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 Головач Святослав Сергеевич

Москва, 2023

# Оглавление

[Введение 3](#_Toc132653887)

[1. Аналитическая часть 4](#_Toc132653888)

[1.1. Постановка задачи 4](#_Toc132653889)

[1.2. Описание используемых методов 11](#_Toc132653890)

[1.3. Разведочный анализ данных 21](#_Toc132653891)

[2. Практическая часть 26](#_Toc132653892)

[2.1. Предобработка данных 26](#_Toc132653893)

[2.2 Разработка, обучение и тестирование десяти моделей (включая нейросеть) для прогнозирования трёх целевых признаков по отдельности. 27](#_Toc132653894)

[2.3 Написание нейронной сети для прогнозирования соотношения матрица-наполнитель 33](#_Toc132653895)

[2.4. Разработка приложения 36](#_Toc132653896)

[2.5. Создание удаленного репозитория 36](#_Toc132653897)

[Заключение 37](#_Toc132653898)

[Список использованных источников и литературы 38](#_Toc132653899)

[Приложение А. Скриншоты веб-приложения 40](#_Toc132653900)

# Введение

Современный мир требует от нас постоянного развития, и наука и технологии являются главными средствами, позволяющими нам это делать. В частности, разработка новых материалов – ключевой фактор, определяющий технологический прогресс, который приводит к созданию новых устройств и техники, которые недавно еще были невозможными. Однако, создание новых материалов – сложный процесс, который требует больших затрат времени и ресурсов. В связи с этим возникает необходимость в разработке методов и технологий, которые позволят сократить время и средства, необходимые для создания новых материалов. Одним из основных методов является прогнозирование конечных свойств новых материалов. Это позволяет ускорить процесс исследований, уменьшить затраты на эксперименты и сделать процесс создания новых материалов более эффективным и безопасным.

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

Композиционными называются материалы, в которых имеет место сочетание двух (или более) химически разнородных компонентов (фаз) с четкой границей раздела между ними. Это неоднородные по химическому составу и структуре материалы.

Структура композиционных материалов представляет собой матрицу (основной компонент), содержащую в своем объеме или армирующие элементы, часто называемые наполнителем. Матрица и наполнитель разделены границей (поверхностью) раздела. Наполнитель равномерно распределен в матрице и имеет заданную пространственную ориентацию.

Композиционные материалы характеризуются совокупностью свойств, не присущих каждому в отдельности взятому компоненту. За счет выбора армирующих элементов, варьирования их объемной доли в матричном материале, а также размеров, формы, ориентации и прочности связи по границе «матрица-наполнитель», свойства композиционных материалов можно регулировать в значительных пределах.

Возможно получить композиты с уникальными эксплуатационными свойствами. Этим обусловлено широкое применение композиционных материалов в различных областях техники.

Учитывая такое широкое распространение и высокую потребность в новых материалах, тема данной работы является очень актуальной.

Стоимость производства композитного материала высока. Зная характеристики компонентов, невозможно рассчитать свойства композита. Значит для получения заданных свойств требуется большое количество испытаний различных комбинаций. Сократить время и затраты на создание определенного материала могла бы помочь система поддержки производственных решений, построенная на принципах машинного обучения.

# 1. Аналитическая часть

## 1.1. Постановка задачи

В данной работе исследуется композит с матрицей из базальтопластика и нашивками из углепластика. От специалистов в предметной области был получен датасет, содержащий данные о свойствах матрицы и наполнителя, производственных параметрах и свойствах готового композита. От нас, как специалистов в машинном обучении, требуется разработать модели, прогнозирующие значения некоторых свойств в зависимости от остальных. Также требуется разработать приложение, делающее удобным использование данных моделей специалистом предметной области.

Датасет состоит из двух файлов: X\_bp.xlsx (признаки базальтопластика) и Х\_nup.xlsx (признаки углепластика).

Файл X\_bp.xlsx содержит 1023 строки, индекс и 10 признаков.

Файл X\_nup.xlsx содержит 1040 строк индекс и 3 признака.

В условии задания сказано, что файлы требуют объединения с типом INNER по индексу. Прежде чем объёдинять таблицы я решил визуально проанализировать их в Excel на рисунке 1.

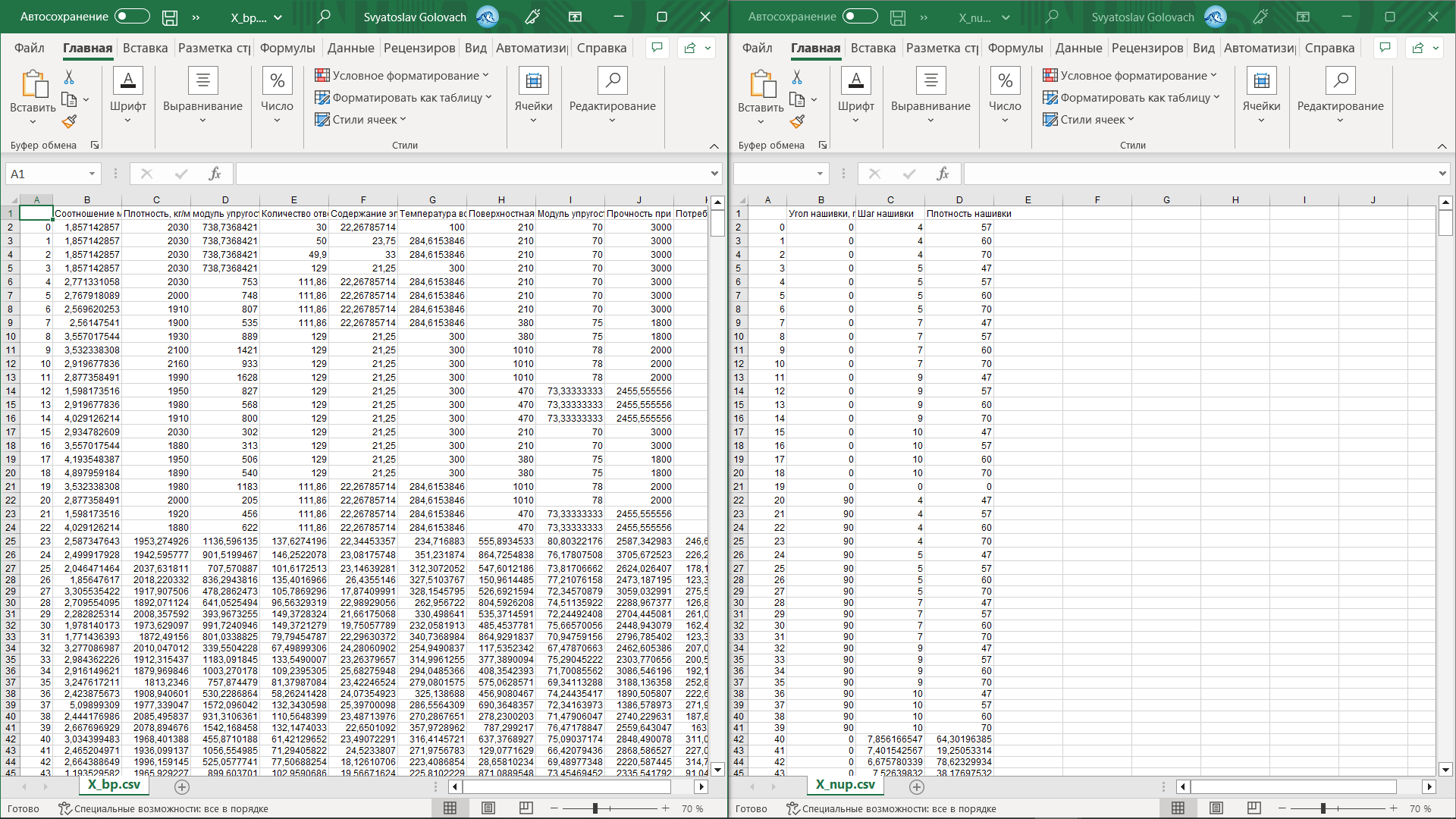


Рисунок 1 – Значения в X\_bp.xlsx и в Х\_nup.xlsx

Заметим, что первые строки обеих датасетов были будто бы искуственно созданы, так как в них есть целые и повторяющиеся значения. В то время как практически весь датасет состоит из чисел с плавающей точкой. Нам уже известно, что X\_bp.xlsx имеет 1023 строки, а X\_nup.xlsx имеет 1040 строк, и мы можем наблюдать что ровно 23 строки в первом файле и ровно 40 во втором нас смущают своими значениями. Если мы удалим эти значения, то оба датасета бутут иметь одинаковое количество строк, но соединить по индексам методом INNER, без потерь, уже не получится, так как один датасет начнётся с 24 строки, а другой с 41. Поэтому нам придётся пожертвовать ещё 17 строками в X\_bp.xlsx, с начала таблицы, и тогда обе таблицы начнутся с 41 строчки. Но вместе с этим, концы таблиц тоже будут несостыковываться по индексам, тогда подравняем по меньшему индексу. То есть урежем датасет до длины меньшего.

Таким образом мы самостоятельно выбрали что нам нужно оставить, а что удалить. В то время как метод INNER, по-своему, автоматически отсекает лишние строки. Дальнейшие исследования проводим с объединенным датасетом, содержащим 983 строки и 13 признаков.

