



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

Лазарев Михаил Юрьевич

Москва, 2022

## Оглавление

ОГЛАВЛЕНИЕ .....	2
ВВЕДЕНИЕ. ....	4
1. АНАЛИЗ ИСХОДНЫХ ДАННЫХ. ....	5
1.1. ОПИСАНИЕ ЗАДАЧИ И ИСХОДНЫХ ДАННЫХ. ....	5
1.2. РАЗВЕДОЧНЫЙ АНАЛИЗ И ВИЗУАЛИЗАЦИЯ ДАННЫХ.....	6
1.3. МЕТОДЫ РЕШЕНИЯ И ПРЕДОБРАБОТКА ДАННЫХ .....	11
2. РАЗРАБОТКА МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ.....	18
2.1. РАЗРАБОТКА И ОБУЧЕНИЕ МОДЕЛЕЙ .....	18
2.2. ТЕСТИРОВАНИЕ МОДЕЛЕЙ .....	19
2.3. ПРИЛОЖЕНИЕ И РЕПОЗИТОРИЙ .....	25
ЗАКЛЮЧЕНИЕ.....	27



## **Введение.**

Композитными называют искусственно созданные материалы, состоящие из нескольких слоев: слоя-наполнителя и слоя-матрицы. Сочетание в одном материале слоев с разными свойствами позволяет получить новый продукт с качествами, отличными от характеристик каждого слоя в отдельности. Эффективность данного подхода обеспечивается за счёт технологии производства и совмещения свойств взаимодополняющих друг друга материалов.

Современные возможности искусственного интеллекта в прогнозировании данных помогают при решении задач симуляции характеристик материалов и получившихся при этом композитов, что позволяет избежать различных физических испытаний и соответствующих затрат времени и финансов.

Машинное обучение представляет собой набор различных методов, при помощи которых имея конкретные исходные данные можно «натренировать» модель и предсказать определённые события или свойства объектов, связанных с этими данными. При этом современные технологии, использующие нейронные сети, позволяют ускорить и усовершенствовать процессы, направленные на изучение данных.

## **1. Анализ исходных данных.**

### **1.1. Описание задачи и исходных данных.**

Целью настоящей работы является прогнозирование характеристик «Модуля упругости при растяжении, ГПа» и «прочность при растяжении, МПа» и рекомендовать «соотношение матрица- наполнитель» материалов при помощи методов машинного обучения и построение моделей.

Данные предоставлены Центром НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана).

Имеются данные о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим и т.д.). Один набор данных «X\_br» содержит (10 переменных вещественного типа) содержащей 1023 строки данных. Второй набор данных «X\_nir» (3 переменные вещественного типа) содержащей 1040 строк данных. Данные представлены в Excel – файлах. Интересующие выходные характеристики (3 выходные переменные) содержатся в наборе «X\_br». Для анализа данных будет использоваться язык программирования Python. Этому способствует простота языка, а также большое разнообразие открытых библиотек.

Для машинного обучения предварительной обработки и анализа данных будут использоваться библиотеки Python – Pandas, Scikit-learn и Matplotlib.

## 1.2. Разведочный анализ и визуализация данных

Предварительный анализ данных с целью выявления наиболее общих зависимостей, закономерностей и тенденций, характера и свойств анализируемых данных, законов распределения анализируемых величин. При разведочном анализе учитывается и сравнивается большое число признаков, а для поиска закономерностей используются самые разные методы. Результаты разведочного анализа помогут в разработке наилучшей стратегии углубленного анализа, выдвижение гипотез, уточнение особенностей применения тех или иных математических методов и моделей.

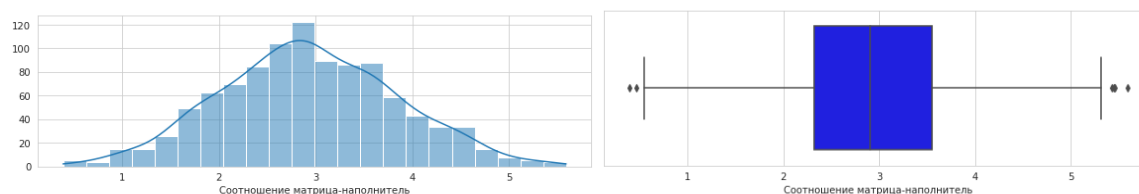
С помощью библиотеки Pandas загружаем и объединяем данные по типу «INNER».

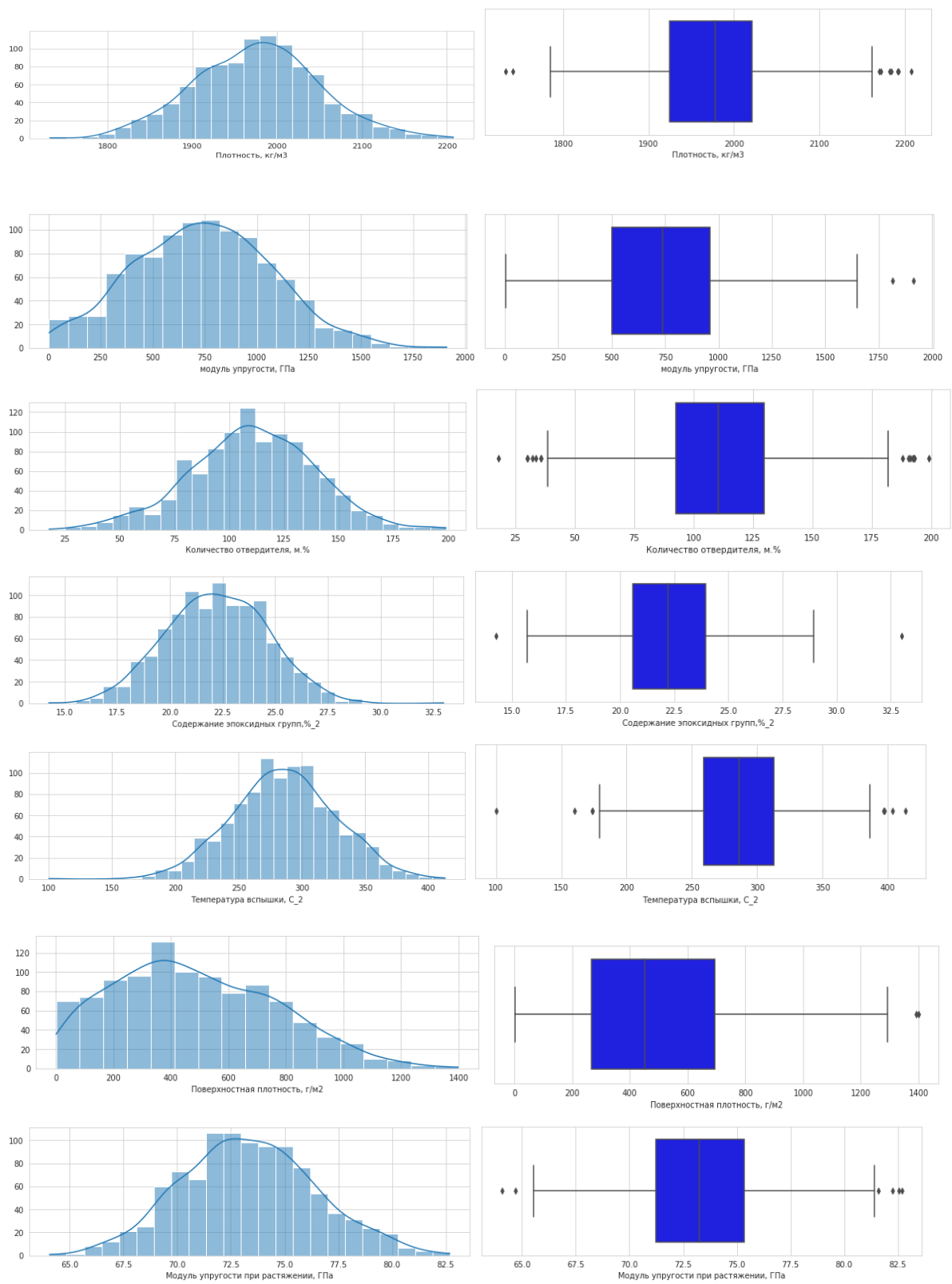
	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп, %_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0.0	1.857143	2030.0	738.736842	30.00	22.267857	100.000000	210.0	70.0	3000.0	220.0	0.0	4.0	57.0
1.0	1.857143	2030.0	738.736842	50.00	23.750000	284.615385	210.0	70.0	3000.0	220.0	0.0	4.0	60.0
2.0	1.857143	2030.0	738.736842	49.90	33.000000	284.615385	210.0	70.0	3000.0	220.0	0.0	4.0	70.0
3.0	1.857143	2030.0	738.736842	129.00	21.250000	300.000000	210.0	70.0	3000.0	220.0	0.0	5.0	47.0
4.0	2.771331	2030.0	753.000000	111.86	22.267857	284.615385	210.0	70.0	3000.0	220.0	0.0	5.0	57.0

```
df.shape
(1023, 13)
```

Рисунок 1 - Основные параметры

Часть данных 17 строк невозможно включить в таблицу из-за различия количества строк в наборах данных в процессе компоновки. Исходный датафрейм содержит 1023 строки с входными параметрами и 13 колонок переменных как показано на рисунке 1. Визуализация гистограмм и диаграмм («ящик с усами») позволяют наглядно увидеть характер распределения переменных.





