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 Заргарян Карен Гургенович

Москва, 2023

**Оглавление**

[**Введение** 3](#_Toc133281962)

[1. **Аналитическая часть** 4](#_Toc133281963)

[**1.1.** **Постановка задачи** 4](#_Toc133281964)

[**1.2.** **Методы разведочного анализа данных (EDA)** 5](#_Toc133281965)

[**1.3.** **Методы предобработки данных (ETL)** 6](#_Toc133281966)

[**1.4.** **Методы прогнозирования** 7](#_Toc133281967)

[**1.5.** **Метрики оценки качества прогнозирования** 12](#_Toc133281968)

[**1.6.** **Pipline машинного обучения** 14](#_Toc133281969)

[2. **Практическая часть** 15](#_Toc133281970)

[**2.1.** **Разведочный анализ данных** 15](#_Toc133281971)

[**2.2.** **Разработка и обучение модели** 20](#_Toc133281972)

[**2.3.** **Построение нейронной сети для прогнозирования соотношения матрица-наполнитель** 23](#_Toc133281973)

[**2.4.** **Разработка приложения** 26](#_Toc133281974)

[**Заключение** 29](#_Toc133281975)

[**Список литературы и веб ресурсы** 30](#_Toc133281976)

# **Введение**

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

Композиционные материалы - это искусственно созданные материалы, состоящие из нескольких других с четкой границей между ними. Композиты обладают теми свойствами, которые не наблюдаются у компонентов по отдельности. При этом композиты являются монолитным материалом, т. е. компоненты материала неотделимы друг от друга без разрушения конструкции в целом. Яркий пример композита - железобетон. Бетон прекрасно сопротивляется сжатию, но плохо растяжению. Стальная арматура внутри бетона компенсирует его неспособность сопротивляться сжатию, формируя тем самым новые, уникальные свойства.

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

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).

# **Аналитическая часть**

## **Постановка задачи**

В данной работе исследуется композит с матрицей из базальтопластика и нашивками из углепластика. Для работы получен датасет, содержащий данные о свойствах матрицы и наполнителя, производственных параметрах и свойствах готового композита. Требуется разработать модели, прогнозирующие значения некоторых свойств в зависимости от остальных. Так же требуется разработать приложение, позволяющее использовать результирующую обученную модель специалистом предметной области.

В соответствии с заданием требуется:

1) Изучить теоретические основы и методы решения поставленной задачи.

2) Провести разведочный анализ предложенных данных. Необходимо нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек. Необходимо также для каждой колонке получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков.

3) Провести предобработку данных (удаление шумов, нормализация и т.д.).

4) Обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении. При построении модели необходимо 30% данных оставить на тестирование модели, на остальных происходит обучение моделей. При построении моделей провести поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10.

5) Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель.

6) Разработать приложение с графическим интерфейсом или интерфейсом командной строки, которое будет выдавать прогноз, полученный в задании 4 или 5 (один или два прогноза, на выбор учащегося).

7) Оценить точность модели на тренировочном и тестовом датасете.

8) Создать репозиторий в GitHub / GitLab и разместить там код исследования. Оформить файл README.

Датасет состоит из двух файлов в формате MS Excel: X\_bp (составляющая из базальтопластика) и Х\_nup (составляющая из углепластика).

Файл X\_bp содержит 10 признаков и индекс, всего строк: 1023.

Файл X\_nup содержит 3 признака и индекс, всего строк: 1040.

Для получения итогового датасета для анализа необходимо объединить два входных файла типом INNER по индексу. После объединения часть строк из файла X\_nup была отброшена в связи с неперсекающимися индексами в файле X\_bp. Итоговой датасет содержит 13 признаков и 1023 строки (объектов).

Все признаки имеют вещественный тип. Пропусков в данных нет, все признаки по всем объектам имеют вещественные значения. Все признаки, кроме «Угол нашивки», являются количественными. Признак «Угол нашивки» принимает только два значения 0 град и 90 град. Это категориальный признак, состоящий из двух категорий.

## **Методы разведочного анализа данных (EDA)**

Аббревиатура процесса разведочного анализа данных EDA расшифровывается как «Exploratory Data Analysis».

Для разведочного анализа исходных данных с целью формирования представления об их составе и структуре, а также определения необходимых преобразовании в настоящей работе используются следующие методы из библиотек Numpy, Pandas и Seaborn:

- pd.read\_excel – загрузка исходных таблиц в Notebook;

- .join – объединение исходных таблиц в один датасет;

- .shape() – размерность датасета;

- .info() –наименования столбцов и размерность датасета, тип данных в датасете (int, float, str и т.д.);

- .describe() – минимальные, средние, максимальные значения, стандартные отклонения, процентные перцентили в разрезе столбцов датасета;

- .isnull().sum() – анализ пустых NuN ячеек в датасете;

- sns.heatmap – визуализация корреляции переменных между собой;

- .corr() – список корреляционных значений переменных;

- sns.boxplot – визуализация и анализ наличия выбросов;

- sns.histplot – визуализация и анализ распределения значений переменных;

- sns.pairplot – попарная визуализация и анализ распределения значений переменных.

## **Методы предобработки данных (ETL)**

Аббревиатура процесса предобработки данных ETL расшифровывается как «Extract, Transformer, Load».

В настоящей работе для приведения исходных данных к формату, удобному для их дальнейшей обработки моделями машинного обучения, применяется метод стандартизации StandartScaler из библиотеки Sklearn.preprocessing – масштабирование данных путем их приведения к интервалу со средним нулевым значением и стандартным отклонением равным единице с целью соблюдения единого диапазона значений и масштаба.

Для удобства работы с переменными выполнено частичное переименование переменных посредством применения метода .rename().

Также для разделения массива данных на тестовую и обучающую выборки для процесса обучения, тестирования и оценки результатов применяется метод train\_test\_split из библиотеки Sklearn.model\_selection.

## **Методы прогнозирования**

Прогнозирование конечных свойств композиционных материалов является задачей регрессии в машинном обучении, поэтому для прогнозирования значений модуля упругости при растяжении применяется модель Linear Regression из библиотеки Sklearn.linear\_model и ансамблевая модель Random Forest Regression из библиотеки Sklearn.ensemble.

Для прогнозирования значений прочности при растяжении применяется модель Support Vector Regression из библиотеки Sklearn и модель Decision Tree Regression из библиотеки Sklearn.tree.

Для прогнозирования соотношения матрица-наполнитель выполняется построение искусственной последовательной нейронной сети.

Для поиска гиперпараметров моделей используются методы поиска по сетке Grid Search и случайного поиска Random Search, а для поиска гиперпараметров нейронной сети используется Keras Tuner.

Рассмотрим описание, достоинства и недостатки принятых к использованию методов прогнозирования.

