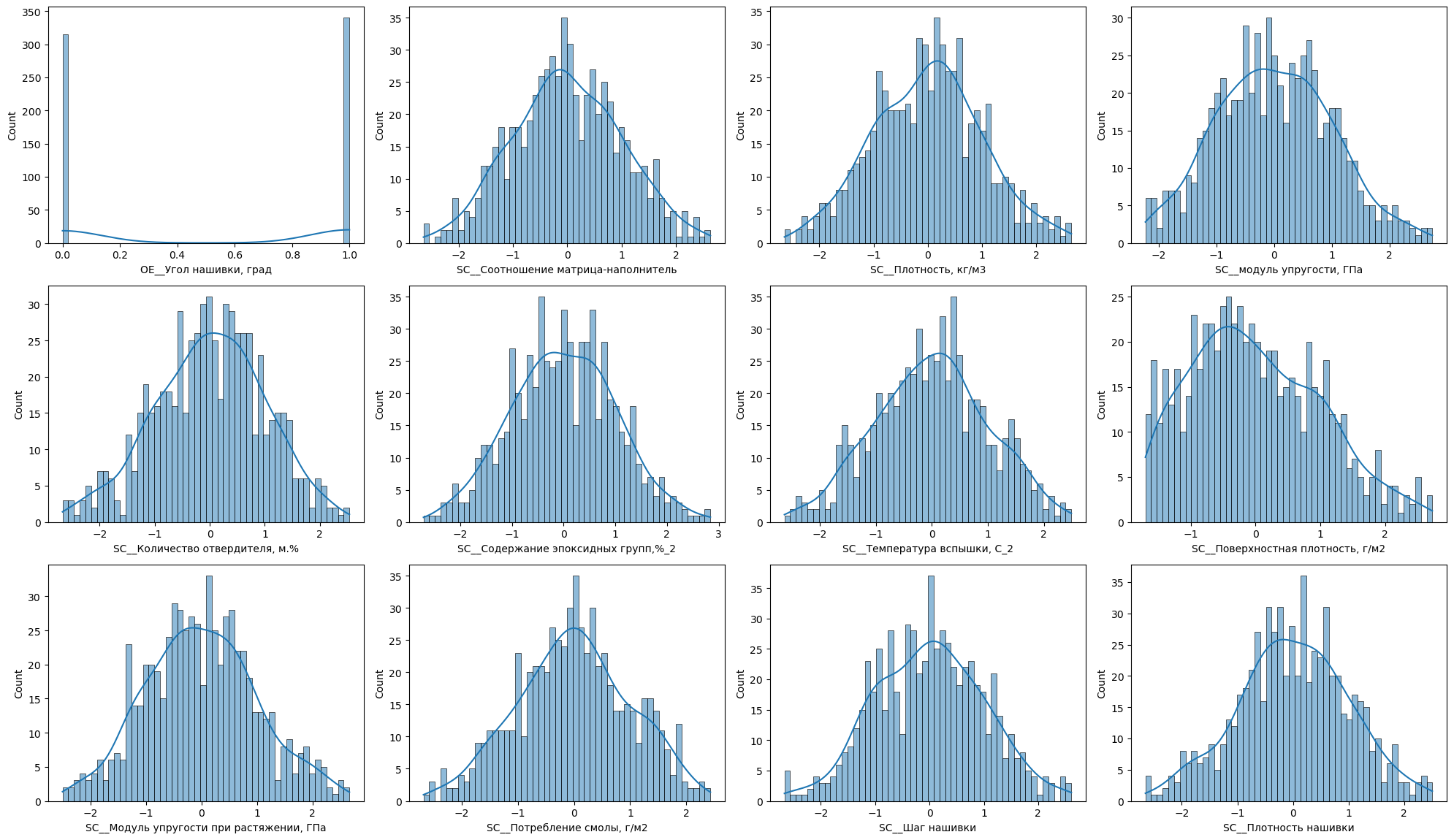


После этого выполняем разделение на тестовые и тренировочные значения методом train\_test\_split. Согласно требованиям инженерного кейса, тестовая 30% данных оставлено на тестирование модели

Рисунок 14 – Размерность тестовых датафреймов и тренировочных датафреймов



Рисунок 15 – Гистограммы распределения переменных после нормализации



* + 1. Обработка параметров для задачи прогнозирования соотношения матрица-наполнитель

Для выполнения задачи прогнозирования модуля упругости при растяжении разделим датафрейм на два по входным и выходному признаку.

Показатели входных переменных до обработки:

Рисунок 16 – Статистические показатели входных переменных до нормализации

****

Для нормализации данных, к числовым параметрам был применен метод StandardScaler(), для категориального параметра был использован кодировщик OrdinalEncoder (), чтобы параметр принял значения 0 и 1.