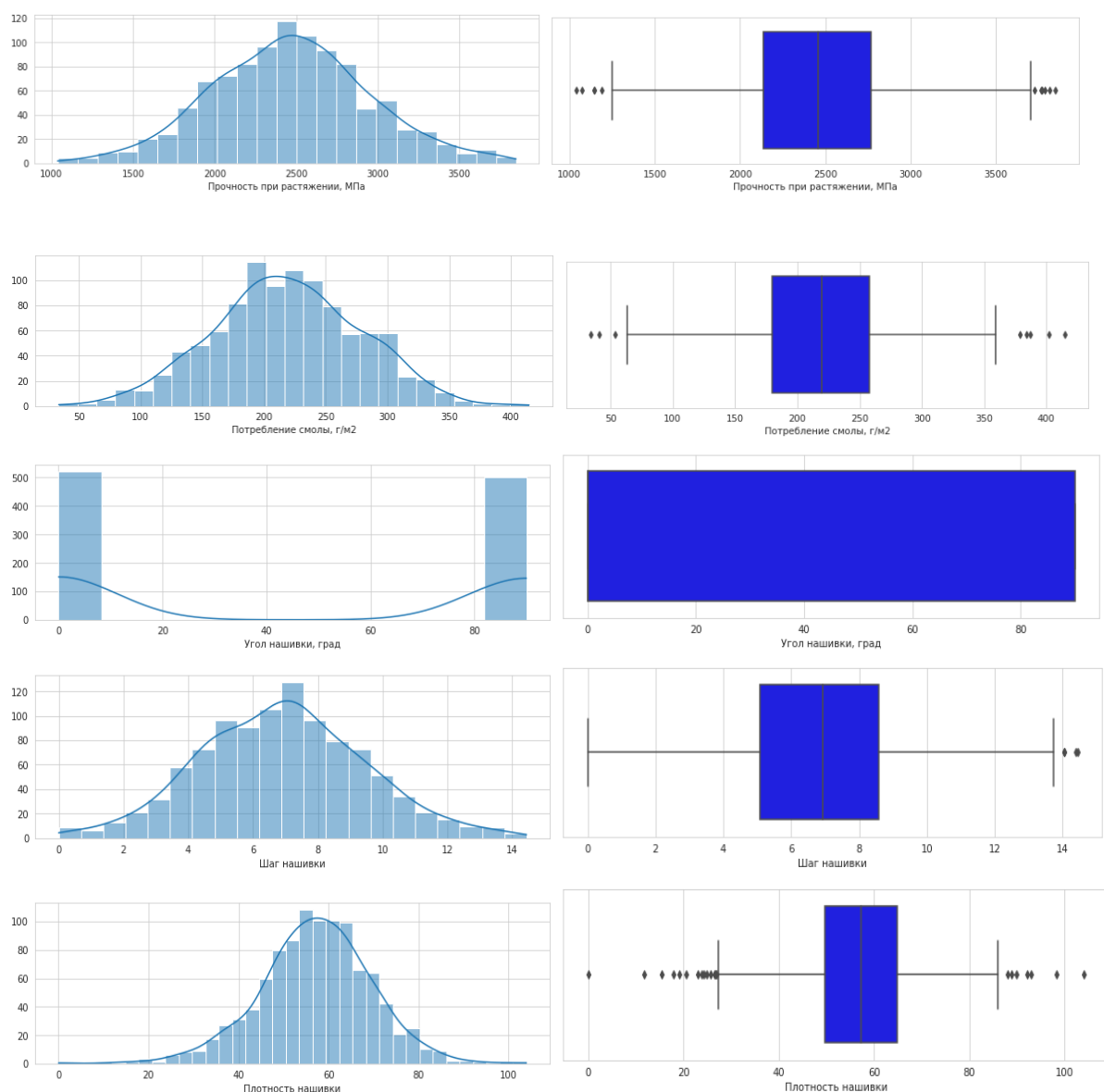


Рисунок 2 – Гистограммы распределения и диаграммы размаха

Все переменные имеют нормальное распределение рисунок 2, кроме «Угол нашивки, град», где всего два значения.

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп, %_2	Температура вспыхки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
count	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000
mean	2.930366	1975.734888	739.923233	110.570769	22.244390	285.882151	482.731833	73.328571	2466.922843	218.423144	44.252199	6.899222	57.153929
std	0.913222	73.729231	330.231581	28.295911	2.406301	40.943260	281.314690	3.118983	485.628006	59.735931	45.015793	2.563467	12.350969
min	0.389403	1731.764635	2.436909	17.740275	14.254985	100.000000	0.603740	64.054061	1036.856605	33.803026	0.000000	0.000000	0.000000
25%	2.317887	1924.155467	500.047452	92.443497	20.608034	259.066528	266.816645	71.245018	2135.850448	179.627520	0.000000	5.080033	49.799212
50%	2.906878	1977.621657	739.664328	110.564840	22.230744	285.896812	451.864365	73.268805	2459.524526	219.198882	0.000000	6.916144	57.341920
75%	3.552660	2021.374375	961.812526	129.730366	23.961934	313.002106	693.225017	75.356612	2767.193119	257.481724	90.000000	8.586293	64.944961
max	5.591742	2207.773481	1911.536477	198.953207	33.000000	413.273418	1399.542362	82.682051	3848.436732	414.590628	90.000000	14.405222	103.988901

Рисунок 3 – Описательная статистика



На таблице с рисунка 3 описательная статистика, также видно, что у переменной «Поверхностная плотность, г/м2» медианное значение меньше среднего. Диаграммы размаха («ящик с усами») кроме «Угол нашивки, град» показывают незначительные выбросы. Так же был проверен датафрейм на уникальные значения. Для удаления выбросов был применён метод перцентиля `pmpy.percentile()`.

Для анализа корреляционных признаков были построены корреляционная тепловая карта и диаграмма рассеивания.

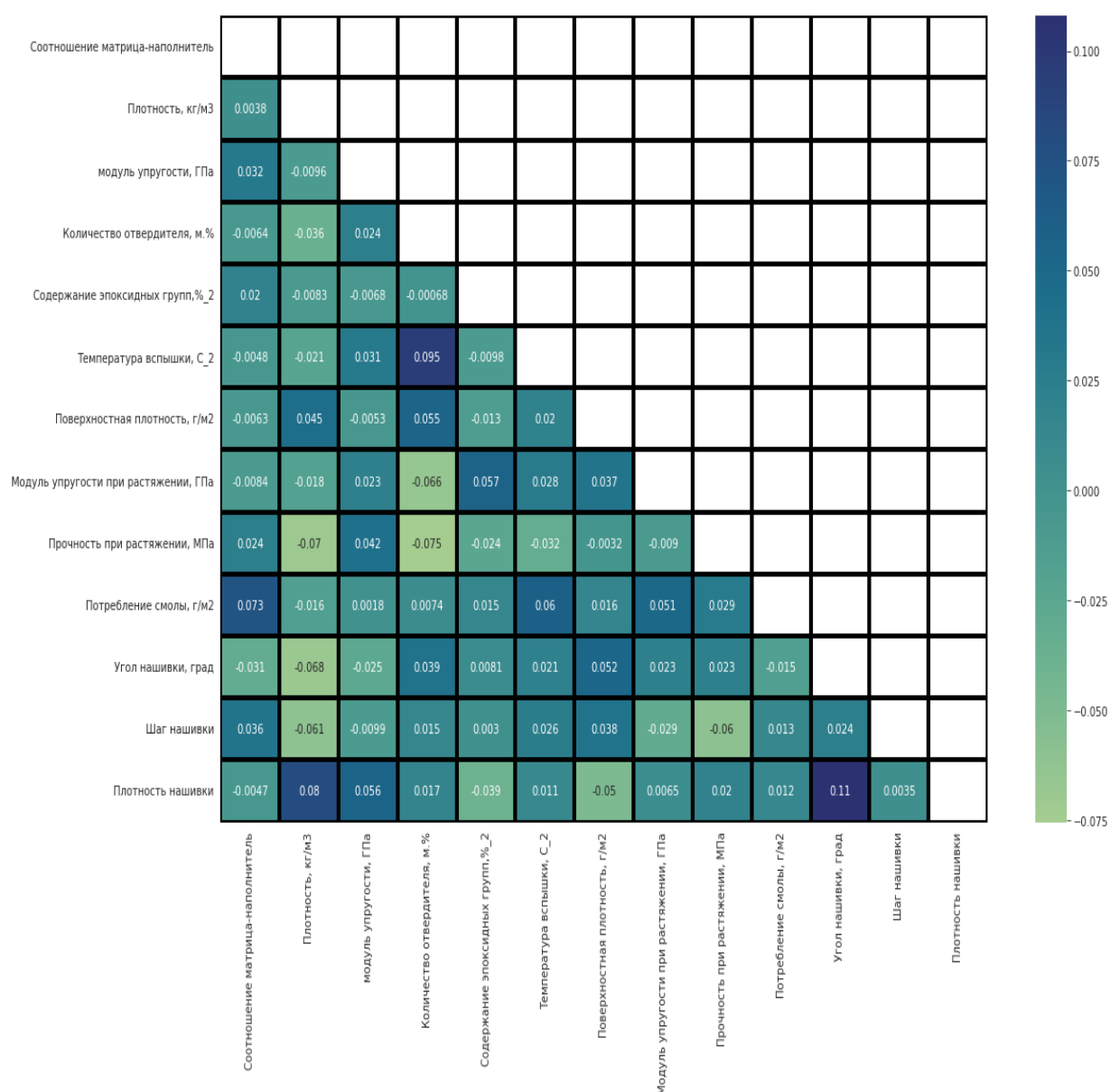
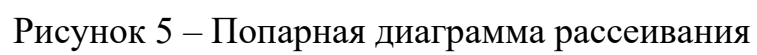
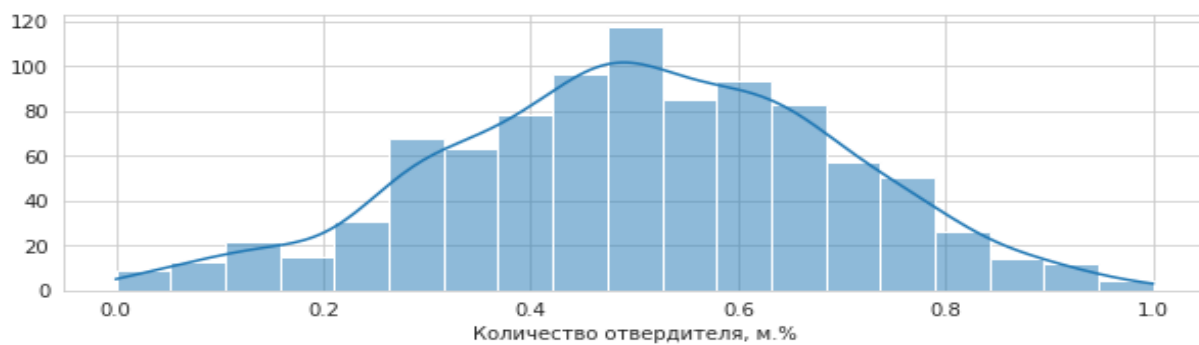
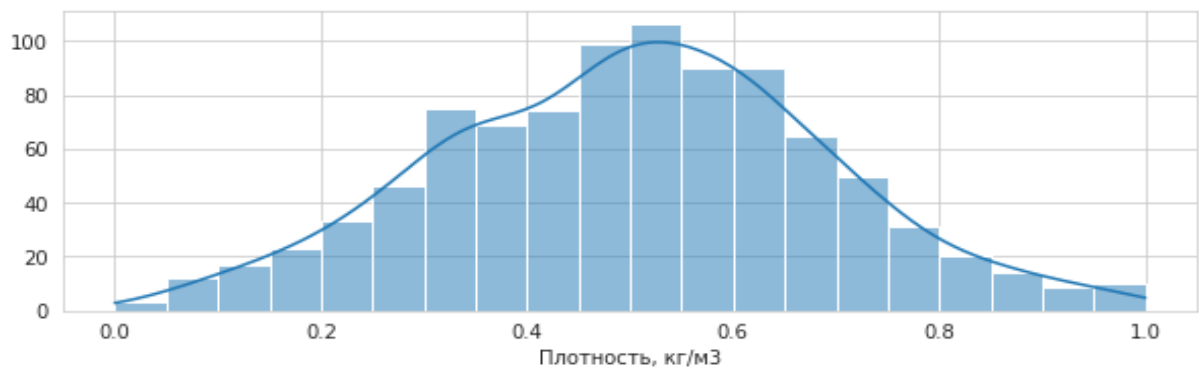
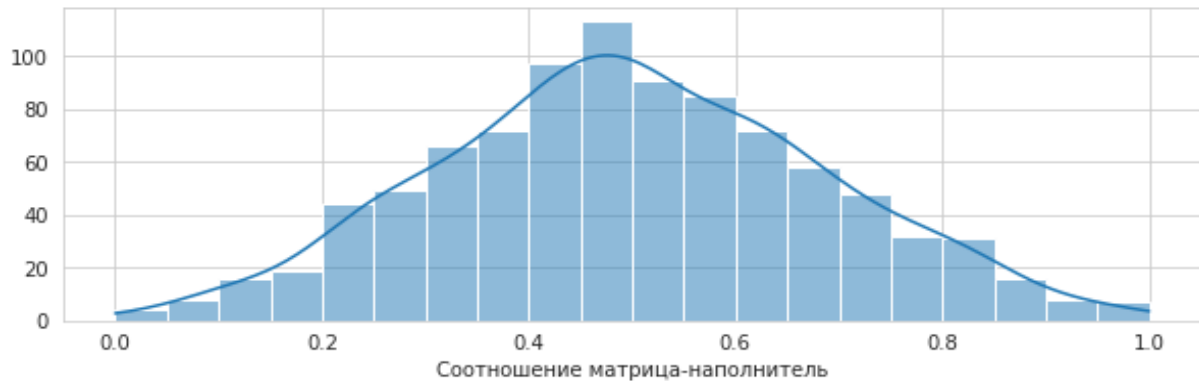