1. Linear Regression.

Множественная линейная регрессия - это модель линейной регрессии, которая оценивает взаимосвязь между несколькими независимыми переменными (признаками) и одной зависимой переменной.

Используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной «y» от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) «x» с линейной функцией зависимости.

Линейная регрессия использует метод, известный как обычный метод наименьших квадратов, чтобы найти наиболее подходящее уравнение регрессии.

1. Decision Tree Regression.

Регрессия дерева решений строит регрессионную модель в виде древовидной структуры. Структура дерева представляет собой «листья» и «ветки». На рёбрах («ветках») дерева решения записаны признаки, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах — признаки, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе. Каждый лист представляет собой значение целевой переменной, изменённой в ходе движения от корня по рёбрам дерева до листа. Каждый внутренний узел сопоставляется с одной из входных переменных. Алгоритм вычисляет информационный прирост для каждой характеристики и выбирает ту, которая дает наивысшее значение.

1. Random Forest Regression.

Алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга и метод случайных подпространств. Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество решения, но за счёт их большого количества результат получается хорошим. Прогнозирование конечного значения выполняется за счет расчета среднего значения по всем построенным деревьям.

1. Support Vector Regression.

Метод классификации опорных векторов может быть расширен для решения задач регрессии. Этот метод называется регрессией опорных векторов.

Модель, созданная с помощью классификации опорных векторов (зависит только от подмножества обучающих данных, поскольку функция затрат для построения модели не заботится о точках обучения, которые лежат за пределами поля. Аналогично, модель, созданная с помощью регрессии опорных векторов, зависит только от подмножества обучающих данных, поскольку функция стоимости игнорирует выборки, прогнозирование которых близко к их цели.

5. Gradient Boosting Regressor

Градиентный бустинг, является представителем ансамблевых методов, идея которого заключается в итеративном процессе последовательного построения частных моделей решающего дерева. Каждая новая модель обучается с использованием информации об ошибках, сделанных на предыдущем этапе, а результирующая функция представляет собой линейную комбинацию всего ансамбля моделей с учетом минимизации любой штрафной функции.

Бустинг, использующий деревья решений в качестве базовых алгорит-мов, называется градиентным бустингом над решающими деревьями. Он от-лично 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. 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 нейросети бывают многослойными: выход одного слоя передается следующему так, как описано выше. На выходе нелинейность не применяется.

Нейронные сети тренируются с помощью метода стохастического градиента и алгоритма обратного распространения ошибки.

В таблице 1 отражены основные преимущества и недостатки используемых методов прогнозирования

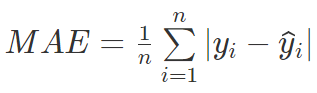
Таблица 1 - Преимущества и недостатки используемых методов

|  |  |  |
| --- | --- | --- |
| Регрессионный метод | Преимущества | Недостатки |
| Linear Regression | * быстрое моделирование (особенно если отсутствует большой объём данных); * легко объясняема и интерпретируема; * достаточно хорошо работает на большом наборе данных. | * сложности проектирования в случае нелинейных данных, необходимость наличия информации о структуре данных и взаимосвязи между переменными. |
| Decision Tree Regression | * подходит для изучения сложных линейных и нелинейных отношений; * основные алгоритмы просты в понимании и реализации. Границы решений, которые создаются во время обучения, легко понять; * отсутствие требования масштабирования данных. | * склонность к переобучению. Завершенная модель дерева решений может быть чрезмерно сложной и содержать ненужную структуру; * плохие результаты на небольших наборах данных. |
| Random Forest Regression | * подходит для изучения сложных линейных и нелинейных отношений; * основные алгоритмы просты в понимании и реализации. Границы решений, которые создаются во время обучения, легко понять; * отсутствие требования масштабирования данных. | * склонность к переобучению. Завершенная модель дерева решений может быть чрезмерно сложной и содержать ненужную структуру. * используя большие случайные леса для достижения более высокой производительности, расходуется память и время. |
| Support Vector Regression | * эффективен в пространствах с высокой размерностью. * эффективен в случаях, когда количество измерений больше, чем количество выборок. * использует подмножество обучающих точек в функции принятия решений (называемых опорными векторами), поэтому он также экономит память. * универсальность: для функции принятия решения могут быть указаны различные [функции ядра](https://scikit-learn.org/stable/modules/svm.html#svm-kernels). Предоставляются общие ядра, но также возможно указать пользовательские ядра. * хорошо адаптируется; * хорошо работает для нелинейных задач; * отсутствие предвзятости к объекту, отличающемуся от других. | * если количество функций намного больше, чем количество выборок чувствителен к чрезмерной подгонке при выборе [функций ядра](https://scikit-learn.org/stable/modules/svm.html#svm-kernels), термин регуляризации имеет решающее значение. * SVM напрямую не предоставляют оценки вероятности, они рассчитываются с использованием дорогостоящей пятикратной перекрестной проверки * требуется масштабирование данных; * труден для интерпретации. |
| Нейронная сеть | • эффективна при моделировании сложных нелинейных отношений ввиду многослойности;  • гибкость: не нужно беспокоиться о структуре данных в нейронных сетях;  • производительность растет с увеличением тренировочных данных. | * возможная сложность архитектуры и модели в целом; * требование тщательной настройки гиперпараметров и скорости обучения. * для достижения высокой производительности нейронным сетям необходимо огромное количество данных, и в результате, как правило, нейросети уступают другим ML алгоритмам в тех случаях, когда данных мало. |

## **Метрики оценки качества прогнозирования**

Для оценки точности и качества работы выбранных моделей прогнозирования и нейронных сетей применяются следующие метрики:

* + MAE (средняя абсолютная ошибка) - определяет среднее абсолютное расстояние между прогнозируемыми и целевыми значениями – то, насколько число в прогнозе разошлось с реальным числом. Данную ошибку удобно трактовать – погрешность измеряется в тех же единицах, что и значения целевой переменной.

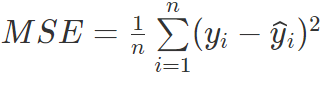
 (1)

где n – кол-во элементов;

у – реальное целевое значение;

ŷ – предсказанной целевое значение.

* + МSE (средняя квадратичная ошибка) - определяет среднеквадратичную ошибку между прогнозируемыми и целевыми значениями. Настроена на отражение влияния именно больших ошибок на качество модели. Менее удобна для понимания ввиду измерения в квадратных единицах. Данная метрика обычно применяется для сравнения моделей между собой.

 (2)

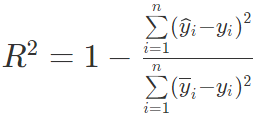
где:

n – кол-во элементов;

у – реальное целевое значение;

ŷ – предсказанной целевое значение.

