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 Григорьев Антон Михайлович

Москва, 2023

**Содержание**

[Введение 4](#_Toc101899125)

[1. Аналитическая часть 6](#_Toc101899126)

[1.1. Постановка задачи 6](#_Toc101899127)

[1.2 Описание используемых методов 9](#_Toc101899128)

[1.2.1 Линейная регрессия 10](#_Toc101899129)

[1.2.2 Регрессия k-ближайших соседей 11](#_Toc101899130)

[1.2.3 Случайный лес 11](#_Toc101899131)

[1.2.4 Многослойный перцептрон 12](#_Toc101899132)

[1.2.5 Лассо регрессия 13](#_Toc101899133)

[1.2.6 Метрики качества моделей 13](#_Toc101899134)

[1.3. Разведочный анализ данных 14](#_Toc101899135)

[2. Практическая часть 16](#_Toc101899136)

[2.1. Предобработка данных 16](#_Toc101899137)

[2.2 Разработка и обучение регрессионных моделей 17](#_Toc101899138)

[2.3 Результаты работы моделей 18](#_Toc101899139)

[2.3.1 Линейная регрессия 19](#_Toc101899140)

[2.3.2 Регрессия k-ближайших соседей 20](#_Toc101899141)

[2.3.3 Случайный лес 21](#_Toc101899142)

[2.3.4 Многослойный перцептрон 22](#_Toc101899143)

[2.3.5 Лассо регрессия 24](#_Toc101899144)

[2.4 Разработка нейронной сети для прогнозирования соотношения матрица-наполнитель 26](#_Toc101899145)

[2.5. Разработка приложения 28](#_Toc101899146)

[2.6. Создание удаленного репозитория 31](#_Toc101899147)

[Заключение 32](#_Toc101899148)

[Список использованных источников 33](#_Toc101899149)

# Введение

Композиционные материалы представляют собой сочетание нескольких химически разнородных компонентов с границей раздела между ними. За счет выбора комбинаций армирующих компонентов и наполнителей, изменения их объемных долей, размеров, формы, ориентации и прочности связи по границе раздела результирующие свойства композита могут меняться в значительных пределах. Возможность получения материалов с уникальными свойствами не присущими отдельным компонентам обуславливает широкое применение композиционных материалов в различных областях техники. Примеры композита – железобетон (сочетание стали арматуры и камня бетона), древесно-волокнистая плита ДВП (сочетание древесной основы – щепы и полимерного связующего).

Традиционно разработка композитных материалов – это долгосрочный процесс, так как по характеристикам отдельных компонентов невозможно рассчитать итоговые свойства композита. Для получения заданных свойств требуется большое количество испытаний различных комбинаций, что делает актуальной задачу прогнозирования успешное решение которой позволило бы снизить расходы и трудозатраты по разработке новых материалов.

Сформулируем цель работы: создание системы прогнозирования параметров композитного материала на основе методов машинного обучения. Для достижения поставленной цели необходимо решить следующие задачи:

1. изучить теоретические основы и методы машинного обучения (задача регрессии);
2. провести разведочный анализ экспериментальных данных;
3. выполнить подготовку обучающих и тестовых выборок (предобработку данных);
4. сформировать и обучить несколько моделей для прогноза модуля упругости при растяжении и прочности при растяжении;
5. разработать нейронную сеть, для рекомендации (прогноза) соотношения матрица-наполнитель;
6. разработать приложение с графическим интерфейсом для выполнения прогноза соотношения «матрица-наполнитель» по произвольно введённым пользователем входным параметрам;
7. оценить точность модели на тренировочном и тестовом наборах данных.

# 1. Аналитическая часть

## 1.1. Постановка задачи

Необходимо сделать модели для прогноза модуля упругости при растяжении, прочности при растяжении и соотношения «матрица-наполнитель» по входным параметрам из файлов с данными о параметрах базальтопластика X\_bp.xlsx и нашивки углепластика X\_nup.xlsx.

Результирующий набор данных содержит 13 признаков и 1023 строки (Рисунок 1). Пропусков в данных нет. Все признаки, кроме «Угол нашивки», являются непрерывными, количественными, имеют вещественный тип. «Угол нашивки» принимает только два значения и будет закодирован как категориальный признак.