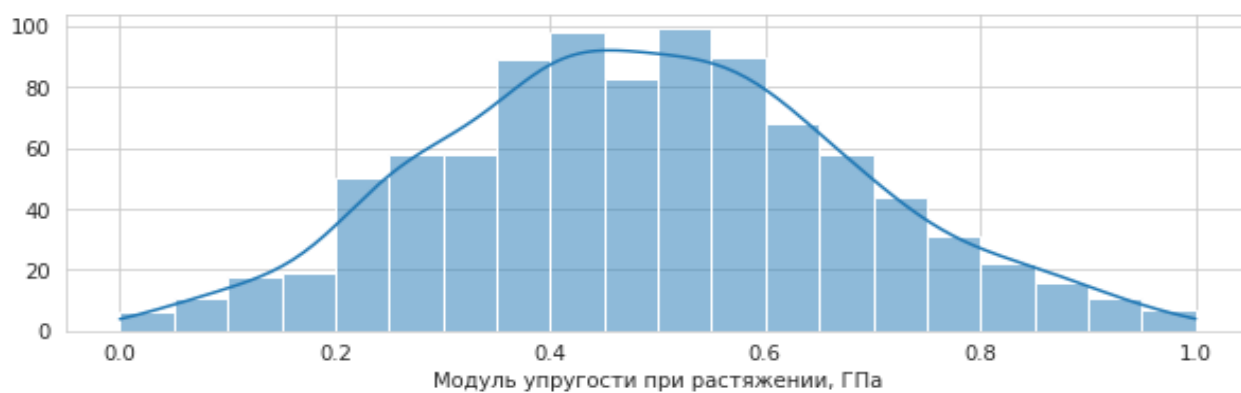
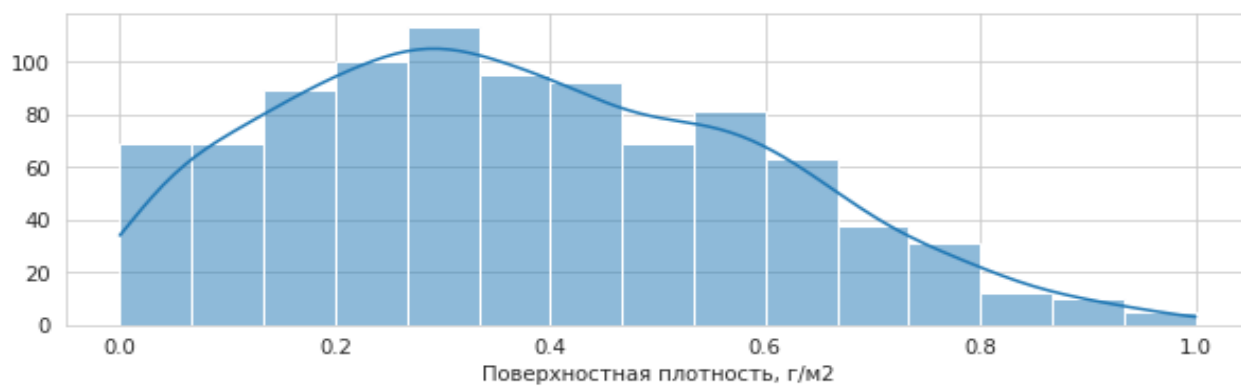
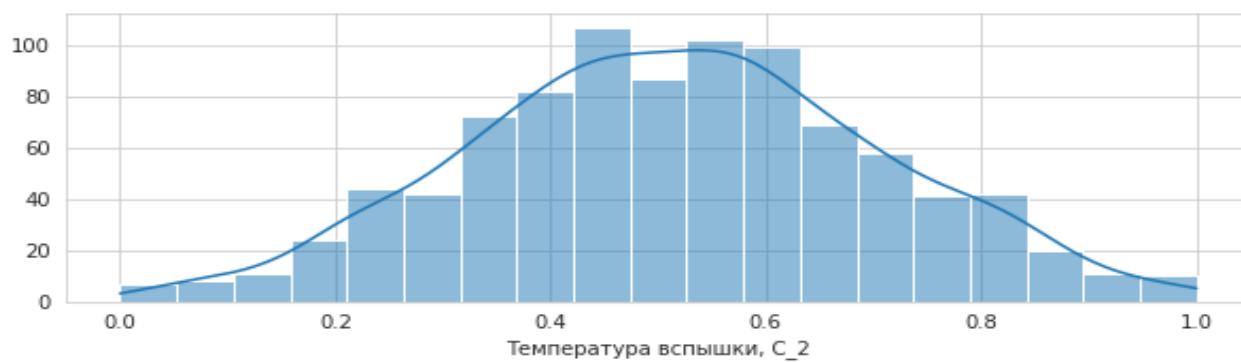
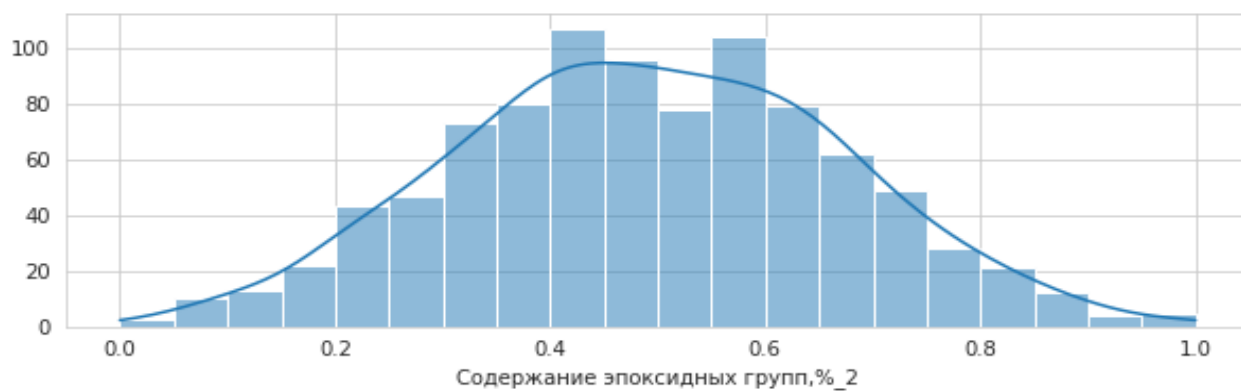
Рисунок 4 – Тепловая карта



### 1.3. Методы решения и предобработка данных

По полученным данным видно, что корреляция минимальна и линейной связи между признаками нет. Для нормализации датасета был применён метод MinMaxScaler.





