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 Сысоев Павел Иванович

Москва, 2022

**Содержание**

[1 Введение 3](#_Toc105950822)

[2 Основная часть 5](#_Toc105950823)

[2.1 Аналитическая часть 5](#_Toc105950824)

[2.1.1 Постановка задачи 5](#_Toc105950825)

[2.1.2 Описание используемых методов 6](#_Toc105950826)

[2.2 Практическая часть 11](#_Toc105950827)

[2.2.1 Разведочный анализ данных 11](#_Toc105950828)

[2.2.2 Предобработка данных 15](#_Toc105950829)

[2.2.3 Разработка и обучение моделей 17](#_Toc105950830)

[2.2.4 Тестирование моделей 21](#_Toc105950831)

[2.2.5 Разработка нейронной сети, которая рекомендует соотношение матрица-наполнитель 23](#_Toc105950832)

[2.2.6 Разработка приложения 27](#_Toc105950833)

[2.2.7 Создание удаленного репозитория 28](#_Toc105950834)

[3 Заключение 29](#_Toc105950835)

[4 Библиографический список 30](#_Toc105950836)

[5 Приложения 33](#_Toc105950837)

# **1 Введение**

Композиционные материалы – это искусственно созданные материалы, состоящие из нескольких других с чёткой 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).

Перспективным современным материалом является базальтопластик (далее – БП) – композитный материал на основе базальтовых волокон и органического связующего. В качестве армирующей основы в БП служат базальтовые нити, ткани, холсты, маты, а связующей матрицей – неорганические полимеры.

БП обладает уникальными характеристиками, благодаря которым их применяют в качестве строительных материалов, арматуры и конструкций, ответственных изделий в машино- и авиастроении, кровельных рулонированных материалов, магистральных трубопроводов и др.

Базальтовое волокно производят из различных горных пород близких по химическому составу - базальта, базанитов, амфиболитов, габродиабазов или их смесей, средний химический состав которых включает (% по массе): SiO2 (47,5‑55,0); TiO2 (1,36-2,0); AlО3 (14,0-20,0); Fе2O3 + FeО (5,38-13,5); МnО (0,25-0,5); МgО (3,0-8,5); CaО (7,-11,0); Na2O (2.7-7,5); К2О (2,5-7,5); Р2О5 (не более 0,5); SО3 (не более 0,5); прочие породы (не более 5) [1].

Развитие производств базальтовых волокон обусловлено сочетанием следующих определяющих факторов:

- комплекс механических свойств: термостойкость, хемостойкость, биоинертность, негорючесть. БП выдерживает длительное воздействие температуры до 700°С и кратковременное воздействие до 1000 °С (стекловолокно теряет прочность при температуре выше 300 °С)[2].

- долговечность, коррозионно-, щелоче- и кислотостойкость, стабильность состояния. Изделия из этого материала служат более 100 лет без потери качеств.

- наличие в разных регионах России неограниченных запасов сырья. Базальт – горная порода, составляющая 30% земной коры, его запасы неисчерпаемы.

Свойства БП определяются как характеристиками применяемого волокна, так и свойствами полимерного связующего. Задачей данного исследования является создание модели машинного обучения, позволяющей прогнозировать зависимость ряда конечных свойств получаемого композиционного материала БП.

Созданные прогнозные модели могут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками композитов.

# **2 Основная часть**

**2.1 Аналитическая часть**

**2.1.1 Постановка задачи**

Решение задачи машинного обучения состоит из следующих основных этапов: формулирование проблемы, сбор и предобработка данных, разработка и построение модели машинного обучения, тестирование, оценка качества, постобработка результатов, внедрение эксплуатацию.

Решаемая нами задача имеет следующей вид: имеется множество объектов 𝑋 – элементов выборки, и множество ответов 𝑌 – переменных параметров элементов выборки.

В нашем случае объектами являются конкретные образцы композиционного материала, а переменными параметрами – их физические свойства, зафиксированные по результатам измерительных испытаний.

Предполагается, что существует функциональная зависимость 𝑓: 𝑋 → 𝑌 между объектами и ответами, но она неизвестна. Известна лишь совокупность пар вида (объект, ответ), называемая выборкой данных. Требуется найти приближенный вид этой 𝑓 путём построения аппроксимирующей функции. Или другими словами, по известным значениям 𝑦𝑖 некоторой неизвестной функции в заданных точках 𝑥𝑖 (𝑖 = 1, . . . , 𝑙) предсказать значения этой функции в других точках.

Такое прогнозирование конечных свойств новых материалов на основании имеющихся данных испытаний относится к задаче регрессии.

Оценка качества модели машинного обучения, решающей задачу регрессии, заключается в определении величины отклонения (ошибки) ответа, который предсказывает модель от реального значения показателя. Чем меньше такое отклонение, тем лучше модель.

Существует несколько метрик для определения ошибки модели предсказания. Для оценки качества моделей регрессии мы будем использовать метрики:

Коэффициент детерминации (R2) – квадрат коэффициента корреляции. показывает насколько хорошо регрессионная модель описывает данные. R2 равный 1, означает что функция идеально ложится на все точки – данные идеально описаны моделью. Метрика помогает понять, какую долю данных модель смогла объяснить.

Средняя абсолютная ошибка (МАЕ) – сумма модулей разницы между прогнозом и реальным значением для всех объектов деленная на число всех объектов. Преимуществом данной метрики является удобство интерпретации, т.к. погрешность измеряется в тех же единицах, что и значения целевых переменных. Подходит для нашей задачи, когда требуется сравнить разные модели, предсказывающие одно и то же по одинаковым признакам.

**2.1.2 Описание используемых методов**

Для решения поставленной задачи мы рассмотрим следующие методы:

1) Линейная регрессия – это метод, используемый для моделирования отношений между одной или несколькими независимыми входными переменными функции и выходной зависимой переменной.

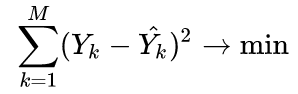
Регрессия выбирает из семейства функций, функцию с минимальной функцией потерь, характеризующей насколько сильно пробная функция отклоняется от значений в заданных точках.

Цель регрессии — найти коэффициенты этой линейной комбинации, и тем самым определить регрессионную функцию, которую также называют моделью.

На практике линия регрессии чаще всего представлена в виде линейной функции, наилучшим образом приближающей искомую кривую:

 (1)

Поиск коэффициентов регрессии выполняется с помощью метода наименьших квадратов, когда минимизируется сумма квадратов отклонений реально наблюдаемых Y от их оценок Ŷ (имеются в виду оценки с помощью прямой линии, претендующей на то, чтобы представлять искомую регрессионную зависимость):

, (2)

где M – объём выборки.

Этот подход основан на том известном факте, что фигурирующая в приведённом выражении сумма принимает минимальное значение именно для того случая, когда

. (3)

Также нами будет рассмотрена полиномиальная регрессия, которая позволяет сделать модель нелинейной комбинацией входных переменных, т.е. среди них могут быть экспоненциальные переменные: синус, косинус и т. п.

Преимущества линейной регрессии:

- высокая скорость;

- модель может работать на очень больших выборках;

- линейная регрессия хорошо изучена и достаточно проста в понимании, имеет небольшое количество гиперпараметров.

Недостатки линейной регрессии:

- в случае нелинейных данных полиномиальную регрессию трудно спроектировать. Необходимо иметь дополнительную информацию о структуре данных и взаимосвязи между переменными.

- плохо работают в задачах, в которых зависимость ответов от признаков сложная, нелинейная.

2) Случайный лес. Суть алгоритма случайного леса состоит в использовании ансамбля решающих деревьев. Из-за большого количества решающих деревьев, результат модели значительно улучшается по сравнению с единичным решающим деревом.

Алгоритм построения случайного леса, состоящего из N деревьев, выглядит следующим образом. Для каждого n = 1, …, N сгенерировать выборку Xn с помощью статистического бутстрэпа, построить решающее дерево bn по выборке Xn:

– по заданному критерию выбирается лучший признак, по которому выполняется разбиение в дереве, и так до исчерпания выборки;

– дерево строится, пока в каждом листе не более nmin объектов или пока не будет достигнута определённая высота дерева;

– при каждом разбиении сначала выбирается m случайных признаков из n исходных, и оптимальное разделение выборки ищется только среди них.

Итоговый классификатор для задачи регрессии выбирает решение по среднему значению.

Преимущества алгоритма:

– имеет высокую точность предсказания. Не требует тщательной настройки параметров, хорошо работает со значениями по умолчанию.

– редко переобучается. На практике добавление деревьев только улучшает композицию. В случае наличия проблемы переобучения, она преодолевается путем усреднения или объединения результатов различных деревьев решений.

– способен эффективно обрабатывать данные с большим числом признаков и классов.

– одинаково хорошо обрабатывает как непрерывные, так и дискретные признаки

Недостатки алгоритма:

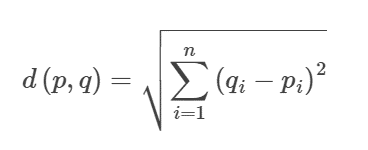
– для реализации алгоритма случайного дерева требуется значительный объем вычислительных ресурсов. Построение случайного леса отнимает больше времени, чем деревья решений или линейные алгоритмы.

3) Метод k-ближайших соседей (k Nearest Neighbors, kNN) – популярный алгоритм классификации, который используется в разных типах задач машинного обучения. На интуитивном уровне суть метода проста: посмотри на соседей вокруг, какие из них преобладают, таковым ты и являешься.

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

Основной математической составляющей алгоритма k-ближайших соседей является евклидово расстояние – метрика в евклидовом пространстве, расстояние между двумя точками, вычисляемое по теореме Пифагора. Проще говоря, это наименьшее возможное расстояние между точками A и B.

