ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР МГТУ им. Н. Э. Баумана

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»

Тема: Прогнозирование конечных свойств новых материалов (композиционных материалов)

Слушатель Степанова Василиса Валерьевна

Москва, 2023

Содержание

Введение

Композиционные материалы — это искусственно созданные материалы, состоящие из нескольких других с четкой границей между ними. Композиты обладают теми свойствами, которые не наблюдаются у компонентов по отдельности. При этом композиты являются монолитным материалом, т. е. компоненты материала неотделимы друг от друга без разрушения конструкции в целом. Яркий пример композита - железобетон. Бетон прекрасно сопротивляется сжатию, но плохо растяжению. Стальная арматура внутри бетона компенсирует его неспособность сопротивляться сжатию, формируя тем самым новые, уникальные свойства. Современные композиты изготавливаются из других материалов: полимеры, керамика, стеклянные и углеродные волокна, но данный принцип сохраняется. У такого подхода есть и недостаток: даже если мы знаем характеристики исходных компонентов, определить характеристики композита, состоящего из этих компонентов, достаточно проблематично. Для решения этой проблемы есть два пути: физические испытания образцов материалов, или прогнозирование характеристик. Суть прогнозирования заключается в симуляции представительного элемента объема композита, на основе данных о характеристиках входящих компонентов (связующего и армирующего компонента). На входе имеются данные о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.). На выходе необходимо спрогнозировать ряд конечных свойств получаемых композиционных материалов. Кейс основан на реальных производственных задачах Центра НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана).

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

Традиционно разработка композитных материалов является долгосрочным процессом, так как из свойств отдельных компонентов невозможно рассчитать конечные свойства композита. Для достижения определенных характеристик требуется большое количество различных комбинированных тестов, что делает насущной задачу прогнозирования успешного решения, снижающего затраты на разработку новых материалов.

1. Аналитическая часть

1.1 Постановка задачи

Для исследовательской работы были даны 2 файла: X_bp.xlsx (с данными о параметрах, состоящий из 1023 строк и 10 столбцов данных) и X_nup.xlsx (с данными нашивок, состоящий из 1040 строк и 3 столбцов данных).

Для разработки моделей по прогнозу модуля упругости при растяжении, прочности при растяжении и соотношения матрица-наполнитель нужно объединить 2 файла. Объединение по типу INNER, поэтому часть информации (17 строк таблицы X_nup.xlsx) не имеет соответствующих строк в таблице X_bp.xlsx и будет удалена.

Описание признаков объединенного датасета приведено в таблице 1. Все признаки имеют тип float64, то есть вещественный. Пропусков в данных нет. Все признаки, кроме «Угол нашивки», являются непрерывными, количественными. «Угол нашивки» принимает только два значения и будет рассматриваться как категориальный признак.

Таблица 1 — Описание признаков датасета

Название	Файл	Тип данных	Непустых значений	Уникальных значений
Соотношение матрица-наполнител	X_bp	float64	1023	1014
Плотность, кг/м3	X_bp	float64	1023	1013
модуль упругости, ГПа	X_bp	float64	1023	1020
Количество отвердителя, м.%	X_bp	float64	1023	1005
Содержание эпоксидных групп,%_2	X_bp	float64	1023	1004

1	2	3	4	5
Температура вспышки, С_2	X_bp	float64	1023	1003
Поверхностная плотность, г/м2	X_bp	float64	1023	1004
Модуль упругости при растяжении, ГПа	X_bp	float64	1023	1004
Прочность при растяжении, МПа	X_bp	float64	1023	1004
Потребление смолы, г/м2	X_bp	float64	1023	1003
Угол нашивки, град	X_nup	float64	1023	2
Шаг нашивки	X_nup	float64	1023	989
Плотность нашивки	X_nup	float64	1023	988

Также необходимо провести разведочный анализ данных, нарисовать гистограммы распределения каждой из переменной, диаграммы boxplot (ящик с усами), попарные графики рассеяния точек.

Для каждой колонки получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков; сделать предобработку: удалить шумы и выбросы, сделать нормализацию и стандартизацию.

Обучить несколько моделей для прогноза модуля упругости при растяжении и прочности при растяжении. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель. Разработать приложение с графическим интерфейсом, которое будет выдавать прогноз соотношения матрица-наполнитель. Оценить точность модели на тренировочном и тестовом датасете. Создать репозиторий в GitHub и разместить код исследования. Оформить файл README.

1.2 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, поэтому для наилучшего решения были исследованы (и некоторые из них применены) следующие методы:

1. линейная регрессия (Linear regression);
2. лассо регрессия (Lasso);
3. гребневая регрессия (Ridge);
4. эластичная регрессия (ElasticNet);
5. К-ближайших соседей (KNeighborsRegressor);
6. дерево решений (DecisionTreeRegressor);
7. случайный лес (RandomForest);
8. градиентный бустинг (GradientBoostingRegressor);
9. градиентный бустинг (AdaBoostRegressor);
10. стохастический градиентный спуск (SGDRegressor);
11. метод опорных векторов (Support Vector Regression);
12. многослойный перцептрон.

1. Линейная регрессия (Linear regression) — это алгоритм машинного обучения, основанный на контролируемом обучении, рассматривающий зависимость между одной входной и выходными переменными. Это один из самых простых и эффективных инструментов статистического моделирования. Она определяет зависимость переменных с помощью линии наилучшего соответствия. Модель регрессии создаёт несколько метрик. R2, или коэффициент детерминации, позволяет измерить, насколько модель может объяснить дисперсию данных. Если R-квадрат равен 1, это значит, что модель описывает все данные. Если же R-квадрат равен 0,5, модель объясняет лишь 50 процентов дисперсии данных. Оставшиеся отклонения не имеют объяснения. Чем ближе R2 к единице, тем лучше.

Достоинства метода: быстр и прост в реализации; легко интерпретируем, имеет меньшую сложность по сравнению с другими алгоритмами.

Недостатки метода: моделирует только прямые линейные зависимости; требует прямую связь между зависимыми и независимыми переменными; выбросы оказывают огромное влияние, а границы линейны.

Чтобы улучшить Линейную модель путем обмена некоторой этой дисперсии с предвзятостью, чтобы уменьшить нашу общую ошибку. Это происходит при помощи регуляризации, в которой модифицируется функция стоимости, чтобы ограничить значения коэффициентов. Это позволяет изменить чрезмерную дисперсию на некоторое смещение, потенциально уменьшая общую ошибку.

2. Лассо регрессия (Lasso) – это линейная модель, которая оценивает разреженные коэффициенты. Это простой метод, позволяющий уменьшить сложность модели и предотвратить переопределение, которое может возникнуть в результате простой линейной регрессии. Данный метод вводит дополнительное слагаемое регуляризации в оптимизацию модели. Это даёт более устойчивое решение. В регрессии лассо добавляется условие смещения в функцию оптимизации для того, чтобы уменьшить коллинеарность и, следовательно, дисперсию модели. Но вместо квадратичного смещения, используется смещение абсолютного значения. Лассо регрессия хорошо прогнозирует модели временных рядов на основе регрессии, таким как авторегрессии.

Достоинства метода: легко полностью избавляется от шумов в данных; быстро работает; не очень энергоёмко; способно полностью убрать признак из датасета; доступно обнуляет значения коэффициентов.

Недостатки метода: часто страдает качество прогнозирования; выдаёт ложное срабатывание результата; случайным образом выбирает одну из коллинеарных переменных; не оценивает правильность формы взаимосвязи

между независимой и зависимой переменными; не всегда лучше, чем пошаговая регрессия.

Лассо-регрессию следует использовать, когда есть несколько характеристик с высокой предсказательной способностью, а остальные бесполезны. Она обнуляет бесполезные характеристики и оставляет только подмножество переменных.

3. Гребневая регрессия (Ridge) – это регрессия, которая добавляет дополнительный штраф к функции стоимости, но вместо этого суммирует квадраты значений коэффициентов (норма L-2) и умножает их на некоторую постоянную лямбду. По сравнению с Лассо этот штраф регуляризации уменьшит значения коэффициентов, но не сможет принудительно установить коэффициент равным 0. Это ограничивает использование регрессии гребня в отношении выбора признаков. Однако, когда p> n, он способен выбрать более n релевантных предикторов, если необходимо, в отличие от Лассо. Он также выберет группы коллинеарных элементов, которые его изобретатели назвали «эффектом группировки».

Как и в случае с Лассо, мы можем варьировать лямбду, чтобы получить модели с различными уровнями регуляризации, где лямбда = 0 соответствует OLS, а лямбда приближается к бесконечности, что соответствует постоянной функции.

Анализ регрессии Лассо, как и Риджа показывают, что ни один метод не лучше, чем другой; нужно попробовать оба метода, чтобы определить, какой использовать.

Ридж-регрессию лучше применять, когда предсказательная способность набора данных распределена между различными характеристиками. Ридж-регрессия не обнуляет характеристики, которые могут быть полезны при составлении прогнозов, а просто уменьшает вес большинства переменных в модели.

4. Эластичная сеть (ElasticNet) – это регрессия, которая включает в себя термины регуляризации как L-1, так и L-2. Это дает преимущества регрессии Лассо и Риджа. 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 Ridge.

Elastic Net поставляется с дополнительными издержками на определение двух лямбда-значений для оптимальных решений.

Компромисс смещения дисперсии — это компромисс между сложной и простой моделью, в которой промежуточная сложность, вероятно, является наилучшей.

Лассо, Ридж-регрессия и Эластичная сеть — это модификации обычной линейной регрессии наименьших квадратов, которые используют дополнительные штрафные члены в функции стоимости, чтобы сохранить значения коэффициента небольшими и упростить модель.

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

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

Elastic Net сочетает в себе регрессию Лассо и Риджа, что потенциально приводит к модели, которая является простой и прогнозирующей.

5. Метод ближайших соседей - К-ближайших соседей (kNN - k Nearest Neighbours) ищет ближайшие объекты с известными значениями целевой переменной и основывается на хранении данных в памяти для сравнения с новыми элементами. Алгоритм находит расстояния между запросом и всеми примерами в данных, выбирая определенное количество примеров (k), наиболее близких к запросу, затем голосует за наиболее часто встречающуюся метку (в случае задачи классификации) или усредняет метки (в случае задачи регрессии).

Достоинства метода: прост в реализации и понимании полученных результатов; имеет низкую чувствительность к выбросам; не требует построения модели; допускает настройку нескольких параметров; позволяет делать дополнительные допущения; универсален; находит лучшее решение из возможных; решает задачи небольшой размерности.

Недостатки метода: замедляется с ростом объёма данных; не создаёт правил; не обобщает предыдущий опыт; основывается на всем массиве доступных исторических данных; невозможно сказать, на каком основании строятся ответы; сложно выбрать близость метрики; имеет высокую зависимость результатов классификации от выбранной метрики; полностью перебирает всю обучающую выборку при распознавании; имеет вычислительную трудоемкость.

6. Дерево решений (DecisionTreeRegressor) – метод автоматического анализа больших массивов данных. Это инструмент принятия решений, в котором используется древовидная структура, подобная блок-схеме, или модель решений и всех их возможных результатов, включая результаты, затраты и полезность. Дерево принятия решений - эффективный инструмент интеллектуального анализа данных и предсказательной аналитики. Алгоритм дерева решений подпадает под категорию контролируемых алгоритмов обучения. Он работает как для непрерывных, так и для категориальных выходных переменных. Правила генерируются за счёт обобщения множества отдельных наблюдений (обучающих примеров), описывающих предметную область. Регрессия дерева решений отслеживает особенности объекта и обучает модель в структуре дерева прогнозированию данных в будущем для получения значимого непрерывного вывода. Дерево решений один из вариантов решения регрессионной задачи, в случае если зависимость в данных не имеет очевидной корреляции.

Достоинства метода: помогают визуализировать процесс принятия решения и сделать правильный выбор в ситуациях, когда результаты одного

решения влияют на результаты следующих решений, создаются по понятным правилам; просты в применении и интерпретации; заполняют пропуски в данных наиболее вероятным решением; работают с разными переменными; выделяют наиболее важные поля для прогнозирования.