Показатели входных переменных после обработки:

Рисунок 17 – Статистические показатели входных переменных после нормализации



Рисунок 18 – Статистические показатели выходной переменной

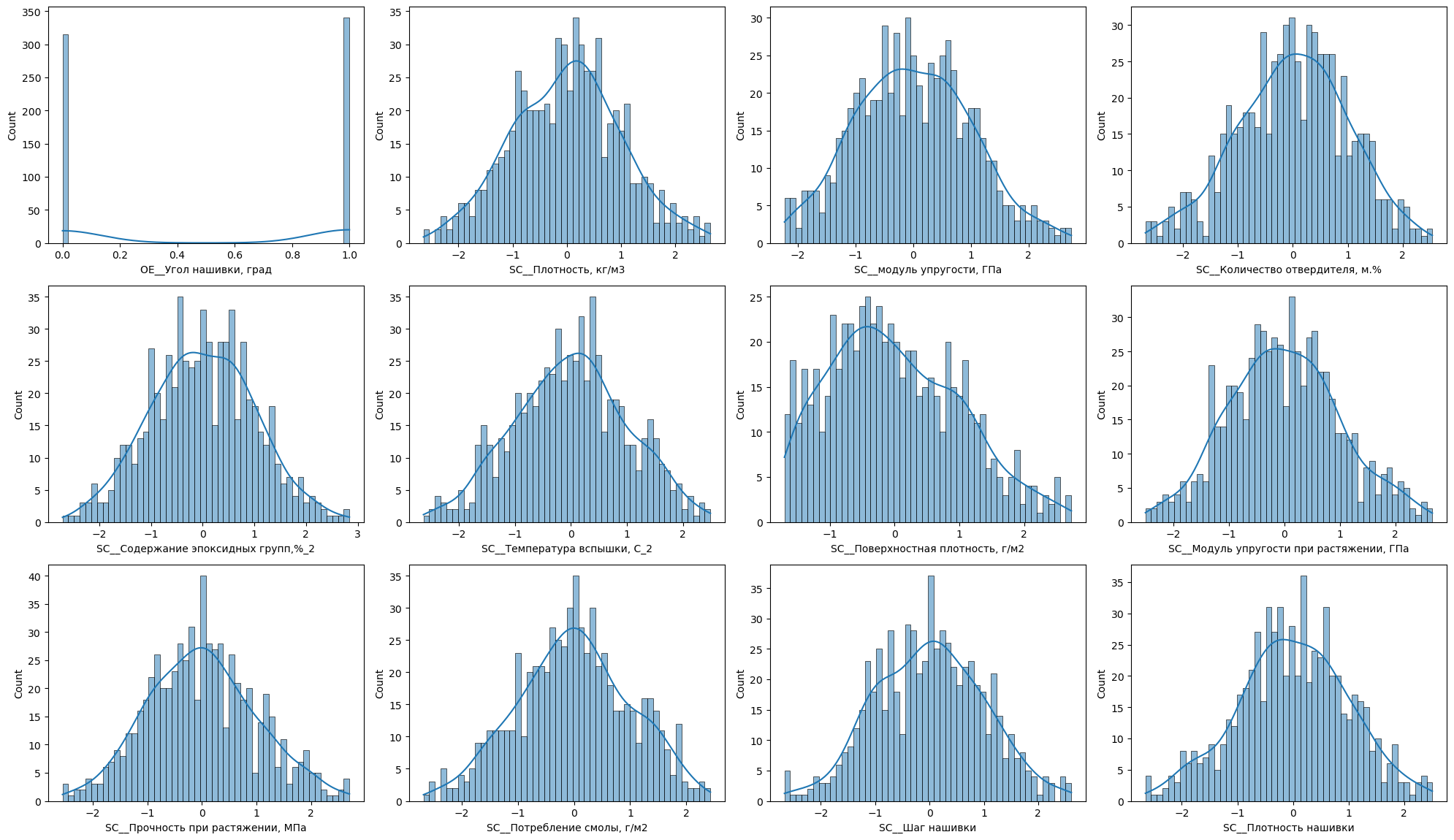


После этого выполняем разделение на тестовые и тренировочные значения методом train\_test\_split. Согласно требованиям инженерного кейса, тестовая 30% данных оставлено на тестирование модели

Рисунок 19– Размерность тестовых датафреймов и тренировочных датафреймов



Рисунок 20 – Гистограммы распределения переменных после нормализации

****

* 1. Разработка и обучение модели
     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 для прогнозирования модуля упругости при растяжении

На первом этапе разработки, необходимо выбрать базовую модель в качестве точки отсчета. Показатели этой модели можно будет сравнивать с показателями любого другого используемого алгоритма. В качестве базовой модели выбор пал на DummyRegressor()

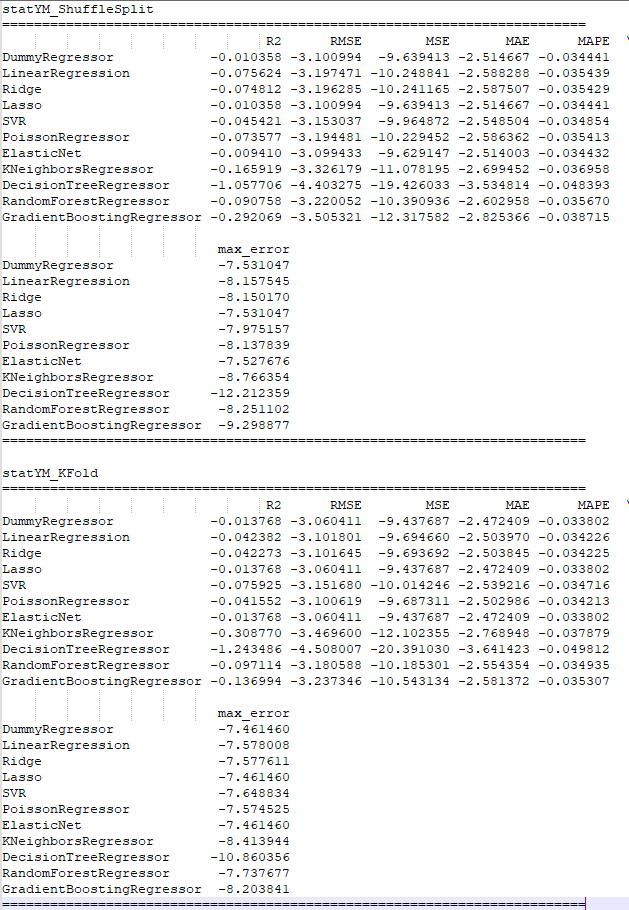
Список моделей, которые будут использоваться для решения задачи 1 и задачи 2:

* + - 1. Гребневая регрессия: Ridge()
      2. Регрессия Лассо: Lasso()
      3. Метод опорных векторов: SVR()
      4. Метод k-ближайших соседей: KneighborsRegressor()
      5. Деревья решений: DecisionTreeRegressor()
      6. Случайный лес: RandomForestRegressor()
      7. Эластичная сеть: ElasticNet()
      8. Градиентный бустинг: GradientBoostingRegressor()

Также для поиска оптимальных гиперпараметров для обучающихся алгоритмов я использую оценщики HalvingGridSearchCV() и GridSearchCV()

Результаты работы выбранных моделей с гиперпараметрами по умолчанию:

Рисунок 21 – Метрики работы моделей с гиперпараметрами по умолчанию



Ни одна модель не выдала хорошего результата. Большая часть алгоритмов выдало показатели хуже, чем базовая модель.

Проведем поиск лучших гиперпараметров для наших моделей.

Рисунок 22 - Результат подбора гиперпараметров для модели Ridge

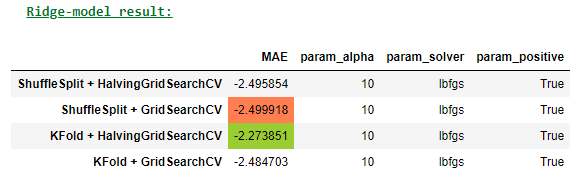


Рисунок 23 - Результат подбора гиперпараметров для модели Lasso

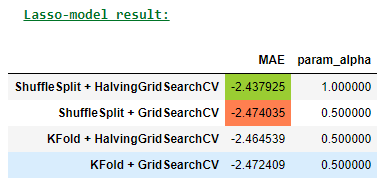


Рисунок 24 - Результат подбора гиперпараметров для модели SVR

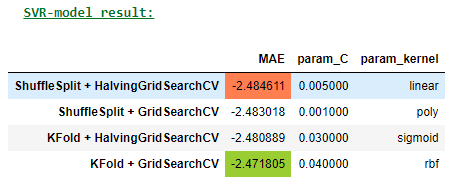


Рисунок 25 - Результат подбора гиперпараметров для модели KNeighborsRegressor