Описание признаков объединенного датасета приведено на рисунке 2. Все признаки имеют тип float64 (числа с плавающей точкой), кроме «Угол нашивки, град». «Угол нашивки, град» имеет тип int64 (целочисленные значения). Все признаки, кроме «Угол нашивки, град», являются непрерывными, количественными. «Угол нашивки, град» принимает только два значения. Пропусков в данных нет.

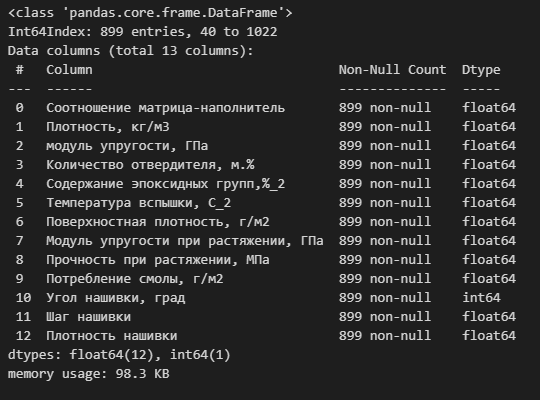
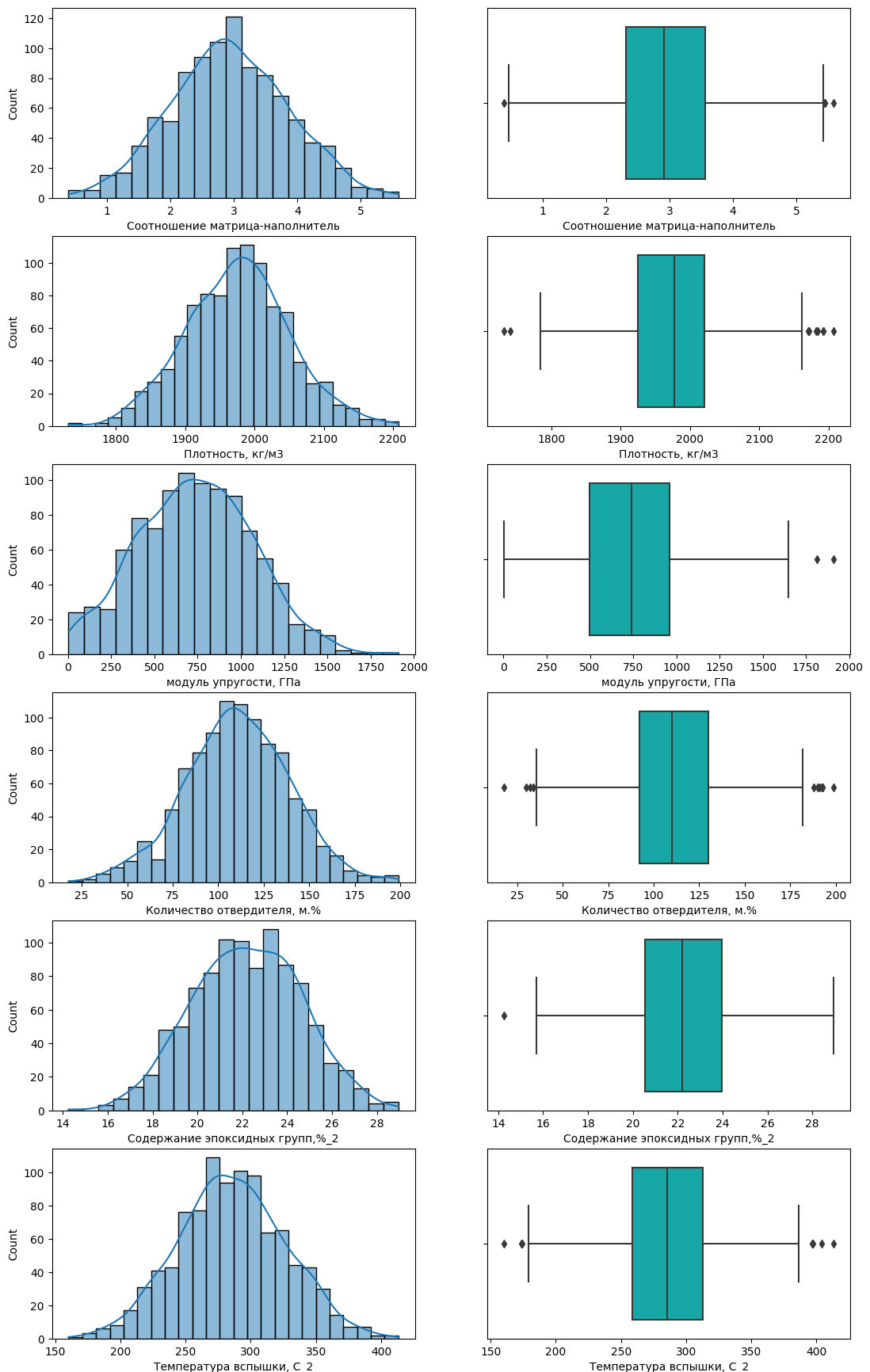


Рисунок 2 — Описание признаков датасета

Гистограммы распределения переменных и диаграммы «ящик с усами» приведены на рисунке 3. По ним видно, что почти все признаки, кроме «Угол нашивки, град», имеют нормальное распределение и принимают неотрицательные значения. «Угол нашивки, град» принимает значения только – 0 и 90.

 Изображение выглядит как орган

Автоматически созданное описание

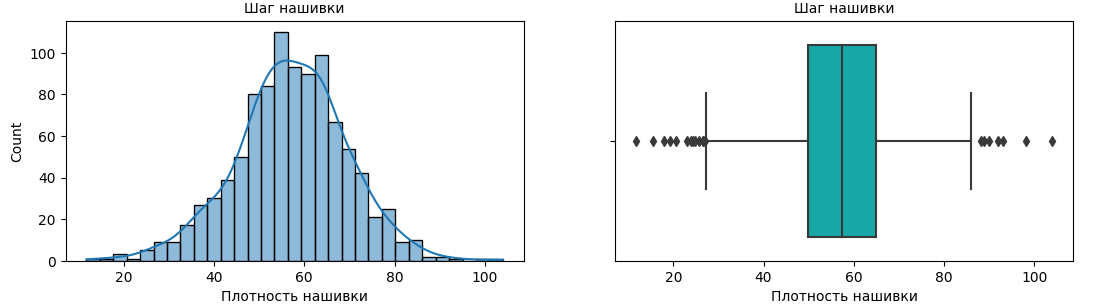


Рисунок 3 - Гистограммы распределения переменных

и диаграммы «ящик с усами»

В задании было сказано, что датасет был предварительно подготовлен, поэтому пропуски отсутствуют. Нас интересует описательная статистика датасета. Она представлена на рисунке 4. Она в численном виде отражает то, что мы видим на гистограммах.

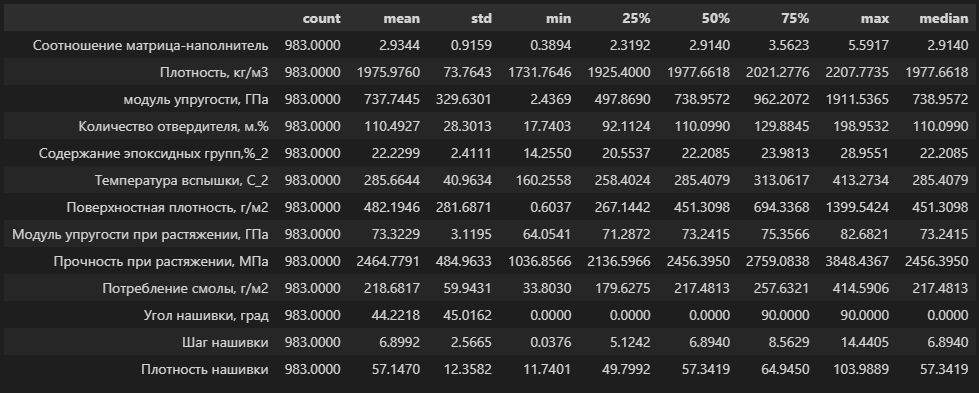


Рисунок 4 — Описательная статистика признаков датасета

Попарные графики рассеяния точек приведены на рисунке 5. По графикам рассеяния можно увидеть, что некоторые точки стоят далеко от общего облака.