Недостатки метода: ошибаются при классификации с большим количеством классов и небольшой обучающей выборкой; имеют нестабильный процесс (изменение в одном узле может привести к построению совсем другого дерева); имеет затратные вычисления; необходимо обращать внимание на размер; ограниченное число вариантов решения проблемы.

7. Случайный лес (RandomForest) — это множество решающих деревьев. Универсальный алгоритм машинного обучения с учителем, представитель ансамблевых методов. Если точность дерева решений оказалось недостаточной, мы можем множество моделей собрать вместе.

Достоинства метода: не переобучается; не требует предобработки входных данных; эффективно обрабатывает пропущенные данные, данные с большим числом классов и признаков; имеет высокую точность предсказания и внутреннюю оценку обобщающей способности модели, а также высокую параллелизуемость и масштабируемость.

Недостатки метода: построение занимает много времени; сложно интерпретируемый; не обладает возможностью экстраполяции; может недообучаться; трудоёмко прогнозируемый; иногда работает хуже, чем линейные методы.

8. Градиентный бустинг (Gradient Boosting) — это ансамбль деревьев решений, обученный с использованием градиентного бустинга. В основе данного алгоритма лежит итеративное обучение деревьев решений с целью минимизировать функцию потерь. Основная идея градиентного бустинга: строятся последовательно несколько базовых классификаторов, каждый из которых как можно лучше компенсирует недостатки предыдущих. Финальный классификатор является линейной композицией этих базовых классификаторов.

Достоинства метода: новые алгоритмы учатся на ошибках предыдущих; требуется меньше итераций, чтобы приблизиться к фактическим прогнозам; наблюдения выбираются на основе ошибки; прост в настройке темпа обучения и применения; легко интерпретируем.

Недостатки метода: необходимо тщательно выбирать критерии остановки, или это может привести к переобучению, наблюдения с наибольшей ошибкой появляются чаще; слабее и менее гибок чем нейронные сети.

9. Градиентный бустинг (AdaBoost) – это алгоритм, который работает по принципу перевзвешивания результатов. Есть деревья решений, а ансамбль из них — это градиентный бустинг, задача решается с помощью градиентного спуска. Алгоритм AdaBoost учится на ошибках, больше концентрируясь на сложных участках, с которыми он столкнулся в процессе предыдущей итерации обучения. На каждой итерации дается вес алгоритмам. Каждый новый алгоритм корректирует ошибки предыдущих до получения хорошего результата. Все прогнозы объединяются с помощью голосования для получения окончательного прогноза.

Достоинства метода: AdaBoost легко реализовать, достаточно класса моделей и их количества. Он итеративно исправляет ошибки слабого классификатора и повышает точность путем объединения слабых учащихся. Можно использовать многие базовые классификаторы с AdaBoost. Не склонен к переоснащению.

Недостатки метода: AdaBoost чувствителен к шумным данным. AdaBoost обучается дольше линейной регрессии, классификация дольше чем при использовании логистической регрессии. На AdaBoost сильно влияют отклонения, так как он пытается идеально подогнать каждую точку. AdaBoost работает медленнее и чуть хуже, чем XGBoost. Но легче в понимании.

10. Стохастический градиентный спуск (SGDRegressor) — это простой, но очень эффективный подход к подгонке линейных классификаторов и регрессоров под выпуклые функции потерь. Этот подход подразумевает

корректировку весов нейронной сети, используя аппроксимацию градиента функционала, вычисленную только на одном случайном обучающем примере из выборки.

Достоинства метода: эффективен; прост в реализации; имеет множество возможностей для настройки кода; способен обучаться на избыточно больших выборках.

Недостатки метода: требует ряд гиперпараметров; чувствителен к масштабированию функций; может не сходиться или сходиться слишком медленно; функционал многоэкстремален; процесс может «застрять» в одном из локальных минимумов; возможно переобучение.

11. Метод опорных векторов (Support Vector Regression) – этот бинарный линейный классификатор был выбран, потому что он хорошо работает на небольших датасетах. Данный алгоритм – это алгоритм обучения с учителем, использующихся для задач классификации и регрессионного анализа, это контролируемое обучение моделей с использованием схожих алгоритмов для анализа данных и распознавания шаблонов. Учитывая обучающую выборку, где алгоритм помечает каждый объект, как принадлежащий к одной из двух категорий, строит модель, которая определяет новые наблюдения в одну из Категорий. Модель метода опорных векторов – отображение данных точками в пространстве, так что между наблюдениями отдельных категорий имеется разрыв.

Каждый объект данных представляется как вектор (точка) в p-мерном пространстве. Он создаёт линию или гиперплоскость, которая разделяет данные на классы.

Достоинства метода: для классификации достаточно небольшого набора данных. При правильной работе модели, построенной на тестовом множестве, вполне возможно применение данного метода на реальных данных. Эффективен при большом количестве гиперпараметров. Способен обрабатывать случаи, когда гиперпараметров больше, чем количество

наблюдений. Существует возможность гибко настраивать разделяющую функцию. Алгоритм максимизирует разделяющую полосу, которая, как подушка безопасности, позволяет уменьшить количество ошибок классификации.

Недостатки метода: неустойчивость к шуму, поэтому в работе была проведена тщательнейшая работа с выбросами, иначе в обучающих данных шумы становятся опорными объектами-нарушителями и напрямую влияют на построение разделяющей гиперплоскости; для больших наборов данных требуется долгое время обучения; достаточно сложно подбирать полезные преобразования данных; параметры модели сложно интерпретировать, поэтому были рассмотрены и другие методы.

12. Многослойный персептрон (MLPRegressor) — это алгоритм обучения с учителем, который изучает функцию $f(\cdot):Rm \rightarrow Ro$ обучением на наборе данных, где m — количество измерений для ввода и o — количество размеров для вывода.

Это искусственная нейронная сеть, имеющая 3 или более слоёв персептронов. Эти слои - один входной слой, 1 или более скрытых слоёв и один выходной слой персептронов.

Достоинства метода: построение сложных разделяющих поверхностей; возможность осуществления любого отображения входных векторов в выходные; легко обобщает входные данные; не требует распределения входных векторов; изучает нелинейные модели.

Недостатки метода: имеет невыпуклую функцию потерь; разные инициализации случайных весов могут привести к разной точности проверки; требует настройки ряда гиперпараметров; чувствителен к масштабированию функций.

Используемые метрики качества моделей: R2, RMSE, MAE, MAPE.

R2 (коэффициент детерминации) измеряет долю дисперсии, объяснённую моделью, в общей дисперсии целевой переменной. Если он близок к единице,

то модель хорошо объясняет данные, если же он близок к нулю, то качество прогноза идентично средней величине целевой переменной (т.е. очень низкое). Отрицательные значение коэффициента детерминации означают плохую объясняющую способность модели.

RMSE (среднеквадратическая ошибка) – 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 (средняя абсолютная ошибка) – метрика, которая сообщает нам среднюю абсолютную разницу между прогнозируемыми значениями и фактическими значениями в наборе данных. Чем ниже MAE, тем лучше модель соответствует набору данных.

MAPE (средняя абсолютная ошибка в процентах) – это весовой MAE. Чем ниже показатель MAPE тем лучше. За счёт того, что показатель измеряется в процентах его можно легко складывать по разным рядам. Можно даже рассчитать MAPE и изучить его распределение, используя инструменты статистического анализа.

1.3 Разведочный анализ данных

Прежде чем передать данные в работу моделей машинного обучения, необходимо обработать и очистить их. Необработанные данные могут содержать искажения и пропущенные значения и способны привести к неверным результатам по итогам моделирования. Но безосновательно удалять что-либо тоже неправильно. Именно поэтому сначала набор данных надо изучить.

	count	mean	std	min	25%	50%	75%	max	median
Соотношение матрица-наполнитель	1023.0000	2.9304	0.9132	0.3894	2.3179	2.9069	3.5527	5.5917	2.9069
Плотность, кг/м3	1023.0000	1975.7349	73.7292	1731.7646	1924.1555	1977.6217	2021.3744	2207.7735	1977.6217
модуль упругости, ГПа	1023.0000	739.9232	330.2316	2.4369	500.0475	739.6643	961.8125	1911.5365	739.6643
Количество отвердителя, м.%	1023.0000	110.5708	28.2959	17.7403	92.4435	110.5648	129.7304	198.9532	110.5648
Содержание эпоксидных групп,%_2	1023.0000	22.2444	2.4063	14.2550	20.6080	22.2307	23.9619	33.0000	22.2307
Температура вспышки, С_2	1023.0000	285.8822	40.9433	100.0000	259.0665	285.8968	313.0021	413.2734	285.8968
Поверхностная плотность, г/м2	1023.0000	482.7318	281.3147	0.6037	266.8166	451.8644	693.2250	1399.5424	451.8644
Модуль упругости при растяжении, ГПа	1023.0000	73.3286	3.1190	64.0541	71.2450	73.2688	75.3566	82.6821	73.2688
Прочность при растяжении, МПа	1023.0000	2466.9228	485.6280	1036.8566	2135.8504	2459.5245	2767.1931	3848.4367	2459.5245
Потребление смолы, г/м2	1023.0000	218.4231	59.7359	33.8030	179.6275	219.1989	257.4817	414.5906	219.1989
Угол нашивки, град	1023.0000	44.2522	45.0158	0.0000	0.0000	0.0000	90.0000	90.0000	0.0000
Шаг нашивки	1023.0000	6.8992	2.5635	0.0000	5.0800	6.9161	8.5863	14.4405	6.9161
Плотность нашивки	1023.0000	57.1539	12.3510	0.0000	49.7992	57.3419	64.9450	103.9889	57.3419

Рисунок 1 - Описательная статистика датасета

Цель разведочного анализа - получение первоначальных представлений о характерах распределений переменных исходного набора данных, формирование оценки качества исходных данных (наличие выбросов), выявление характера взаимосвязи между переменными с целью последующего выдвижения гипотез о наиболее подходящих для решения задачи моделях машинного обучения.

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

Гистограммы распределения переменных и диаграммы «ящик с усами» приведены на рисунках 2-4. По ним видно, что все признаки, кроме «Угол нашивки», имеют нормальное распределение и принимают неотрицательные значения. «Угол нашивки» принимает значения: 0, 90.

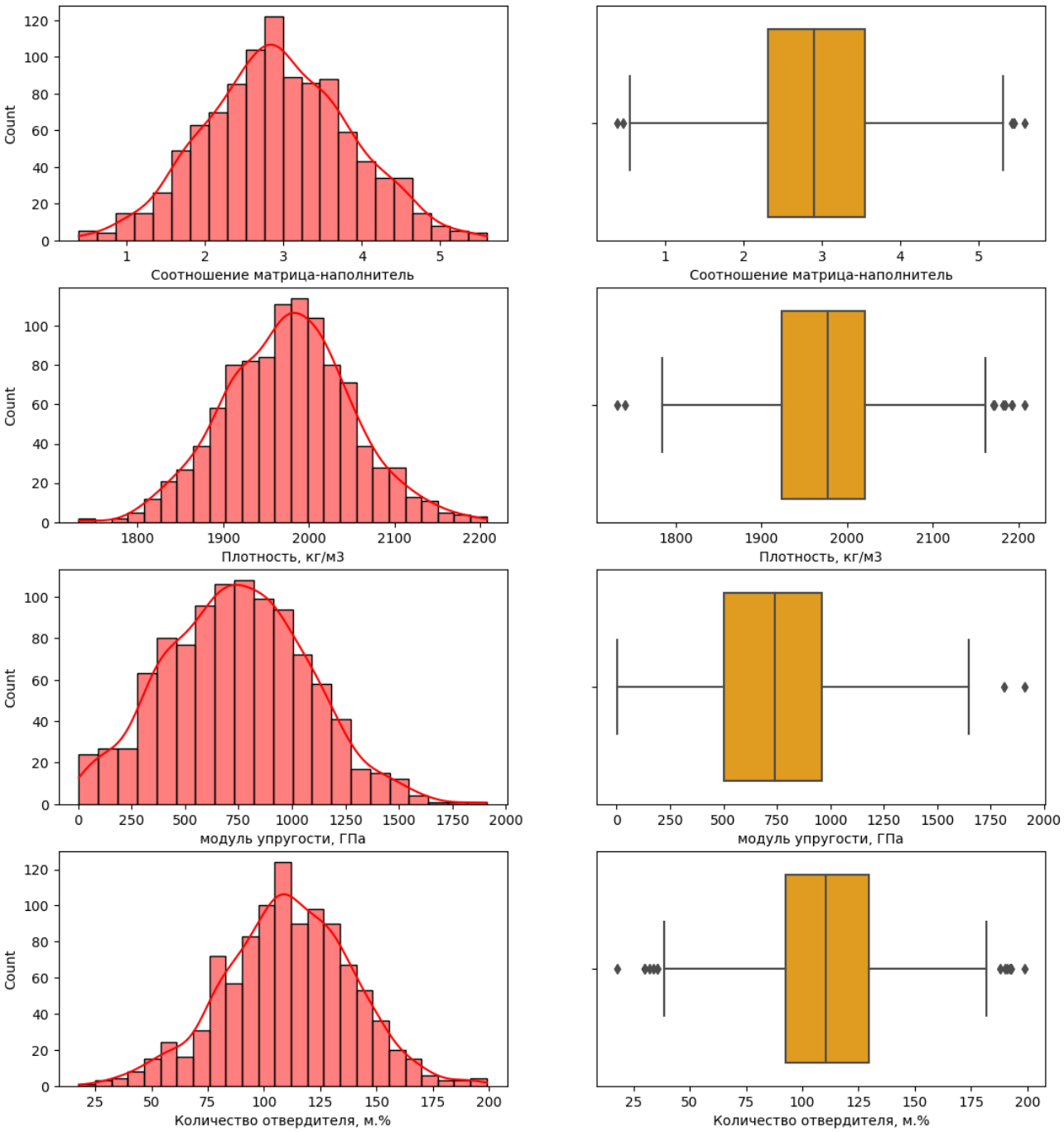

Рисунок 2 - Гистограммы распределения переменных и диаграммы «ящик с усами»