Формула вычисления Евклидова расстояния:

 (4)

Другой важной составляющей метода является нормализация. Разные атрибуты обычно обладают разным диапазоном представленных значений в выборке. К примеру, атрибут А представлен в диапазоне от 0.01 до 0.05, а атрибут Б представлен в диапазоне от 500 до 1000). В таком случае значения дистанции могут сильно зависеть от атрибутов с большими диапазонами. Поэтому данные в большинстве случаев проходят через нормализацию.

Плюсы алгоритма:

- простая реализация;

- можно адаптировать под нужную задачу выбором метрики или ядра. Ядро может задавать операцию сходства для сложных объектов, а сам подход kNN остаётся тем же.

- неплохая интерпретация, можно объяснить, почему тестовый пример был классифицирован именно так.

Минусы алгоритма:

- алгоритм работает значительно медленнее при увеличении объема выборки, предикторов или независимых переменных.

- требуется предварительная работа с данными: нормализация, обработка категориальных признаков.

- зависимость от выбранной метрики расстояния между примерами. Выбор по умолчанию евклидового расстояния чаще всего ничем не обоснован.

- нет теоретических оснований выбора определённого числа соседей – только перебор.

В случае малого числа соседей метод чувствителен к выбросам, то есть склонен переобучаться;

4) Полносвязная нейронная сеть. Нейронная сеть состоит из взаимосвязанных групп узлов, называемых нейронами. Входные данные передаются в эти нейроны в виде линейной комбинации со множеством переменных. Значение, умножаемое на каждую функциональную переменную, называется весом. Затем к этой линейной комбинации применяется нелинейность, что даёт нейронной сети возможность моделировать сложные нелинейные отношения. Чаще всего нейронные сети бывают многослойными: выход одного слоя передаётся следующему так, как описано выше. На выходе нелинейность не применяется.

Нейронные сети тренируются с помощью алгоритма обратного распространения ошибки.

Преимущества:

– так как нейронные сети могут быть многослойными, они эффективны при моделировании сложных нелинейных отношений;

– не нужно беспокоиться о структуре данных в нейронных сетях, которые очень гибки в изучении почти любого типа переменных признаков;

– исследования постоянно показывают: чем больше обучающих данных подают нейронной сети, тем производительнее она становится.

Недостатки:

– из-за сложности этой модели, её нелегко понять и реализовать;

– нейронные сети требуют тщательной настройки гиперпараметров и скорости обучения;

– для достижения высокой производительности нейронным сетям необходимо огромное количество данных, и в результате, как правило, нейросети уступают другим алгоритмам машинного обучения в тех случаях, когда данных мало.

**2.2 Практическая часть**

Код практической части выполнен в среде Jupiter Notebook на Google Colaboratory. Для решения задач практической части использованы библиотеки: numpy, pandas, matplotlib, seaborn, sklearn, tensorflow.

Исходные данные представлены в виде двух файлов в формате «.xlsx»: «X\_bp.xlsx» и «X\_nup.xlsx», содержащих параметры базальтопластика и нашивок углепластиковых соответственно.

При загрузке данных в среду выполнения произведено объединение файлов (электронных таблиц) по индексу с помощью метода .join с параметром inner.

Объединённая электронная таблица (исходный набор данных) содержит информацию о 13 параметрах для 1023 элементов. Пропусков в исходном наборе данных не выявлено. Исходный набор данных определён в коде как «raw\_dataset».

**2.2.1 Разведочный анализ данных**

На входе имеются данные о свойствах (параметрах) композиционных материалов. Набор данных имеет 13 параметров, информация о параметрах представлена в Таблице 1.

Таблица 1 – параметры объектов входных данных

|  |  |  |
| --- | --- | --- |
| Наименование параметра  (в соответствии с входными данными) | Количество непустых значений | Тип данных |
| 1 | 2 | 3 |
| 1. Соотношение матрица-наполнитель | 1023 | float64 |
| 2. Плотность, кг/м3 | 1023 | float64 |
| 3. модуль упругости, ГПа | 1023 | float64 |
| 4. Количество отвердителя, м.% | 1023 | float64 |
| 5. Содержание эпоксидных групп,%\_2 | 1023 | float64 |
| 6. Температура вспышки, С\_2 | 1023 | float64 |
| 7. Поверхностная плотность, г/м2 | 1023 | float64 |
| 8. Модуль упругости при растяжении, ГПа | 1023 | float64 |
| Продолжение таблицы 1 | | |
| 1 | 2 | 3 |
| 9. Прочность при растяжении, МПа | 1023 | float64 |
| 10. Потребление смолы, г/м2 | 1023 | float64 |
| 11. Угол нашивки, град | 1023 | float64 |
| 12. Шаг нашивки | 1023 | float64 |
| 13. Плотность нашивки | 1023 | float64 |

При построении моделей машинного обучения в качестве выходных параметров применяются переменные «Модуль упругости при растяжении, ГПа» и «Прочность при растяжении, МПа». Все остальные параметры рассматриваются как входные.

Для первичного анализа данных использован метод библиотеки pandas «.describe», который возвращает основные статистические показатели заданного массива данных. Результат применения метода представлен на рисунке 1.

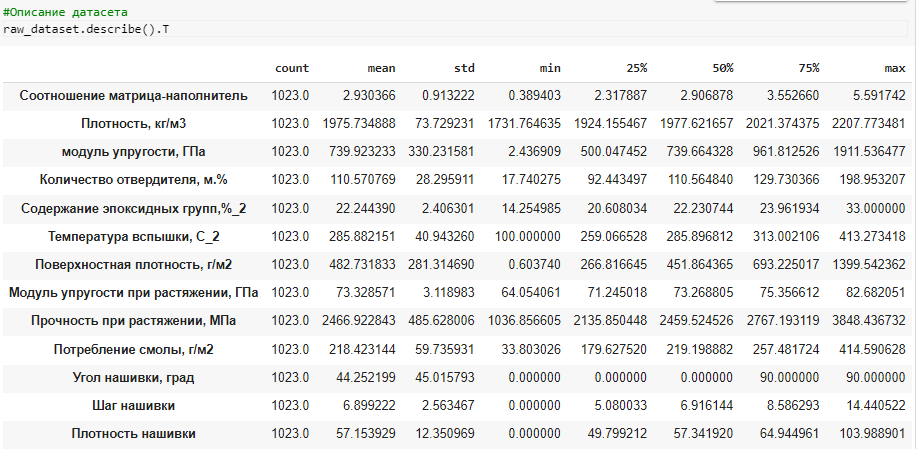


Рисунок 1 – Первичный анализ данных

Как видно из рисунка, минимальные и максимальные значения параметров имеют большой разброс – как от 0 до 6, так и от 2466 до 3848, поэтому мы можем сделать вывод о том, что требуется выполнить нормализацию данных.

С помощью библиотек matplotlib и seaborn выполнено построение гистограммы распределения и диаграммы boxplot («ящик с усами») для каждого параметра.

Для 12 из 13 параметров распределение имеет вид нормального распределения.

Для параметра «Поверхностная плотность, г/м2» наблюдается смещение распределения «влево» (рисунок 2)

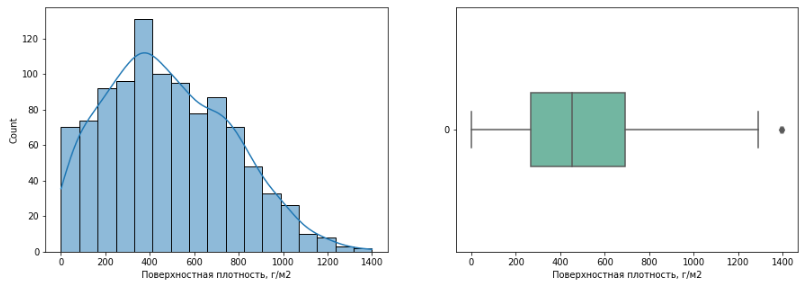


Рисунок 2 – Распределение для параметра «Поверхностная плотность, г/м2»

Параметр «Угол нашивки, град» принимает дискретные значения 0 и 90 (рисунок 3).

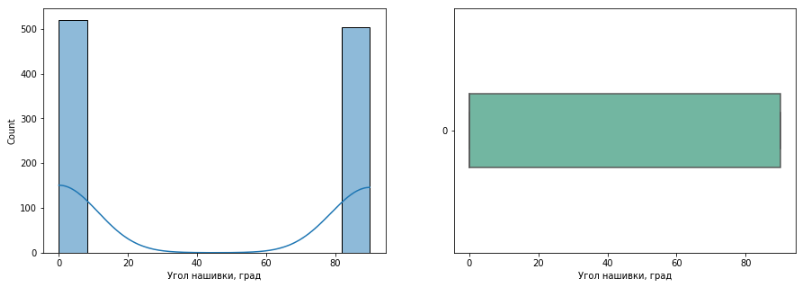


Рисунок 3 – Распределение для параметра «Угол нашивки, град»

Диаграмма boxplot позволяет выявить наличие выбросов в данных. Анализ диаграмм показывает, что требуется выполнить очистку данных от выбросов (рисунок 4).

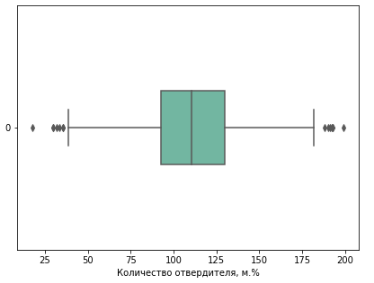
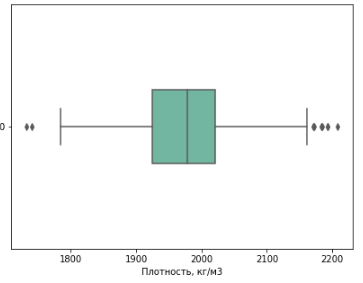


Рисунок 4 – Примеры графиков Boxplot с выбросами

С помощью библиотеки seaborn выполнено построение попарных графиков рассеивания точек (Приложение 1). Из построенных графиков видно, что в представленных данных нет каких-либо ярко выраженных зависимостей и закономерностей взаимного влияния показателей.