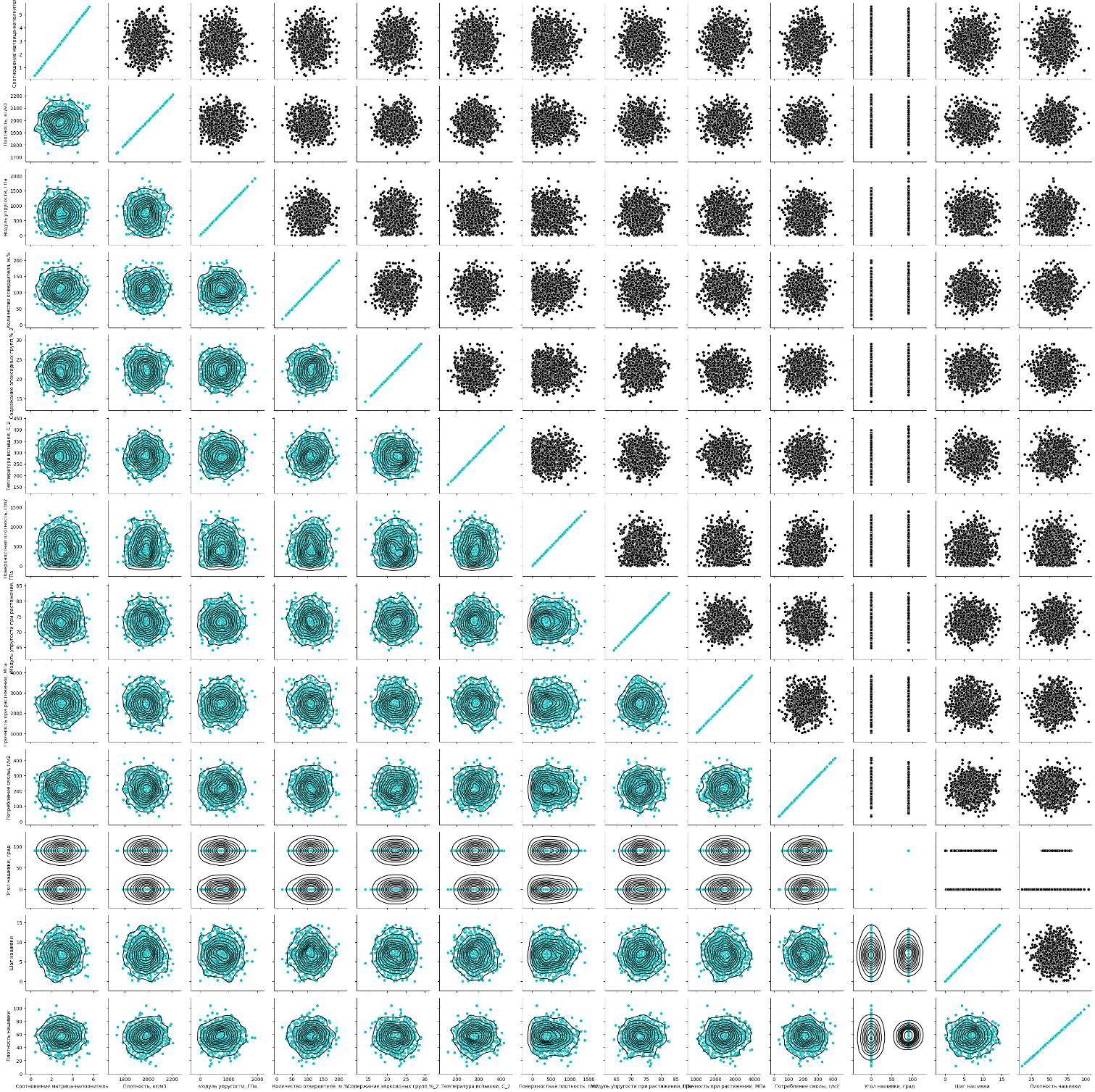


Рисунок 5 — Попарные графики рассеяния точек

Нам известно 2 метода выявления выбросов для признаков с нормальным распределением, это:

* метод 3-х сигм;
* метод межквартильного интервала.

Воспользовавшись этими методами на нашем датасете было найдено:

* 22 выброса методом 3-х сигм;
* 88 выбросов методом межквартильного интервала расстояний.

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

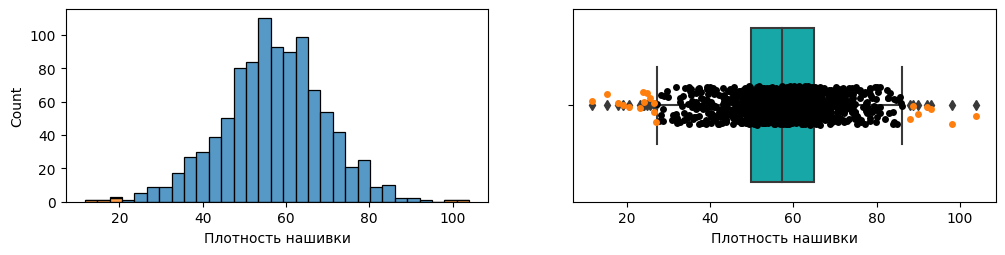


Рисунок 6 — Пример выбросов

Для наиболее эффективной работы модели воспользуемся методом межквартильного интервала. Значения определенные как выбросы удаляем. После чего удаляем строки с пропущенными значениями. По итогу, в датасете осталось 899 строк.

## 1.2. Описание используемых методов

Предсказание значений вещественной, непрерывной переменной — это задача регрессии. Эта зависимая переменная должна иметь связь с одной или несколькими независимыми переменными, называемых также предикторами или регрессорами. Регрессионный анализ помогает понять, как «типичное» значение зависимой переменной изменяется при изменении независимых переменных.

В настоящее время разработано множество методов регрессионного анализа. Например, простая и множественная линейная регрессия. Эти модели являются параметрическими в том смысле, что функция регрессии определяется конечным числом неизвестных параметров, которые оцениваются на основе данных.

1. **Линейная регрессия (LinearRegression)**

Простая линейная регрессия имеет место, если рассматривается зависимость между одной входной и одной выходной переменными. Для этого определяется уравнение регрессии и строится соответствующая прямая, известная как линия регрессии:

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

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

где n - число входных переменных.

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

Линейная регрессия — первый тщательно изученный метод регрессионного анализа. Его главное достоинство — простота. Такую модель можно построить и рассчитать даже без мощных вычислительных средств. Простота является и главным недостатком этого метода.

1. **Лассо (LASSO) регрессия**

LASSO регрессия — это метод регуляризации линейной регрессии, где коэффициенты модели ограничены с помощью штрафа L1 нормы. Это означает, что модель минимизирует сумму квадратов ошибок, при этом ограничивая величину суммы абсолютных значений коэффициентов. Этот метод позволяет автоматически отбирать наиболее значимые признаки и устранять шум в данных.

Преимущества LASSO регрессии заключаются в её способности к сокращению переобучения и возможность выбора наиболее значимых признаков. Кроме того, LASSO регрессия соответствует принципу "мотыги Мора", что означает уменьшение размерности модели за счёт исключения менее важных признаков.

Формула для LASSO регрессии: минимизация суммы квадратов ошибок с ограничением суммы абсолютных значений коэффициентов. Математически это записывается как:

min ||y - Xw||^2 + λ||w||\_1

Где:

* y - вектор значений зависимой переменной;
* X - матрица значений признаков;
* w - вектор коэффициентов модели;
* ||.||\_2 - норма L2;
* ||.||\_1 - норма L1;
* λ - коэффициент регуляризации, который отвечает за величину штрафа.

Таким образом, LASSO регрессия становится полезной моделью при работе с данными, где существует большое число признаков, которые могут вносить шум в модель.

1. **Гребневая (Ridge) регрессия**

Ridge регрессия — это метод множественной линейной регрессии, который используется для предсказания значений зависимой переменной на основе значений нескольких независимых переменных. Она отличается от обычной линейной регрессии тем, что в её формуле используется дополнительный параметр - коэффициент регуляризации.

Преимуществами Ridge регрессии являются её способность справляться с проблемами мультиколлинеарности, когда существует сильная корреляция между независимыми переменными. Ridge регрессия также помогает предотвратить переобучение модели, что позволяет увеличить её обобщающую способность. Кроме того, Ridge регрессия может улучшить качество модели, если данные содержат выбросы.

Формула для Ridge регрессии выглядит следующим образом:

Y = β0 + β1X1 + β2X2 + ... + βnXn + λΣ(βi^2)

где Y - зависимая переменная, Xi - независимые переменные, βi - коэффициенты регрессии, λ - коэффициент регуляризации, который контролирует вклад регуляризации в общую ошибку модели. Чем больше значение λ, тем сильнее регуляризация и меньше свободы модели.

Гребневая регрессия — вариация линейной регрессии, очень похожая на регрессию LASSO. Она также применяет сжатие и хорошо работает для данных, которые демонстрируют сильную мультиколлинеарность. Самое большое различие между ними в том, что гребневая регрессия использует регуляризацию L2, которая взвешивает ошибки по их квадрату, чтобы сильнее наказывать за более значительные ошибки. Регуляризация позволяет интерпретировать модели. Если коэффициент стал 0 (для Lasso) или близким к 0 (для Ridge), значит данный входной признак не является значимым.

1. **ElasticNet регрессия**

ElasticNet регрессия является одним из методов машинного обучения, который решает проблему мультиколлинеарности (сильной корреляции между факторами) в задачах регрессии. Данный метод базируется на двух видах регуляризации – L1 и L2, что позволяет улучшить качество модели и снизить влияние шума в данных. Такой алгоритм широко применяется в бизнесе, а также в научных исследованиях и финансово-экономических моделях.

Преимущества ElasticNet регрессии заключаются в ее универсальности и гибкости настройки. Формулы L1 и L2 регуляризации могут быть заданы произвольно, что позволяет аналитику настраивать метод под конкретные данные. Кроме того, данный метод предназначен для поиска оптимальных коэффициентов регрессии с учетом равновесия между шумом и сигналом в данных. Таким образом, улучшается точность прогнозирования и общего качества модели.

Формула ElasticNet регрессии представляет собой сумму двух элементов – L1 регуляризации (Lasso) и L2 регуляризации (Ridge). Они задаются соответствующим образом в формуле регрессии и зависят от настроек коэффициента α, который контролирует баланс между двумя методами регуляризации. Формула регрессии выглядит следующим образом:

y = β0 + β1x1 + β2x2 + … + βnxn + ε

где y – предиктор, βi – коэффициенты регрессии, xi – факторы регрессии, а ε – ошибка модели.

1. **Случайный лес**

Случайный лес (RandomForest) — представитель ансамблевых методов.