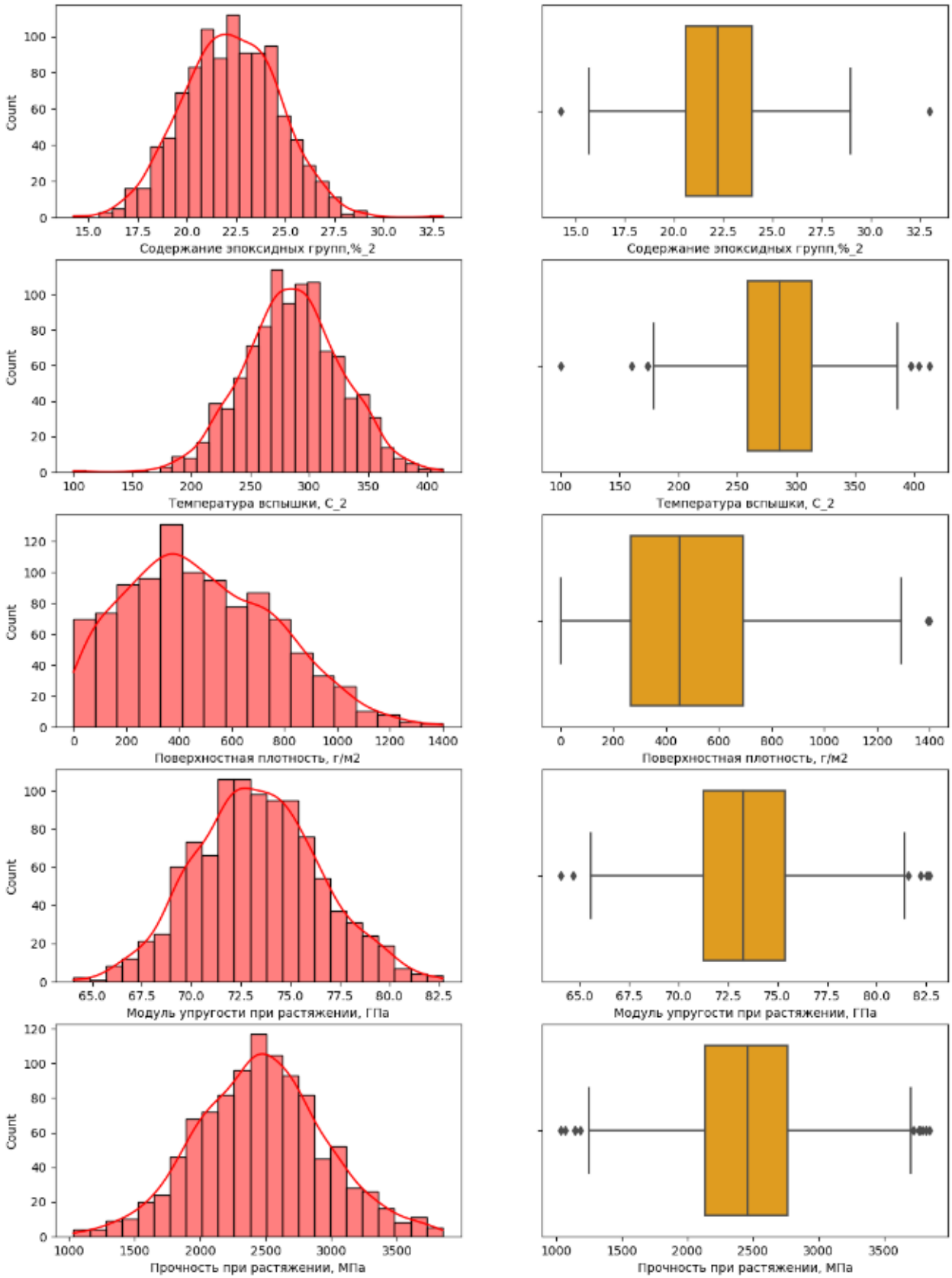

Рисунок 3 - Гистограммы распределения переменных и диаграммы «ящик с усами»

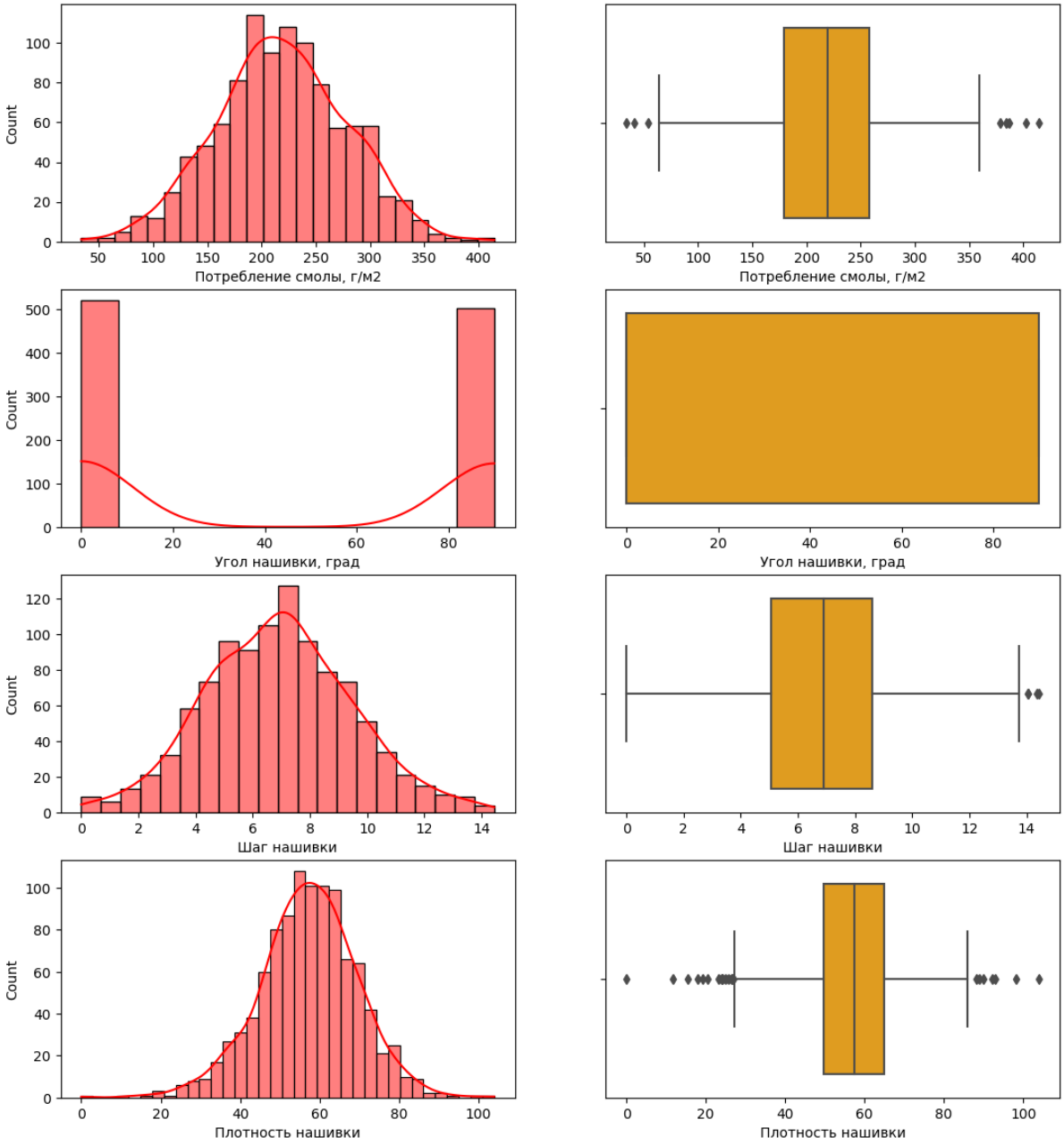

Рисунок 4 - Гистограммы распределения переменных
и диаграммы «ящик с усами»

Попарные графики рассеяния точек приведены на рисунке 5. По графикам рассеяния мы видим, что некоторые точки отстоят далеко от общего облака. Так визуально выглядят выбросы — аномальные, некорректные значения данных, выходящие за пределы допустимых значений признака.

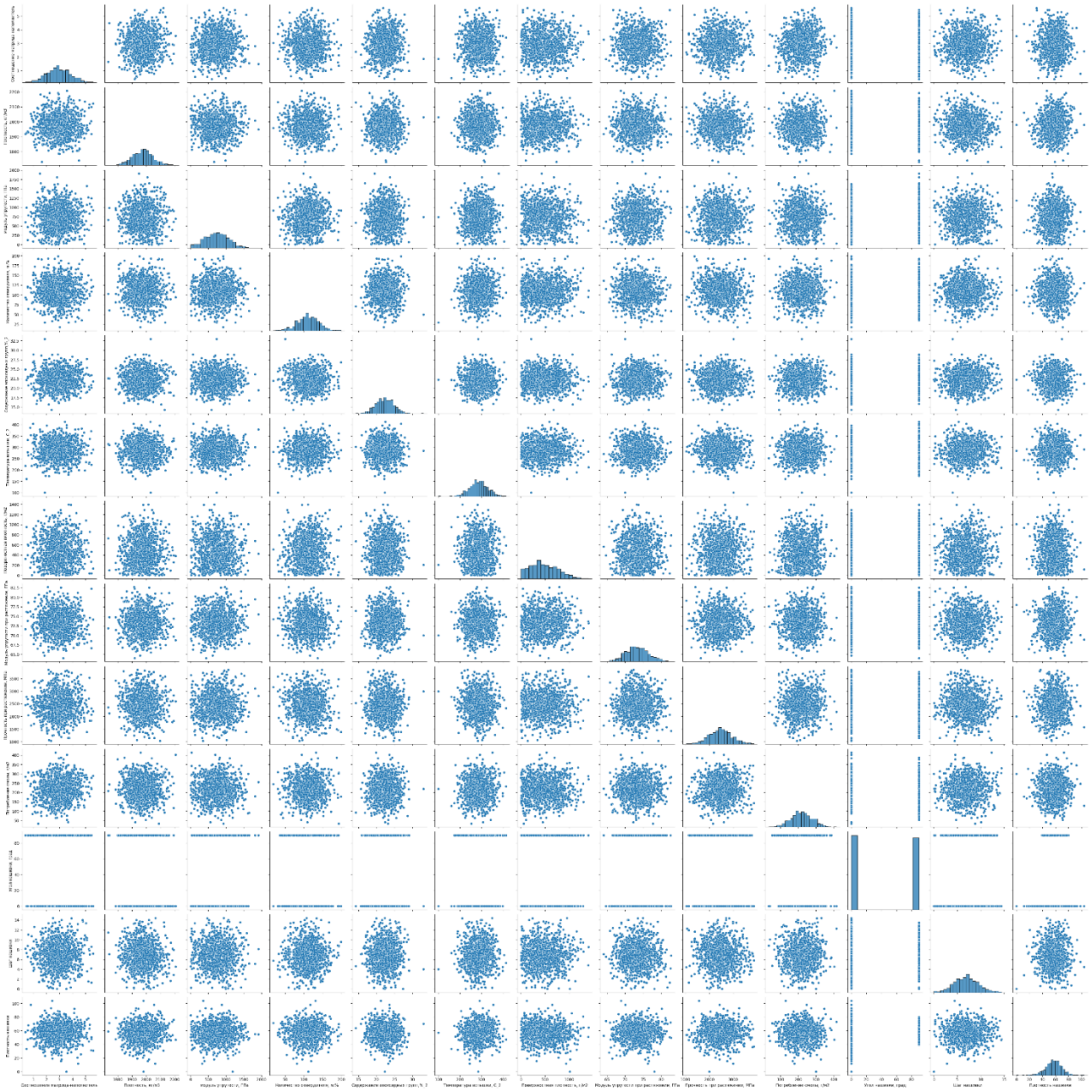

Рисунок 5 - Попарные графики рассеяния точек

Есть следующие методы выявления выбросов для признаков с нормальным распределением: метод 3-х сигм; метод межквартильных расстояний.