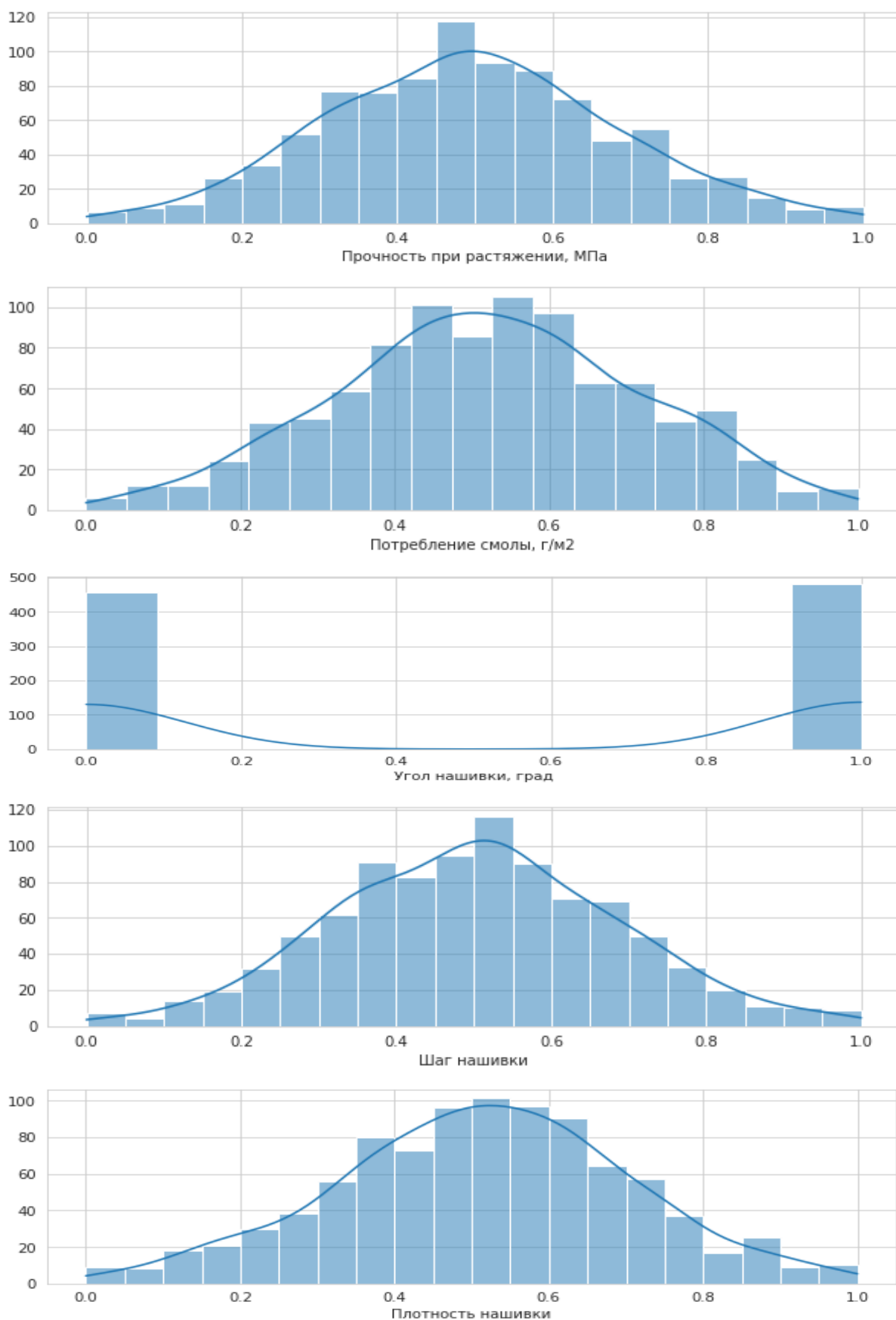
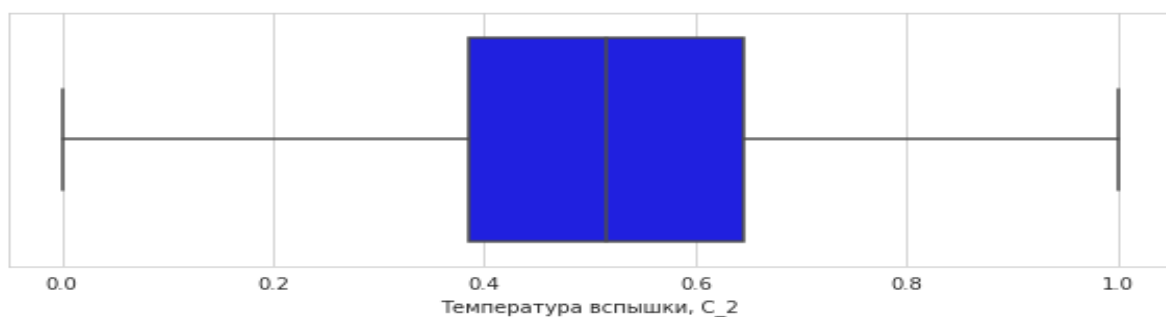
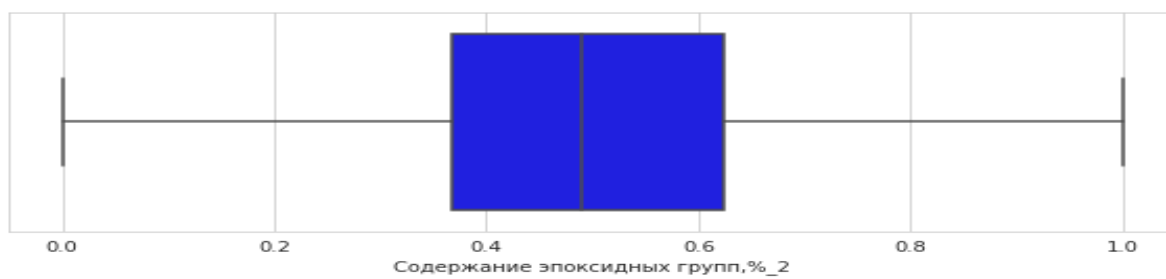
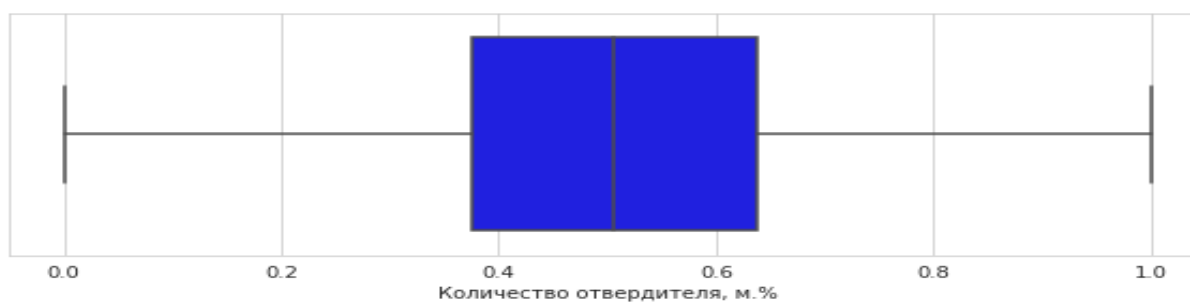
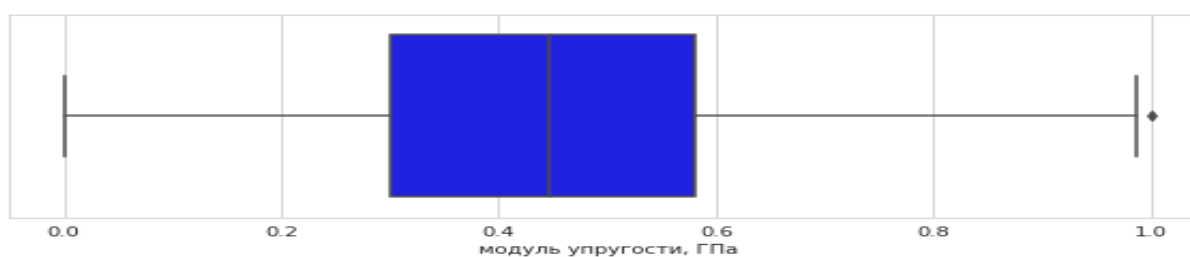
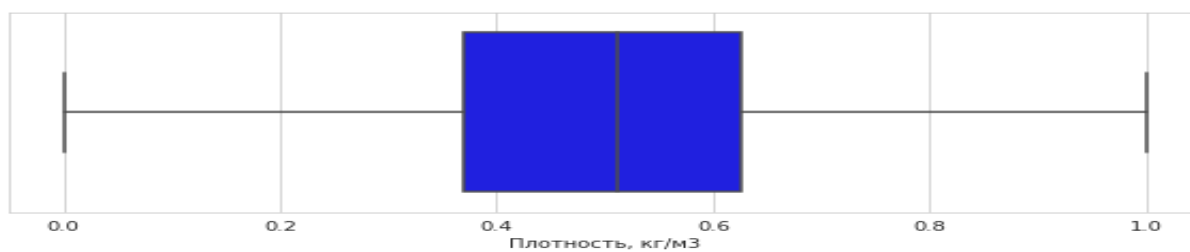
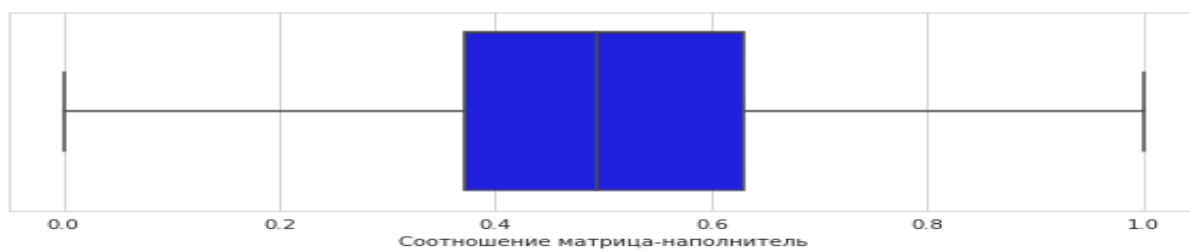