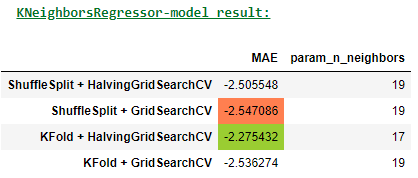


Рисунок 26 - Результат подбора гиперпараметров для модели DecisionTreeRegressor

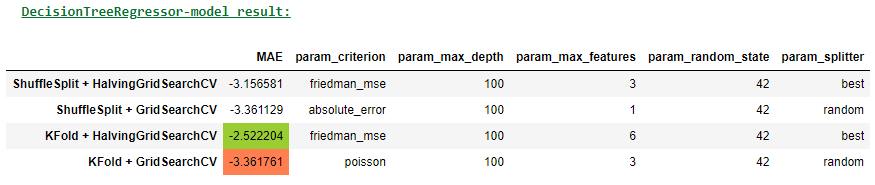


Рисунок 27 - Результат подбора гиперпараметров для модели RandomForestRegressor

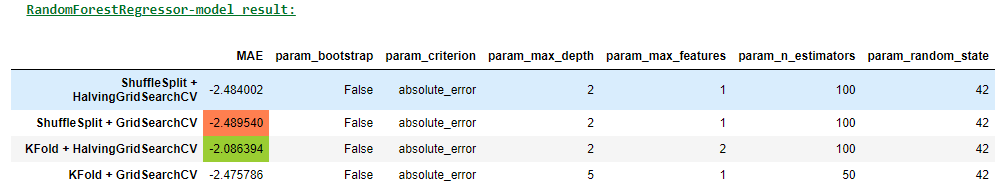


Рисунок 28 – Результат подбора гиперпараметров для модели ElasticNet

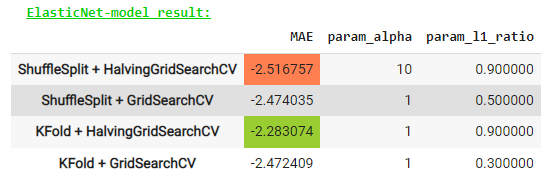
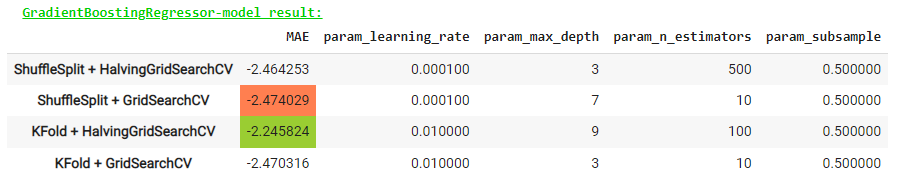
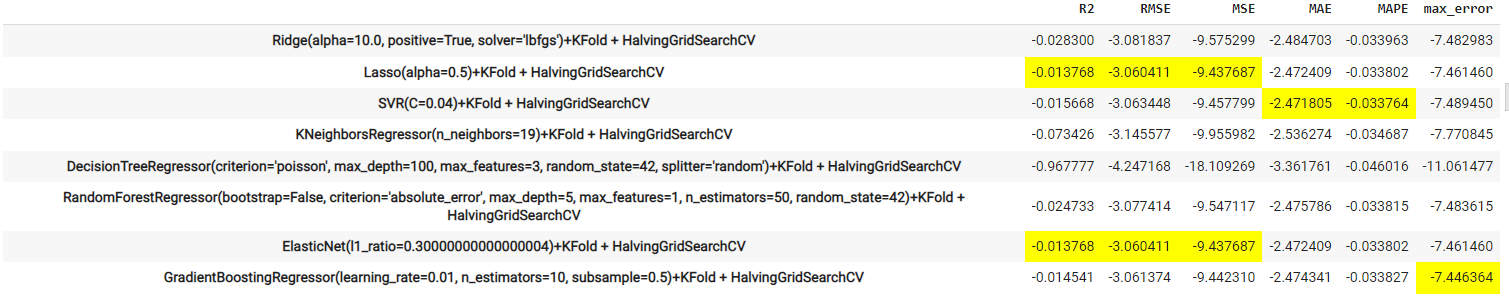


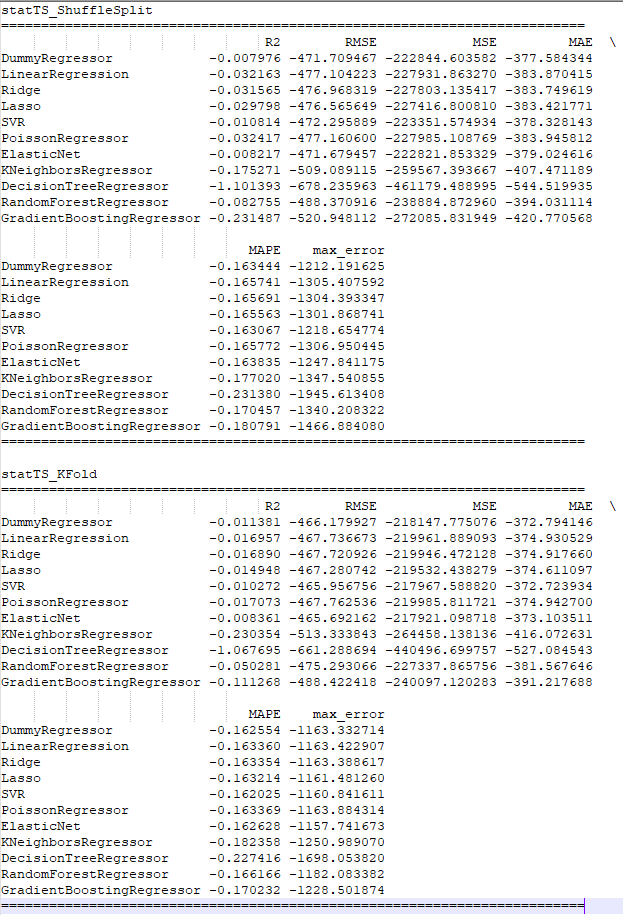
Рисунок 29 – Результаты подбора гиперпараметров для модели GradientBoostingRegressor

Рисунок 30 – Сводная таблица статистических результатов лучших моделей

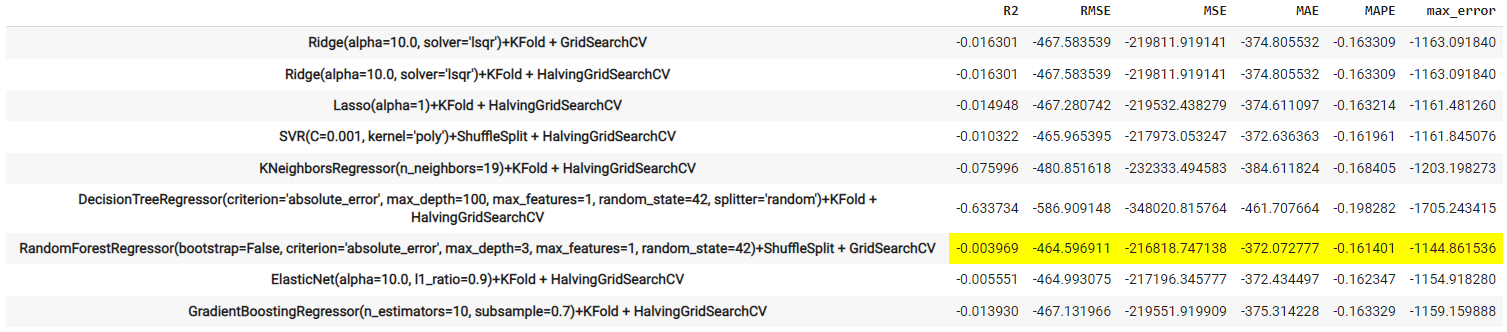
* + 1. Разработка и обучение моделей для прогнозирования прочности при растяжении

Результаты работы выбранных моделей с гиперпараметрами по умолчанию:

Рисунок 31 – Метрики работы моделей с гиперпараметрами по умолчанию



Ни одна из выбранных мной моделей не соответствует данным. Базовая модель снова выдает лучшие показатели. После выполнения подбора гиперпарметров по сетке с перекрестной проверкой, получили метрики, приведенные на рисунке 32.