Применив эти методы на нашем датасете, было найдено:

– методом 3-х сигм — 24 выброса;

– методом межквартильных расстояний — 93 выброса.

Пример выбросов на гистограмме распределения и диаграмме «ящик с усами» приведен на рисунке 6.

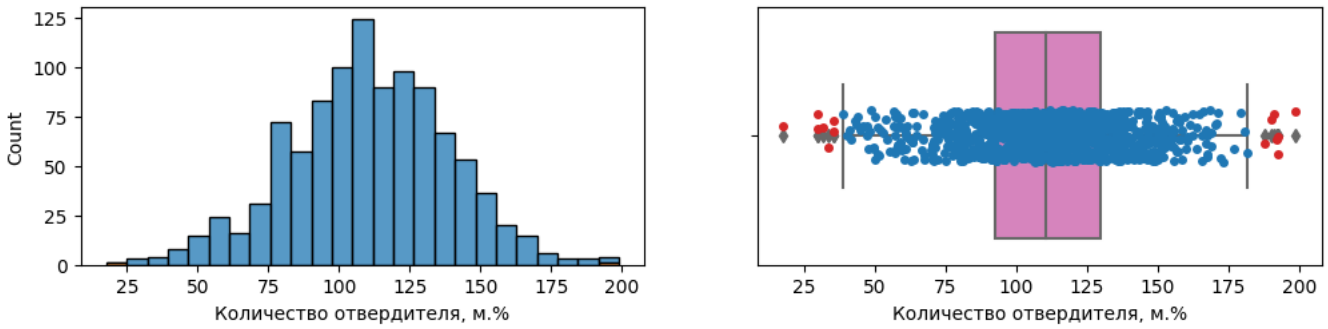

Рисунок 6 – Пример выбросов

В данном случае удалим выбросы способом межквартильного расстояния для максимальной чистоты, так как используем методы чувствительные к выбросам. Данные очищались 3 раза до полного исчезновения выбросов. После этого в датасете осталось 922 строки и 13 признаков-переменных.

На рисунке 5 мы видели график попарного рассеяния точек. По форме «облаков точек» мы не заметили зависимостей, которые станут основой работы моделей. Помочь выявить связь между признаками может матрица корреляции, приведенная на рисунке 7.

Тепловая карта показывает практически отсутствие корреляции между признаками и целевыми переменными.

Максимальная корреляция между плотностью нашивки и углом нашивки 0.11, значит нет зависимости между этими данными. Корреляция между всеми параметрами очень близка к 0, корреляционные связи между переменными не наблюдаются.

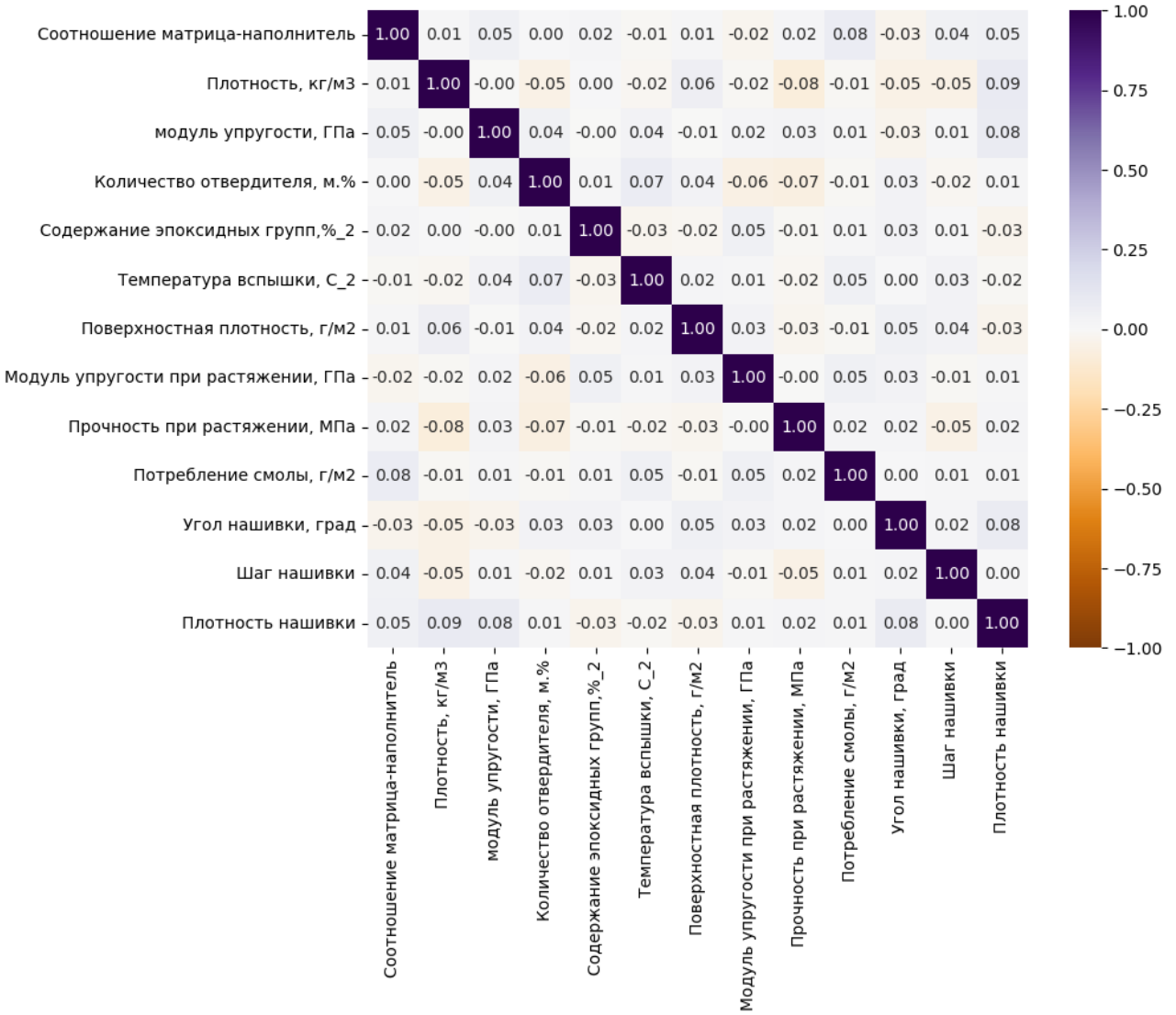

Рисунок 7 - Тепловая карта с корреляцией данных

2. Практическая часть

2.1 Разбиение и предобработка данных

Для прогнозирования модуля упругости при растяжении признаки датасета были разделены на входные и выходные, а строки - на тренировочное и тестовое множество. Размерности полученных наборов данных показаны на рисунке 8. Описательная статистика входных признаков до и после предобработки показана на рисунке 9. Описательная статистика выходного признака показана на рисунке 10.

```
x1_train: (645, 11) y1_train: (645, 1)
x1_test: (277, 11) y1_test: (277, 1)
```

Рисунок 8 - Размерности тренировочного и тестового множеств после разбиения для 1-й задачи

```
# Описательная статистика входных данных до предобработки
show_statistics(x1_train_raw)
```

	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
min	0.547391	1784.482245	2.436909	40.304806	15.695894	188.918674	1.894093	72.530873	0.000000	0.037639	28.661632
max	5.314144	2161.565216	1628.000000	179.645962	28.955094	385.697799	1288.691844	359.052220	90.000000	13.571921	86.012427
mean	2.925725	1973.514328	735.549446	111.418752	22.277356	286.441169	483.129553	217.871279	44.930233	6.948415	57.848569
std	0.906983	71.158440	324.420003	26.337565	2.378254	37.919237	282.682289	56.915998	45.034870	2.500522	11.048670

```
# Описательная статистика входных данных после предобработки
show_statistics(pd.DataFrame(x1_train, columns=(x1_continuous + x_categorical)))
```

	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	Шаг нашивки	Плотность нашивки	Угол нашивки, град
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
mean	0.498942	0.501301	0.450990	0.510359	0.496369	0.495594	0.373979	0.507259	0.510613	0.508919	0.499225
std	0.190273	0.188708	0.199574	0.189015	0.179366	0.192699	0.219679	0.198645	0.184755	0.192651	0.500387

Рисунок 9 – Описательная статистика входных признаков до и после предобработки для 1-й задачи

```
# Описательная статистика выходной переменной
show_statistics(pd.DataFrame(y1_train, columns=[
```

	Модуль упругости при растяжении, ГПа
min	65.979990
max	81.203147
mean	73.342384
std	3.027444

Рисунок 10 - Описательная статистика выходного признака для 1-й задачи

Для прогнозирования прочности при растяжении признаки датасета были разделены на входные и выходные, а строки - на тренировочное и тестовое множество. Размерности полученных наборов данных показаны на рисунке 11. Описательная статистика входных признаков до и после предобработки показана на рисунке 12. Описательная стати-стика выходного признака показана на рисунке 13.

```
x2_train: (645, 11) y2_train: (645, 1)
x2_test: (277, 11) y2_test: (277, 1)
```

Рисунок 11 – Размерности тренировочного и тестового множеств после разбиения для 2-й задачи

```
# Описательная статистика входных данных до предобработки
show_statistics(x2_train_raw)
```

	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
min	0.547391	1784.482245	2.436909	40.304806	15.695894	188.918674	1.894093	72.530873	0.000000	0.037639	28.661632
max	5.314144	2161.565216	1628.000000	179.645962	28.955094	385.697799	1288.691844	359.052220	90.000000	13.571921	86.012427
mean	2.925725	1973.514328	735.549446	111.418752	22.277356	286.441169	483.129553	217.871279	44.930233	6.948415	57.848569
std	0.906983	71.158440	324.420003	26.337565	2.378254	37.919237	282.682289	56.915998	45.034870	2.500522	11.048670

```
# Описательная статистика входных данных после предобработки
show_statistics(pd.DataFrame(x2_train, columns=(x2_continuous + x_categorical)))
```

	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	Шаг нашивки	Плотность нашивки	Угол нашивки, град
min	-0.044392	-0.048759	-0.013532	0.011430	-0.027459	0.046176	0.001000	-0.036906	-0.017166	-0.065695	0.000000
max	1.004025	1.004371	1.025125	0.984755	1.003704	0.998209	0.997948	1.015031	0.988080	1.000469	1.000000
mean	0.478708	0.479177	0.454892	0.508175	0.484379	0.517998	0.373838	0.496698	0.496125	0.476897	0.499225
std	0.199485	0.198734	0.207289	0.183973	0.184956	0.183456	0.219009	0.208962	0.185724	0.205397	0.500387

Рисунок 12 – Описательная статистика входных признаков до и после предобработки для 2-й задачи

```
# Описательная статистика выходной переменной
show_statistics(pd.DataFrame(y2_train, columns
```