Если точность дерева решений оказалось недостаточной, мы можем множество моделей собрать в коллектив. Формула итогового решателя (3) — это усреднение предсказаний отдельных деревьев.

Diagram

Description automatically generated with medium confidence

где

N – количество деревьев;

i – счетчик для деревьев;

b – решающее дерево;

x – сгенерированная нами на основе данных выборка.

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

Преимущества случайного леса:

* высокая точность предсказания;
* редко переобучается;
* практически не чувствителен к выбросам в данных;
* одинаково хорошо обрабатывает как непрерывные, так и дискретные признаки, данные с большим числом признаков;
* высокая параллелизуемость и масштабируемость.

Из недостатков можно отметить, что его построение занимает больше времени. А также теряется интерпретируемость.

1. **Метод k-ближайших соседей (KNeighborsRegressor)**

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.

Преимущество k-neighbors регрессии заключается в ее простоте и универсальности. Алгоритм может использоваться для решения широкого круга задач в области машинного обучения и анализа данных, таких как прогнозирование цен на недвижимость, распознавание рукописных символов, определение возраста или пола на фотографии, анализ данных геопозиционирования и т.д.

Формула k-neighbors регрессии выглядит следующим образом:

y = 1/ k ∑ yₘ,

где y - значение целевой переменной, k - количество ближайших соседей, yₘ - значение данной переменной у каждого из соседей. Количество соседей, на основе которого происходит прогнозирование, задается заранее и является параметром алгоритма. Чем меньше значение k, тем более гибкой будет модель и тем больше она будет склонна к переобучению. При увеличении значения k, модель становится более устойчивой и менее склонной к переобучению.

В случае использования метода для регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны. Для реализации метода необходима метрика расстояния между объектами. Используется, например, эвклидово расстояние для количественных признаков или расстояние Хэмминга для категориальных. Этот метод — пример непараметрической регрессии.

1. **Метод опорных векторов для регрессии (SVR)**

Метод опорных векторов (SVR) — 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.

Чаще всего он применяется в постановке бинарной классификации.

Основная идея 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.

Решается задача оптимизации.

Для вычислений используется ядерная функция, получающая на вход два вектора и возвращающая меру сходства между ними:

* линейная;
* полиномиальная;
* гауссовская (rbf).

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

Преимущество метода — его хорошая изученность.

Недостатки:

* чувствительность к выбросам;
* отсутствие интерпретируемости.

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.

Чтобы построить алгоритм градиентного бустинга, нам необходимо выбрать базовый алгоритм и функцию потерь или ошибки (loss). Loss-функция —это мера, которая показывает, насколько хорошо предсказание модели соответствуют данным. Используя градиентный спуск и обновляя предсказания, основанные на скорости обучения (learning rate), ищем значения, на которых loss минимальна.

Бустинг, использующий деревья решений в качестве базовых алгоритмов, называется градиентным бустингом над решающими деревьями. Он отлично работает на выборках с «табличными», неоднородными данными и способен эффективно находить нелинейные зависимости в данных различной природы. На настоящий момент это один из самых эффективных алгоритмов машинного обучения. Благодаря этому он широко применяется во многих конкурсах и промышленных задачах. Он проигрывает только нейросетям на однородных данных (изображения, звук и т. д.).

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

1. **Деревья решений**

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

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

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

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

При формировании правила для разбиения в очередном узле дерева необходимо выбрать атрибут, по которому это будет сделано

Для регрессии критерием является 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.

1. **Нейронная сеть**

Нейронная сеть — это последовательность нейронов, соединенных между собой связями. Структура нейронной сети пришла в мир программирования из биологии. Вычислительная единица нейронной сети — нейрон или персептрон.

У каждого нейрона есть определённое количество входов, куда поступают сигналы, которые суммируются с учётом значимости (веса) каждого входа.

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

Также 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.

## 1.3. Разведочный анализ данных

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

На рисунке 4 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.

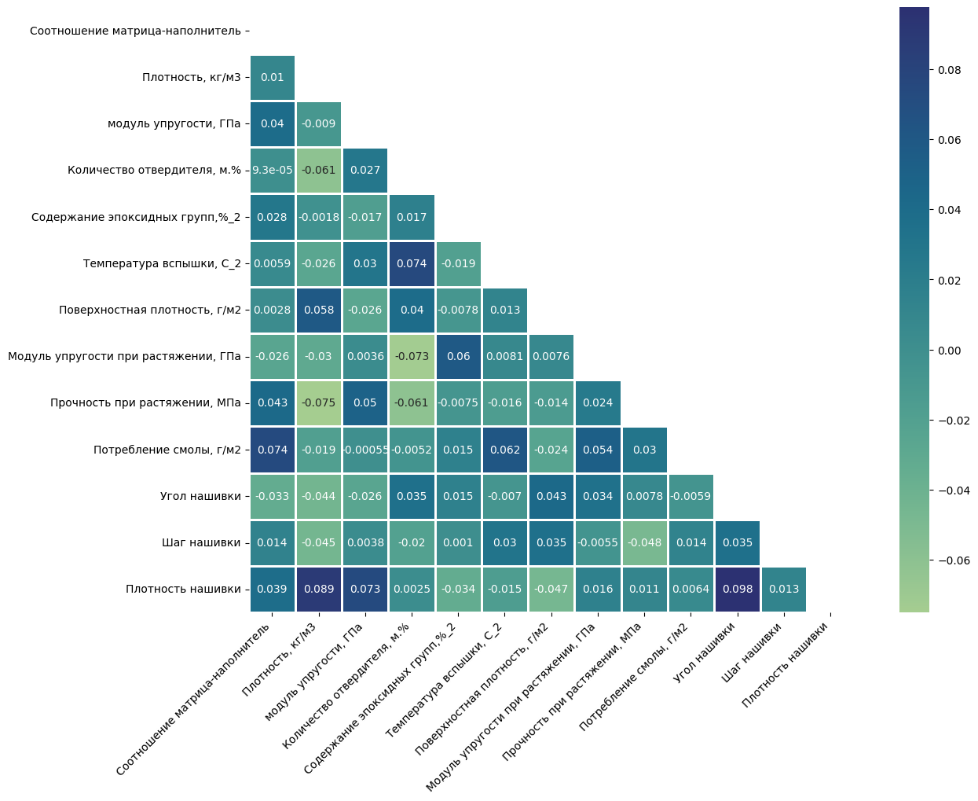


Рисунок 7 — Матрица корреляции

По матрице корреляции мы видим, что все коэффициенты корреляции близки к нулю, что означает отсутствие линейной зависимости между признаками. В связи с такой незначительной корреляцией было решено перейти к рассмотрению данного нам датасета ещё раз.

Проанализировав ещё раз данный нам датасет, мы пришли к выводу что в столбце "Температура вспышки, С\_2" имеется всего 13 значений, соответствующих стандарту определенному ASTM International (англоязычная ассоциация по стандартизации), который гласит нам о том, что температура вспышки С2 измеряется круглым термометром с точностью до 0,5 °C. Таким образом, температура вспышки С2 обычно округляется до ближайшего целого числа. Однако, иногда для более точных измерений могут применяться термометры с плавающей точкой, и тогда температура вспышки С2 может быть указана с точностью до десятых или сотых градуса Цельсия. В то время как в нашем датасете имеются нефизичные значения температуры вспышки. Исходя из этого, мы проделали всю процедуру подготовки и анализа данных заново.

Сперва мы соединили два данных изначально датасета, X\_bp.xlsx и X\_nup.xlsx, без каких-либо манипуляций над ними, только используя метод INNER. Получился датасет, размера, представленного на рисунке 8.

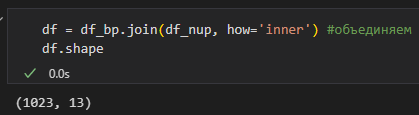


Рисунок 8 — Размерность нового датасета

Затем очистили наш датасет от нефизичных величин (рисунок 9).

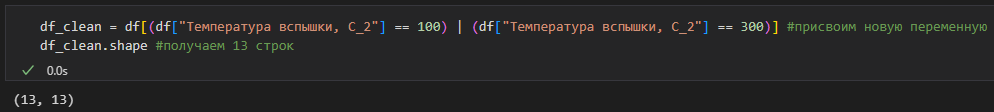


Рисунок 9 — Очистка от нефизичных величин

После чего мы обнаружили что в столбце «Угол нашивки, град» осталось всего одно уникальное значение равное 0. Было принято решение удалить данный столбец так как он больше не влияет на выборку.

На рисунке 10 представлена размерность датасета после удаления лишнего столбца.



Рисунок 10 — Размер чистого датасетика

В ходе разведочного анализа очищенного датасета, было обнаружено, что признаки имеют разный масштаб. На рисунке 11 представлена описательная статистика очищенного датасета, а затем применение метода MinMaxScaler чтобы это исправить.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11 — Описательная статистика очищенного датасета с последующим применением метода MinMaxScaler

Применив метод MinMaxScaler, мы нормализовали (шкалировали/масштабировали) данные в нашем мини-датасете. Это действие полезно тем, что теперь мы можем визуально сравнивать наши признаки между собой и также можем подавать наш нормализованный датасет на вход машинному обучению.

Взглянем на описательную статистику очищенного датасета, после нормализации данных, на рисунке 12 и сравним с рисунком 11, на котором наш датасет отображён до нормализации.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 — Описательная статистика очищенного датасета после нормализации.