Рисунок 32 – Метрики работы моделей после подбора гиперпараметров

Подбор гиперпараметров помог получить модель, превосходящую по показателям базовую. Однако, все равно все модели плохо предсказывают выходное значение. Не удалось добиться коэффициента детерминации больше нуля. Лучшие результат дал рандомный лес.

* 1. Тестирование моделей
     1. Тестирование лучшей модели для прогноза модуля упругости при растяжении

Оптимизация гиперпараметров по сетке с перекрестной проверкой, получили метрики, приведенные на рисунке 3. Подбирая гиперпараметры, можно улучшить предсказание выбранной модели. Однако, полученные результаты говорят о том, что все модели плохо обучаются на исходных данных - не удалось получить положительное значение R2. В качестве лучшей модели выбрана модель Lasso(). На рисунке 31 приведена визуализация работы лучшей модели на тестовых данных.

Полученный график показывает, что данные, выдаваемые лучшей моделью совсем не соответствуют исходным.

Рисунок 33 – Визуализация результатов модели Lasso

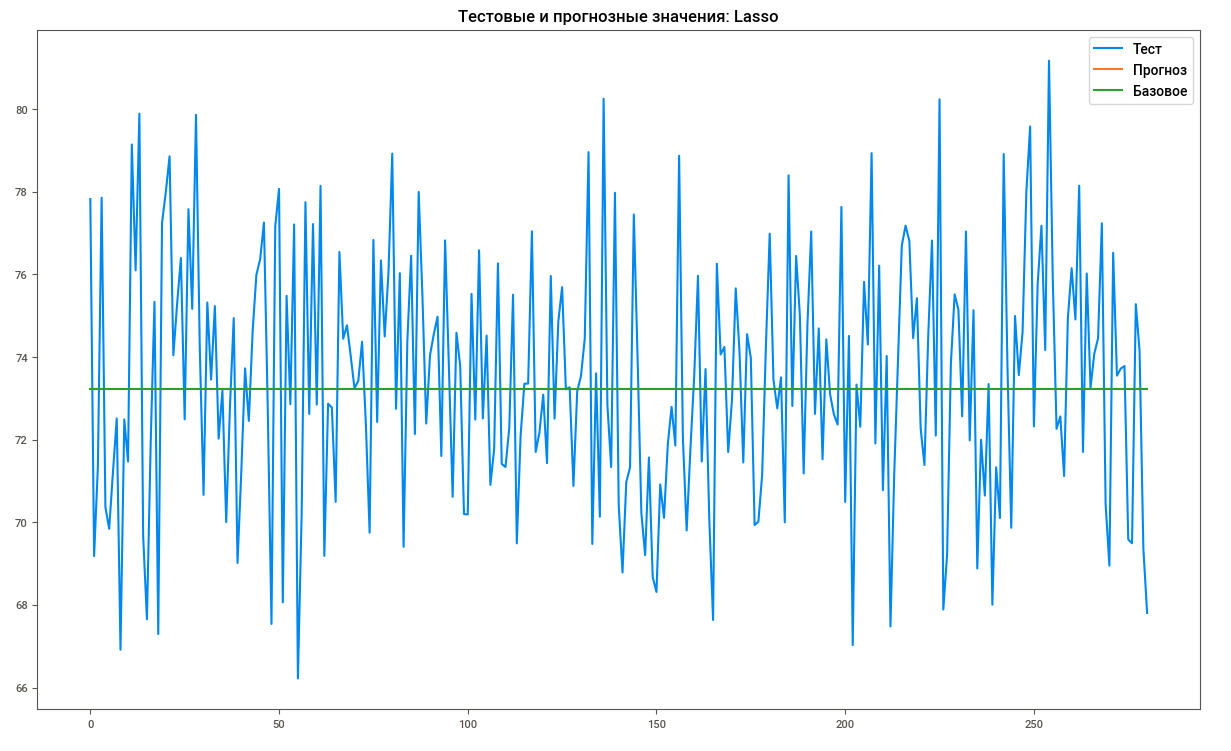
****

Рисунок 34 – Метрики работы лучшей модели на тестовых данных

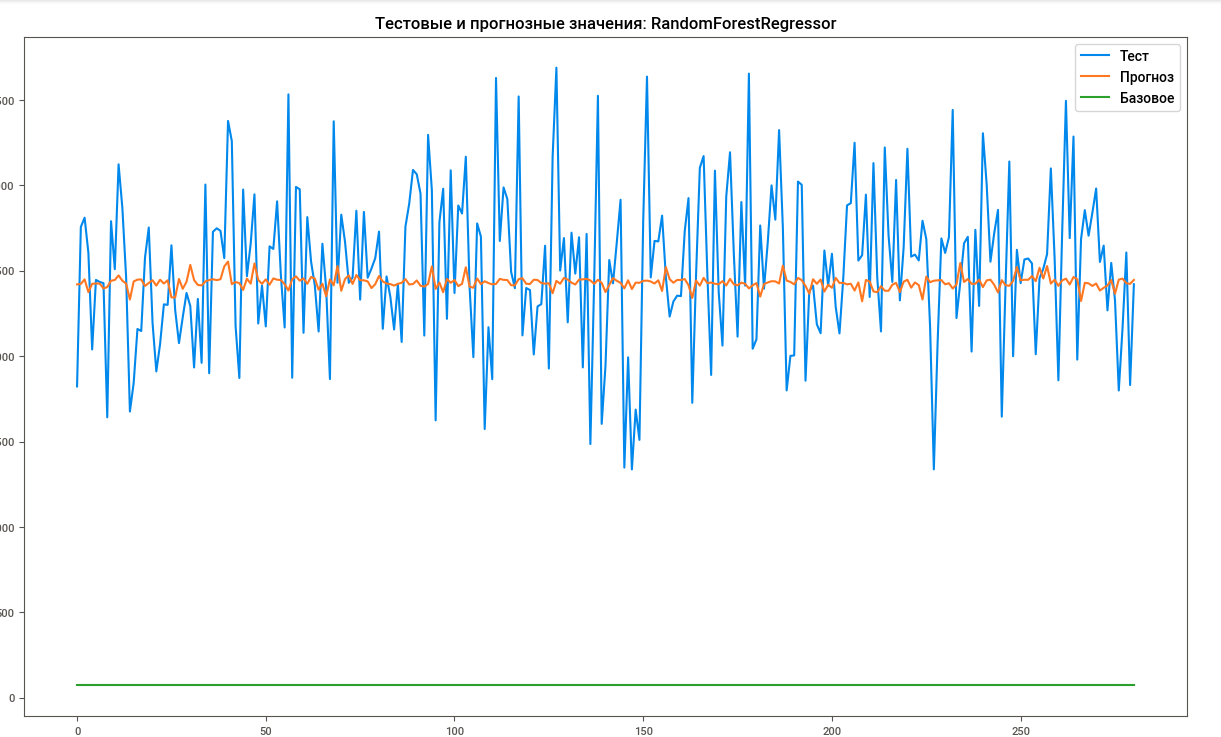


Метрики, указанные на рисунке 34, подтверждают, что полученная модель не лучше базовой. Не удалось получить модели, которая могла бы спрогнозировать качественный результат.

* + 1. Тестирование лучшей модели для прогноза прочности при растяжении

На рисунке 35 приведена визуализация работы лучшей модели на тестовых данных

Рисунок 35 – Визуализация результатов модели RandomForestRegressor