	Прочность при растяжении, МПа
min	1250.392802
max	3636.892992
mean	2466.696221
std	459.451353

Рисунок 13 – Описательная статистика выходного признака для 2-й задачи

Для прогнозирования соотношения матрица-наполнитель признаки датасета были разделены на входные и выходные, а строки - на тренировочное и тестовое множество. Размерности полученных наборов данных показаны на рисунке 14. Описательная статистика входных признаков до и после предобработки показана на рисунке 15. Описательная стати-стика выходного признака показана на рисунке 16.

```
x3_train: (645, 12) y3_train: (645, 1)
x3_test: (277, 12) y3_test: (277, 1)
```

Рисунок 14 – Размерности тренировочного и тестового множеств после разбиения для 3-й задачи

```
# Описательная статистика входных данных до предобработки
show_statistics(x3_train_raw)
```

	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
min	1784.482245	2.436909	40.304806	15.695894	188.918674	1.894093	65.979990	1250.392802	72.530873	0.000000	0.037639
max	2161.565216	1628.000000	179.645962	28.955094	385.697799	1288.691844	81.203147	3636.892992	359.052220	90.000000	13.571921
mean	1973.514328	735.549446	111.418752	22.277356	286.441169	483.129553	73.342384	2466.696221	217.871279	44.930233	6.948415
std	71.158440	324.420003	26.337565	2.378254	37.919237	282.682289	3.027444	459.451353	56.915998	45.034870	2.500522

```
# Описательная статистика входных данных после предобработки
show_statistics(pd.DataFrame(x3_train, columns=(x3_continuous + x_categorical)))
```

	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	Шаг нашивки	Плотность нашивки
min	-0.048759	-0.013532	0.011430	-0.027459	0.046176	0.001000	0.012265	-0.061113	-0.036906	-0.017166	-0.065695
max	1.004371	1.025125	0.984755	1.003704	0.998209	0.997948	1.015299	0.992257	1.015031	0.988080	1.000469
mean	0.479177	0.454892	0.508175	0.484379	0.517998	0.373838	0.497363	0.475748	0.496698	0.496125	0.476897
std	0.198734	0.207289	0.183973	0.184956	0.183456	0.219009	0.199474	0.202796	0.208962	0.185724	0.205397

Рисунок 15 – Описательная статистика входных признаков до и после предобработки для 3-й задачи

```
# Описательная статистика выходной переменной
show_statistics(pd.DataFrame(y3_train, columns=
```

	Соотношение матрица-наполнитель
min	0.547391
max	5.314144
mean	2.925725
std	0.906983

Рисунок 16 - Описательная статистика выходного признака для 3-й задачи

2.2 Разработка и обучение моделей для прогнозирования модуля упругости при растяжении

Для подбора лучшей модели для этой задачи я взяла следующие модели:

- LinearRegression — линейная регрессия;
- Ridge — гребневая регрессия;
- Lasso — лассо-регрессия;

- SVR — метод опорных векторов;
- KneighborsRegressor — метод ближайших соседей;
- DecisionTreeRegressor — деревья решений;
- RandomForestRegressor — случайный лес.

В качестве базовой модели взят DummyRegressor, возвращающий среднее значение целевого признака.

Метрики работы выбранных моделей с гиперпараметрами по умолчанию, полученные с помощью перекрестной проверки на тестовом множестве, приведены на рисунке 17.

Ни одна из выбранных мной моделей не оказалась подходящей для наших данных.

Коэффициент детерминации R2 близок к 0 для линейных моделей и метода опорных векторов. Значит, они не лучше базовой модели. И остальные метрики у них примерно совпадают с базовой моделью.

Гораздо хуже линейных моделей с гиперпараметрами по умолчанию отработали метод ближайших соседей и деревья решений.

Случайный лес отработал лучше, чем одно дерево решений, но хуже, чем линейные модели.

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.011479	-3.018468	-2.434764	-0.033263	-7.216742
LinearRegression	-0.022599	-3.034619	-2.453669	-0.033520	-7.222509
Ridge	-0.021050	-3.032391	-2.451847	-0.033495	-7.212702
Lasso	-0.011479	-3.018468	-2.434764	-0.033263	-7.216742
SVR	-0.050196	-3.073227	-2.486431	-0.033930	-7.562206
KNeighborsRegressor	-0.252309	-3.350693	-2.673319	-0.036522	-8.485262
DecisionTreeRegressor	-1.317465	-4.505732	-3.660577	-0.050026	-11.406496
RandomForestRegressor	-0.081437	-3.117793	-2.525298	-0.034499	-7.342254

Рисунок 17 – Результаты моделей с гиперпараметрами по умолчанию

	R2	RMSE	MAE	MAPE	max_error
Ridge(alpha=60, positive=True, solver='lbfgs')	-0.008194	-3.013781	-2.433792	-0.033249	-7.173646
Lasso(alpha=0.15)	-0.011479	-3.018468	-2.434764	-0.033263	-7.216742
SVR(C=0.03)	-0.009769	-3.016089	-2.434354	-0.033249	-7.220076
KNeighborsRegressor(n_neighbors=29)	-0.033495	-3.050098	-2.470094	-0.033784	-7.219134
DecisionTreeRegressor(max_depth=1, max_features=1, random_state=3128, splitter='random')	-0.011451	-3.018141	-2.426078	-0.033145	-7.182651
RandomForestRegressor(bootstrap=False, criterion='absolute_error', max_depth=2, max_features=1, n_estimators=50, random_state=3128)	-0.011578	-3.018652	-2.438736	-0.033310	-7.189810

Рисунок 18 – Результаты моделей после подбора гиперпараметров

После выполнения подбора гиперпарметров по сетке с перекрестной проверкой, получили метрики, приведенные на рисунке 18.

Можно сделать вывод, что, подбирая гиперпараметры, можно значительно улучшить предсказание выбранной модели.

Все модели крайне плохо описывают исходные данные - не удалось добиться положительного значения R2. Самая лучшая модель дает коэффициент детерминации близкий к нулю, что соответствует базовой модели.

Линейные модели совпадают с базовой моделью. Их характеристики улучшились, но не значительно.

Метод опорных векторов в процессе подбора гиперпараметры лучшим ядром выбрал линейное и отработал аналогично линейным моделям.

Метод ближайших соседей увеличением количества соседей радикально улучшил качество работы. Но его лучшие результаты все равно немного, но отстают от линейных моделей.

Деревья решений при кропотливом подборе параметров превзошли результат линейной модели. Но они не являются объясняющей зависимость моделью.

Собирая деревья в ансамбли, можно улучшать характеристики. Но подбор параметров для леса затруднен тем, что это затратный по времени процесс. По этой причине мне не удалось получить комбинацию параметров для леса, которая была был лучше дерева решений.

Поэтому в качестве лучшей модели выбираю модель Ridge. На рисунке 19 приведена визуализация работы лучшей модели на тестовом множестве.

Сложно визуализировать регрессию в многомерном пространстве. Но даже на таком графике мы видим, насколько не соответствует лучшая модель исходным данным и насколько она неудачна.

Метрики работы лучшей модели на тестовом множестве и сравнение с базовой отражены на рисунке 20. Они подтверждают: полученная модель хуже базовой. Результат исследования отрицательный. Не удалось получить модели, которая могла бы оказать помощь в принятии решений специалисту предметной области.

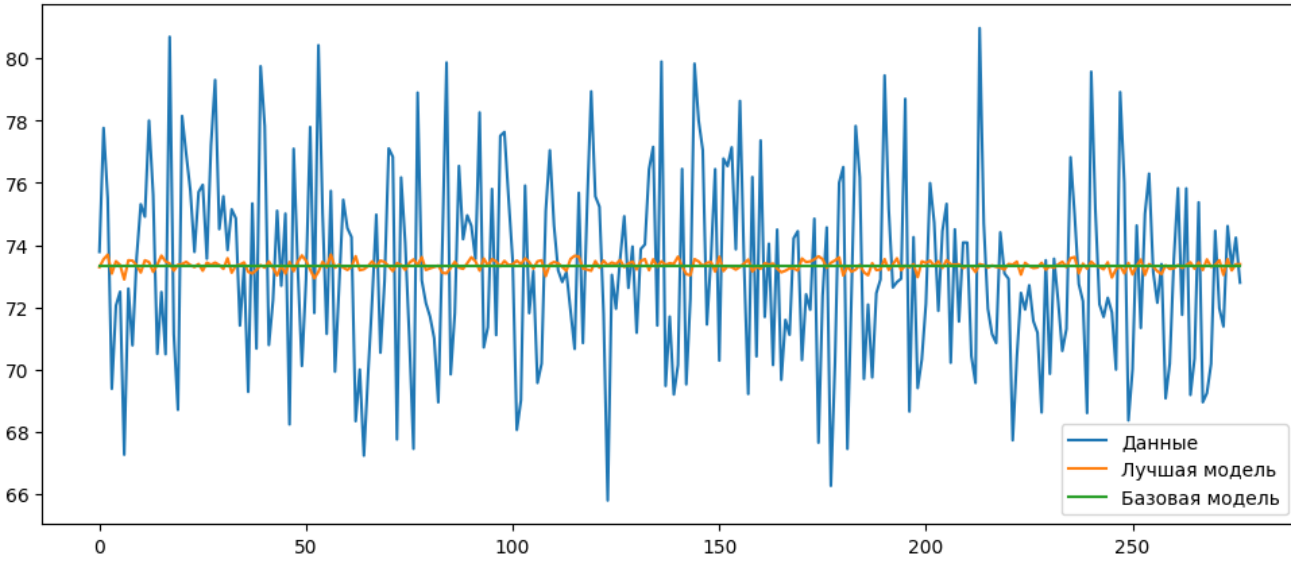

Рисунок 19 - Визуализация работы модели

	R2	RMSE	MAE	MAPE	max_error
Базовая модель	-0.001840	-3.023023	-2.480618	-0.033942	-7.628576
Лучшая модель (модель Ridge)	-0.007314	-3.031272	-2.486880	-0.034029	-7.581602

Рисунок 20 - Метрики работы лучшей модели на тестовом множестве

2.3 Разработка и обучение моделей для прогнозирования прочности при растяжении

Для подбора лучшей модели для этой задачи я взяла следующие модели:

- LinearRegression — линейная регрессия;

- Ridge — гребневая регрессия;
- Lasso — лассо-регрессия;
- SVR — метод опорных векторов;
- DecisionTreeRegressor — деревья решений;
- GradientBoostingRegressor — случайный лес.

В качестве базовой модели взят DummyRegressor, возвращающий среднее значение целевого признака.

Метрики работы выбранных моделей с гиперпараметрами по умолчанию, полученные с помощью перекрестной проверки на тестовом множестве, приведены на рисунке 21.

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.014754	-458.517408	-366.664655	-0.160630	-1095.981614
LinearRegression	-0.017429	-459.567129	-368.992333	-0.161357	-1121.299260
Ridge	-0.016246	-459.284035	-368.747797	-0.161270	-1119.398974
Lasso	-0.009446	-457.664727	-367.497218	-0.160752	-1112.987433
SVR	-0.012648	-458.069725	-366.556719	-0.160477	-1093.217827
DecisionTreeRegressor	-1.101336	-648.893181	-524.028712	-0.223497	-1648.797617
GradientBoostingRegressor	-0.121957	-482.055962	-389.345452	-0.169754	-1194.692003

Рисунок 21 — Результаты моделей с гиперпараметрами по умолчанию

Ни одна из выбранных мной моделей не соответствует данным. R2 близок к 0 для линейных моделей и метода опорных векторов. Lasso-регрессия показала результаты чуть лучше базовой модели.

Гораздо хуже линейных моделей с гиперпараметрами по умолчанию отработали деревья решений.

Градиентный бустинг с параметрами по умолчанию отработал лучше дерева. Он тоже соответствует базовой модели.

После выполнения подбора гиперпарметров по сетке с перекрестной проверкой, получили метрики, приведенные на рисунке 22.