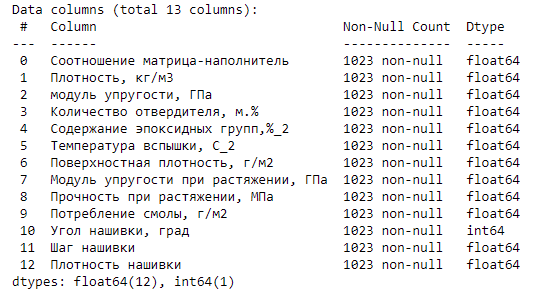


Рисунок 1 – Данные для анализа

Гистограммы распределения переменных (для примера в тексте приведены первые две переменные датасета, остальные находятся в коде) и диаграммы «ящик с усами» приведены на рисунке 2. По ним видно, что все признаки, кроме «Угол нашивки», имеют распределение близкое к нормальному и принимают неотрицательные значения. «Угол нашивки» принимает значения: 0, 90. По боксплотам видно наличие выбросов. Далее при предобработке данных выбросы будут удалены.

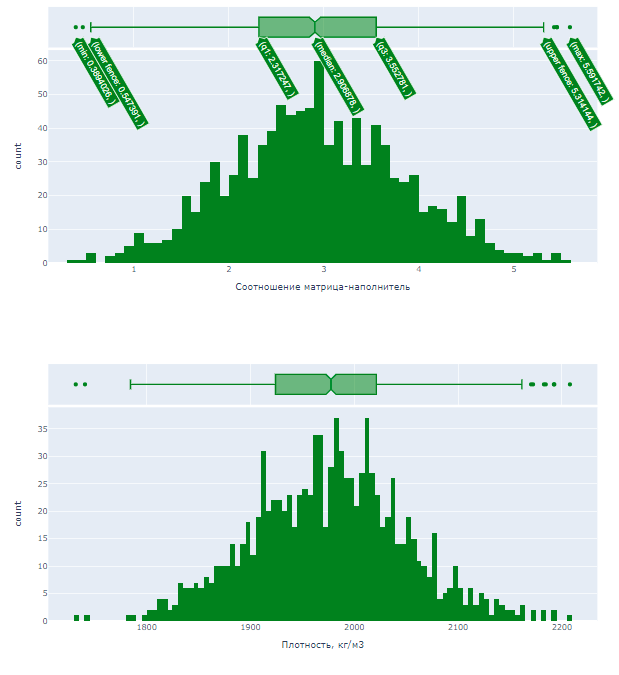


Рисунок 2 - Гистограммы распределения переменных (на примере первых двух переменных датасета)

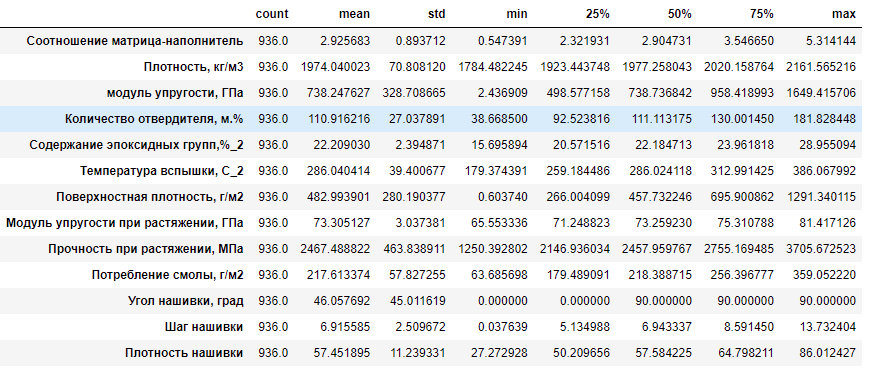
Описательная статистика (медиана, среднее, стандартное отклонение, минимум, максимум, квартили) представлена на рисунке 3.

Рисунок 3 — Описательная статистика признаков датасета

Попарные графики рассеяния точек приведены на рисунке 4. По ним также видны точки, отстоящие от основного облака данных – выбросы. Также по графикам можно сказать, что какая-либо зависимость между переменными выражена слабо.