Сразу можно заметить, как поменялись все значения нашей выборки. Теперь мы можем приступать к обучению наших моделей, делить наши данные на тренировочные и тестовые и обучать модели различными методами.

Помимо нормализации существует ещё и стандартизация, но мы не будем применять данный метод в нашей работе, поэтому перейдём далее к решению нашей задачи.

После построения матрицы корреляции на основе нашего очищенного и отмасштабированного датасета, мы смогли обнаружить некоторые сильные зависимости, которые представлены на рисунке 13.

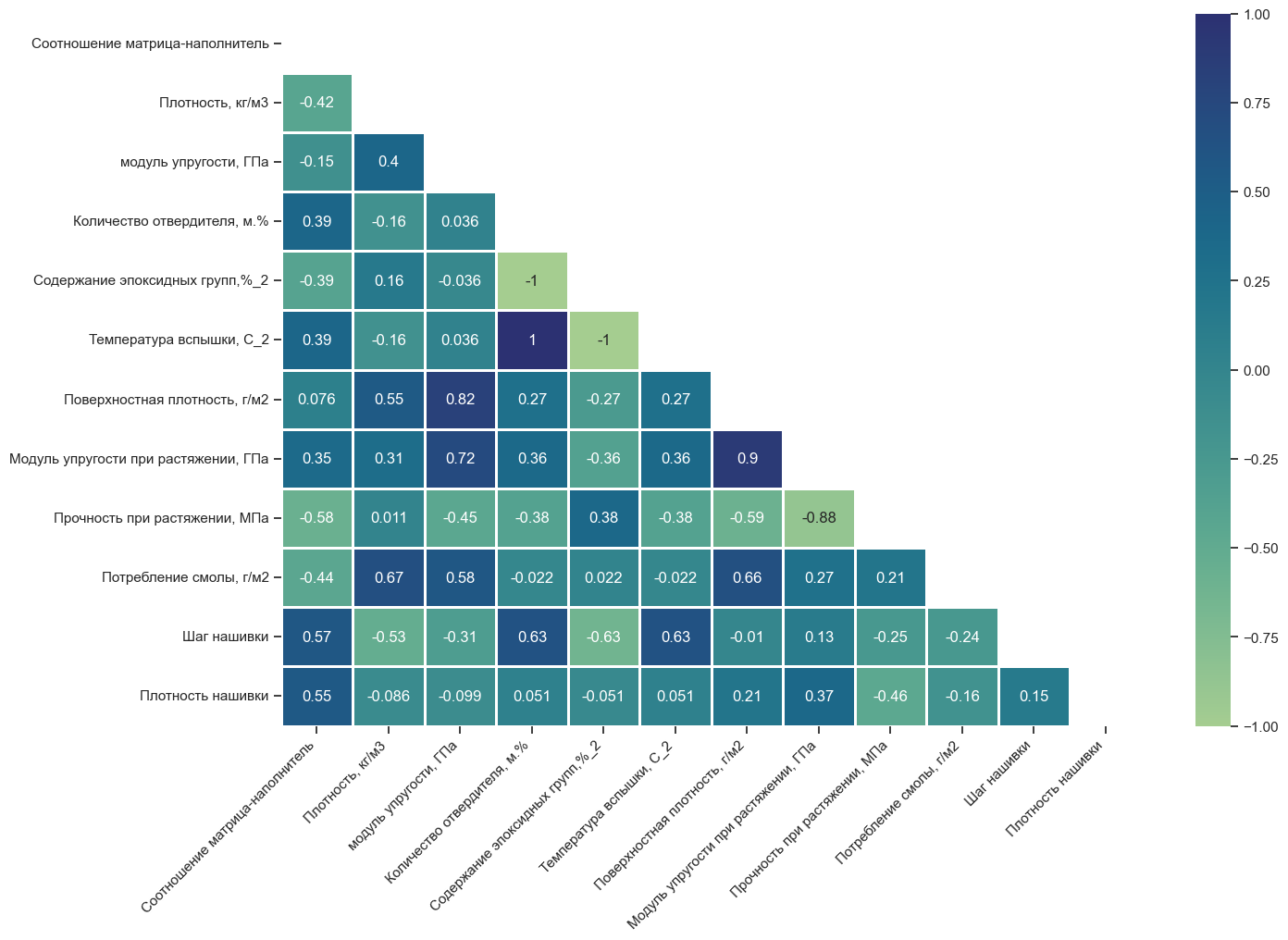


Рисунок 13 — Матрица корреляции на основе очищенного и отмасштабированного датасета

Теперь мы можем наблюдать сильную зависимость у некоторых признаков, хоть у нас и очень мало данных, но для обучения моделей вполне подойдут данные в таком количестве с явно присутствующей корреляцией. Конечно, наши модели будут непригодны для промышленного использования, это очевидно. В любом случае, мы воспользуемся десятью методами (включая нейросеть) для прогнозирования трёх целевых признаков:

* «Модуль упругости при растяжении, ГПа»;
* «Прочность при растяжении, МПа»;
* «Соотношение матрица-наполнитель».

# 2. Практическая часть

## 2.1. Предобработка данных

Цель предварительной обработки данных — обеспечить корректную работу моделей. Её необходимо выполнять перед тем, как мы будем подавать данные на вход машинному обучению.

Поскольку наш набор данных очень маленький, то мы не будем работать с выбросами. Всю предварительную обработку данных мы уже проделали, ещё на этапе разведочного анализа данных, поэтому взглянем на графики ядерной оценки плотности наших признаков до и после нормализации данных, которые представлены на рисунках 14 и 15, а затем сразу же перейдём к разработке, обоучению и тестированию всех наших моделей.

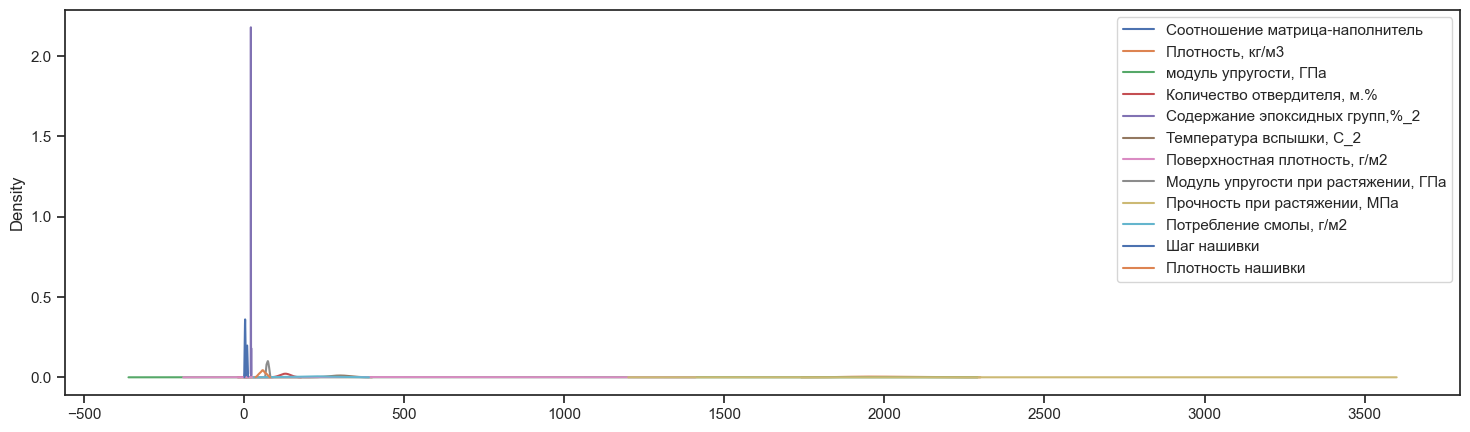


Рисунок 14 — График ядерной оценки плотности признаков очищенного датасета до нормализации

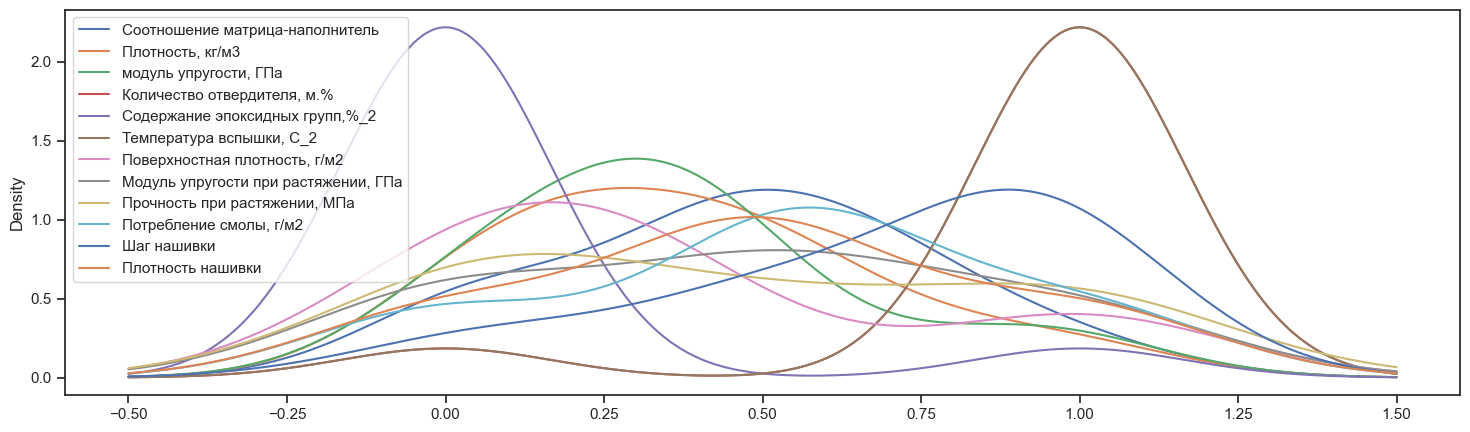


Рисунок 15 — График ядерной оценки плотности признаков очищенного датасета после нормализации

## 2.2 Разработка, обучение и тестирование десяти моделей (включая нейросеть) для прогнозирования трёх целевых признаков по отдельности.