	R2	RMSE	MAE	MAPE	max_error
Ridge(alpha=60, random_state=3128, solver='sag')	-0.012026	-458.047100	-366.642494	-0.160565	-1094.747442
Lasso(alpha=5)	-0.008467	-457.198647	-366.086794	-0.160291	-1090.943083
SVR(C=0.2)	-0.012996	-458.138246	-366.569324	-0.160528	-1093.961228
DecisionTreeRegressor(max_depth=1, max_features=3, random_state=3128)	-0.020410	-459.733570	-365.998753	-0.160581	-1107.058869
GradientBoostingRegressor(loss='absolute_error', max_depth=1, max_features=11, n_estimators=50, random_state=3128)	-0.022204	-460.206446	-368.817375	-0.161594	-1105.850304

Рисунок 22 — Результаты моделей после подбора гиперпараметров

Подбор гиперпараметров - интересный процесс. Но нам он не помог получить модель, превосходящую базовую. Все модели крайне плохо описывают исходные данные. Не удалось добиться коэффициента детерминации, больше нуля.

Линейные методы после подбора немного улучшили характеристики.

Метод опорных векторов отработал аналогично линейным моделям.

Деревья решений как и градиентный бустинг после подбора параметров улучшили результат по сравнению с параметрами по умолчанию.

Но лучший результат дает всё равно Lasso-регрессия.

На рисунке 23 приведена визуализация работы лучшей модели на тестовом множестве.

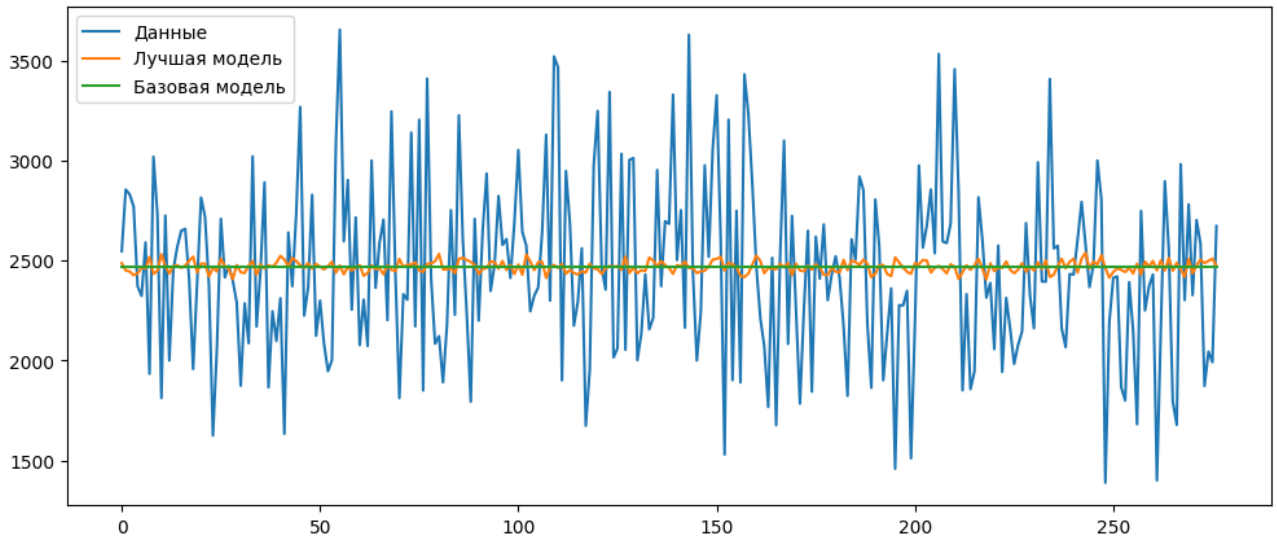

Рисунок 23 — Визуализация работы модели

Визуализируя результаты Lasso-регрессии с выбранными параметрами, мы видим, насколько они плохи и далеки от исходных данных. Примерно такую же картину мы видели при визуализации лучшей модели для модуля упругости при растяжении.

Метрики работы лучшей модели на тестовом множестве и сравнение с базовой отражены на рисунке 24. Мы видим, что на тестовом множестве Lasso-регрессия всё равно справилась хуже базовой, результат исследования отрицательный. Не удалось получить модели, которая могла бы оказать помощь в принятии решений специалисту предметной области.

	R2	RMSE	MAE	MAPE	max_error
Базовая модель	-0.001555	-439.676848	-350.354301	-0.151168	-1187.738138
Лучшая модель (модель Lasso)	-0.009342	-441.382723	-350.450799	-0.151207	-1179.461001

Рисунок 24 - Метрики работы лучшей модели на тестовом множестве

2.4 Разработка нейронной сети для прогнозирования соотношения матрица-наполнитель

По заданию для соотношения матрица-наполнитель необходимо построить нейросеть. Но для сравнения нам также понадобится базовая модель DummyRegressor, возвращающая среднее целевого признака.

1. MLPRegressor из библиотеки sklearn.

Строю нейронную сеть с помощью класса MLPRegressor следующей архитектуры:

- слоев: 8;
- нейронов на каждом слое: 24;
- активационная функция: relu;
- оптимизатор: adam;
- пропорция разбиения данных на тестовые и валидационные: 30%;

- ранняя остановка, если метрики на валидационной выборке не улучшаются;
- количество итераций: 5000.

Нейросеть обучилась за 266мс и 64 итерации. График обучения приведен на рисунке 25. Визуализация результатов, полученных нейросетью, приведена на рисунке 26.

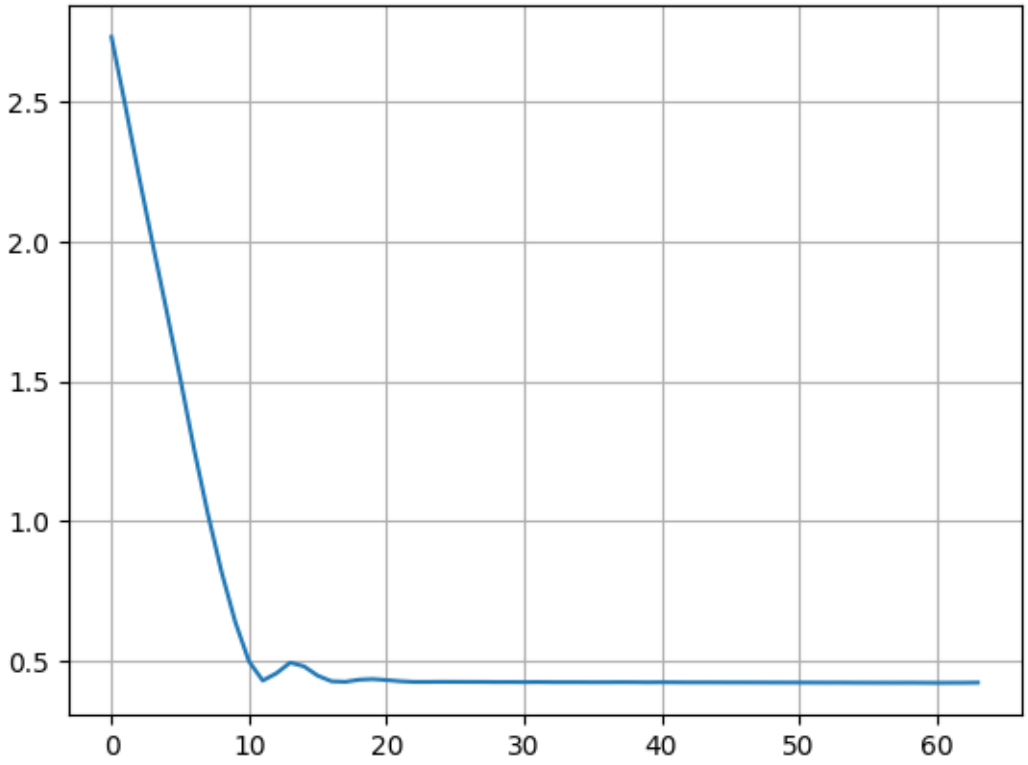

Рисунок 25 — График обучения MLPRegressor

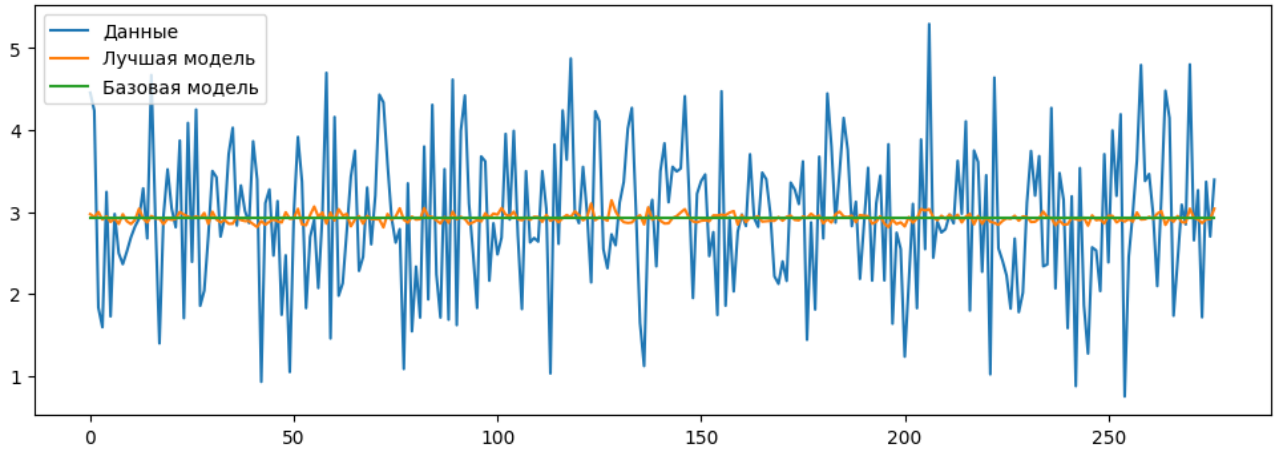

Рисунок 26 — Визуализация работы модели

Метрики работы нейросети MLPRegressor на тестовом множестве и сравнение с базовой моделью отражены на рисунке 27. Ошибка нейросети составляет 29,7%, а коэффициент детерминации чуть больше нуля.

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.000074	-0.868130	-0.691547	-0.296759	-2.370117
MLPRegressor	0.007322	-0.864914	-0.694770	-0.296923	-2.260243

Рисунок 27 — Метрики работы нейросети MLPRegressor на тестовом множестве

2. Нейросеть из библиотеки tensorflow.

Строю нейронную сеть с помощью класса keras.Sequential со следующими параметрами:

- входной слой для 12 признаков;
- выходной слой для 1 признака;
- скрытых слоев: 8;
- нейронов на каждом скрытом слое: 24;
- активационная функция скрытых слоев: relu;
- оптимизатор: Adam;
- loss-функция: MeanAbsolutePercentageError.

Архитектура нейросети приведена на рисунке 28.

```
 Layer (type)                 Output Shape              Param #
=================================================================
 dense_1 (Dense)              (None, 24)                312

 dense_2 (Dense)              (None, 24)                600

 dense_3 (Dense)              (None, 24)                600

 dense_4 (Dense)              (None, 24)                600

 dense_5 (Dense)              (None, 24)                600

 dense_6 (Dense)              (None, 24)                600

 dense_7 (Dense)              (None, 24)                600

 dense_8 (Dense)              (None, 24)                600

 out (Dense)                  (None, 1)                 25

=================================================================
Total params: 4,537
Trainable params: 4,537
Non-trainable params: 0
```

Рисунок 28 — Архитектура нейросети в виде summary

Запускаю обучение нейросети со следующими параметрами:

– пропорция разбиения данных на тестовые и валидационные: 30%;

– количество эпох: 50.

– раннюю остановку не использую.

График обучения приведен на рисунке 29, ошибка — в таблице 2.