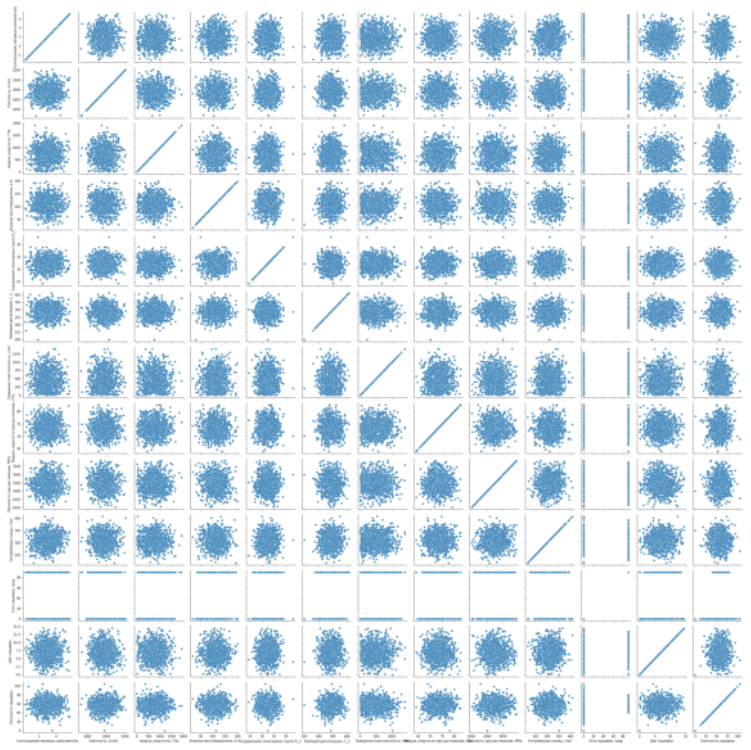


Рисунок 4 — Попарные диаграммы рассеяния точек

## 1.2 Описание используемых методов

Предсказание значений вещественной, непрерывной переменной — это задача регрессии. Эта зависимая переменная должна иметь связь с одной или несколькими независимыми переменными, называемых также предикторами или регрессорами. Регрессионный анализ помогает понять, как значение зависимой переменной изменяется при изменении независимых переменных. Для построения моделей использовались библиотеки машинного обучения для Python: ScikitLearn, Tensorflow и их встроенные модули.

## 1.2.1 Линейная регрессия

За базовую модель для прогнозирования всех искомых параметров принята линейная регрессия. LinearRegression соответствует линейной модели с коэффициентами w = (w1, …, wp), чтобы минимизировать остаточную сумму квадратов между наблюдаемыми целями в наборе данных и целями, предсказанными линейным приближением.

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

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

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

где n - число входных переменных.

Очевидно, что в данном случае модель будет описываться не прямой, а гиперплоскостью. Коэффициенты уравнения множественной линейной регрессии подбираются так, чтобы минимизировать сумму квадратов отклонения реальных точек данных от этой гиперплоскости, [8].

## 1.2.2 Регрессия k-ближайших соседей

Метод ближайших соседей (kNN - kNearestNeighbours) - метод решения задач классификации и [задач регрессии](https://learnmachinelearning.fandom.com/ru/wiki/%D0%A0%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D1%8F), основанный на поиске [ближайших](https://learnmachinelearning.fandom.com/ru/wiki/%D0%9C%D0%B5%D1%82%D1%80%D0%B8%D0%BA%D0%B8) объектов с известными значения [целевой переменной](https://learnmachinelearning.fandom.com/ru/wiki/%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81_%D1%83%D1%87%D0%B8%D1%82%D0%B5%D0%BB%D0%B5%D0%BC_(Supervised_learning)). Для целевой переменной метод предполагает найти ближайшие к нему объекты x1, x2…xk и построить прогноз по их меткам, то есть определить границы классов и выстроить гиперплоскость регресcии. Метка, назначенная целевой переменной, вычисляется на основе среднего значения меток ее ближайших соседей [5].

## 1.2.3 Случайный лес

Случайный лес (RandomForest) — представитель ансамблевых методов, комбинирующий множество отдельных деревьев решений. Формула итогового решателя — это усреднение предсказаний отдельных деревьев.

,

где

N – количество деревьев;

i – счетчик для деревьев;

b – решающее дерево;

x – сгенерированная нами на основе данных выборка.

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

Преимущества случайного леса:

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

Из недостатков можно отметить, что его построение занимает больше времени. Так же теряется интерпретируемость.

## 1.2.4 Многослойный перцептрон

Нейронная сеть — это последовательность нейронов, соединенных между собой связями. Нейронные сети применяются для решения задач регрессии, классификации, распознавания образов и речи, компьютерного зрения и других. На настоящий момент это самый мощный, гибкий и широко применяемый инструмент в машинном обучении. Персептроны пытаются имитировать функциональность человеческого мозга для решения задач. Вычислительная единица нейронной сети — нейрон или персептрон.

У каждого нейрона есть определённое количество входов, куда поступают сигналы, которые суммируются с учётом значимости (веса) каждого входа. Смещение – это дополнительный вход для нейрона, который всегда равен 1 и, следовательно, имеет собственный вес соединения. Так же у нейрона есть функция активации, которая определяет выходное значение нейрона. Она используется для того, чтобы ввести нелинейность в нейронную сеть. Примеры активационных функций: relu, сигмоида.