* + R2 (коэффициент детерминации) - показывает, какую долю разнообразия данных модель смогла объяснить. Метрика просто интерпретируема: модель, для которой R2 больше 0,5, является удовлетворительной. Если R2 больше 0,8, то модель рассматривается как очень хорошая. Значения, меньшие 0,5, говорят о том, что модель некачественна.

 (3)

где:

n – кол-во элементов;

у – реальное целевое значение;

ŷ – предсказанной целевое значение.

Кроме того, для оценки общей точности алгоритмов прогнозирования используется оценка отношения средней абсолютной ошибки к среднему значению фактической целевой переменной:

accuracy = 100 – (MAE / y.mean() \*100) (4)

где:

y.mean() – арифметическое среднее реального целевое значения

## **Pipline машинного обучения**

Обобщая вышесказанное, можно визуализировать последовательность и шаги процесса машинного обучения, реализованного в настоящей работе.

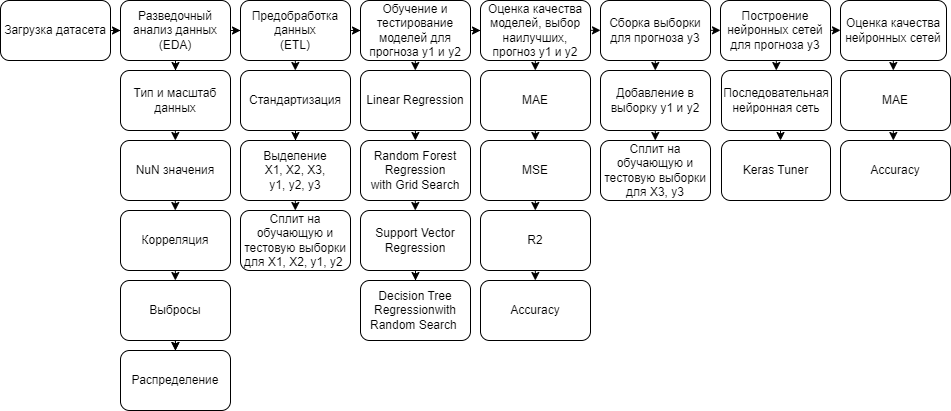
****

Рисунок 1 - Pipline машинного обучения

# **Практическая часть**

## **Разведочный анализ данных**

Разведочный анализ данных показал, что в датасете отсутствуют пропущенные пустые NuN значения, признаки представлены числовыми целыми (int64) и числовыми дробными (float64) значениями, строчные значения (str) отсутствуют, соответственно, отсутствует необходимость их категорирования.

Диапазон разброса минимальных и максимальных значений переменных достаточно велик, минимальное значение в датасете равно «0», максимальное равно «3848».

Матрица корреляции, а также числовые вычисления показывают слабую взаимосвязь переменных между собой, минимальные и максимальные значения корреляций признаков по отношению к целевым переменным исчисляются сотыми долями.

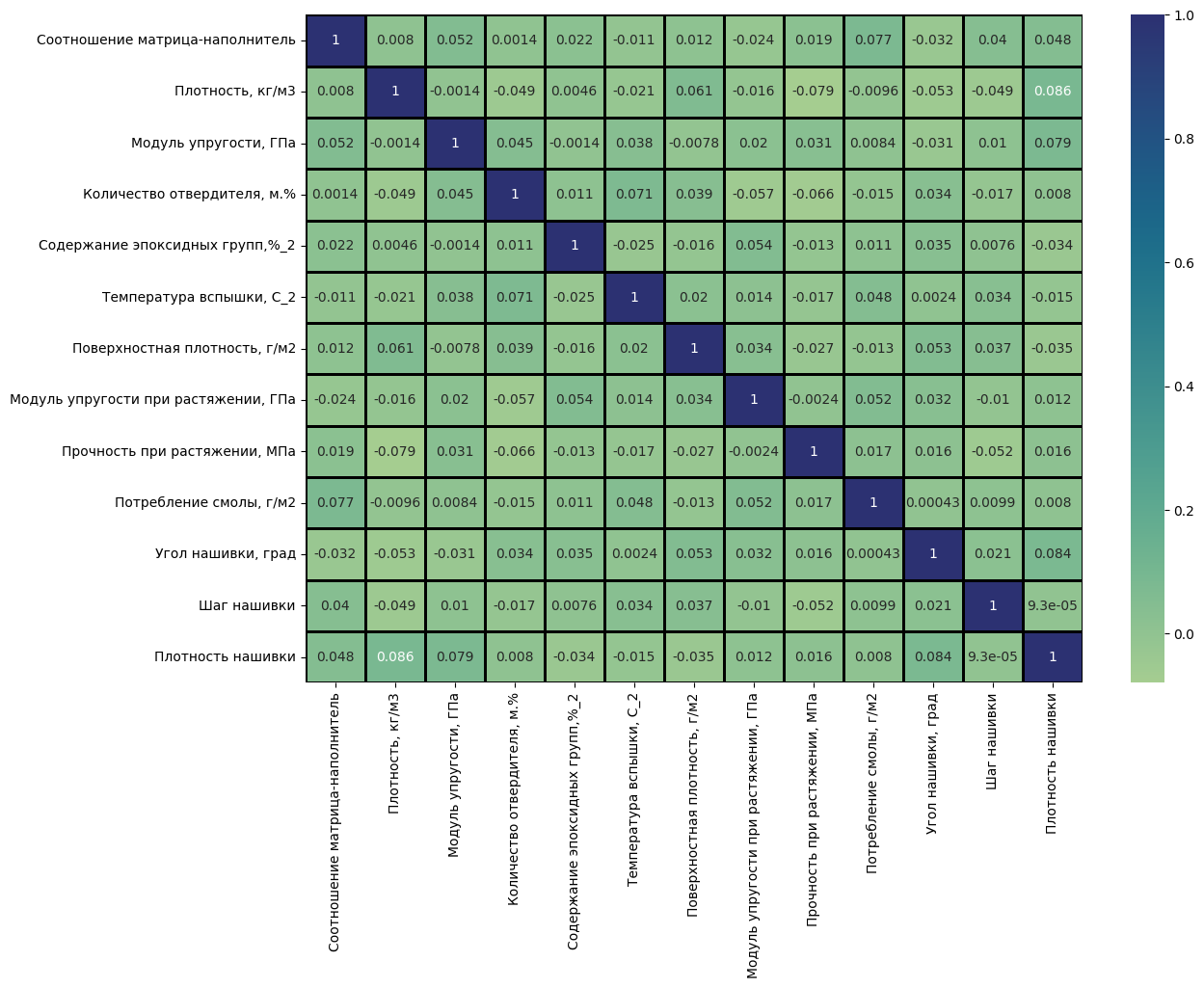
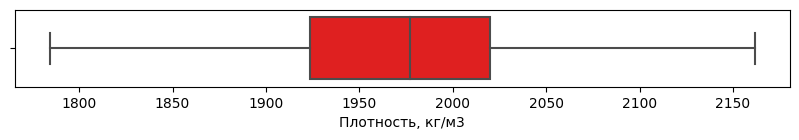


Рисунок 2 - Матрица корреляции

Анализ выбросов показал наличие значений вне минимума и максимума диапазона распределения по всем признакам. Удаления выбросов было осуществлено методом перцентиля (межквартильных расстояний).