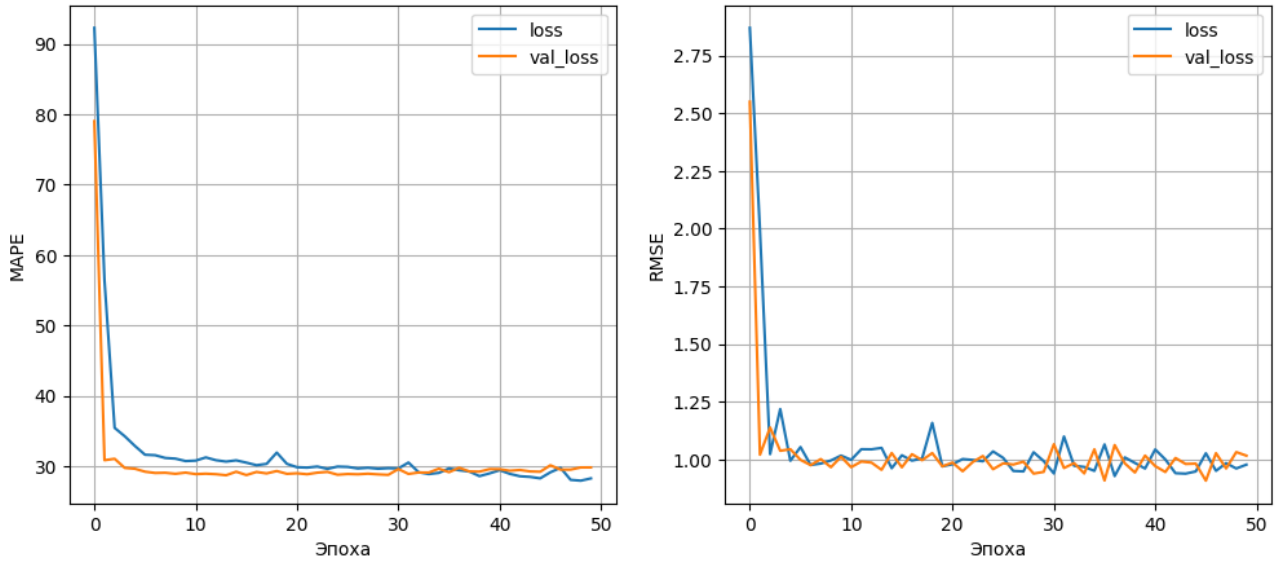

Рисунок 29 — График обучения нейросети

Видно, что примерно до 35 эпохи обучение шло хорошо, а потом сеть начала переобучаться. Значение loss на тестовых выборках продолжило уменьшаться, а на валидационной начало расти.

Одним из способов борьбы с переобучением может быть ранняя остановка обучения, если val_loss начинает расти. Для этого в tensorflow используются callbacks. Попробую взять нейросеть с той же архитектурой и запустить обучение с ранней остановкой. График обучения приведен на рисунке 30, а ошибка — в таблице 2.

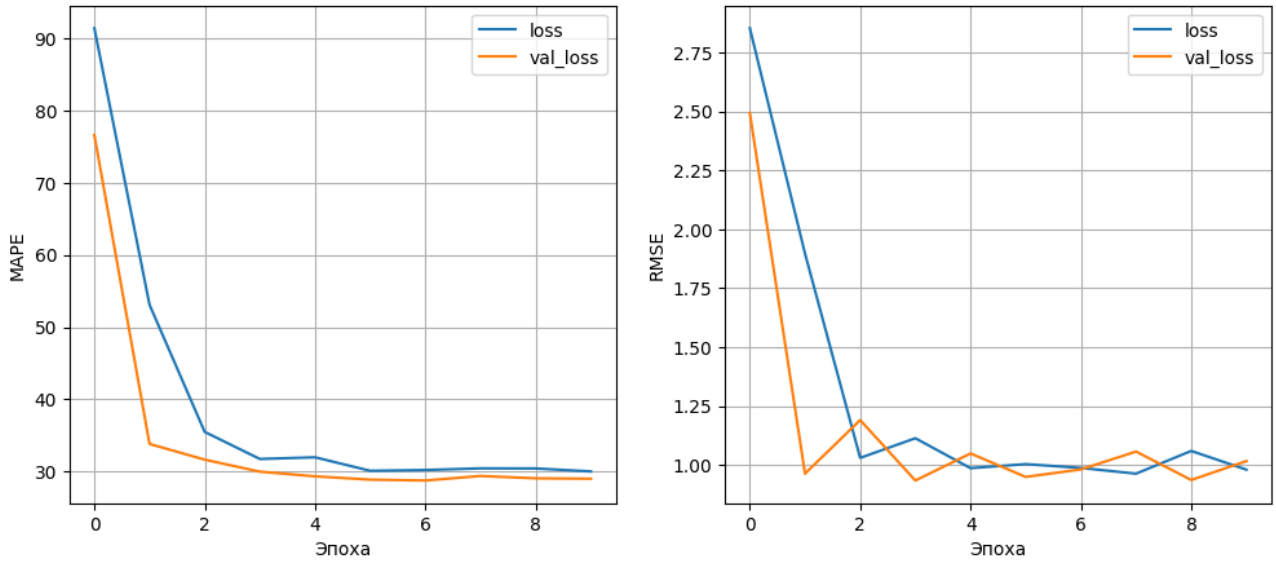

Рисунок 30 — График обучения нейросети с ранней остановкой

Еще одним методом борьбы с переобучением является добавление Dropout-слоев. Построим модель аналогичной архитектуры, только после каждого скрытого слоя добавим слой Dropout с параметром 0.05. Такой слои слои выключат 5% случайных нейронов на каждом слое.

График обучения приведен на рисунке 32, а ошибка — в таблице 2. Видно, что Dropout-слои справились с переобучением.

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

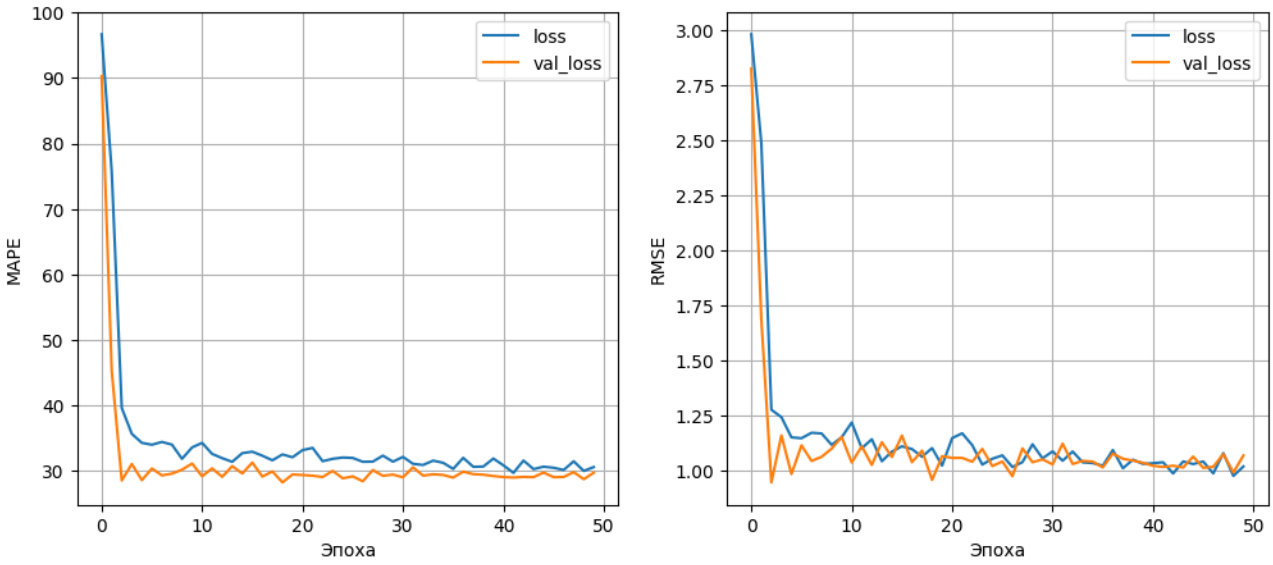

Рисунок 31 — График обучения нейросети с Dropout-слоем

Таблица 2. Борьба с переобучением нейросети

	Эпох	Ошибка на тестовых данных, %	Время обучения, с
Нейросеть переобученная	50	28.83	4.06
Нейросеть с ранней остановкой	10	29.34	1.84
Нейросеть с dropout-слоями	50	29.47	4.77

Визуализация результатов работы нейросетей отображена на рисунках 32-34, а их метрики — на рисунке 35.

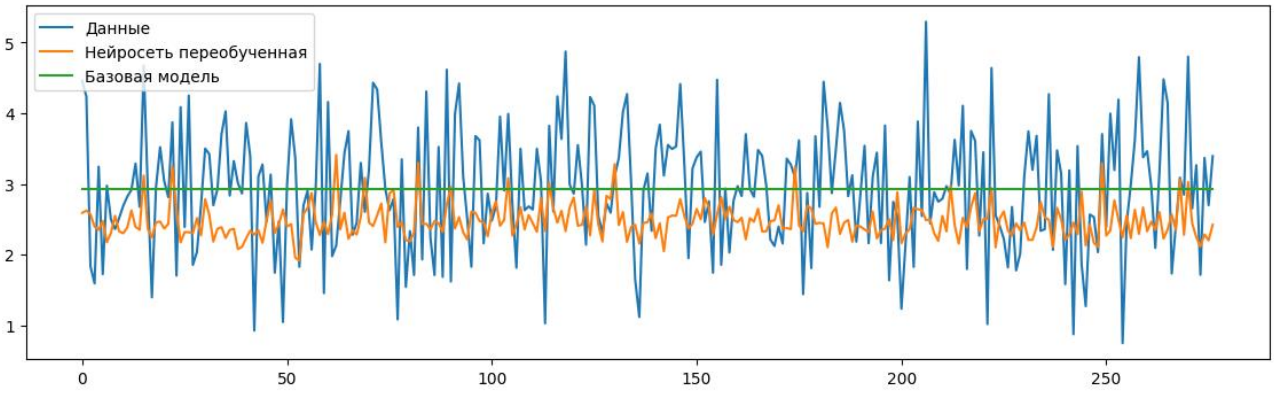

Рисунок 32 — Визуализация результатов работы нейросети с переобучением

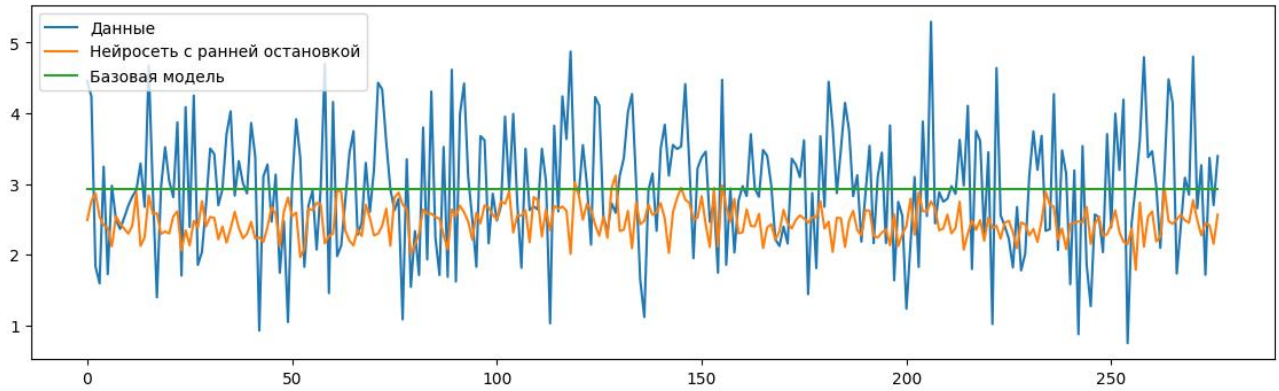

Рисунок 33 — Визуализация результатов работы нейросети с ранней остановкой

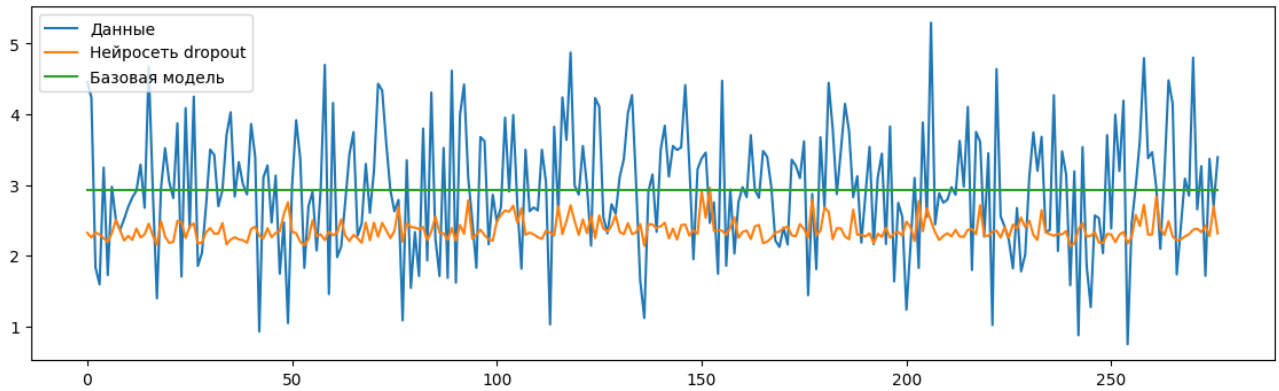

Рисунок 34 — Визуализация результатов работы нейросети с Dropout-слоем

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.000074	-0.868130	-0.691547	-0.296759	-2.370117
Нейросеть переобученная	-0.282342	-0.983039	-0.802209	-0.288258	-2.807613
Нейросеть с ранней остановкой	-0.337591	-1.003992	-0.816470	-0.293582	-2.862930
Нейросеть dropout	-0.432122	-1.038864	-0.847236	-0.294678	-2.807533

Рисунок 35 — Метрики работы нейросетей на тестовом множестве

Визуализация результатов показывает, что нейросеть из библиотеки tensorflow старалась подстроиться к данным. Выглядят результаты «похоже», но метрики разочаровывают. Лучшая обобщающая способность и меньшие значения ошибок на тестовом множестве оказались у нейросети, которая переобучилась, мне не удалось уменьшить ошибку. Но и она предсказывает гораздо хуже базовой модели.