Метрики работы лучшей модели на тестовых данных и сравнение с базовой моделью отражены на рисунке 36. Несмотря на то, что рандомный лес предсказывает результаты чуть лучше базовой модели, результат предсказывающей способности отрицательный. Не удалось получить модели, которая могла бы спрогнозировать качественный результат.

Рисунок 36 – Метрики работы лучшей модели на тестовых данных

* 1. Нейросеть для рекомендации соотношения матрица-наполнитель

Для построения моделей с высокой точностью, я создала базовый класс гипермодели. В данный класс можно передать гиперпараметры, для их оптимизации.

Построение нейросети выполнено с помощью последовательной модели Sequential(). Данная модель подходит для простого стека слоев, где каждый слой имеет ровно один входной тензор и один выходной тензор. Для Sequential были выбраны следующие параметры:

* входной слой для 12 признаков
* выходной слой для 1 признака
* скрытых слоев: 2
* слоев нормализации: 2

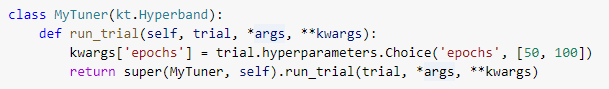
Подробное описание класса гипермодели указано на рисунке 37

Рисунок 37 – Настройки класса гипермодели



Также был создан собственный класс тюнера для Hyperband Tuner, т.к. я хотела также поэкспериментировать с количеством эпох. Описание класса тюнера находится на рисунке 38

Рисунок 38 –Класс гипертюнера



Результат работы тюнера представлен на рисунке 39-40

Рисунок 39 –Результат поиска тюнера

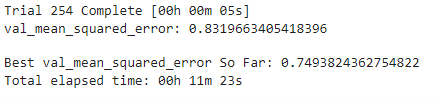
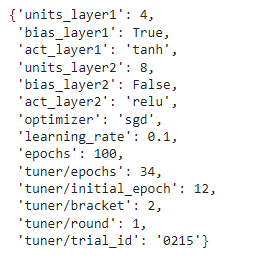


Рисунок 40 –Гиперпараметры, предложенные тюнером



Архитектура нейросети приведена на рисунках 41 и 42.