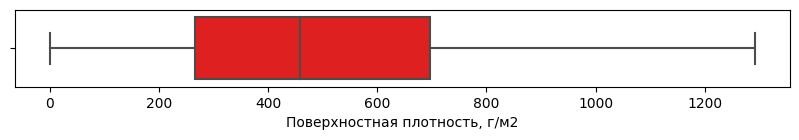




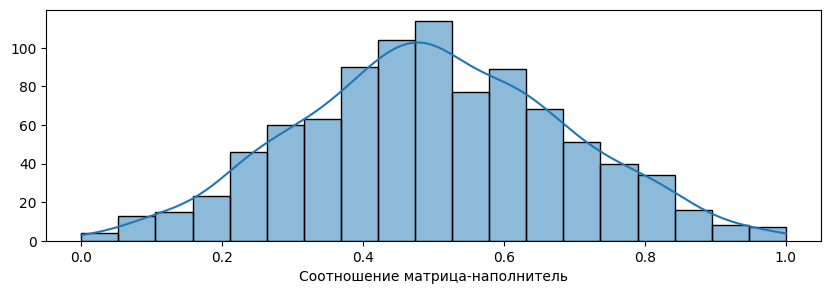


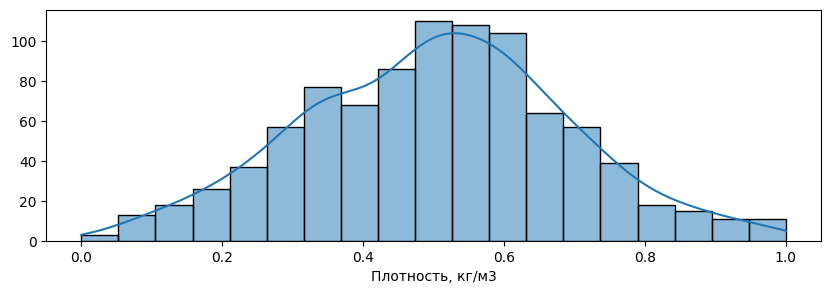
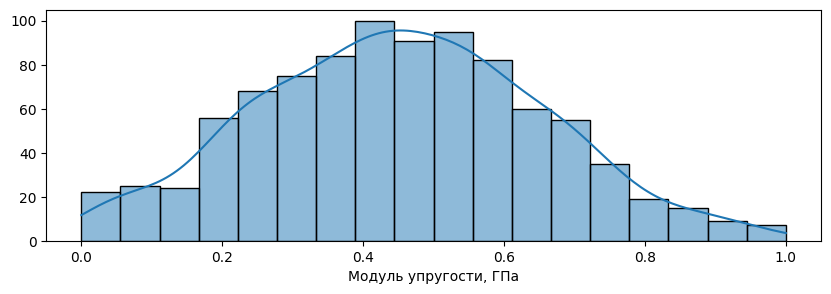
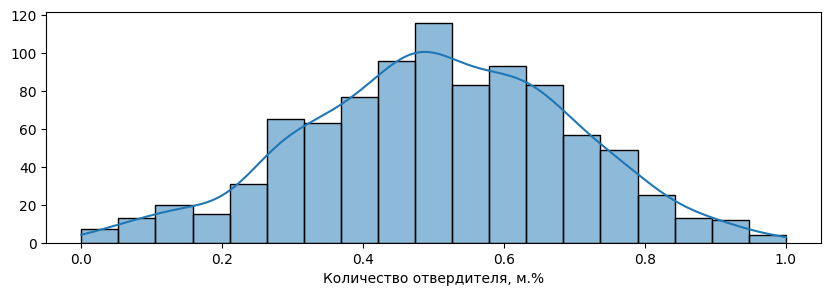
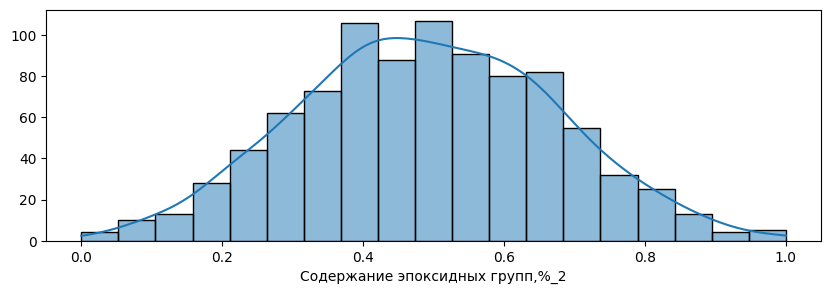
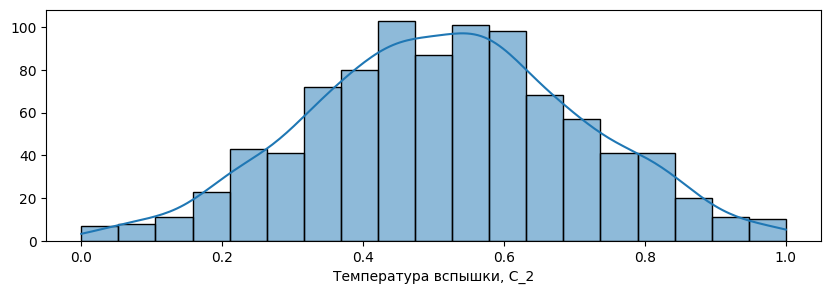
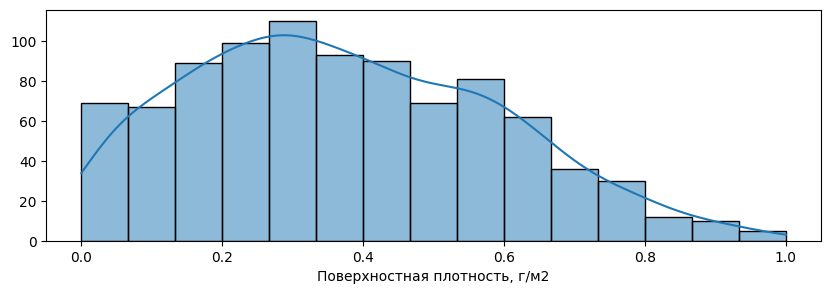
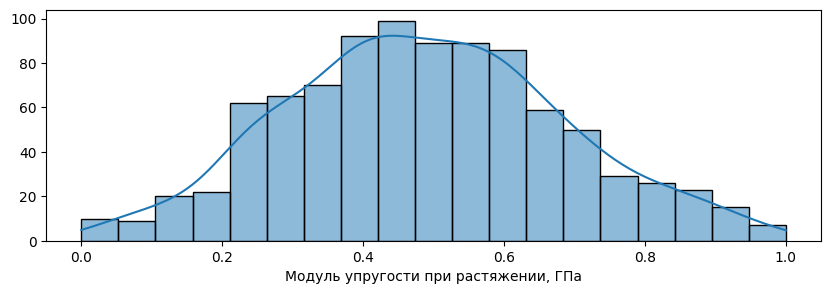
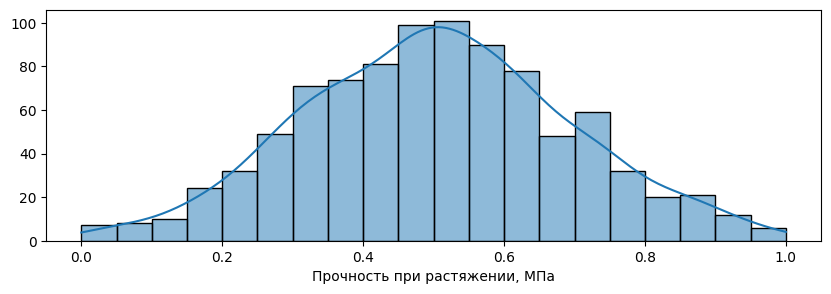
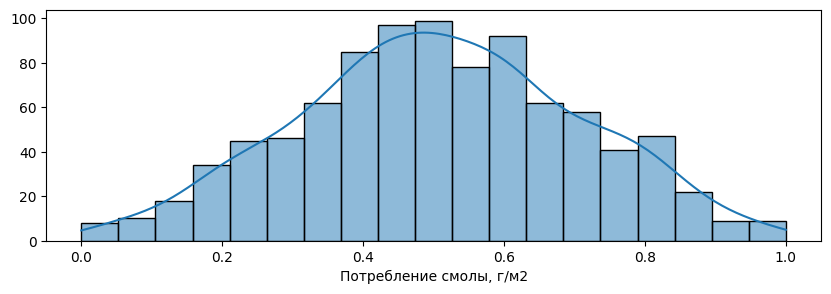
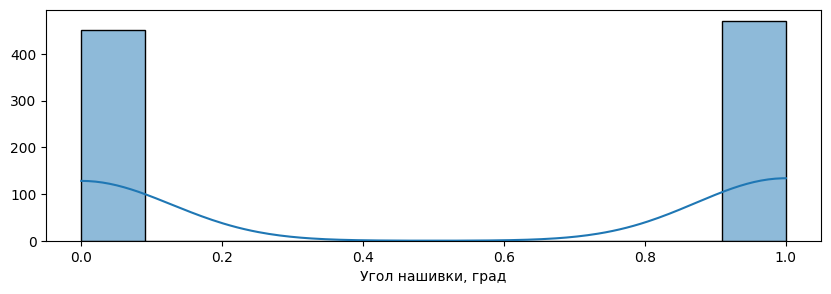
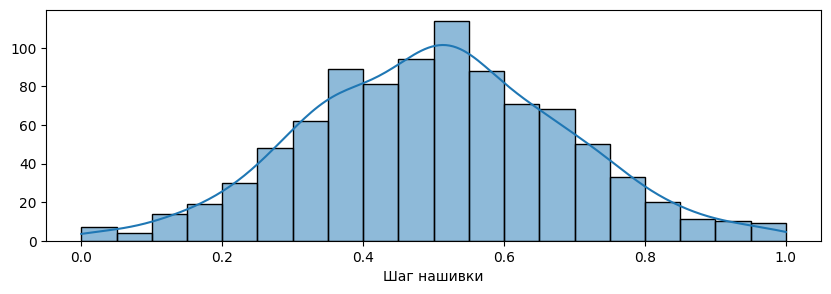
 