Тепловая карта коэффициентов корреляции также показывает слабую зависимость между параметрами исследуемых объектов (рисунок 5).

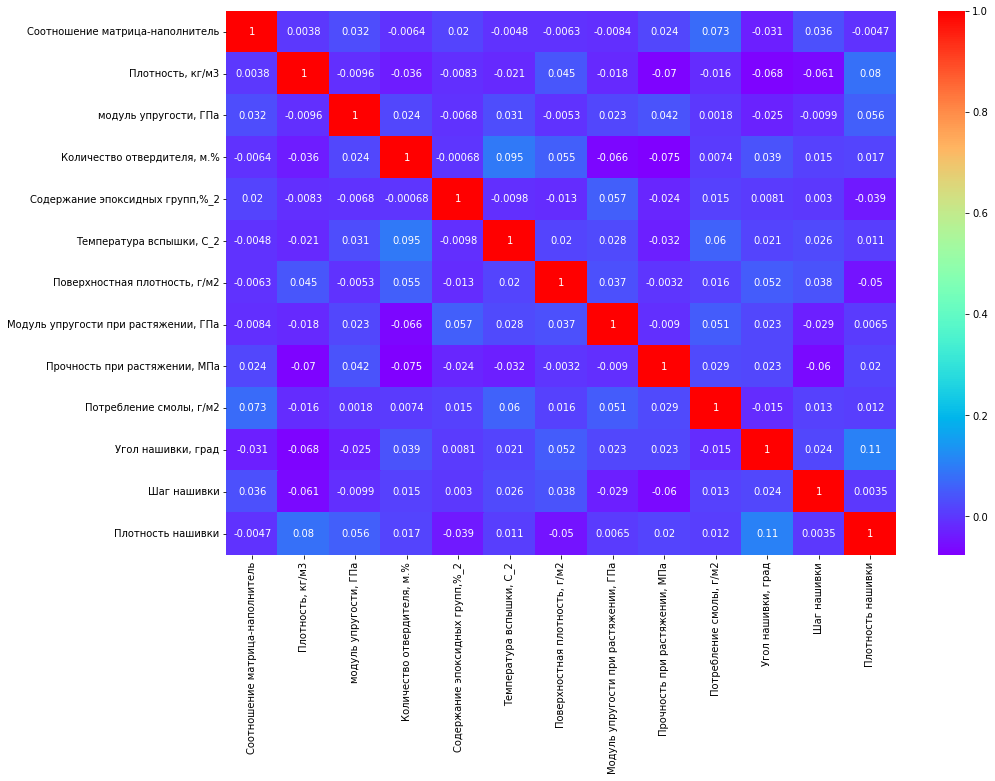


Рисунок 5 – Тепловая карта коэффициентов корреляции

**2.2.2 Предобработка данных**

В первую очередь, выполним очистку данных от выбросов. Выбросы – это данные, которые существенно отличаются от других наблюдений. Они могут соответствовать реальным отклонениям, но могут быть и просто ошибками.

Удаление выбросов выполнено с помощью межквартильных диапазонов

IQR = Quartile3 – Quartile1

Для выявления выбросов для каждого параметра определяются значения, находящиеся выше и ниже нормального диапазона наборов данных, т.е. за пределами верхней и нижней границы:

- верхняя граница = Q3+1.5\*IQR;

- нижняя граница = Q1-1.5\*IQR.

Строки с выбросами исключены из набора данных и дальнейшего исследования (Рисунок, нормализованные данные). В результате удаления выборки, удалено 87 элементов, размерность очищенных данных составляет 936 на 13. Повторное формирование диаграммы boxplot показывает отсутствие выбросов (рисунок 6).

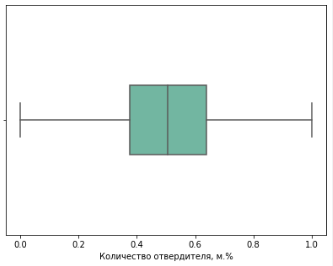
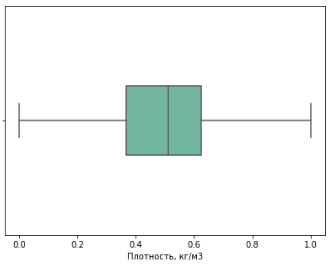


Рисунок 6 – Графики Boxplot без выбросов

Построим графики распределения для каждого параметра, чтобы удостовериться в необходимости нормализации данных (рисунок 7)

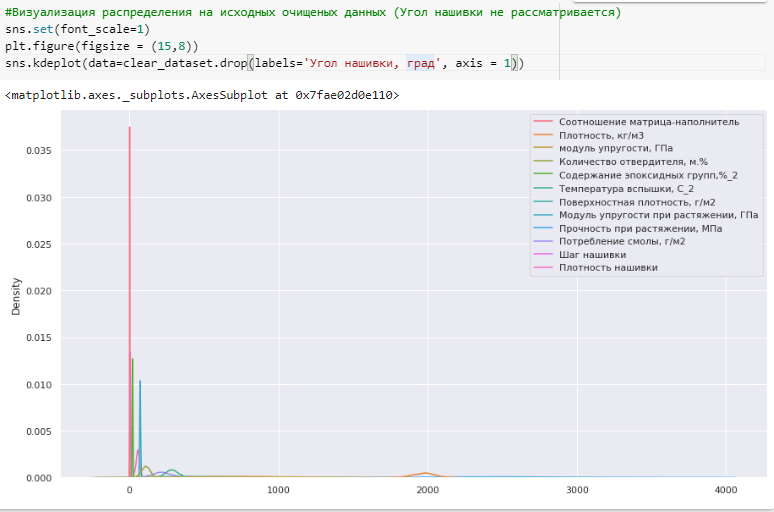


Рисунок 7 – Графики распределения исходных данных

Нормализация данных проведена с помощью класса MinMaxScaler библиотеки sklearn с диапазоном нормализованных значений от 0 до 1. Результат нормализации представлен на рисунке 8



Рисунок 8 – Анализ нормализованных данных

После проведённой нормализации данные принимают сопоставимые значения (рисунок 9).

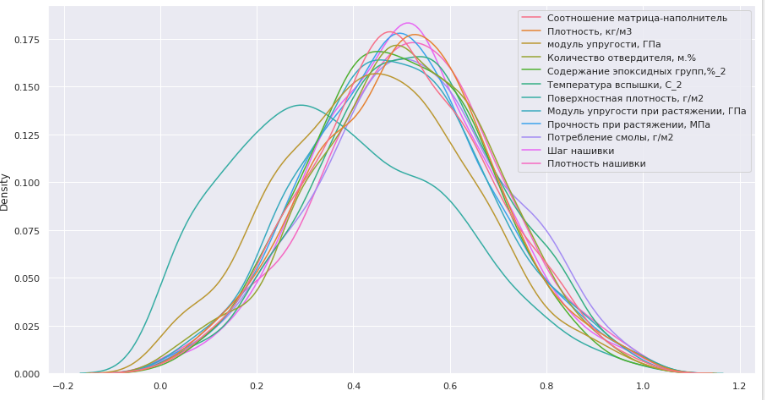


Рисунок 9 – Графики распределения нормализованных данных

Для обучения и проверки качества моделей, набор данных будет разделён на обучающую и тестовую выборки объёмом 70% и 30% соответственно с помощью метода train\_test\_split библиотеки sklearn.

**2.2.3 Разработка и обучение моделей**

Для упрощения процесса разработки моделей машинного обучения, они будут строиться и обучаться на основе функции ml\_model\_func, которая принимает на вход следующие параметры:

- regressor – класс метода машинного обучения, выбранный алгоритм;

- param\_grid\_func – словарь, содержащий набор гиперпараметров модели;

- data – данные, которые будут подаваться в модель;

- y\_features – выходные параметры модели.

В функцию зашиты следующие действия:

1) разбиение данных (data) на обучающую и тестовую выборки, при этом применяется параметр random\_state для того, чтобы фактор случайного разбиения данных не влиял на оценку качества построенных моделей;

2) создание модели (regr = regressor);

3) подбор гиперпараметров с помощью GridSearchCV;

4) и обучение модели на тренировочной выборке с помощью метода .fit(X\_train, y\_train);

5) предсказание ответов на лучших параметрах модели методом .predict;

6) визуализация результатов с помощью средств библиотеки matplotlib.

Функция циклом повторяет все эти манипуляции для всех заявленных выходных параметров y\_features. Мы будем подавать в функцию список из двух параметров – «Прочность при растяжении, МПа», «Модуль упругости при растяжении, ГПа».

Перед запуском функции мы также будем объявлять переменную (словарь) param\_grid, которая содержит набор параметров для определённой модели.

1) Построение модели LinearRegression.

Определяем входную переменную regressor = LinearRegression().

Для линейной регрессии мы предоставим GridSearchCV небольшое количество параметров:

fit\_intercept – вычислять отрезок b₀ (True, по умолчанию) или рассматривать его как равный нулю (False).

positive – принудительные положительные значения коэффициентов (True) или нет (False, по умолчанию).

Программный код для построения модели линейной регрессии выглядит следующим образом:

param\_grid = {'fit\_intercept': [True, False],

                'positive': [False, True]}

ml\_model\_func(LinearRegression(), param\_grid,

              norm\_dataset, ['Прочность при растяжении, МПа', 'Модуль упругости при растяжении, ГПа'])

2) Построение модели LinearRegression c применением полиномиальных параметров. Отличие данного метода от рассмотренного выше заключается в применении к входным переменным объекта PolynomialFeatures, который позволяет добавить нелинейность в модель линейной регрессии.