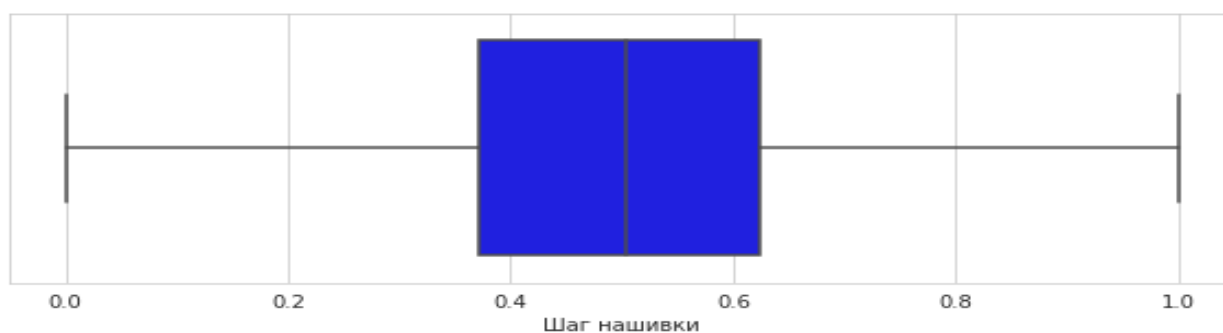
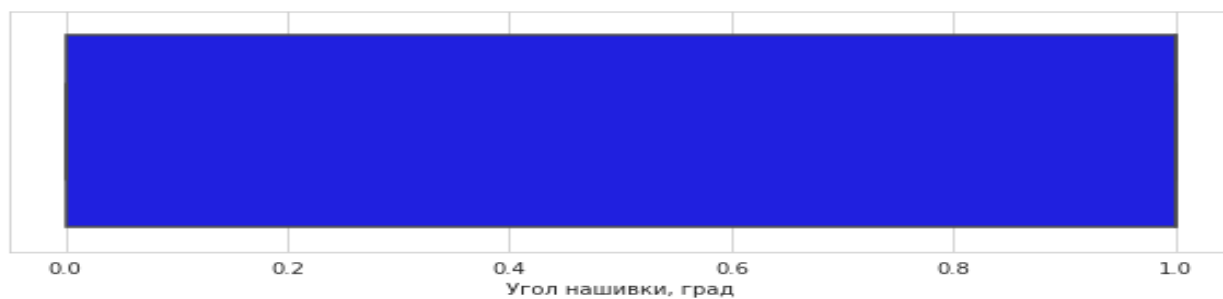
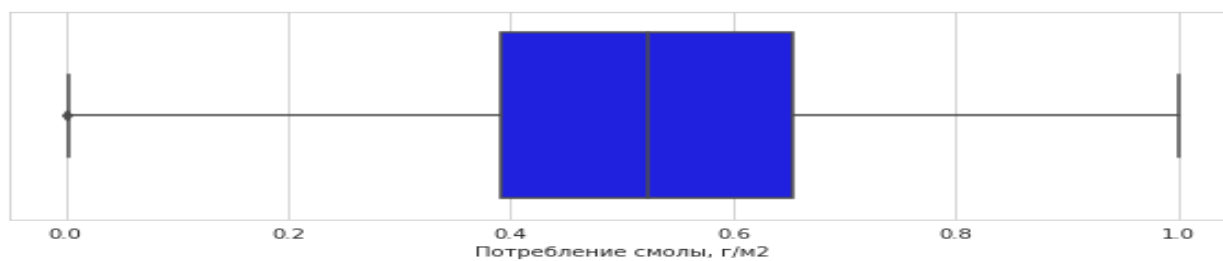
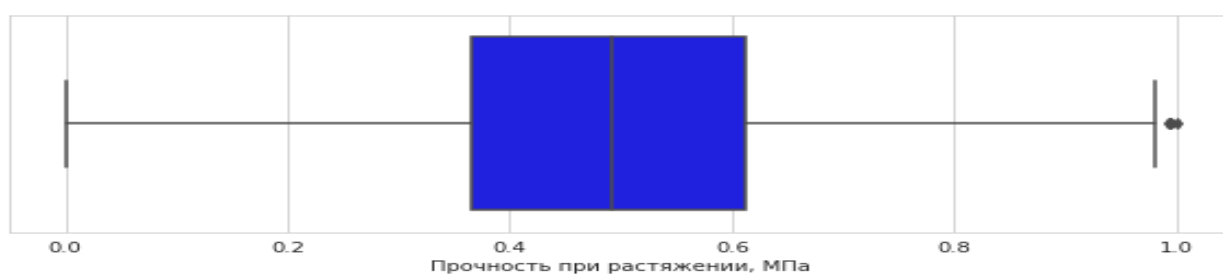
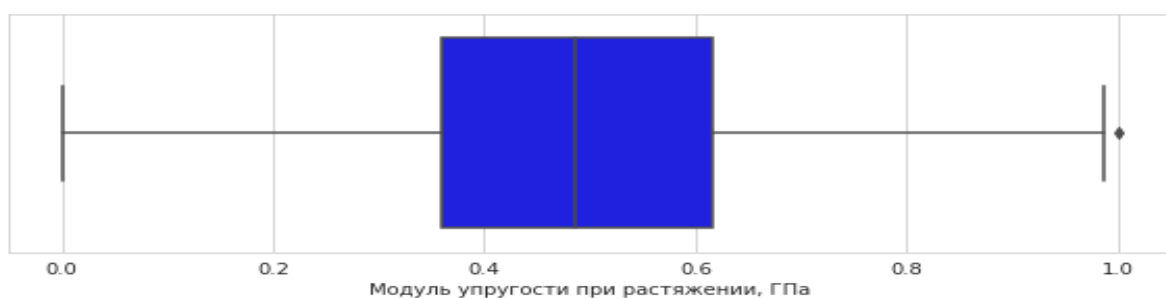
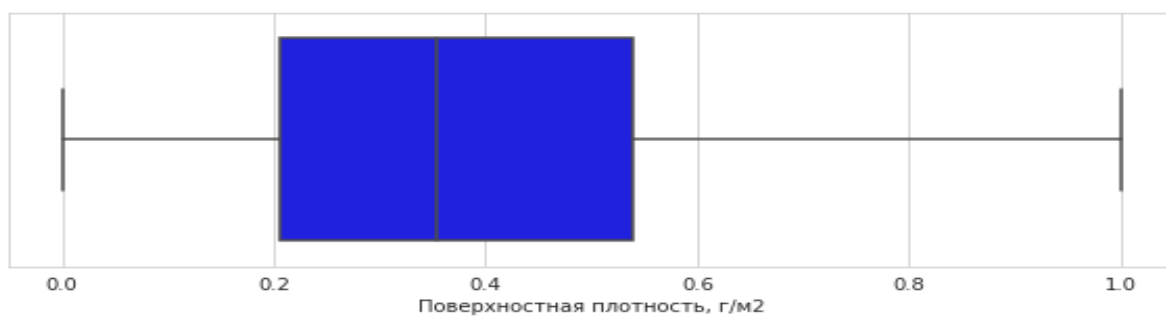