У полносвязной нейросети выход каждого нейрона подается на вход всем нейронам следующего слоя. У нейросети имеется:

* входной слой — его размер соответствует входным параметрам;
* скрытые слои — их количество и размерность определяем специалист;
* выходной слой — его размер соответствует выходным параметрам.

Прямое распространение – это процесс передачи входных значений в нейронную сеть и получения выходных данных, которые называются прогнозируемым значением. Прогнозируемое значение сравниваем с фактическим с помощью функции потери. В методе обратного распространения ошибки градиенты (производные значений ошибок) вычисляются по значениям весов в направлении, обратном прямому распространению сигналов. Значение градиента вычитают из значения веса, чтобы уменьшить значение ошибки. Таким образом происходит процесс обучения. Обновляются веса каждого соединения, чтобы функция потерь минимизировалась. Для обновления весов в модели используются различные оптимизаторы. Количество эпох показывает, сколько раз выполнялся проход для всех примеров обучения, [9,10].

## 1.2.5 Лассо регрессия

Регрессия по методу наименьших квадратов (МНК) часто может стать неустойчивой, то есть сильно зависящей от обучающих данных, что обычно является проявлением тенденции к переобучению. Избежать такого переобучения помогает регуляризация - общий метод, заключающийся в наложении дополнительных ограничений на искомые параметры, которые могут предотвратить излишнюю сложность модели. Смысл процедуры заключается в “лассо-стягивании” в ходе настройки вектора коэффициентов ββ таким образом, чтобы они в среднем оказались несколько меньше по абсолютной величине, чем это было бы при оптимизации по МНК. Метод регрессии Лассо (LASSO, Least Absolute Shrinkage and Selection Operator) заключается во введении дополнительного слагаемого регуляризации в функционал оптимизации модели, что часто позволяет получать более устойчивое решение.

## 1.2.6 Метрики качества моделей

Существует множество различных метрик качества, применимых для регрессии. В этой работе используются:

* R2 или коэффициент детерминации измеряет долю дисперсии, объясненную моделью, в общей дисперсии целевой переменной. Если он близок к единице, то модель хорошо объясняет данные, если же он близок к нулю, то качество прогноза идентично средней величине целевой переменной (т.е. очень низкое). Отрицательные значение коэффициента детерминации означают плохую объясняющую способность модели.
* MSE (Mean Squared Error) или средняя квадратичная ошибка принимает значениях в тех же единицах, что и целевая переменная. Чем ближе к нулю MSE, тем лучше работают предсказательные качества модели.

## 1.3. Разведочный анализ данных

Цель разведочного анализа данных — выявить закономерности в данных. Для корректной работы большинства моделей желательна сильная зависимость выходных переменных от входных и отсутствие зависимости между входными переменными. Для проведения разведочного анализа использованы библиотеки Numpy, Pandas, Matplotlib, Seaborn, Plotly.

Проверку пропусков выполняли методом df.isna().sum(), который показывает количество не нулевых значений. Для визуализации распределения значений по каждому столбцу и взаимосвязи между данными использован px.histogram, sns.boxplot, sns.scatterplot. Метод df.nunique() возвращает количество уникальных значений для каждого столбца.

Отображение корреляционной матрицы выполнено с помощью тепловой карты их значений px.imshow. Удаление выбросов производим по границе межквартильного размаха IQR, по схеме

1. рассчитаем первый и третий квартиль (Q1 и Q3);
2. оценим межквартирный диапазон, IQR-Q1;
3. оценим нижнюю границу, значения меньше 1.5\*IQR;
4. оценим верхнюю границу, значения больше 1.5\*IQR;
5. заменим точки данных, которые лежат за пределами нижней и верхней границы на nan;
6. удалим пропуски nan, если их количество намного меньше количества значений.

Функция dropna() удаляет строки, при axis=0, или столбцы, при axis=1, с значениями NULL или NAN [7].

# 2. Практическая часть

## 2.1. Предобработка данных

* + 1. Цель препроцессинга, или предварительной обработки данных — обеспечить корректную работу моделей.

1. Категориальный признак один - 'Угол нашивки, град'. Он принимает значения 0 и 90. Модели отработают лучше, если мы превратим эти значения в 0 и 1 с помощью LabelEncoder или OrdinalEncoder.

Удаляем выбросы за пределами 1.5 IQR. Значения параметров описательной статистики после удаления выбросов показаны в Таблице 3. От исходных 1023 строк осталось 936 (что вполне достаточно для обучения моделей), выбросы составили примерно 10% от объёма выборки.