Данная модель построена нами без применения функции ml\_model\_func, т.к. имеет частное действие – применение полиномиальных параметров к входным данным:

  X\_trainP = PolynomialFeatures(degree=4).fit\_transform(X\_train)

  X\_testP = PolynomialFeatures(degree=4).fit\_transform(X\_test)

Параметр степени полинома «degree» подбирался ручным перебором.

3) Построение модели RandomForestRegressor.

Определяем входную переменную regressor = RandomForestRegressor().

Важными гиперпараметрами для построения модели являются:

- число деревьев – n\_estimators. Чем больше деревьев, тем лучше качество. однако время настройки и работы Random Forest будут увеличиваться пропорционально количеству деревьев, что может сказаться на производительности;

- критерий расщепления – criterion. В библиотеке sklearn для задач регрессии реализованы два критерия «mse» и «mae», которые являются функциями ошибок Mean Square Error и Mean Absolute Error соответственно;

- число признаков для выбора расщепления – max\_features. При увеличении max\_features увеличивается время построения леса, а деревья становятся похожими друг на друга. В задачах регрессии он по умолчанию равен n/3.

- максимальная глубина деревьев – max\_depth. Чем меньше максимальная глубина, тем быстрее строится и работает алгоритм случайного дерева. При увеличении глубины резко возрастает качество как на обучении модели, так и на ее тестировании. Если позволяют ресурсы, то рекомендуется использовать максимальное значение данного параметра. Неглубокие деревья рекомендуется использовать в задачах со значительным количеством шумовых объектов (выбросов).

Программный код для построения модели случайного леса выглядит следующим образом:

param\_grid = {'n\_estimators': [10, 50, 100],

              'criterion':   ["mse", "mae"],

              'max\_depth':   [5, 15, 100],

              'max\_features': [3, 4, 12],

              'n\_jobs': [-1],

              }

ml\_model\_func(RandomForestRegressor(), param\_grid, norm\_dataset,

              ['Прочность при растяжении, МПа', 'Модуль упругости при растяжении, ГПа'])

4) Построение модели KNeighborsRegressor.

Определяем входную переменную regressor = KNeighborsRegressor()

В модели kNN гиперпараметрами обычно выступают:

- число соседей (n\_neighbors) – не существует конкретного способа определить наилучшее значение для этого параметра, поэтому нам нужно попробовать несколько значений, чтобы найти лучшее из них. Низкое значение k, например, 1 или 2, может привести к эффекту недообучения модели. Высокое значение k может привести к трудностям с производительностью модели, а также повышению риска переобучения. Чаще всего наиболее предпочтительным значением для k является 5, мы попробуем 5, 50 и 150;

- степень для метрики Минковского (p) – 1 или 2 чаще всего будут лучшими.

- веса (weights) – параметр, который позволяет определить степень влияния расстояния между точками: «uniform» - все точки по соседству принимают одинаковые веса и «distance» ближайшие точки будут иметь большее влияние на результат.

Программный код для построения модели kNN выглядит следующим образом:

param\_grid = {'n\_neighbors': [5, 50, 150],

              'weights':   ['uniform', 'distance'],

              'p': [1,2],

              'metric': ["minkowski"],

              'n\_jobs':[-1]

                }

ml\_model\_func(KNeighborsRegressor(), param\_grid,

              norm\_dataset, ['Прочность при растяжении, МПа', 'Модуль упругости при растяжении, ГПа'])

5) Построение нейронной сети для решения задачи регрессии.

Модель нейронной сети является самой сложной для постронения и настройки из рассматриваемых моделей машинного обучения.

Результат работы сети зависит многих параметров: количество слоёв, количество нейронов в слоях, функций активации, алгоритма оптимизации и скорости обучения и др.

Подбор параметров нейронной сети в большинстве случаев проводится перебором и выявлением лучших параметров для каждой конкретной задачи. Для решения большинства задач достаточно одного скрытого слоя, т.к. слишком большая внутренняя структура может сделать выходные значения непредсказуемыми, однако в случае, когда зависимость элементов сложная одного скрытого слоя может быть недостаточно для полноценного обучения модели.

В рамках данной работы мы проверим следующие рекомендации по построению структуры нейронных сетей:

- количество нейронов входного слоя должно соответствовать количеству входных параметров объектов выборки;

- количество нейронов в скрытых слоях должно быть около 2/3 количества нейронов входного слоя или не более половины размера входного слоя.

Таким образом, предлагается рассмотреть и сравнить нейронные сети со структурой 12‑6‑1 и 12‑6-6-1. Для минимизации влияния переобучения будет добавлен слой Dropout после первого скрытого слоя. В качестве функций активации в скрытых слоях будет использоваться relu и линейная функция активации на выходном слое.

**2.2.4 Тестирование моделей**

Результаты тестирования моделей представлены в Таблице 2 и на рисунках 10 и 11.

Таблица 2 – Результаты тестирования моделей

| **Модель** | **Выходной параметр** | **Лучшие параметры** | **Оценка** | |
| --- | --- | --- | --- | --- |
| **MAE** | **R2 score** |
| 1 | 2 | 3 | 4 | 5 |
| Линейная регрессия | Прочность при растяжении | по умолчанию | 0.1446 | -0.0259 |
| Модуль упругости при растяжении | positive=True | 0.1456 | 0.0038 |
| Линейная регрессия с полиномиальными параметрами | Прочность при растяжении | fit\_intercept=False, positive=True  degree = 3 | 0.1441 | 0.0062 |
| Модуль упругости при растяжении | positive=True  degree = 2 | 0.1455 | -0.0006 |
| Случайный лес | Прочность при растяжении | criterion='mse', max\_depth=5, max\_features=3  n\_estimators=50, warm\_start=True | 0.1454 | -0.0176 |
| Модуль упругости при растяжении | criterion='mae', max\_depth=15, max\_features=3 | 0.1456 | -0.0069 |
| К-ближайших соседей | Прочность при растяжении | n\_neighbors=150, p=1, weights='distance' | 0.1456 | -0.0137 |
| Модуль упругости при растяжении | n\_neighbors=150,  p=1 | 0.1458 | 0.0002 |
| Нейронная сеть  12-6-1 | Прочность при растяжении | learning\_rate=0.0035  epochs=20  validation\_split = 0.3  batch\_size=45 | 0.1439 | -0.0053 |
| Модуль упругости при растяжении | 0.1454 | -0.0220 |
| Нейронная сеть  12-6-6-1 | Прочность при растяжении | learning\_rate=0.004  epochs=30  validation\_split = 0.3  batch\_size=45 | 0.1438 | 0.0085 |
| Модуль упругости при растяжении | 0.1438 | 0.0216 |