Рисунок 6 – Гистограммы распределения после нормализации и удаления выбросов





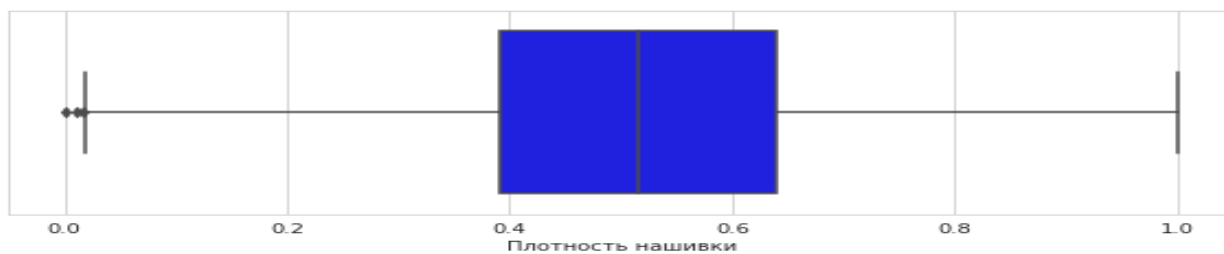


Рисунок 7 – Диаграммы размаха после нормализации и удаления выбросов

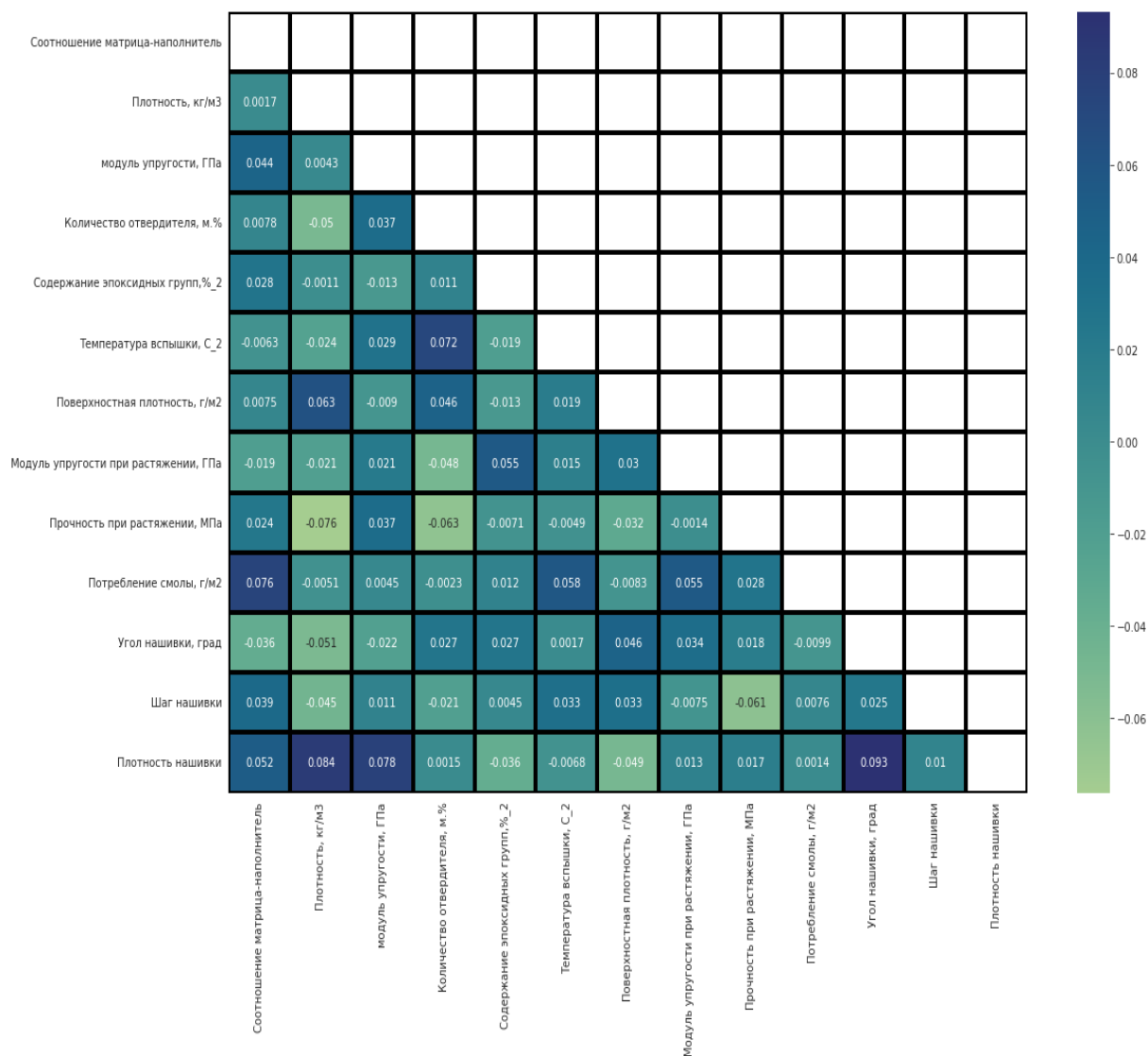


Рисунок 8 – Тепловая карта после нормализации и удаления выбросов



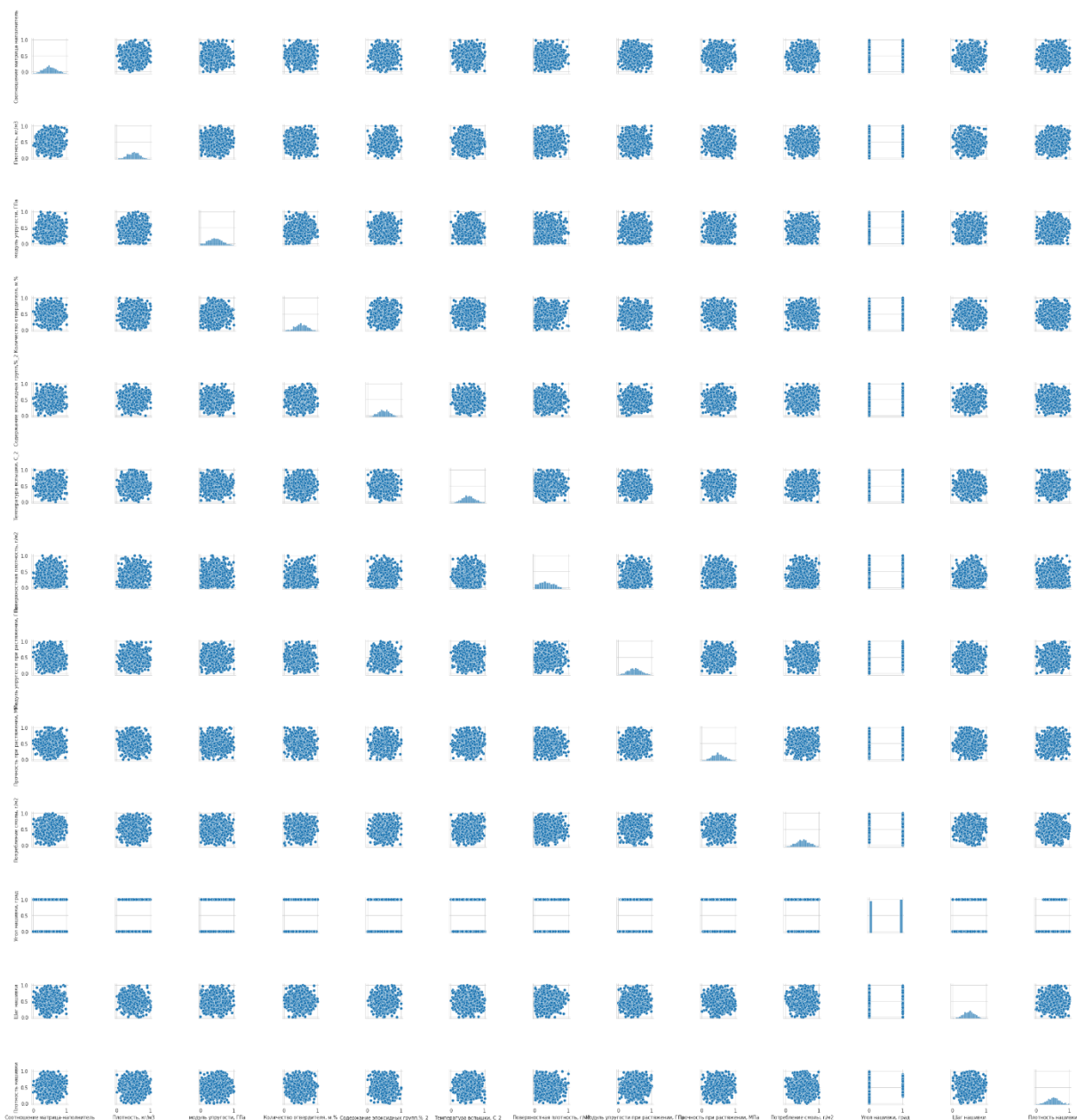


Рисунок 9 – Попарная диаграмма рассеивания после нормализации и удаления выбросов

Для построения прогноза «Модуль упругости при растяжении, гПа %», «Прочность при растяжении %» будут применены следующие модели:

- Метод К-ближайших соседей (KNeighborsRegressor);
- Градиентный бустинг (GradientBoostingRegressor);
- Случайный лес (RandomForestRegressor);
- Линейная регрессия (LinearRegression).

Для рекомендации «Соотношение матрица- накопитель %» будет построена нейронная сеть с использованием Tensorflow.Keras.

## 2. Разработка моделей машинного обучения

### 2.1. Разработка и обучение моделей

При нормализации методом `MinMaxScaler()` из библиотеки `Sklearn` мы преобразовали значения данных в диапазон от 0 до 1.

Для прогнозирования «Модуль упругости при растяжении, гПа %» и «Прочность при растяжении %» применим метод `KNeighborsRegressor`, так как он один из самых простых и распространённых методов. Для сравнения используем методы `GradientBoostingRegressor`, `RandomForestRegressor` и `LinearRegression` с настройками по умолчанию.

Реализации в специализированных библиотеках машинного обучения метода поиска гиперпараметров по сетке позволяют по заданным различным значениям гиперпараметров выбранного класса моделей протестировать возможные сочетания. Для поиска гиперпараметров по сетке использовался метод `GridSearchCv` библиотеки `Sklearn`.

Для прогнозирования «Модуля упругости при растяжении, гПа %» поиск по сетке с перекрестной проверкой показал:

- алгоритм - 'auto';
- количество соседей – 181;
- веса - 'uniform'.

Для прогнозирования «Прочности при растяжении поиск %» по сетке с перекрестной проверкой показал:

- алгоритм - 'brute';
- количество соседей – 93;
- веса - 'distance'.

Для рекомендации «Соотношение матрица-наполнитель %» была построена многослойная полносвязная нейронная сеть, которая содержит в себе: входной Dense-слой со 128 нейронами, последовательно с каждым слоем количество нейронов уменьшается в два раза вплоть до выходного слоя с одним нейроном, активационная функция слоёв – 'selu'. В свою очередь Dense-слои чередуются с `BatchNormalization`. С помощью метода `compile()` настроен процесс компиляции это необходимо перед обучением. В качестве функции ошибки будет использоваться средняя абсолютная ошибка (MAE) и оптимизатор - стохастический градиентный спуска SGD с `learning rate = 0,01`. Данные поделены на тренировочные и тестовые (30%).

## 2.2. Тестирование моделей

Обучение моделей регрессии подобранных с использованием поиска гиперпараметров по сетке, проводились в среде Google Colab. Оценка точности предсказаний проводилась с использованием оставшихся 30 % тестовых данных.

Для оценки качества моделей регрессии использовались специальные показатели:

1)  $R^2\_score$  (коэффициент детерминации) принимает значение от 0 до 1 и показывает долю объяснённой дисперсии объясняемого рода. Чем ближе  $R^2$  к 1, тем меньше доля необъяснённого;

2) Mean Absolut Error (MAE)- средняя абсолютная ошибка, показывает среднее значение абсолютных отклонений между наблюдаемыми и прогнозируемыми значениями;

3) Root Mean Squared Error (RMSE)- среднеквадратичная ошибка, показывает расстояние между двумя точками.

Судя по полученным данным модель KNeighborsRegressor показала себя лучше остальных, однако данные говорят о том, что мы будем получать усреднённые значения.

Результаты полученные при оценке качества моделей нельзя назвать удовлетворительными.

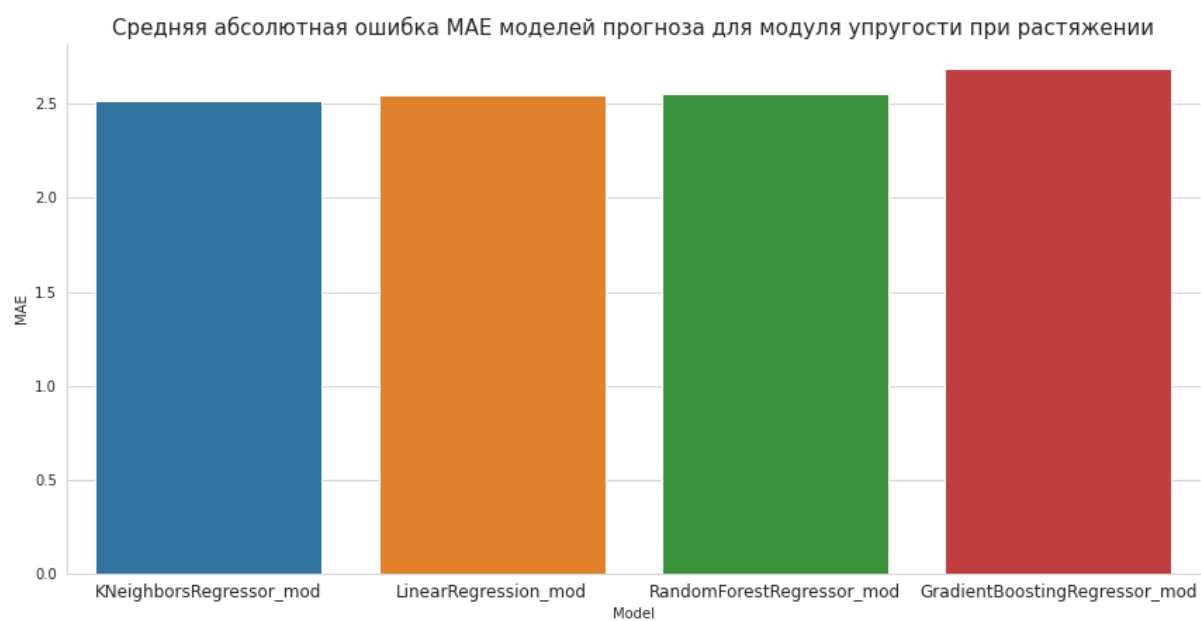
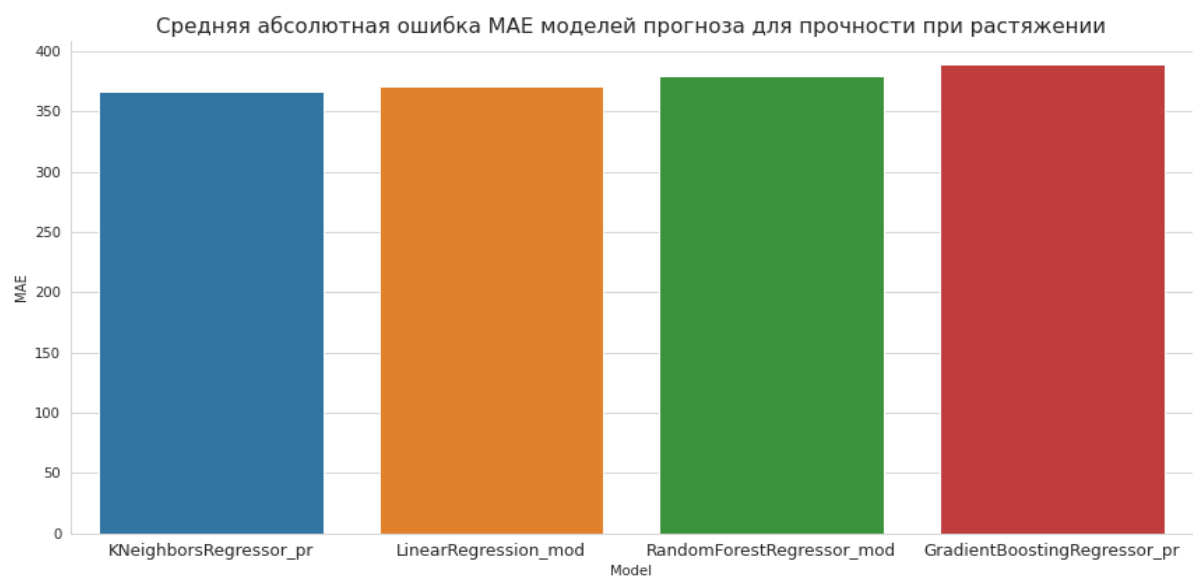