1. **Для прогнозирования признака «Модуль упругости при растяжении, ГПа»**

Для того чтобы воспользоваться нашими методами обучения моделей, нам нужно разделить наш датасет на входную часть, в которую входят все данные кроме целевых. То есть нам нужно исключить весь столбец, который мы будем предсказывать и сохранить это множество как входную (X) часть. Затем нам нужно наоборот, оставить только столбец, в который входят все целевые значения и сохранить данное множество как выходную (y) часть. Эти манипуляции позволят нам применить метод train\_test\_split из библиотеки sklearn, который разделит наши части, ещё раз, на тренировочные и тестовые множества. Все вышеописанные действия разделения выборки на входную, выходную, а также, тренировочную и тестовую часть представлены на рисунке 16.

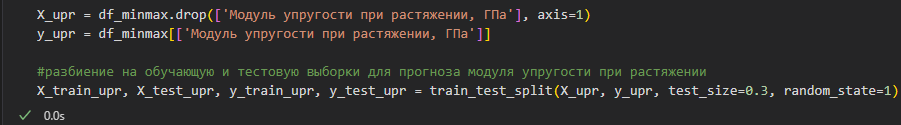


Рисунок 16 — Разделение выборки на входное, выходное, тренировочное и тестовое множество.

Признаки датасета были разделены на входные и выходные (X и y), а строки - на тренировочное и тестовое множество (train и test). Размерности полученных наборов данных показаны на рисунке 17.

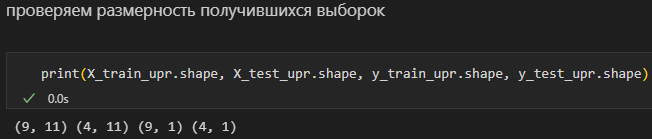


Рисунок 17 — Размерности тренировочных и тестовых множеств

после разбиения

После того как мы произвели разделение, перейдём к разработке и обучению наших моделей с последующим тестированием и сравнением всех моделей сразу.

Из раздела 1.2 нам уже известны все методы, которые мы будем применять. Некоторые методы являются непараметрическими, то есть полностью зависят от данных и не имеют гибких настроек, что не позволяет нам улучшить результаты модели для конкретной задачи путём подбора гиперпараметров. В нашей работе используются и те и другие методы, поэтому взглянем на примеры процесса обучения тех и других моделей. На рисунке 18 представлен пример обучения модели методом классической линейной регрессии (LinearRegression).

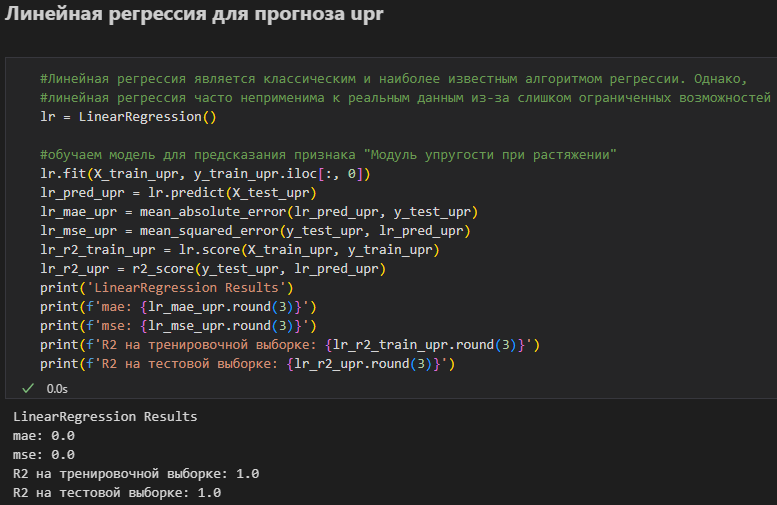


Рисунок 18 — Обучение модели методом линейной регрессии для предсказания признака «Модуль упругости при растяжении, ГПа»

Далее, на рисунке 19, представлен пример обучения модели методом дерева решений (DecisionTreeRegressor), уже с подбором гиперпараметров.

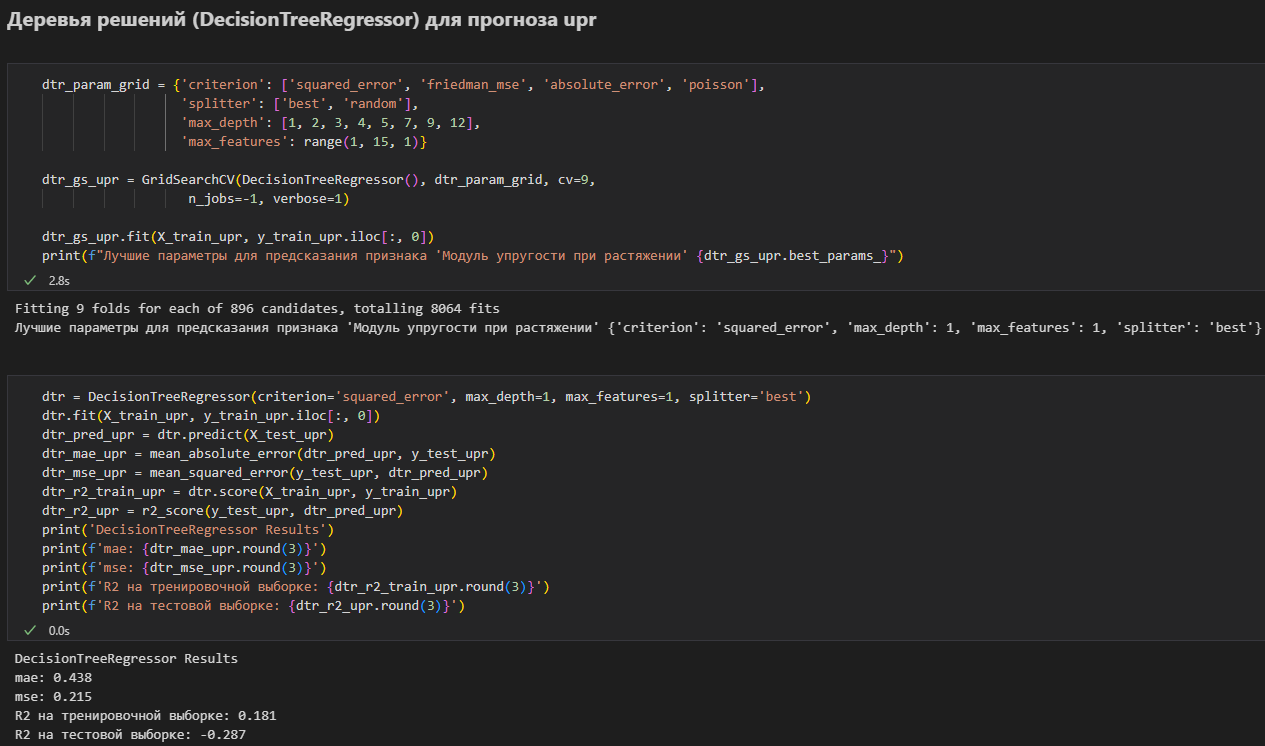


Рисунок 19 — Подбор гиперпараметров и обучение модели дерева решений для предсказания признака «Модуль упругости при растяжении, ГПа»

Для поиска гиперпараметров по сетке мы воспользовались методом GridSearchCV из sklearn. Он получает на вход метод, выбранный нами для решения нашей задачи (регрессор) и набор параметров, для поочерёдной передачи их регрессору, затем выполняет обучение и определяет лучшую комбинацию, что называется гиперпараметрами. Следуя за этим, мы устанавливаем полученные гиперпараметры в наш регрессор и сохраняем это всё как нашу модель.

И так мы разрабатываем и обучаем все модели и переходим к сравнению результатов этих моделей и выбору наилучшей, для последующего сохранения и внедрения в наше веб-приложение, прогнозирующее значения целевых признаков.

Существует множество различных метрик качества. В нашей работе мы используем:

* MAE (Mean Absolute Error) средняя абсолютная ошибка — среднее абсолютное отклонение между прогнозируемыми значениями модели и реальными значениями на тестовых данных;
* MSE (Mean Squared Error) среднеквадратичная ошибка — cреднее арифметическое квадратов разностей между предсказанными и реальными значениями модели машинного обучения;
* R2 или коэффициент детерминации — это статистический показатель, который используется для измерения того, насколько хорошо модель подходит для данных. Он определяет долю дисперсии в данных, которую модель может объяснить. Значение R2 находится в диапазоне от 0 до 1, где 1 означает идеальное соответствие модели с данными, а 0 означает полное расхождение. Если значение R2 низкое, то это может означать, что модель не соответствует данным, либо что данные сами по себе имеют большую ошибку. Отрицательное значение коэффициента детерминации означает, что разработанная вами модель даёт прогноз даже хуже, чем простое усреднение.