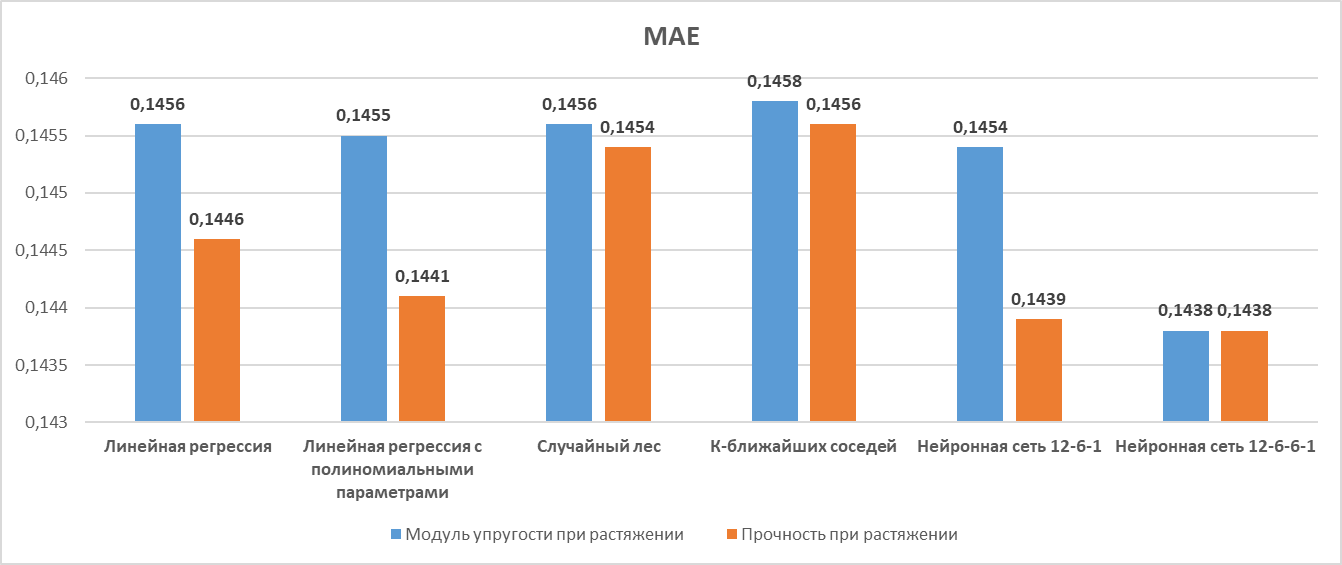


Рисунок 10 – Оценка моделей по средней абсолютной ошибке

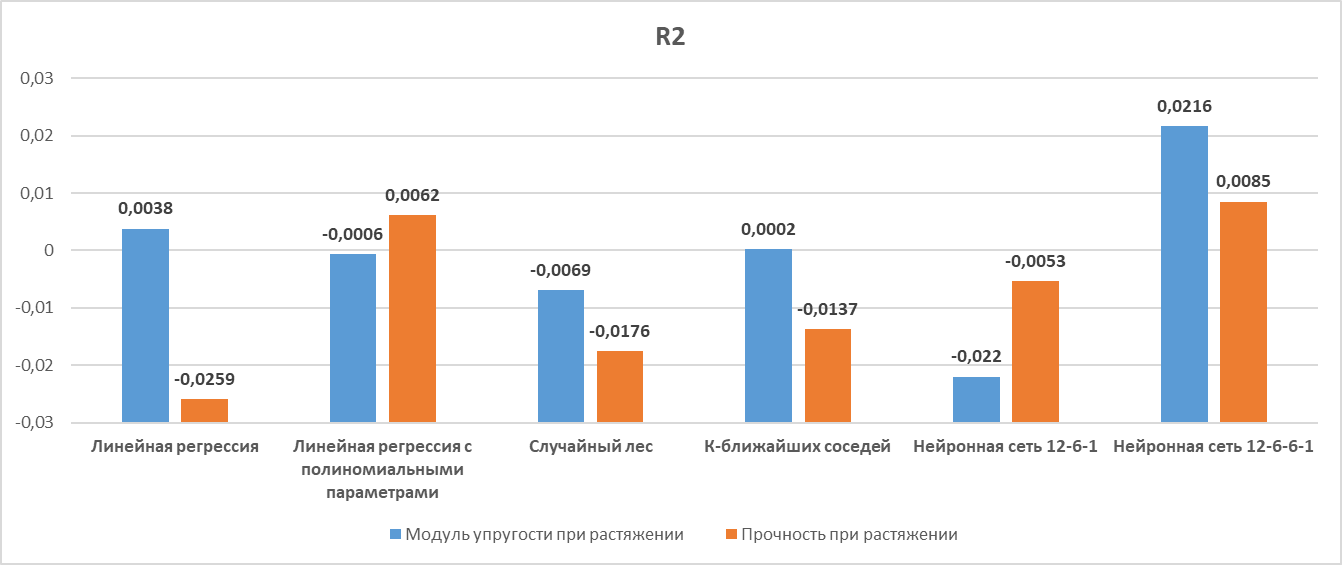


Рисунок 11 – Оценка моделей по коэффициенту детерминации

Лучшие метрики удалось получить на модели нейронной сети с архитектурой 12-6-6-1.

Диаграммы с визуализацией результатов тестирования каждой модели представлены в Приложении 2.

**2.2.5 Разработка нейронной сети, которая рекомендует соотношение матрица-наполнитель**

Для получения рекомендации соотношения матрица-наполнитель мы будем использовать разработанную выше нейронную сеть с параметрами 12‑6‑6‑1:

Архитектура сети и оценка модели на параметре «Соотношение матрица-наполнитель» представлена на рисунке 12.

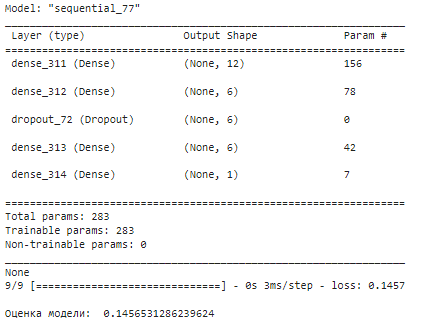


Рисунок 12 – Архитектура нейронной сети для прогноза параметра «Соотношение матрица-наполнитель»

Результаты обучения нейронной сети и прогнозирования значений представлены на рисунках 13 и 14:

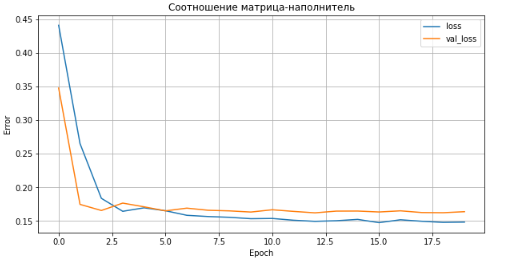


Рисунок 13 – График обучения нейронной сети

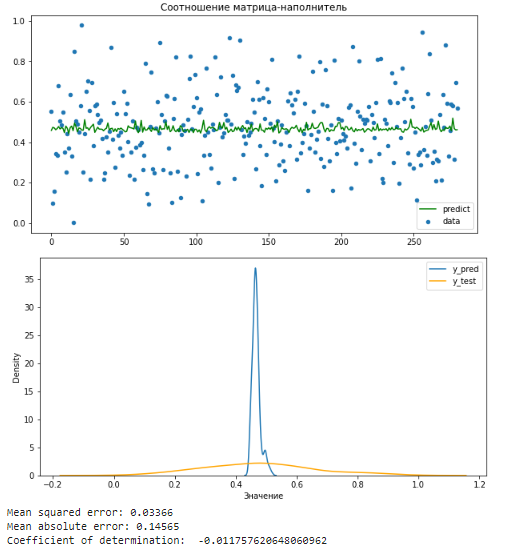


Рисунок 14 – Диаграммы соответствия прогнозных и тестовых данных

Смоделируем работу сети по прогнозированию показателя в тетради Google Colab:

1) Ввод параметров и получение списка с параметрами

print("Введите параметры для прогноза соотношения 'Матрица-наполнитель'")

print("Плотность, кг/м3")

f1 = np.float64(input())

print("Модуль упругости, ГПа")

f2 = np.float64(input())

print("Количество отвердителя, м.%")

f3 = np.float64(input())

print("Содержание эпоксидных групп,%\_2")

f4 = np.float64(input())

print("Температура вспышки, С\_2")

f5 = np.float64(input())

print("Поверхностная плотность, г/м2")

f6 = np.float64(input())

print("Модуль упругости при растяжении, ГПа")

f7 = np.float64(input())

print("Прочность при растяжении, МПа")

f8 = np.float64(input())

print("Потребление смолы, г/м2")

f9 = np.float64(input())

print("Угол нашивки, град")

f10 = np.float64(input())

print("Шаг нашивки")

f11 = np.float64(input())

print("Плотность нашивки")

f12 = np.float64(input())

args = np.array([[f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12]])

Вывод при запуске данного кода представлен на рисунке 15

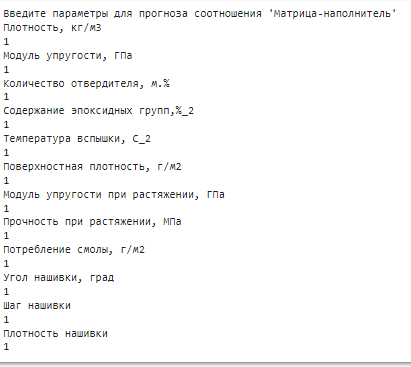


Рисунок 15 – Вывод полей для ввода значений показателей: для всех значений введено произвольное тестовое значение «1»

2) Получение предсказания рекомендуемого соотношения:

pred = nn\_model2.predict(args)

#возвращаем исходный масштаб значения

pred2 = pred \* np.max(clear\_dataset['Соотношение матрица-наполнитель']) + np.min(clear\_dataset['Соотношение матрица-наполнитель'])

print("Рекомендуемое соотношение 'Матрица-наполнитель' =", pred2[0,0])

Получаем на выходе сообщение: «Рекомендуемое соотношение 'Матрица-наполнитель' = 3.4318194».

**2.2.6 Разработка приложения**

В ходе выполнения работы разработано приложение для предсказания параметра прочности при растяжении. Приложение разработано в среде PyCharm с помощью библиотеки FLASK. Модель приложения реализована на методе K-ближайших соседей. На рисунке 16 представлен скриншот запущенного приложения (файл «app.py») на сервере разработки.

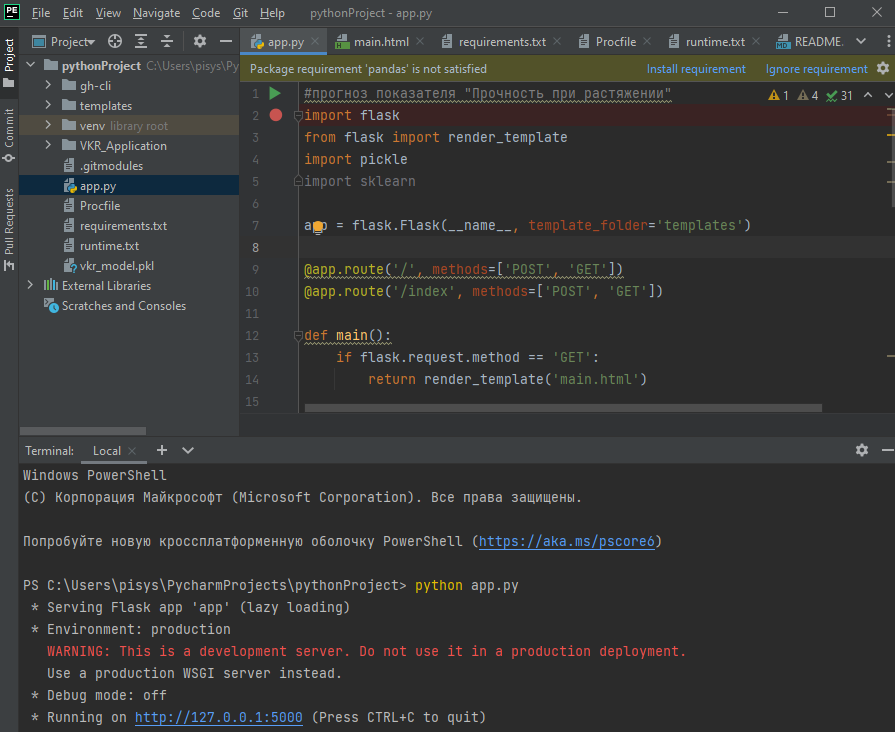


Рисунок 16 – Запущенное приложение

Интерфейс приложения можно посмотреть на рисунке 17.