Рисунок 10 – Средняя абсолютная ошибка

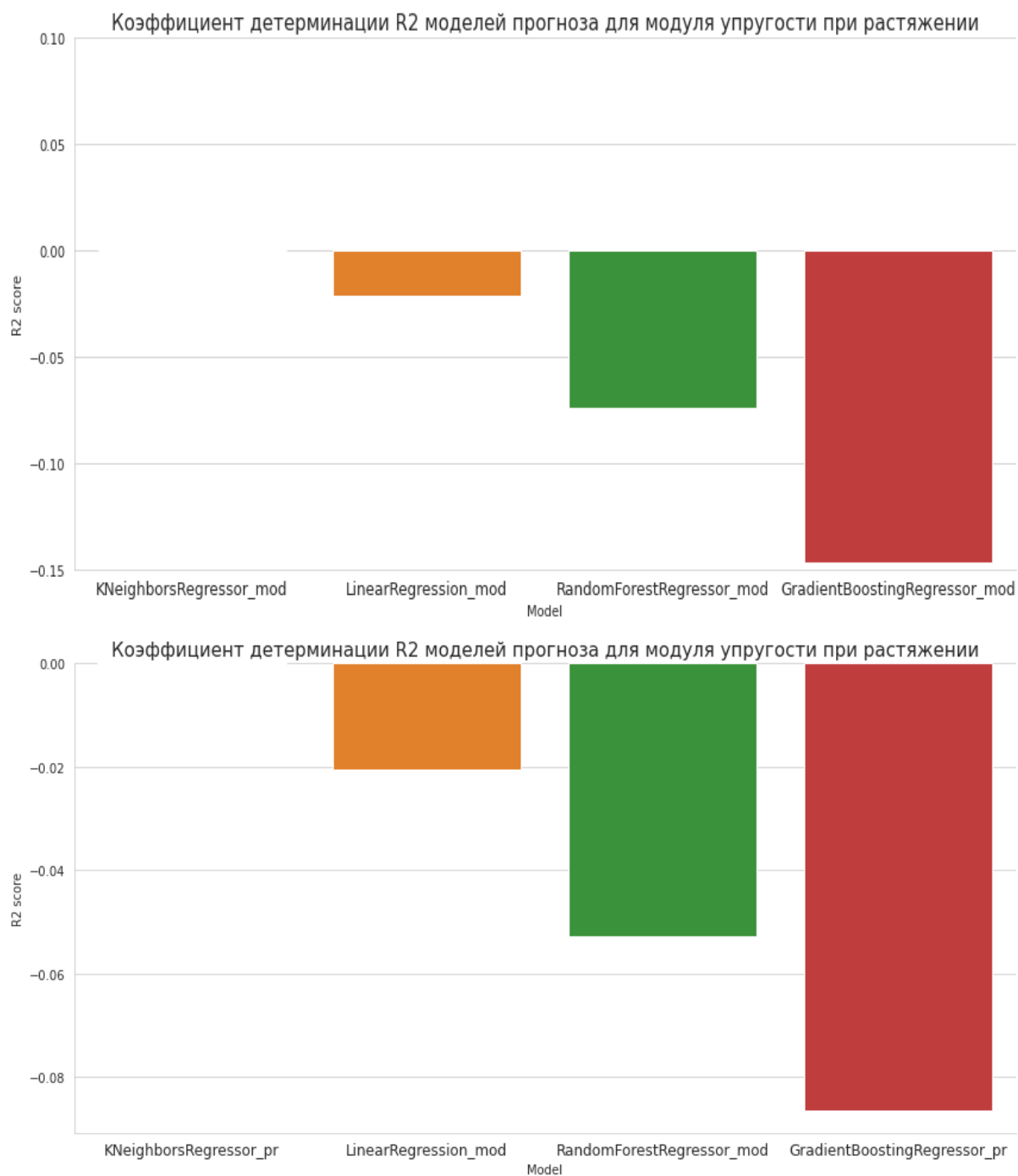


Рисунок 11 – Коэффициент детерминации

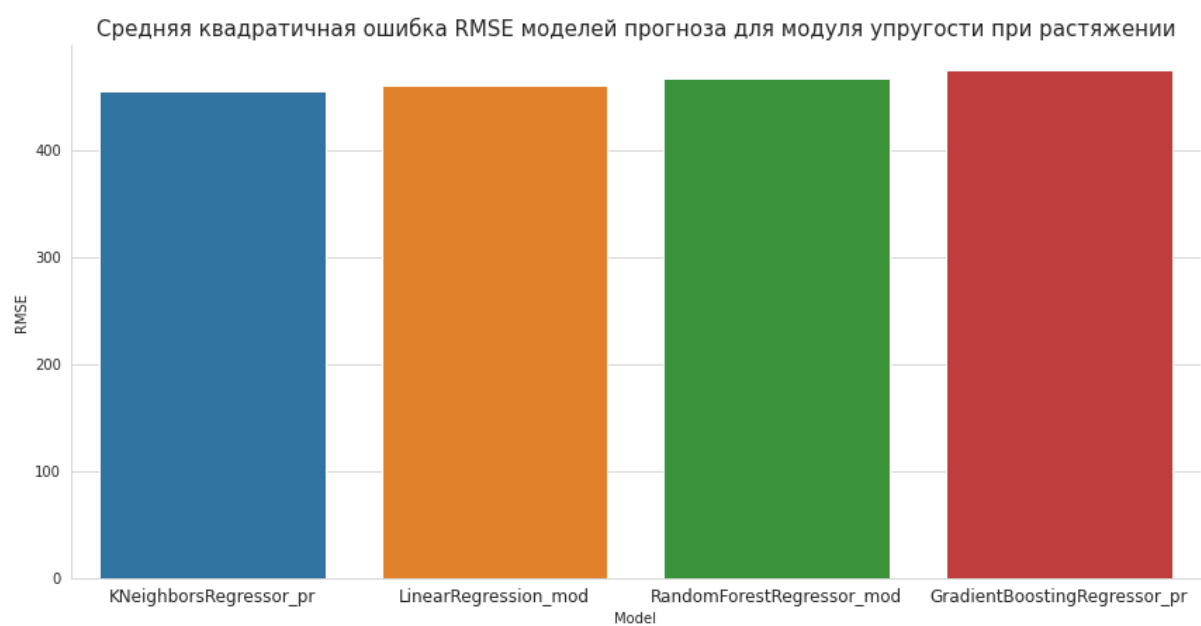
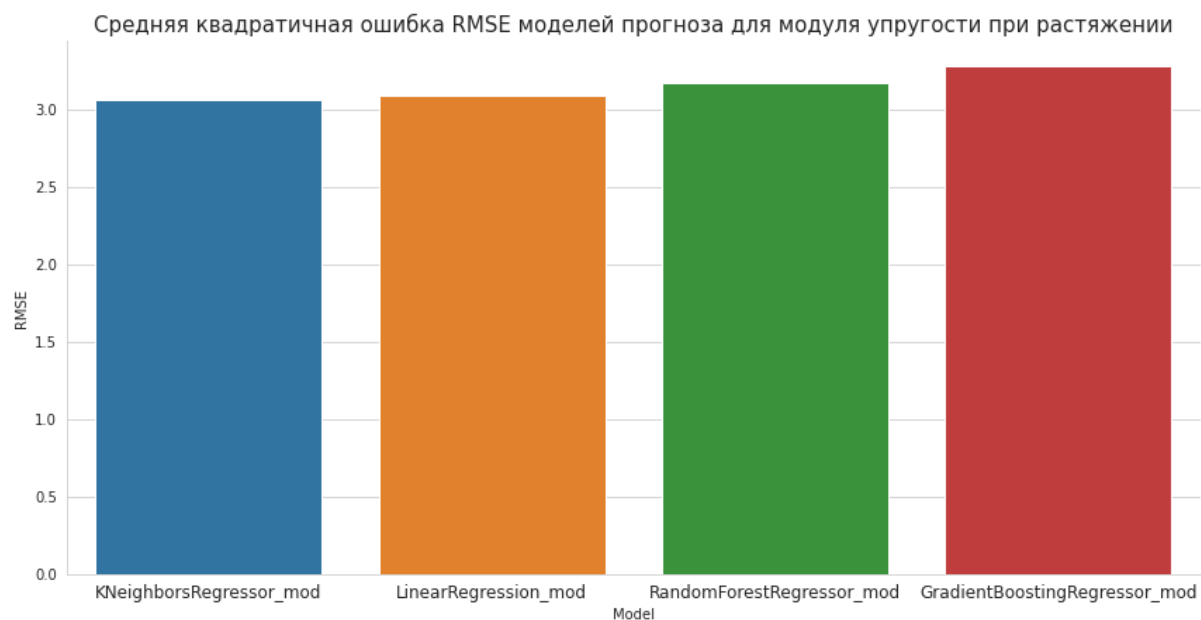