Рисунок 41 – Архитектура предложенной тюнером нейросети в формате summary

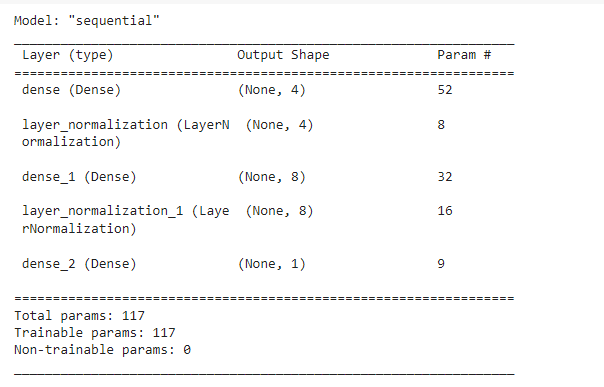


Рисунок 42 – Архитектура предложенной тюнером нейросети в виде графа

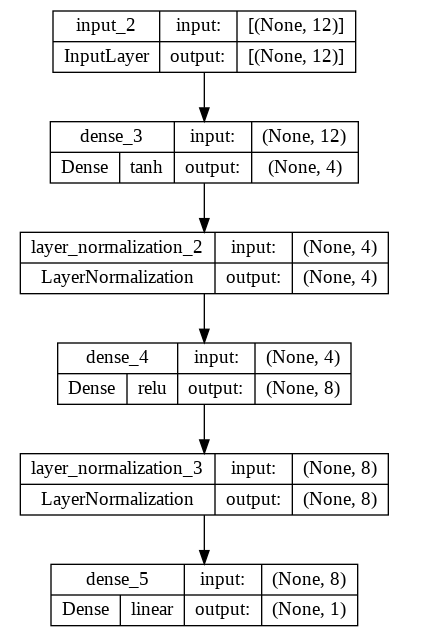


Рисунок 43 – График обучения нейросети, предложенной тюнером

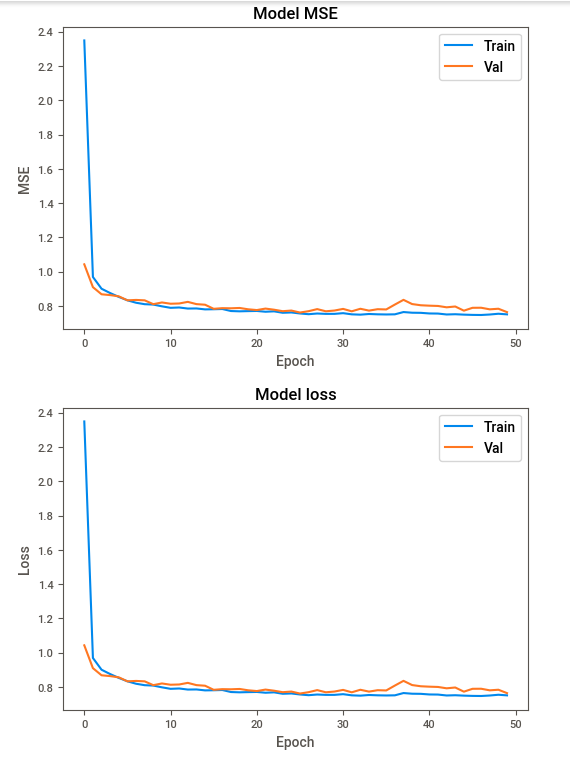


Рисунок 44 – Метрики нейросети, полученные на тестовых значениях



После работы с тюнером Hyperband, было принято решение провести эксперимент с тюнером RandomSearch.

Рисунок 45 –Результат поиска тюнера

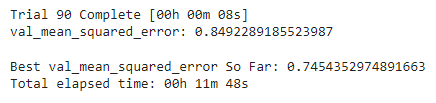
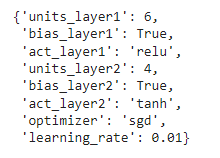


Рисунок 46 –Гиперпараметры, предложенные тюнером



Архитектура нейросети приведена на рисунках 47 и 48.