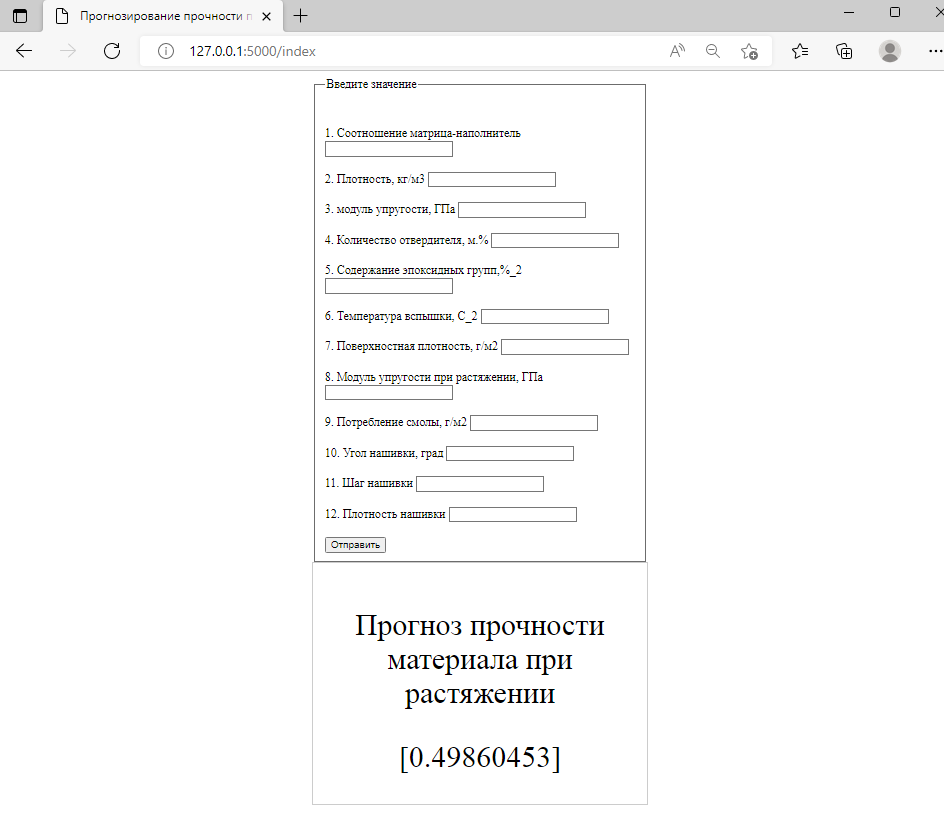


Рисунок 17 – Интерфейс приложения

Код приложения доступен в репозитории Github по ссылке «https://github.com/PaIvSysoev/VKR\_Application» (скриншот на рисунке 18).

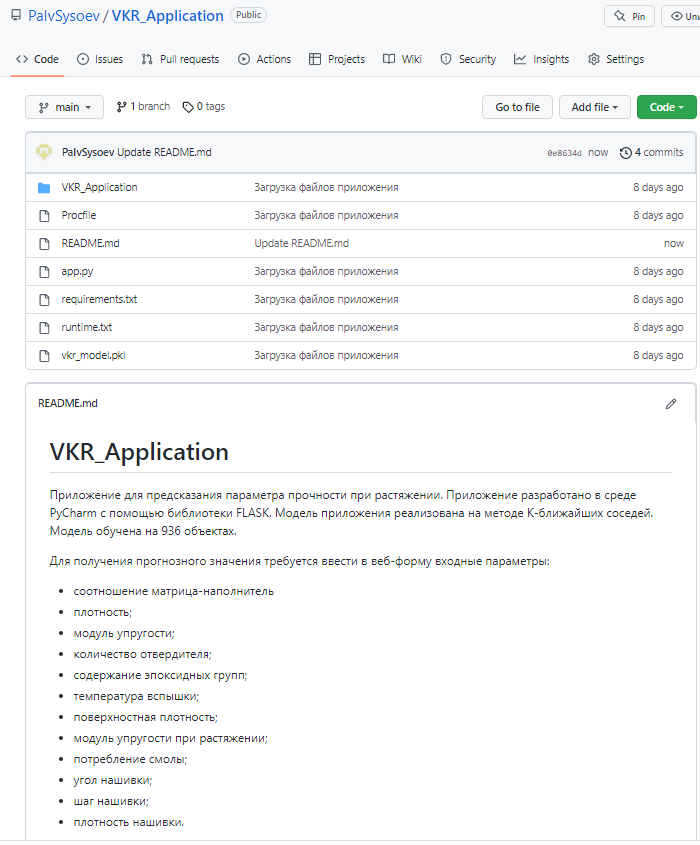


Рисунок 18 – Репозиторий приложения на GitHub

**2.2.7 Создание удаленного репозитория**

Репозиторий для выпускной квалификационной работы создан на ресурсе GitHub и доступен по ссылке: «https://github.com/PaIvSysoev/VKR\_Data\_Science\_MGTU».

Репозиторий содержит программный код, презентацию, исходные данные, пояснительную записку.

# **3 Заключение**

В результате работы были разработаны и протестированы несколько моделей машинного обучения, позволяющих прогнозировать конечные свойства композитных материалов, в том числе: линейная регрессия, случайный лес, метод K-ближайших соседей, полносвязная нейронная сеть.

Разработана нейронная сеть, которая рекомендует соотношение матрица-наполнитель для создания композиционного материала.

Разработано приложение, которое может быть использовано для прогноза параметра прочности при растяжении композиционных материалов.

Модели могут быть использованы для переобучения и решения аналогичных задач на новых входных данных.

# **4 Библиографический список**

1 Артеменко, С.Е. Структура и свойства базальто-, стекло- и углепластиков, сформированных по интеркаляционной технологии / С.Е. Артеменко, О.Г. Васильева, Ю.А. Кадыкова, А.Н. Леонтьев // Полимеры-2004: III Всерос. Каргинская конф. М., 2004. Т. 2. С. 162.

2 Джулли, П.: Библиотека Keras - инструмент глубокого обучения / пер. с англ. А.А. Слинкин. – ДМК Пресс, 2017. – 249 с.

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

4 Жерон, Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. Пер. с англ. - СпБ.: ООО «Альфа-книга»: 2018. – 688 с.

5 Кадыкова, Ю.А. Полимерные композиционные материалы на основе волокон различной химической природы / Ю.А. Кадыкова, А.Н. Леонтьев, О.Г. Васильева, С.Е. Артеменко // Строительные материалы, оборудование, технологии ХХ1 века. 2002. № 6. С. 10-11.

6 Николенко, С., Кадурин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. - СПб.: Питер. - 2020. - 480 с. ISBN: 978-5-4461-1537-2.

7 Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2007. - 1408 с.

8 Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. - М.: Горячая Линия - Телеком. - 2013. - 384 с. ISBN: 978-5-9912-0320-3.

9 Статистическая обработка данных, планирование эксперимента и случайные процессы : учебное пособие для вузов / Берикашвили В. Ш., Оськин С. П. - 2-е изд., испр. и доп. - М.: Юрайт, 2021. - 163 с.

10 Фостер Д. Генеративное глубокое обучение. Творческий потенциал нейронных сетей. - СПб.: Питер. - 2020. - 336 с. - ISBN: 978-5-4461-1566-2.

12 Шитиков В.К., Мастицкий С.Э. (2017) Классификация, регрессия и другие алгоритмы Data Mining с использованием R. 351 с. − Электронная книга, адрес доступа: https://github.com/ranalytics/data-mining.

12 Кашнитский, Ю. Открытый курс машинного обучения. Тема 3. Классификация, деревья решений и метод ближайших соседей [Электронный ресурс] : – Режим доступа: https://habr.com/ru/company/ods/blog/322534/. (дата обращения: 12.06.2022).

13 Константинов, М. Краткий курс машинного обучения или как создать нейронную сеть для решения скоринг задачи [Электронный ресурс] : – Режим доступа: https://habr.com/ru/post/340792/. (дата обращения: 12.06.2022).

14 Масзанский, А. Метод k-ближайших соседей (k-nearest neighbour) [Электронный ресурс] : – Режим доступа: https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19/. (дата обращения: 12.06.2022).

15 Метод обратного распространения ошибки: математика, примеры, код [Электронный ресурс] : – Режим доступа: https://neurohive.io/ru/osnovy-data-science/obratnoe-rasprostranenie/. (дата обращения: 12.06.2022).

16 Радченко, В. Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес [Электронный ресурс] : – Режим доступа: https://habr.com/ru/company/ods/blog/324402. (дата обращения: 12.06.2022).

17 Регрессия в машинном обучении: оптимальный алгоритм [Электронный ресурс] : – Режим доступа: https://proglib.io/p/ml-regression. (дата обращения: 12.06.2022).

18 Федоров, А. Решение задачи регрессии полносвязной нейронной сетью [Электронный ресурс] : – Режим доступа: https://www.bizkit.ru/2019/11/05/14921/. (дата обращения: 12.06.2022).

19 Функции активации в нейронных сетях [Электронный ресурс] : – Режим доступа: http://www.aiportal.ru/articles/neural-networks/activation-function.html. (дата обращения: 12.06.2022).

20 Хлевнюк, А. Основы линейной регрессии [Электронный ресурс] : – Режим доступа: https://habr.com/ru/post/514818/. (дата обращения: 12.06.2022).