Рисунок 12 – Средняя квадратичная ошибка

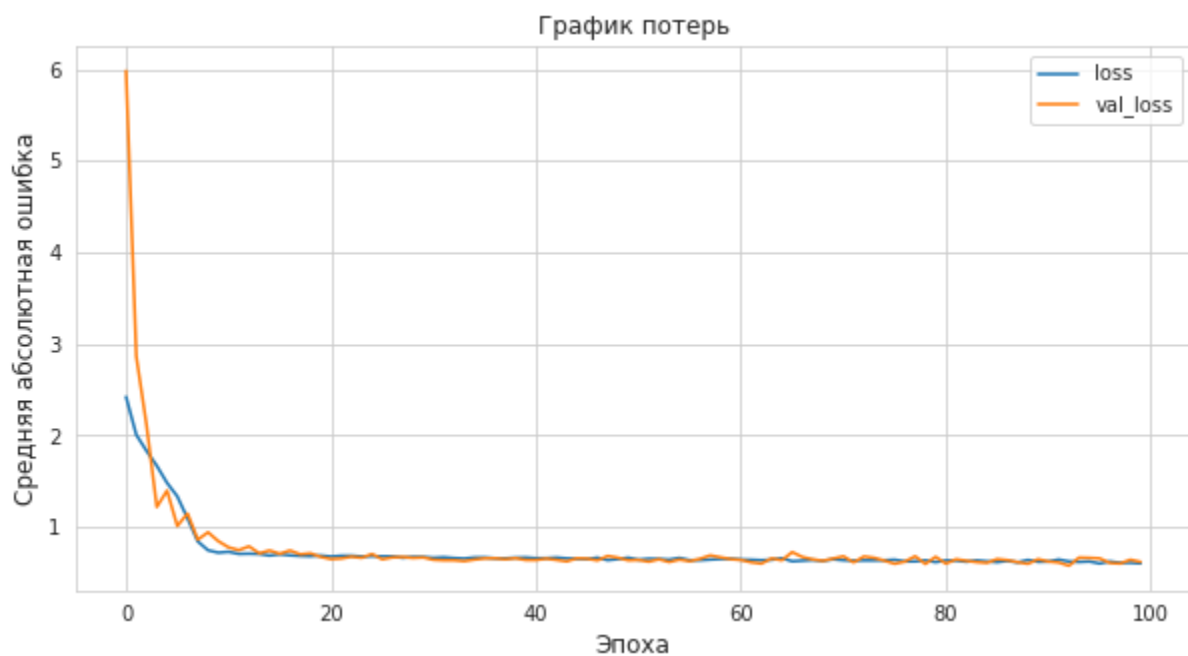


Рисунок 13 – График потерь нейронной сети



Рисунок 14 – Тестовые и прогнозные значений



Рисунок 15 – Рассеяние данных прогнозных и тестовых значений

Результаты нейронной сети тоже не удовлетворительны, как показывает визуализация. На рисунке 13 видно, что потери свелись к минимальному значению, однако дальнейшие итерации показывают, что ошибка прогнозирования показывает незначительные изменения. На рисунке 14 так же видно, что прогнозные значения усредняются относительно тестовых. На рисунке 15 заметен большой разброс данных.



### 2.3. Приложение и репозиторий

Разработано приложение, которое прогнозирует «Соотношение матрица - наполнитель %» на основе разработанной нейронной сети.

Приложение принимает входные данные на основе остальных 12 параметров ( «Плотность, кг/м3 %», «модуль упругости, ГПа %», «Прочность при растяжении, МПа %», «Количество отвердителя, м. %», «Содержание эпоксидных групп,%\_2 %», «Температура вспышки, С\_2 %», «Поверхностная плотность, г/м2 %», «Модуль упругости при растяжении, ГПа %», «Потребление смолы, г/м2 %», «Угол нашивки, град %», «Шаг нашивки %», «Плотность нашивки %») и возвращает соответствующие данные параметра «Соотношение матрица - наполнитель %», на основе сохранённой модели.

Приложение запускается локально, после запуска app.py заходим в браузер и вводим адрес прописанный в командной строке приложения.

#### Расчет соотношения матрица-наполнитель

Введите параметры

Введите Плотность, кг/м3

Введите Модуль упругости, ГПа

Введите Количество отвердителя, м.%

Введите Содержание эпоксидных групп,%\_2

Введите Температура вспышки, С\_2

Введите Поверхностная плотность, г/м2

Введите Модуль упругости при растяжении, ГПа

Введите Прочность при растяжении, МПа

Введите Потребление смолы, г/м2

Введите Угол нашивки, град

Введите Шаг нашивки

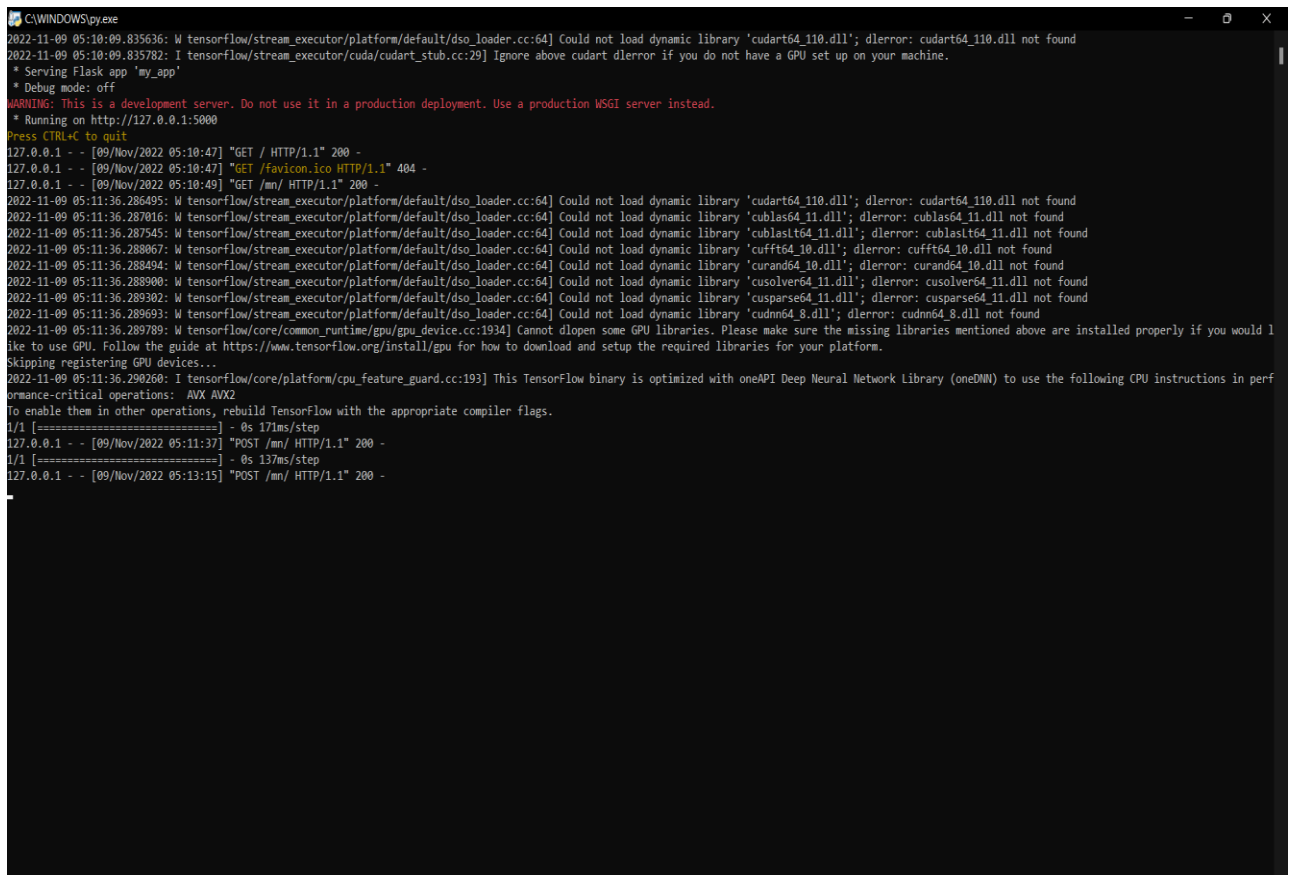
Введите Плотность нашивки

Рассчитать

Сбросить

Спрогнозированное Соотношение матрица-наполнитель для введенных параметров: [[3.256538]]

Рисунок 16 – Графический интерфейс



```
C:\WINDOWS\py.exe
2022-11-09 05:10:09.835636: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-11-09 05:10:09.835782: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
* Serving Flask app "my_app"
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [09/Nov/2022 05:10:47] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Nov/2022 05:10:47] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [09/Nov/2022 05:10:49] "GET /mn/ HTTP/1.1" 200 -
2022-11-09 05:11:36.286495: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-11-09 05:11:36.287016: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublas64_11.dll'; dlerror: cublas64_11.dll not found
2022-11-09 05:11:36.287545: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublaslt64_11.dll'; dlerror: cublaslt64_11.dll not found
2022-11-09 05:11:36.288067: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2022-11-09 05:11:36.288494: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'curand64_10.dll'; dlerror: curand64_10.dll not found
2022-11-09 05:11:36.288908: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusolver64_11.dll'; dlerror: cusolver64_11.dll not found
2022-11-09 05:11:36.289302: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusparse64_11.dll'; dlerror: cusparse64_11.dll not found
2022-11-09 05:11:36.289693: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudnn64_8.dll'; dlerror: cudnn64_8.dll not found
2022-11-09 05:11:36.289789: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1934] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2022-11-09 05:11:36.290268: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
1/1 [=====] - 0s 171ms/step
127.0.0.1 - - [09/Nov/2022 05:11:37] "POST /mn/ HTTP/1.1" 200 -
1/1 [=====] - 0s 137ms/step
127.0.0.1 - - [09/Nov/2022 05:13:15] "POST /mn/ HTTP/1.1" 200 -
```

Рисунок 17 – Работа приложения

Был создан отдельный репозиторий для проекта выпускной работы на GitHub: [https://github.com/HanzoSC/bmstu\\_vkr\\_datascience](https://github.com/HanzoSC/bmstu_vkr_datascience)

- BMSTU\_VKR\_DataScience.ipynb - практическая часть;
- presentation.pptx – презентация;
- app - приложение (для работы приложения нужно скачать папку целиком, для запуска необходимо запустить app.py);
- diplom.pdf - пояснительная записка;
- X\_bp.xlsx, X\_nip.xlsx - исходные данные.

## Заключение

В ходе выполнения данной работы были построены регрессионные модели машинного обучения для предсказания «Модуль упругости, гПа %» и «Прочность при растяжении %» и модель нейронной сети «Соотношение матрица - наполнитель %». В целом все результаты проделанной работы можно улучшить, за счёт применения более глубокого подхода к нормализации данных и доработки моделей как нейронной сети, так и моделей машинного обучения.

## Список используемой литературы и веб ресурсы

1. Кербер М. Л. Полимерные композиционные материалы: структура, свойства, технология: Учебное пособие, 2011, 560с.
2. Бринк Х., Ричардс Дж., Феверолф М. Машинное обучение, 2017, 545 с.
3. Среда разработки Google Colab - Режим доступа: <https://colab.research.google.com/> (дата обращения 7.11.2022)
4. Язык программирования Python - Режим доступа: <https://www.python.org/> (дата обращения 7.11.2022)
5. Библиотека Pandas - Режим доступа: <https://pandas.pydata.org/> (дата обращения 7.11.2022)
6. Библиотека Matplotlib - Режим доступа: <https://matplotlib.org/> (дата обращения 7.11.2022)
7. Библиотека Sklearn - Режим доступа: <https://scikit-learn.org/stable/> (дата обращения 7.11.2022)
8. Библиотека Tensorflow: Режим доступа: <https://www.tensorflow.org/> (дата обращения 7.11.2022)