Рисунок 47 – Архитектура предложенной тюнером нейросети в формате summary

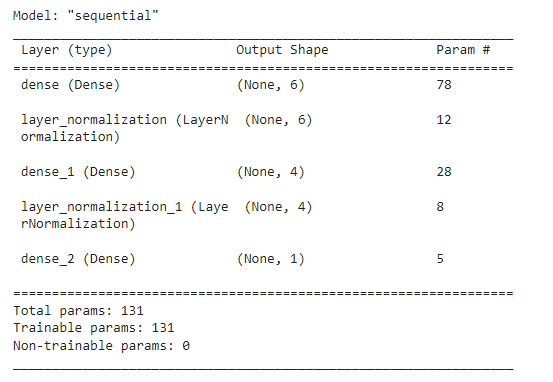


Рисунок 48 – Архитектура предложенной тюнером нейросети в виде графа

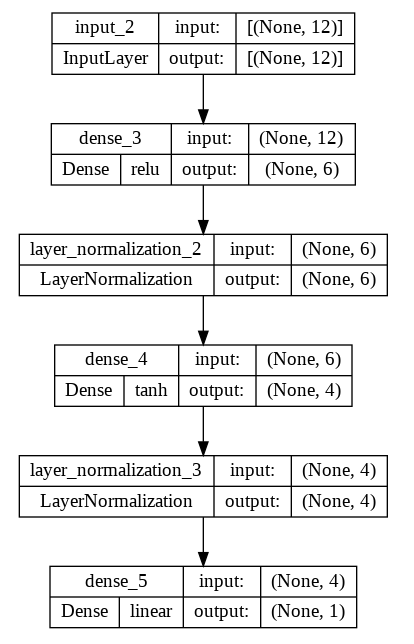


Рисунок 49 – График обучения нейросети, предложенной тюнером

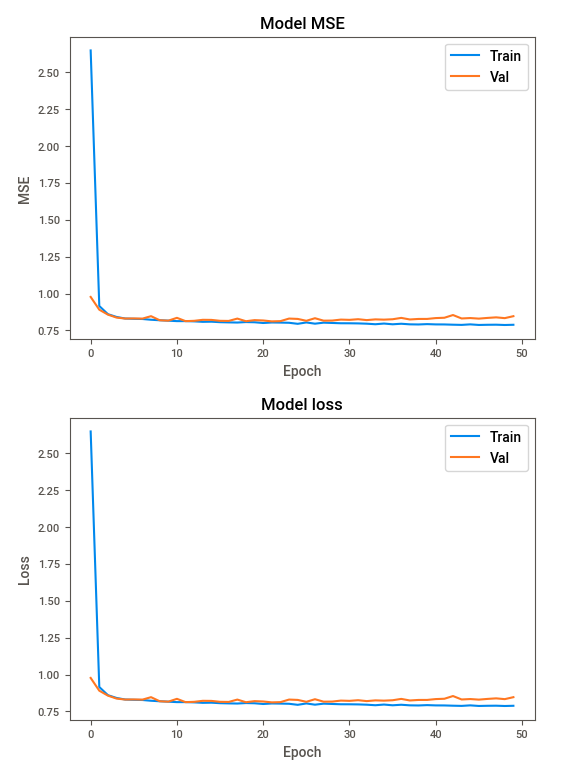


Рисунок 50 – Метрики нейросети, полученные на тестовых значениях



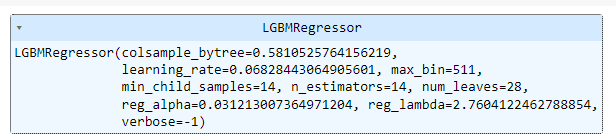
Также я решила попробовать AutoML из библиотеки **FLAML.**

При инициализации AutoML были заданы некоторые параметры, которые помогут в обучении модели:

* time\_budget: время, за которое модель должна быть обучена
* metric: метрики, по которым мы будем смотреть производительность нашей модели
* task: определение задачи (регрессия)

На рисунке 51 представлена модель, предложенная AutoML:

Рисунок 51 – Предложенная AutoML модель



Также с помощью AutoML можно вывести показания важности признаков (рисунок 52)

Рисунок 52 – Показатели важности признаков

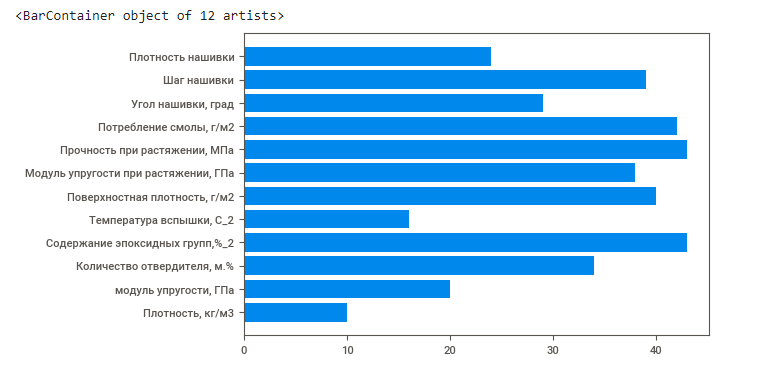


Рисунок 53 – Сводная статистика метрик работы моделей



* 1. Разработка приложения
  2. Создание удаленного репозитория и загрузка результатов работы на него.

Репозиторий на GitHub можно найти по ссылке <https://github.com/whiskeyjackAD/graduation_work>

Заключение

В ходе выполнения выпускной работы были выполнены основные задачи дата сайентиста. В этот список задач входит:

* изучение теоретических методов анализа данных и машинного обучения
* изучение основ предметной области
* извлечение и обработка данных
* проведение разведочного анализа
* выполнение препроцессинга данных для обеспечения корректной работы моделей
* подбор модели машинного обучения и архитектуры нейронной сети
* визуализация работы моделей, алгоритмов и оценка качества их работы
* сохранение и экспорт моделей
* разработка и тестирование веб-приложения

К сожалению, решить предложенный инженерный кейс не удалось.

Для получения более качественных результатов необходимо:

* попробовать различные архитектуры нейросетей и моделей
* провести более тщательное конструирование признаков
* попробовать провести аугментацию данных
* запросить больше входных параметров для моделей

Библиографический список

1 В.А. Струк, Л.С Пинчук, Н.К. Мышкин, В.А. Гольдаде, П.А Витязь. Материаловедение в машиностроении и промышленных технологиях / Издательский дом «Интеллект», 2010 – 536с

2 Сет Вейдман. Глубокое обучение: легкая разработка проектов на Python. / Правообладатель: Питер, 2019. – 272 с

3 Джоэл Грас. Data Science. Наука о данных с нуля / Правообладатель: БХВ-Петербург, 2019 – 411с

4 Дэви Силен, Арно Мейсман, Мохамед Али. Основы Data Science и Big Data. Python и наука о данных / Правообладатель: Питер, 2019 – 336с

5 Франсуа Шолле. Глубокое обучение на Python / Правообладатель: Питер, 2018 – 400с

6 By Amita Kapoor, Antonio Gulli, Sujit Pal. Deep Learning with TensorFlow and Keras - Third Edition / Packt, 2022 – 698

7 By Paul Crickard. Data Engineering with Python / Packt, 2020 -356

8 By Roy Jafari. Hands-On Data Preprocessing in Python / Packt, 2022 – 602

9 By Luis Pedro Coelho, Willi Richert, Matthieu Brucher. Building Machine Learning Systems with Python - Third Edition / Packt, 2018 – 406

Приложение

****

****