Рисунок 3 - Визуализированный анализ выбросов boxplot

Анализ распределения переменных показал наличие в целом распределения переменных по каждому признаку, близкому к нормальному распределению Гаусса, при этом у признака «Поверхностная плотность» наблюдается небольшое смещение значений, а распределение значение признака «Угол нашивки» полностью не соответствует нормальному.



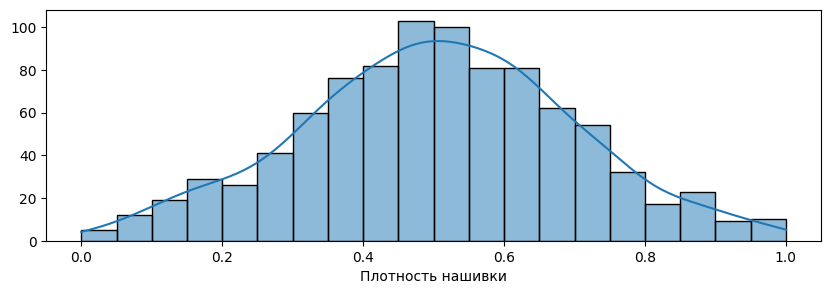
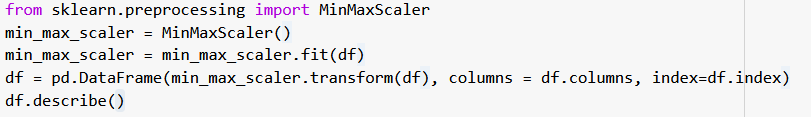


Рисунок 4 - Анализ распределения значений переменных

По полученным данным видно, что корреляция минимальна и линейной связи между признаками нет.

Для нормализации датасета был применён метод MinMaxScaler.



Результат сохранился, корреляции между признаками нет. На этом разведочный анализ закончен.

## **Разработка и обучение модели**

По заданию необходимо обучить несколько моделей для прогноза модуля упругости при растяжении и прочности при растяжении. При построении моделей провести поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10. Для выполнения данной задачи выбран модуль GridSearchCV из библиотеки sklearn, который позволяет обучать модель с подбором гиперпараметров.