Результаты работы наших моделей, для прогноза признака «Модуль упругости при растяжении, ГПа», в виде таблицы, представлены на рисунке 20, а в виде столбчатых диаграмм на рисунке 21.



Рисунок 20 — Результаты моделей для прогноза признака «Модуль упругости при растяжении, ГПа», в виде таблицы

Лучший коэффициент детерминации R2 у нас равен единице, это означает что наша модель идеально подошла для решения нашей задачи. Хуже всего справилась с задачей модель, использующая метод случайного леса.

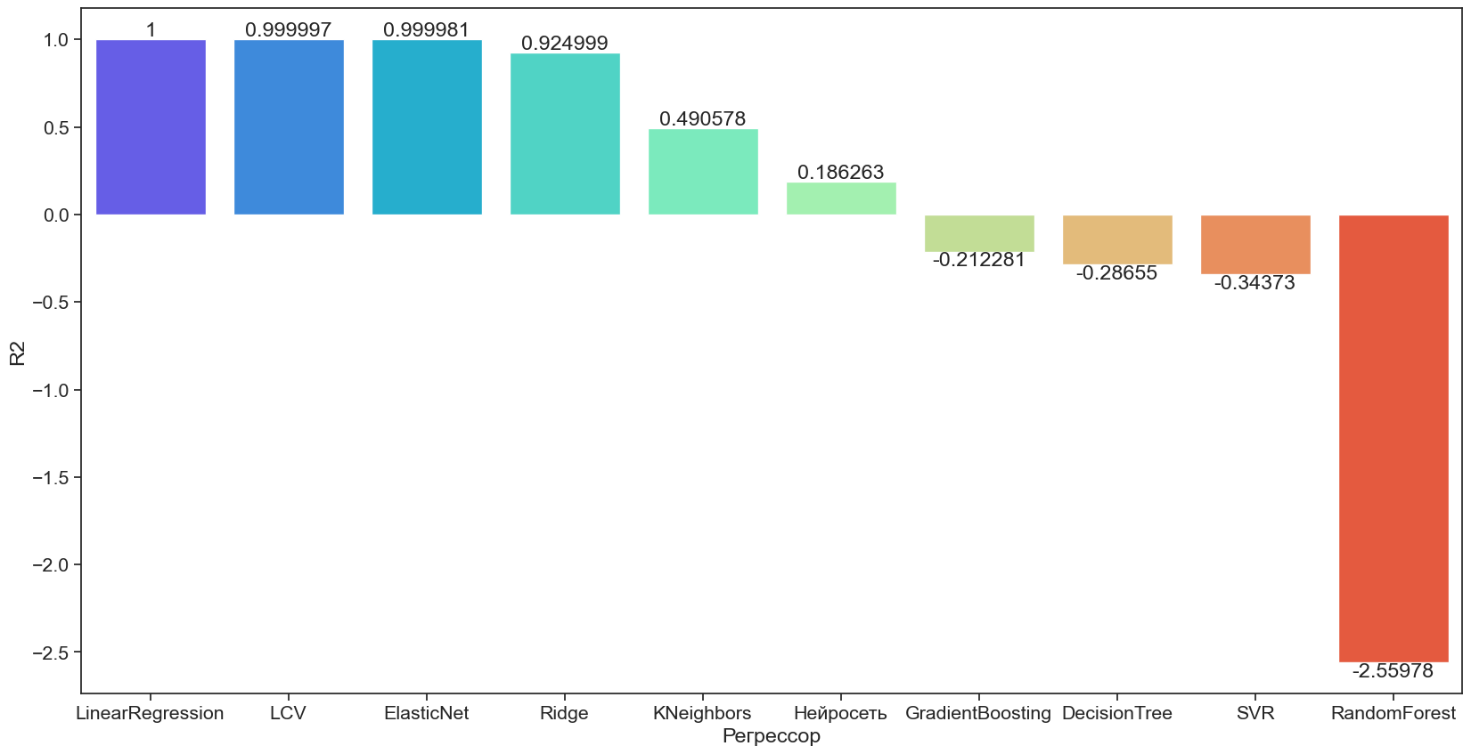


Рисунок 21 — Результаты моделей для прогноза признака «Модуль упругости при растяжении, ГПа», в виде столбчатых диаграмм

Исходя из результатов можно сделать вывод о том, что нам удалось создать модель, прогнозирующую целевой признак «Модуль упругости при растяжении, ГПа». Теперь перейдём к следующей задаче, а именно созданию моделей для прогнозирования признака «Прочность при растяжении, МПа».

1. **Для прогнозирования признака «Прочность при растяжении, МПа»**

В данном разделе мы производим все те же самые действия что и в прошлом, но в качестве целевой переменной у нас уже является признак «Прочность при растяжении, МПа». Поэтому мы покажем, как мы разделили выборку для этой задачи и сразу же перейдём к результатам наших моделей.

На рисунке 22 представлено разделение нашей выборки для прогноза признака «Прочность при растяжении, МПа».

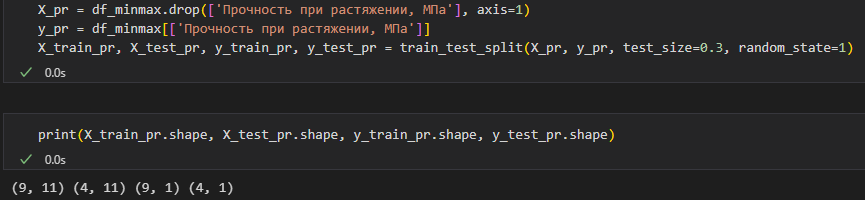


Рисунок 22 — Разделение выборки для прогноза признака «Прочность при растяжении, МПа»

Результаты работы моделей, прогнозирующих «Прочность при растяжении, МПа», в виде таблицы, приведены на рисунке 23.



Рисунок 23 — Результаты моделей, прогнозирующих «Прочность при растяжении, МПа», в виде таблицы

Рисунок под номером 24 отображает результаты тех же моделей, но в виде столбчатых диаграмм.

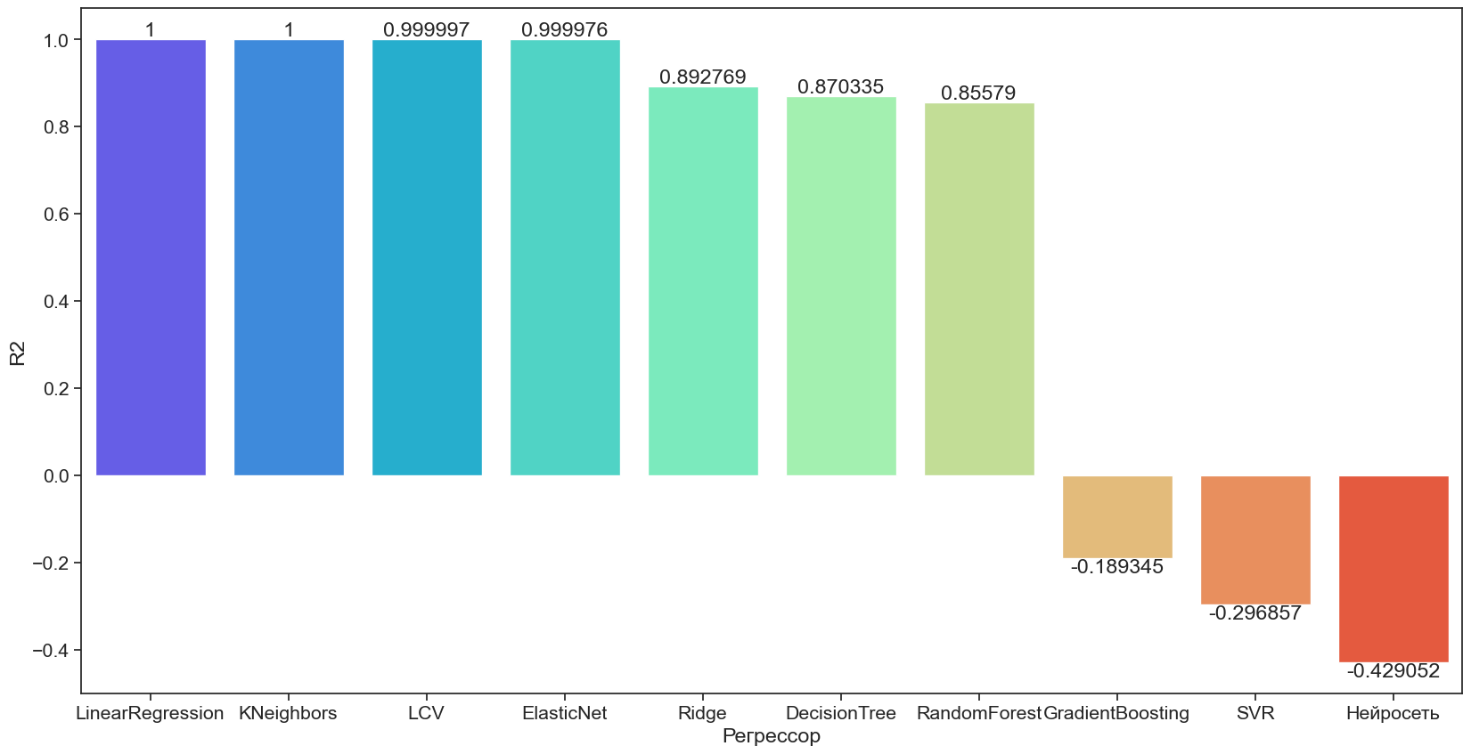


Рисунок 24 — Результаты моделей, прогнозирующих «Прочность при растяжении, МПа», в виде столбчатых диаграмм

Результаты несколько изменились, теперь помимо модели с линейной регрессией, модель, использующая KNeighbors смогла набрать единицу по R2 метрике. В целом, ситуация улучшилась и для большинства остальных методов.