# **5 Приложения**

**Приложение А**

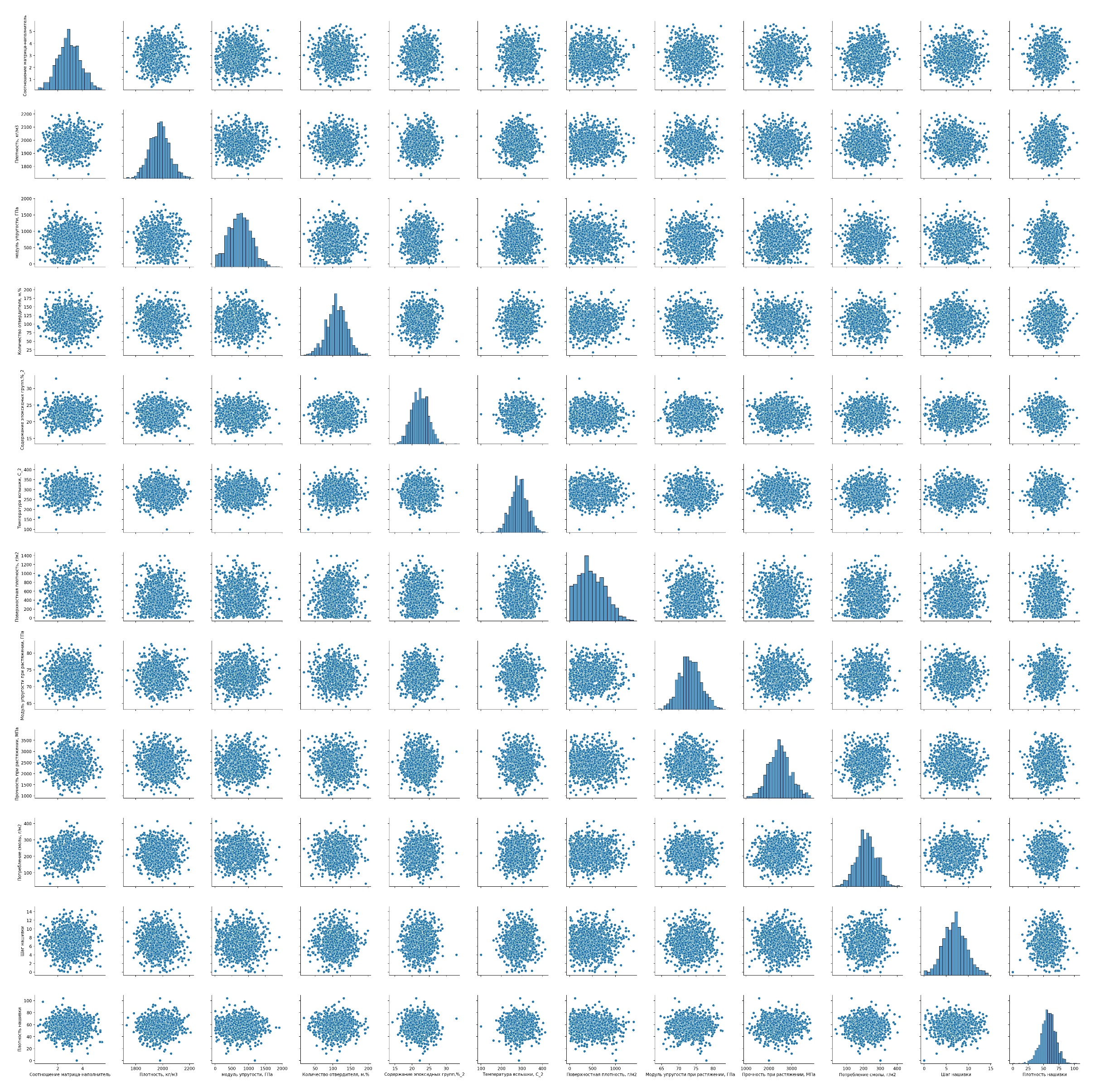
**Попарные графики рассеивания**

Рисунок А.1 – Попарные графики рассеивания

**Приложение Б**

**Результаты обучения и тестирования моделей**

1. Линейная регрессия

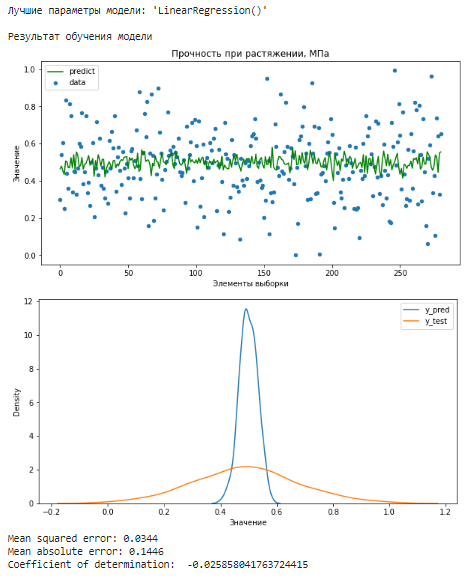
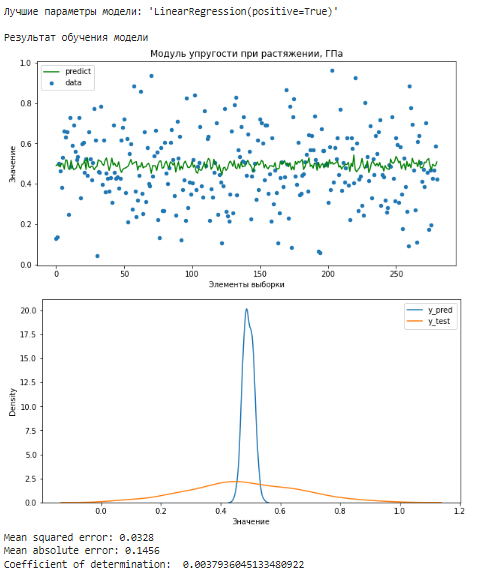


Рисунок Б.1 – Результат тестирования модели линейной регрессии на параметре «Прочность при растяжении»

 Рисунок Б.2 – Результат тестирования модели линейной регрессии на параметре «Модуль упругости при растяжении»

1. Линейная регрессия с полиномиальными параметрами

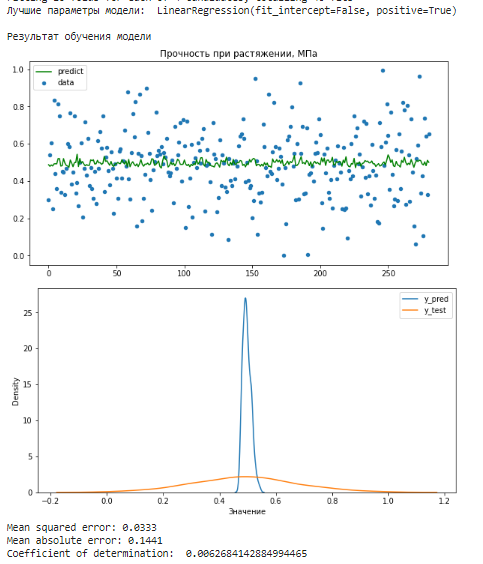


Рисунок Б.3 – Результат тестирования модели линейной регрессии с полиномиальными параметрами на параметре «Прочность при растяжении»

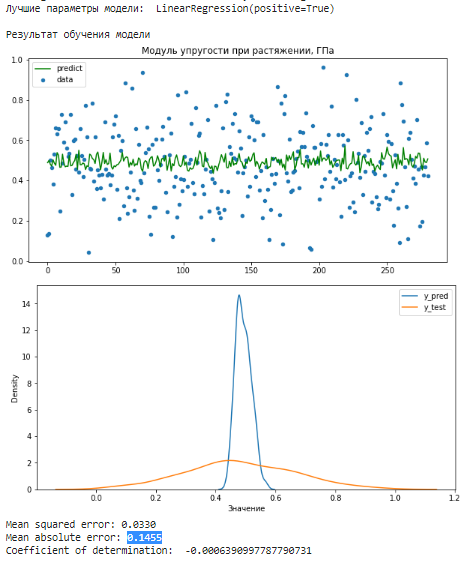


Рисунок Б.4 – Результат тестирования модели линейной регрессии с полиномиальными параметрами на параметре «Модуль упругости при растяжении»

1. Случайный лес

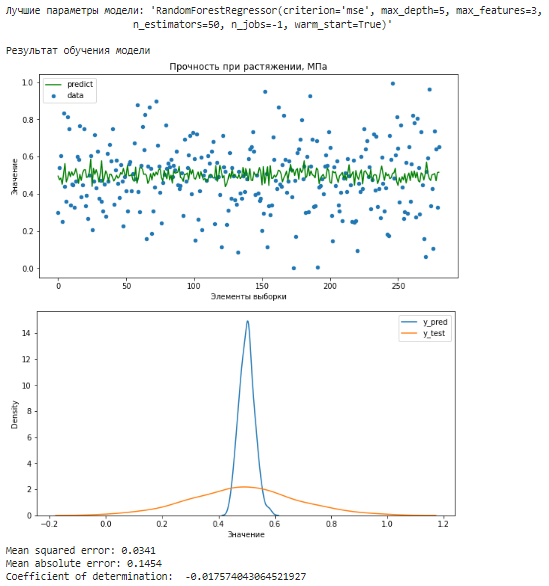


Рисунок Б.5 – Результат тестирования модели «Случайный лес» на параметре «Прочность при растяжении»

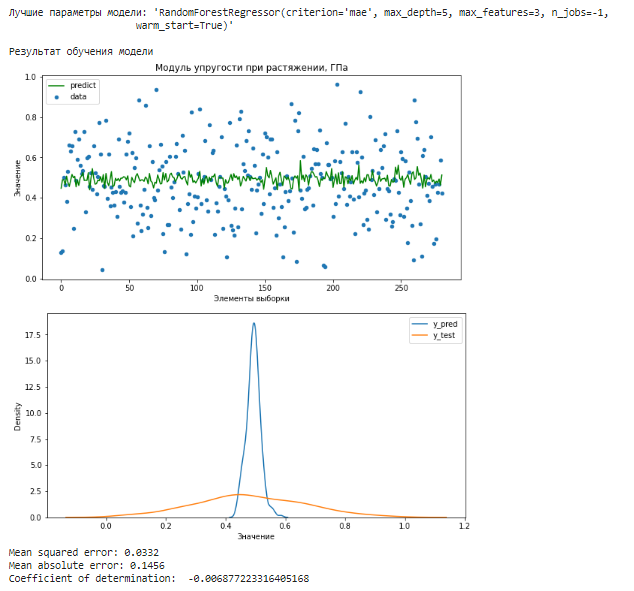


Рисунок Б.6 – Результат тестирования модели «Случайный лес» на параметре «Модуль упругости при растяжении»