2.5 Тестирование модели

Согласно заданию, необходимо сравнить ошибку каждой модели на тренировочной и тестирующей части выборки.

Модель для предсказания модуля упругости при растяжении — Ridge(alpha=60, positive=True, solver='lbfgs'). Сравнение ее ошибок показано на ри-сунке 36.

	R2	RMSE	MAE	MAPE	max_error
Модуль упругости, тренировочный	0.010635	-3.008967	-2.422298	-0.033091	-7.839390
Модуль упругости, тестовый	-0.007314	-3.031272	-2.486880	-0.034029	-7.581602

Рисунок 36 – Сравнение ошибок модели для модуля упругости при растяжении на тренировочном и тестовом датасете

Ridge-регрессия имеет ошибку на тренировочном датасете меньше, чем на тестовом, потому что чему-то все-таки научилось. Но даже на тренировочном датасете оно не нашло закономерности во входных данных. Задачу решить не удалось.

Модель для предсказания прочности при растяжении — Lasso(alpha=5). Сравнение ее ошибок показано на рисунке 37.

	R2	RMSE	MAE	MAPE	max_error
Прочность при растяжении, тренировочный	0.013067	-456.085754	-364.432618	-0.159590	-1266.944961
Прочность при растяжении, тестовый	-0.009342	-441.382723	-350.450799	-0.151207	-1179.461001

Рисунок 37 – Сравнение ошибок модели для прочности на тренировочном и тестовом датасете

Lasso-регрессия также показала ошибку на тестовом множестве больше, чем на тренировочном. Значит, модель не нашла следы зависимости.

Модель для предсказания соотношения матрица-наполнитель — нейросеть MLPRegressor(early_stopping=True, hidden_layer_sizes=(24, 24, 24, 24, 24, 24, 24, 24), max_iter=5000, random_state=3128, validation_fraction=0.3, verbose=True). Сравнение ее ошибок показано на рисунке 38.

	R2	RMSE	MAE	MAPE	max_error
Соотношение матрица-наполнитель, тренировочный	0.007198	-0.903012	-0.728448	-0.315134	-2.401036
Соотношение матрица-наполнитель, тестовый	0.007322	-0.864914	-0.694770	-0.296923	-2.260243

Рисунок 38 – Сравнение ошибок модели для соотношения матрица-наполнитель на тренировочном и тестовом датасете

У нейросети показатели для тестовой выборки чуть лучше показателей тренировочной. Удалось добиться коэффициента детерминации чуть больше нуля, значит модель чему-то обучилась.

Возможно, требуется более тщательное и грамотное построение архитектуры нейронной сети, чтобы получить лучший результат. Но сейчас задача далека от решения.

2.6 Разработка приложения

Несмотря на то, что пригодных к внедрению моделей получить не удалось, можно разработать функционал приложения. Возможно, дальнейшие исследования позволят построить качественную модель и внедрить ее в готовое приложение.

В приложении необходимо реализовать следующие функции:

- ввод входных параметров;
- проверка введенных параметров;
- загрузка сохраненной модели, получение и отображение прогноза выходных параметров.

Решено разработать веб-приложение для прогнозирования соотношения матрица-наполнитель с помощью языка Python, фрейм-ворка Flask и шаблонизатора Jinja.

Эта задача решена не полностью. Удалось разработать только локальное приложение, для загрузки на внешние ресурсы не хватило времени.

Скриншот разработанного веб-приложения приведен на рисунке 39

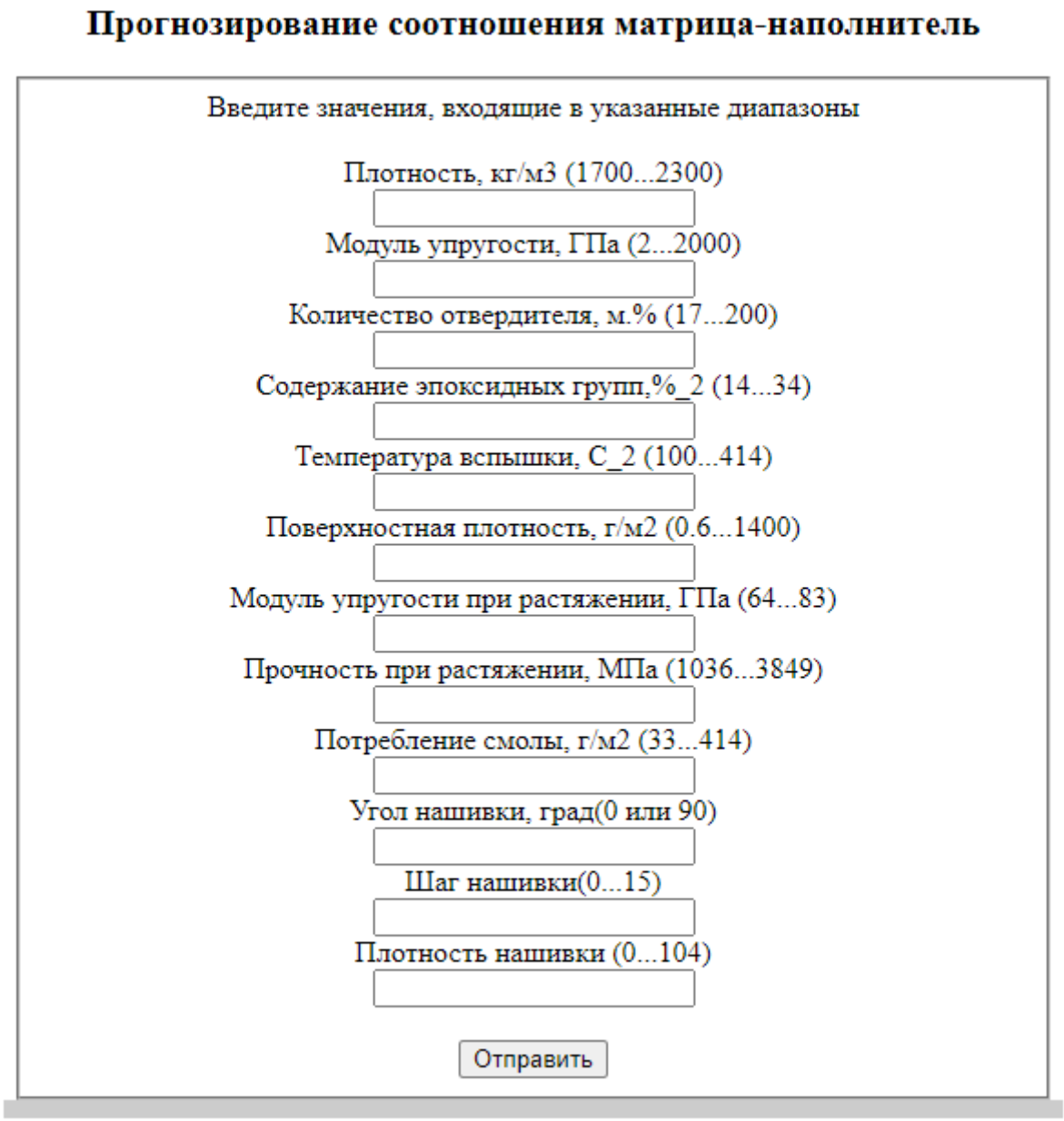

Рисунок 39 – Ввод входных параметров для прогнозирования соотношения матрица-наполнитель

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

2.7. Создание удаленного репозитория

Для данного исследования был создан удаленный репозиторий на GitHub, который находится по адресу https://github.com/VasilisaStepanova/VKR_composites.git.

На него были загружены результаты работы: исследовательский notebook, код приложения.

Заключение

В ходе выполнения данной работы мы прошли практически весь Dataflow pipeline, рассмотрели большую часть операций и задач, которые приходится выполнять специалисту по работе с данными.

Этот поток операций и задач включает:

- изучение теоретических методов анализа данных и машинного обучения;
- изучение основ предметной области, в которой решается задача;
- извлечение и трансформацию данных. Здесь нам был предоставлен готовый набор данных, поэтому через трудности работы с разными источниками и парсингом данных пройти не пришлось;
- проведение разведочного анализа данных статистическими методами;
- DataMining — извлечение признаков из датасета и их анализ;
- разделение имеющихся, в нашем случае размеченных, данных на обучающую, валидационную, тестовую выборки;
- выполнение предобработки (препроцессинга) данных для обеспечения корректной работы моделей;
- построение аналитического решения. Это включает выбор алгоритма решения и модели, сравнение различных моделей, подбор гиперпараметров модели;
- визуализация модели и оценка качества аналитического решения;
- сохранение моделей;

- разработка и тестирование приложения для поддержки принятия решений специалистом предметной области, которое использовало бы найденную модель;
- внедрение решения и приложения в эксплуатацию. Этот блок задач мы тоже пока не затронули.

В этой работе мы имели дело не с учебными наборами данных, которые дают хорошо изученные решения, а с реальной производственной задачей. И, к сожалению, не смогли поставленную задачу решить — не получили моделей, которые бы описывали закономерности предметной области. Я проделала максимум исследований, которые в моей компетенции как начинающего дата-сайентиста, применила большую часть знаний, полученных в ходе прохождения курса.

Возможные причины неудачи:

- нечеткая постановка задачи, отсутствие дополнительной информации о зависимости признаков с точки зрения физики процесса. Незначимые признаки являются для модели шумом, и мешают найти зависимость целевых от значимых входных признаков;
- исследование предварительно обработанных данных. Возможно, на «сырых», не предобработанных данных можно было бы получить более качественные модели, воспользовавшись другими методами очистки и подготовки;
- мой недостаток знаний и опыта. Нейросети являются самым современным подходам к решению такого рода задач. Они

способны находить скрытые и нелинейные зависимости в данных. Но выбор оптимальной архитектуры нейросети является неочевидной задачей.

Дальнейшие возможные пути решения этой задачи могли бы быть:

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

Список использованной литературы

1. Композиционные материалы : учебное пособие для вузов / Д. А. Иванов, А. И. Ситников, С. Д. Шляпин ; под редакцией А. А. Ильина. — Москва : Издательство Юрайт, 2019 — 253 с. — (Высшее образование). — Текст : непосредственный.

2. Машинное обучение без лишних слов / Андрей Бурков — СПб.: Питер, 2020. — 192 с.: ил. — (Серия «Библиотека программиста»).

3. Нейронные сети: полный курс, 2-е издание / Саймон Хайкин : Пер. с англ. — Москва: Издательский дом «Вильям», 2006. — 1104 с. — Парал. тит. англ.

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

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

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

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

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

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

10. Документация по библиотеке keras: – Режим доступа: https://keras.io/.