## 2.3 Написание нейронной сети для прогнозирования соотношения матрица-наполнитель

Для построения нейросети воспользуемся библиотекой tensorflow. Строим полносвязную нейронную сеть, в которую входит 2 скрытых слоя по 8 нейронов, с функцией активации «relu» и один выходной слой c функцией активации «linear». Данная архитектура была выбрана, в связи с тем, что она показалась наиболее удачной нам для решения этой задачи.

В качестве оптимизатора берём «Adam», в качестве функции потерь берём «mse».

Архитектура нейросети приведена на рисунке 25.

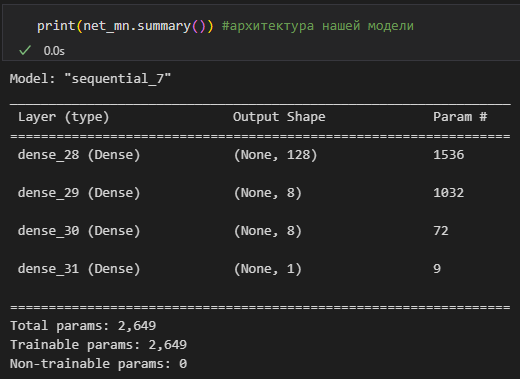


Рисунок 25 — Архитектура нейросети

Запускаю обучение нейросети со следующими параметрами:

* пропорция разбиения данных на тестовые и валидационные: 10%;
* количество эпох: 100.

График обучения нейросети приведен на рисунке 26.

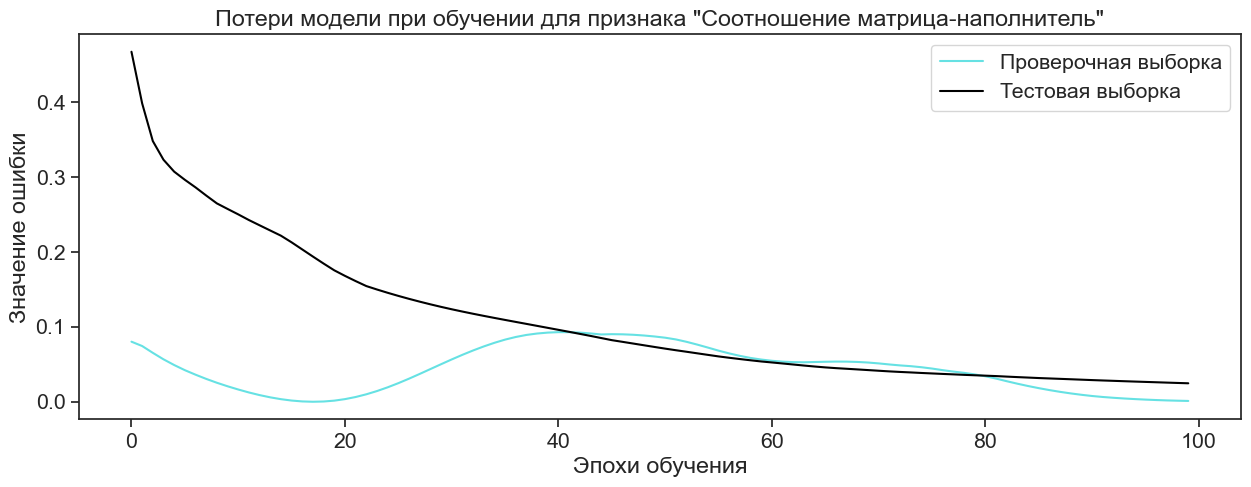


Рисунок 26 — График обучения нейросети

Видно, что значение ошибки валидационной выборки старается аппроксимировать со значением ошибки на тестовой, с каждой эпохой, что является положительным замечанием для нейросети.

На рисунке 27 представлены результаты тестирования нейросети и всех остальных моделей для прогноза признака «Соотношение матрица-наполнитель».



Рисунок 27 — Результаты тестирования нейросети и всех остальных моделей для прогноза признака «Соотношение матрица-наполнитель»

Коэффициент детерминации у всех моделей отрицательный, что говорит нам о непригодности моделей для предсказания признака «Соотношение матрица-наполнитель». Ни одна модель не смогла справиться со своей задачей, но наша поставленная задача была выполнена. Мы написали нейронную сеть, которая рекомендует некоторое соотношение матрица-наполнитель.

Результат работы нашей нейронной сети для этой задачи считать удовлетворительным – нельзя, но в силу того, что нам было дано очень мало пригодных данных, мы можем порадоваться и этим результатам. Таким образом, мы переходим к следующему разделу, где будет описан функционал нашего веб-приложения и краткая инструкция использования.

## 2.4. Разработка приложения

При разработке приложения нам понадобилась библиотека Flask, при помощи которой мы смогли создать и реализовать функционал нашего приложения. Помимо библиотеки Flask мы ещё использовали библиотеку tensorflow и pickle для внедрения наших моделей в приложение.

После запуска приложения мы переходим на главную страницу, которая содержит в себе 3 основных функции:

* Прогнозирование модуля упругости при растяжении;
* Прогнозирование прочности при растяжении;
* Рекомендация соотношения матрица-наполнитель.

Выбираем нужную нам функцию и переходим на страницу заполнения параметров необходимых для вычисления прогноза. Далее заполняем все поля и нажимаем кнопку «Рассчитать», расположенную чуть ниже всех полей для заполнения. После этого получаем спрогнозированное значение для нашего признака. Если нам нужно перейти к расчёту другого признака, то необходимо нажать на кнопку «Вернуться на главную страницу», которая расположена почти в самом низу страницы. После этого выбираем любую другую функцию и переходим на страницу расчёта. Все поля заполняются только числовыми значениями.

Таким образом, данный раздел можно считать выполненным. Скриншоты разработанного веб-приложения приведены в приложении А.

## 2.5. Создание удаленного репозитория

В дополнение к данной работе, был создан удаленный репозиторий на GitHub, который находится по адресу https://github.com/NEEOONU/VKR-DS-2023-BMSTU. На него были загружены все необходимые материалы по нашему заданию. А именно: исследовательский ноутбук, пояснительная записка, файлы приложения, а также необходимые файлы для работы с ноутбуком.

# Заключение

По итогу данной работы можно сделать заключение о том, что нам удалось выполнить все поставленные задачи. Мы создали 30 разных моделей, обучили их и сравнили. После чего нам удалось разработать веб-приложение и внедрить в него лучшие модели, которые предсказывают 3 целевых признака на выбор.

Данные модели, конечно, непригодны для использования в промышленной среде, так как на этапе препроцессинга, мы урезали практически весь датасет и у нас осталось очень мало данных для обучения моделей.

В ходе данной работы мы познакомились с множеством различных статистических методов, использующихся для прогноза. Нам удалось применить каждый выбранный метод в наших моделях, применить гиперпараметры, там, где это возможно и сравнить результаты работы каждого, как в виде таблиц, так и на графиках с диаграммами.

Мы освоили множество методов и библиотек на языке програмирования Python для решения различных задач. Научились писать нейросети, подбирать архитектуры для различных задач и научились создавать приложения для браузера, а также оформлять их.

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

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

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/api/>.

11. Руководство по быстрому старту в flask: – Режим доступа: <https://flask-russian-docs.readthedocs.io/ru/latest/quickstart.html>.

12. Loginom Вики. Алгоритмы: – Режим доступа: <https://wiki.loginom.ru/algorithms.html>.

13. Andre Ye. 5 алгоритмов регрессии в машинном обучении, о которых вам следует знать: – Режим доступа:<https://habr.com/ru/company/vk/blog/513842/>.

14. Alex Maszański. Метод k-ближайших соседей (k-nearest neighbour): – Режим доступа: <https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19>.

15. Yury Kashnitsky. Открытый курс машинного обучения. Тема 3. Классификация, деревья решений и метод ближайших соседей: – Режим доступа: <https://habr.com/ru/company/ods/blog/322534/>.

16. Yury Kashnitsky. Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес: – Режим доступа: <https://habr.com/ru/company/ods/blog/324402/>.

17. Alex Maszański. Машинное обучение для начинающих: алгоритм случайного леса (Random Forest): – Режим доступа: <https://proglib.io/p/mashinnoe-obuchenie-dlya-nachinayushchih-algoritm-sluchaynogo-lesa-random-forest-2021-08-12>.

18. Alex Maszański. Решаем задачи машинного обучения с помощью алгоритма градиентного бустинга: – Режим доступа: <https://proglib.io/p/reshaem-zadachi-mashinnogo-obucheniya-s-pomoshchyu-algoritma-gradientnogo-bustinga-2021-11-25>.

# Приложение А. Скриншоты веб-приложения

Скриншоты веб-приложения, иллюстрирующие его работу. Реализованы были следующие функции:

* Прогнозирование модуля упругости при растяжении;
* Прогнозирование прочности при растяжении;
* Рекомендация соотношения матрица-наполнитель;
* Ввод входных параметров;
* Возвращение на главную страницу.