4. Метод k-ближайших соседей

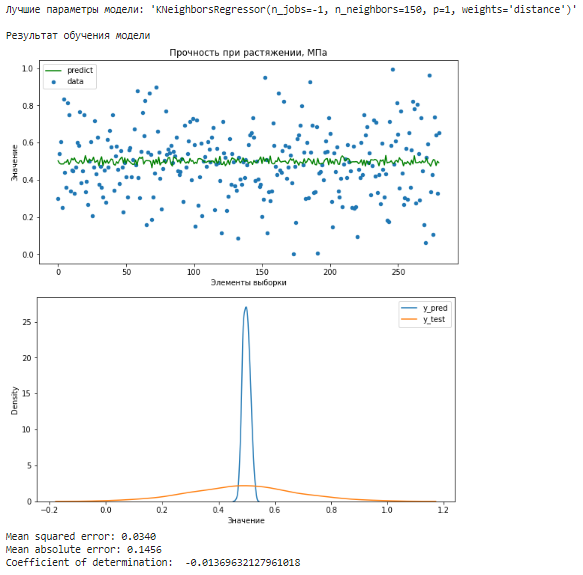


Рисунок Б.7 – Результат тестирования модели «k-ближайших соседей» на параметре «Прочность при растяжении»

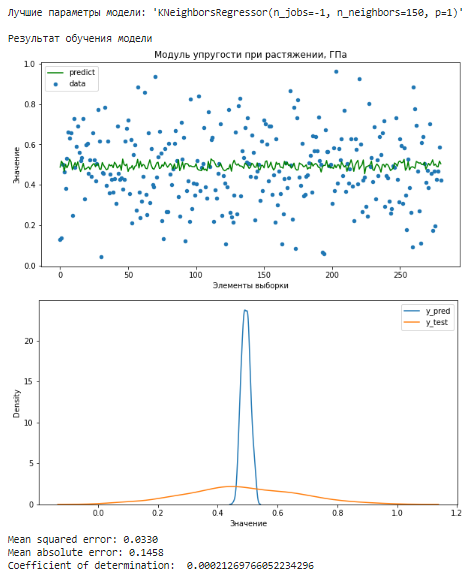


Рисунок Б.8 – Результат тестирования модели «k-ближайших соседей» на параметре «Модуль упругости при растяжении»

5. Нейронная сеть 12-6-1

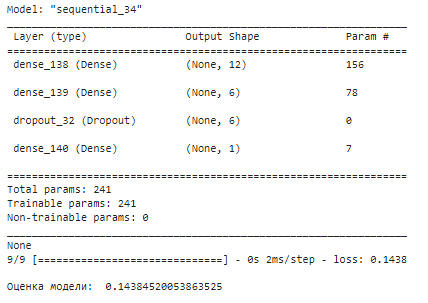


Рисунок Б.9 – Архитектура нейронной сети 12-6-1

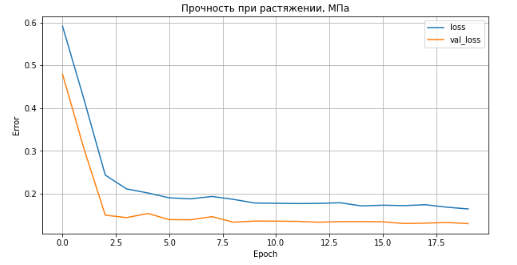


Рисунок Б.10 – График обучения нейронной сети 12-6-1 на параметре «Прочность при растяжении»

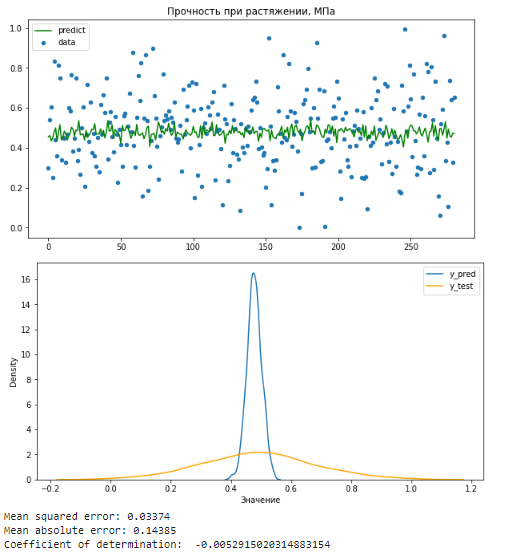


Рисунок Б.11 – Результат тестирования нейронной сети 12-6-1 на параметре «Прочность при растяжении»

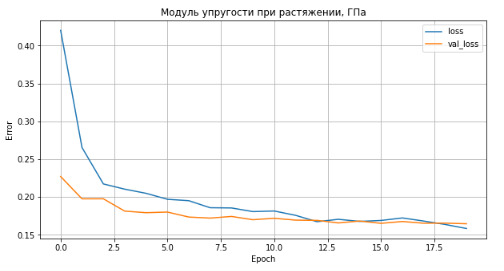


Рисунок Б.12 – График обучения нейронной сети 12-6-1 на параметре «Модуль упругости при растяжении»

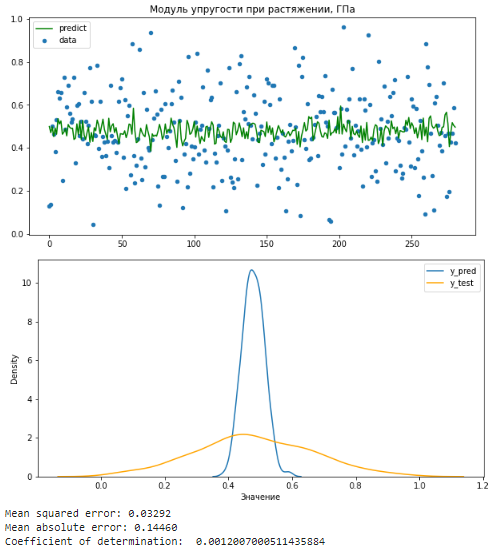


Рисунок Б.13 – Результат тестирования нейронной сети 12-6-1 на параметре «Модуль упругости при растяжении»

6. Нейронная сеть 12-6-6-1

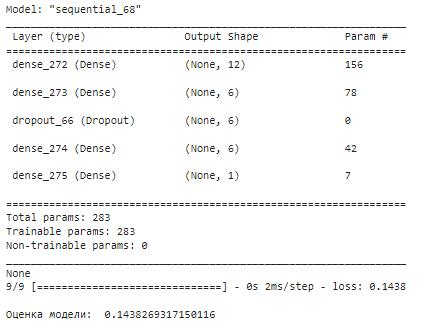


Рисунок Б.14 – Архитектура нейронной сети 12-6-6-1

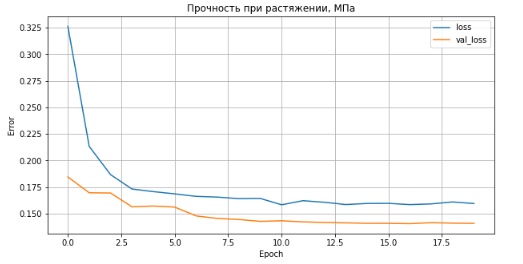


Рисунок Б.15 – График обучения нейронной сети 12-6-6-1 на параметре «Прочность при растяжении»

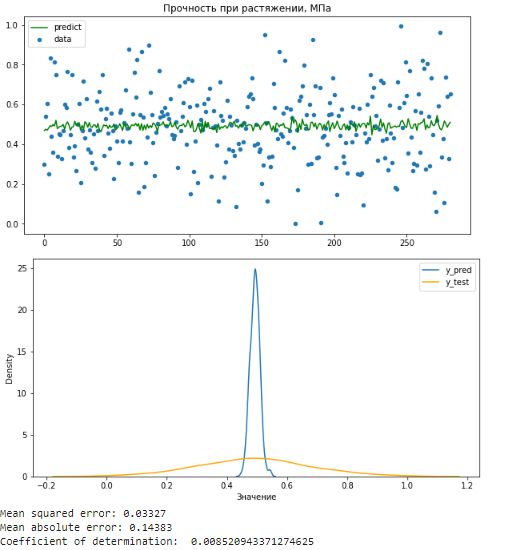


Рисунок Б.16 – Результат тестирования нейронной сети 12-6-6-1 на параметре «Прочность при растяжении»



Рисунок Б.17 – График обучения нейронной сети 12-6-6-1 на параметре «Модуль упругости при растяжении»

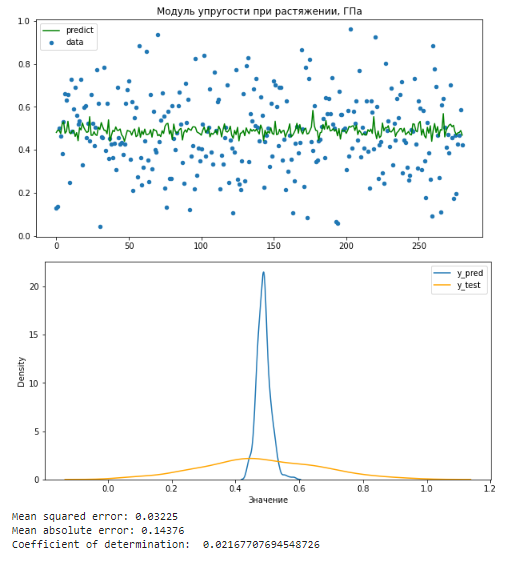


Рисунок Б.18 – Результат тестирования нейронной сети 12-6-6-1 на параметре «Модуль упругости при растяжении»