Список выбранных моделей ниже:

- Метод K-ближайших соседей (KNeighborsRegressor);

- Градиентный бустинг (GradientBoostingRegressor);

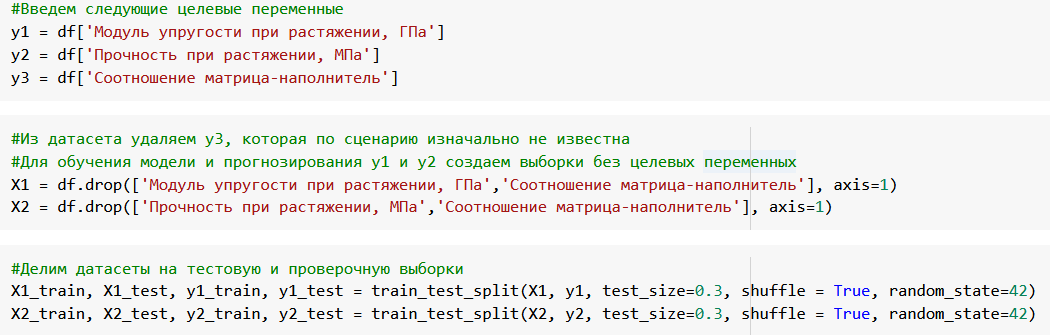
- Случайный лес (RandomForestRegressor);

- Линейная регрессия (LinearRegression).

- Метод опорных векторов (Support Vector Regression);

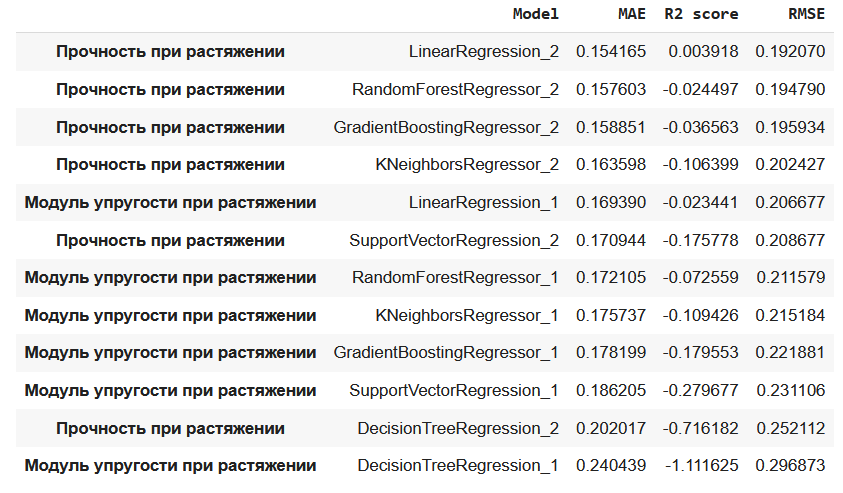
- Дерево решений (Decision Tree Regression).

Разделяем исходный датасет на «X» (feature-переменные) и «у» (target-переменные) и разделим на тестовую и тренировочные выборки:



Для прогнозирования «Модуль упругости при растяжении, ГПа» и «Прочность при растяжении, МПа » применим модели с настройками по умолчанию и соберем метрики в единую таблицу для бенчмаркинга результативности моделей (таблица 2).

Таблица 2 – Бенчмаркинг результативности моделей



Для оценки качества моделей регрессии использовались специальные показатели:

1) R2\_score (коэффициент детерминации) принимает значение от 0 до 1 и показывает долю объяснённой дисперсии объясняемого рода. Чем ближе R2 к 1, тем меньше доля необъяснённого;

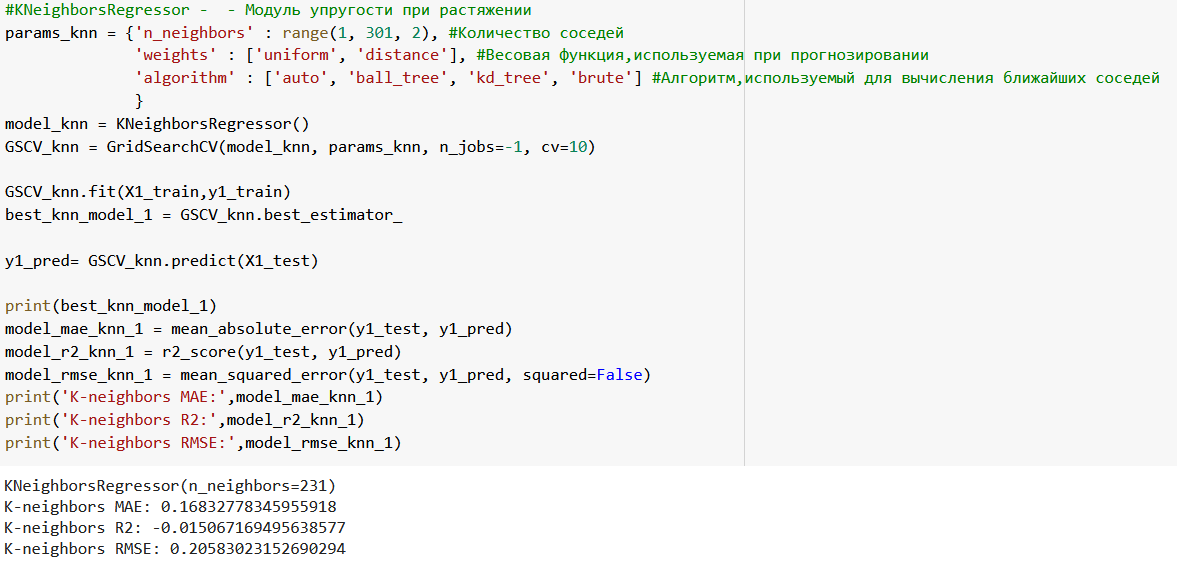
2) Mean Absolut Error (MAE)- средняя абсолютная ошибка, показывает среднее значение абсолютных отклонений между наблюдаемыми и прогнозируемыми значениями;

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

Как видно из таблицы, модели показали практически одинаковую результативность при прогнозировании первой и второй целевых переменных соответственно. При этом коэффициент детерминации практически всегда ниже 0, что свидетельствует о низкой эффективности моделей. Это связано с практически полным отсутствием корреляции между признаками датасета.

Дополнительно сделан поиск гиперпараметров с помощью поиска по сетке с перекрестной проверкой для моделей KNeighborsRegressor и RandomForestRegressor для y1 (Модуль упругости при растяжении) и y2 (Прочность при растяжении).

Пример ниже:

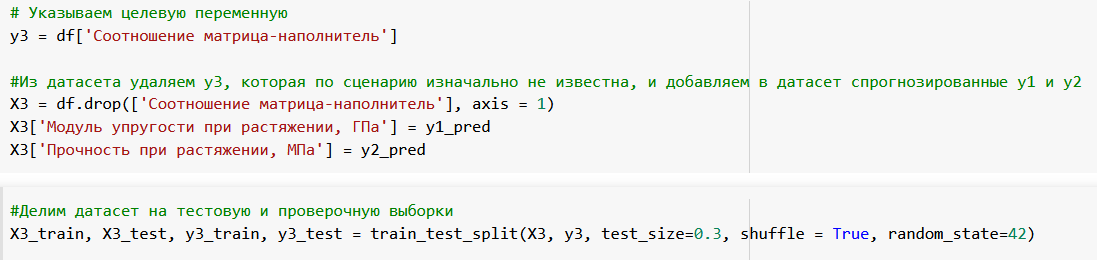


Результаты улучшились незначительно. По результатам подбора гиперпараметров и предыдущих результатов для прогноза y1 выбираем модель KNeighborsRegressor и для y2 выбираем модель LinearRegression.

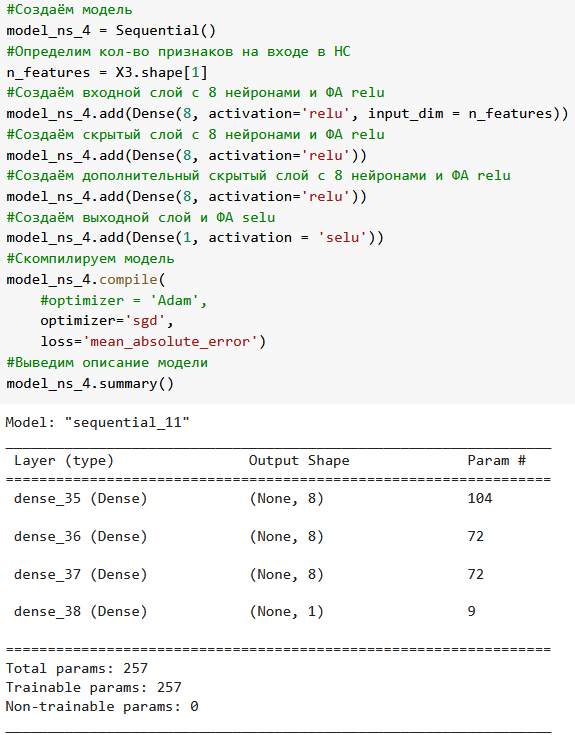


## **Построение нейронной сети для прогнозирования соотношения матрица-наполнитель**

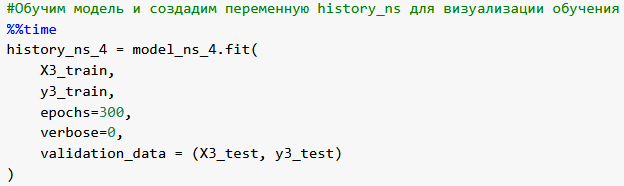
Для построения нейронной сети, прогнозирующей соотношение матрица-наполнитель, по условиям задания предварительно необходимо спрогнозировать значения модуля упругости при растяжении и прочности при растяжении. После выполнения указанных прогнозов, их масштабирования и конкатенации к набору данных Х3 получим датасет с feature-переменными.



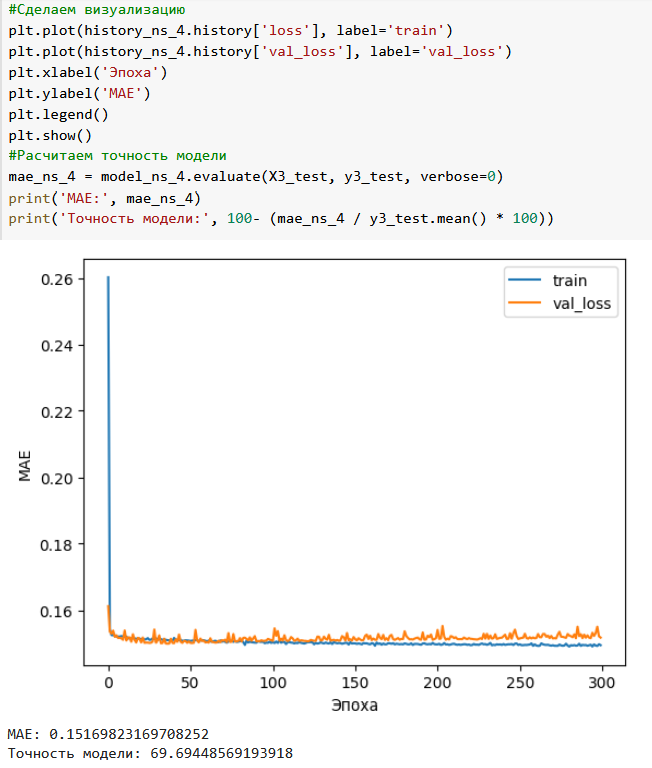
Для построения модели нейронной сети использована библиотека keras tensorflow. Создание НС на основе библиотеки TensorFlow сводится к созданию объекта класса Sequential и включению в него необходимого количества слоев с нужным количеством нейронов. После создания модели её необходимо скомпилировать, при этом происходит конвертация структуры модели в байт-код, назначить оптимизатор, который будет обновлять веса модели при обратном проходе, указать метрику точности модели. Листинг создания НС для данной работы представлен ниже:



Обучение модели происходит по вызову метода fit. Данному методу необходимо передать обучающую выборку, количество эпох обучения и валидационную выборку для оценки точности модели в процессе её обучения.



По результатам обучения сети на 300 эпохах построим график снижения MAE на обучающей и валидационной выборках и оценим точность сети.



Как видно из графика, модель перестала обучаться в самом начале и значение loss функции больше не меняется. Это говорит о том, что модель не нашла закономерностей в представленных ей данных и дальнейшее обучение проводить нет необходимости. Прогноз модели на уровне шума.

## **Разработка приложения**

Разработано приложение, которое прогнозирует «Соотношение матрица - наполнитель» на основе разработанной нейронной сети. На рисунке 5 представлен листинг приложения. Приложение запускается локально, после запуска app.py заходим в браузер и вводим адрес, прописанный в командной строке приложения.

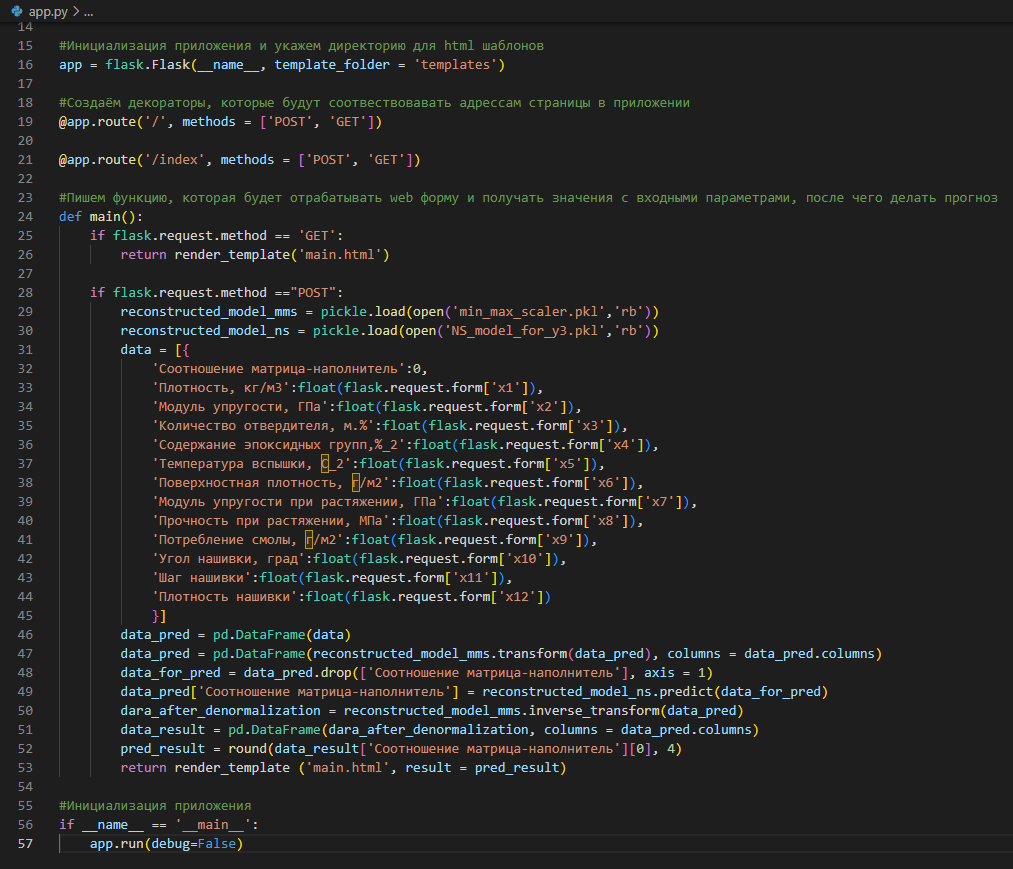
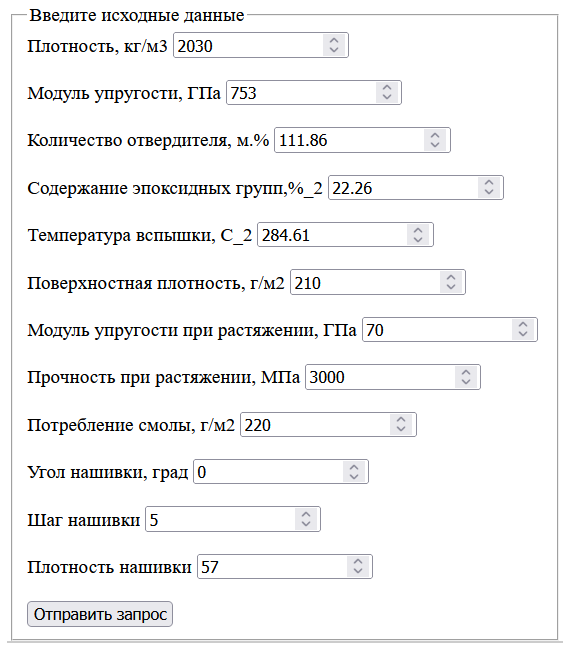


Рисунок 5 - Листинг приложения

На рисунке 6 представлен результат работы приложения. Необходимо поочередно ввести значения входных признаков. При этом происходит нормализация введенных значений, прогноз и его обратная трансформация. После чего выводится прогнозное значение.



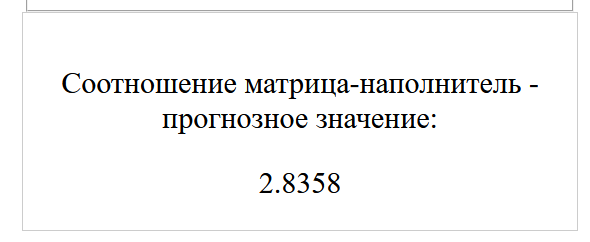


Рисунок 6 - Графический интерфейс приложения

Дополнительно создан отдельный репозиторий для проекта выпускной работы на GitHub: https://github.com/ZKaren88/VKR.git.

# **Заключение**

В настоящей выпускной квалификационной работе были рассмотрены аспекты решения задач регрессии в машинном обучении, а именно: актуальность задачи, шаги и методы разведочного анализа и предобработки данных, некоторые существующие модели прогнозирования и искусственные нейронные сети, алгоритмы автоматического подбора гиперпараметров используемых для прогноза механизмов, а также метрики оценки работы таких механизмов.

Также были рассмотрены способы создания flask-приложения для развертывания модели для общего доступа, был создан репозиторий с материалами проекта на сайте [www. GitHub.com](http://www.gitlab.com).

Рассмотренные алгоритмы предобработки данных и прогнозирования в целом подтверждают свою гибкость и эффективность, при этом некоторые полученные метрики результатов работы моделей и нейронных сетей лишний раз свидетельствуют о необходимости наличия достаточно объемного исходного массива данных с присутствием взаимосвязей и зависимостей элементов внутри для построения действительно близкого к реальным значениям прогноза.

# **Список литературы и веб ресурсы**

* 1. А.В. Протодьяконов, П.А. Пылов, В.Е. Садовников. Алгоритмы Data Science и их практическая реализация на Python: учебное пособие. Москва, Вологда, «Инфра-Инженерия», 2022, 392 с.;
  2. Хайкин Саймон. Нейронные сети: полный курс, 2-е издание. Издательский дом «Вильямс», 2006, 1104 с.;
  3. Дюк В., Самойленко А. Data minig: учебный курс. Издательский дом «Питер», 2001, 368 с.;
  4. Грас Д. Data science: наука о данных с нуля. БВХ-Петербург, 2021, 416 с.;
  5. А. Джулли, с. Пал. Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow. ДМК пресс, 2018, 294 с.;
  6. Аллен Б. Дауни .Основы Python. Научитесь думать как программист. Манн, Иванов и Фербер, 2021, 304 с.;
  7. Б. Любанович. Простой Python. Современный стиль программирования. 2-е изд. СПб.: Питер, 2021, 592 с.;
  8. Рассел Стюарт, Норвиг Питер. Искусственный интеллект: современный подход, 2-е изд.. Издательский дом ‘‘Вильямс’’, 2007, 1408 с.;
  9. Жерон Орельен. Прикладное машинное обучение с помощью Scikit-Learn, Keras и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем, 2-е изд. СПб.: «Диалектика», 2020, 1040 с.;
  10. Справочник по библиотеке Matplotlib: <https://matplotlib.org>;
  11. Справочник по библиотеке Seaborn <https://seaborn.pydata.org>;
  12. Справочник по библиотеке Scikit-Learn <https://scikit-learn.org>;
  13. Справочник по библиотеке TensorFlow <https://www.tensorflow.org>.