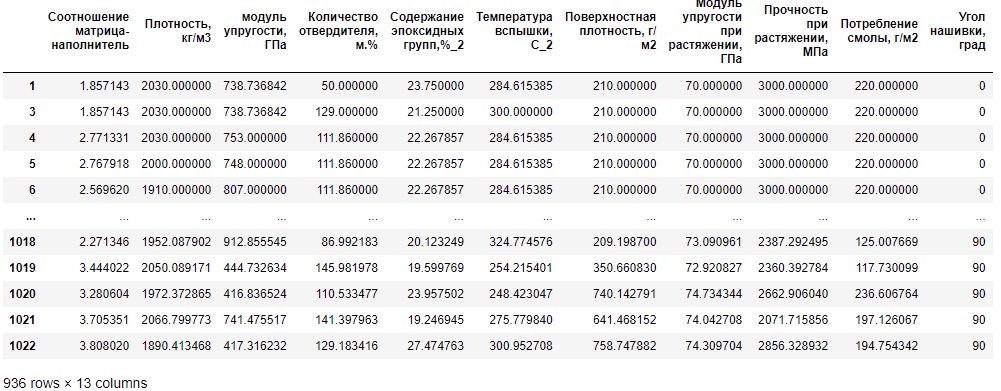


Рисунок 5 – Описательная статистика после удаления выбросов

Также проверим визуально удаление выбросов путем построения боксплотов.

Вещественных количественных признаков у нас большинство. Однако их значения лежат в разных диапазонах. Для сопоставимости данных выполним нормализацию – приведение в диапазон от 0 до 1 с помощью MinMaxScaler.

Построим корреляционную матрицу (Рисунок 6). Так как данные близки к нормальным, используем для расчёта коэффициента используем метод Пирсона. Согласно матрице корреляции коэффициенты корреляции близки к нулю, что означает отсутствие линейной зависимости между признаками. Можно предположить, что качество прогноза линейных моделей будет невысоким.

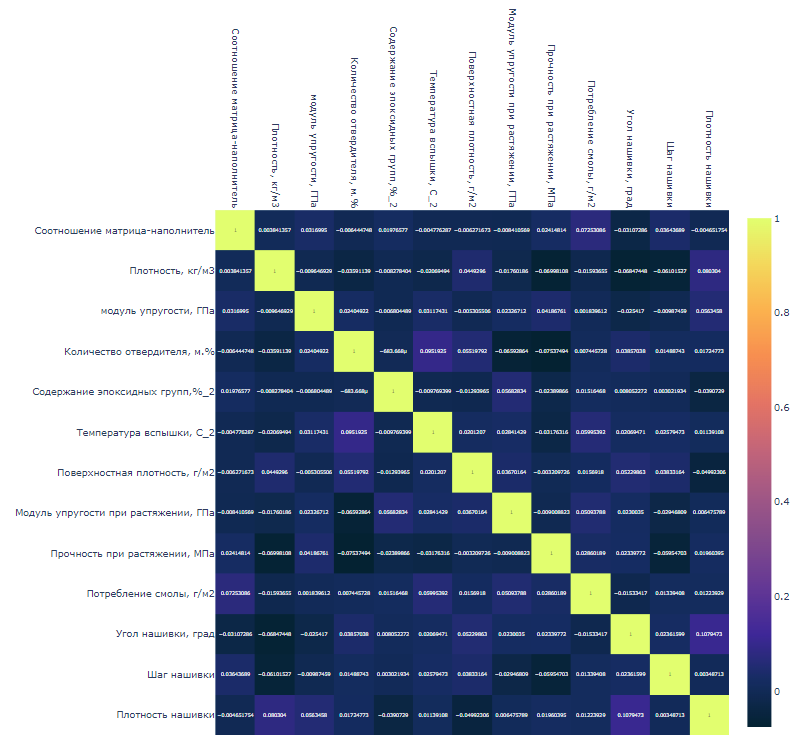


Рисунок 6 – Корреляционная матрица

## 2.2 Разработка и обучение регрессионных моделей

Для прогноза модуля упругости при растяжении и прочности при растяжении использованы модели LinearRegression, kNearestNeighboursRegression, RandomForestRegressor, MLPRegressor, LassoRegressor. Определяем целевые переменные-выходы (Модуль упругости при растяжении, Гпа и Прочность при растяжении, МПа) и входы модели (остальные переменные). Делим данные на тестовую и обучающую выборки методом train\_test\_split. 30% данных оставим на тестирование модели, на остальных происходит обучение моделей. Зерно генератора случайных чисел зададим постоянным для воспроизводимости результатов обучения.

Проверим правильность разбивки:

df\_normalized.shape[0] - x\_train\_1.shape[0] - x\_test\_1.shape[0] = 0.

Разбивка верна.

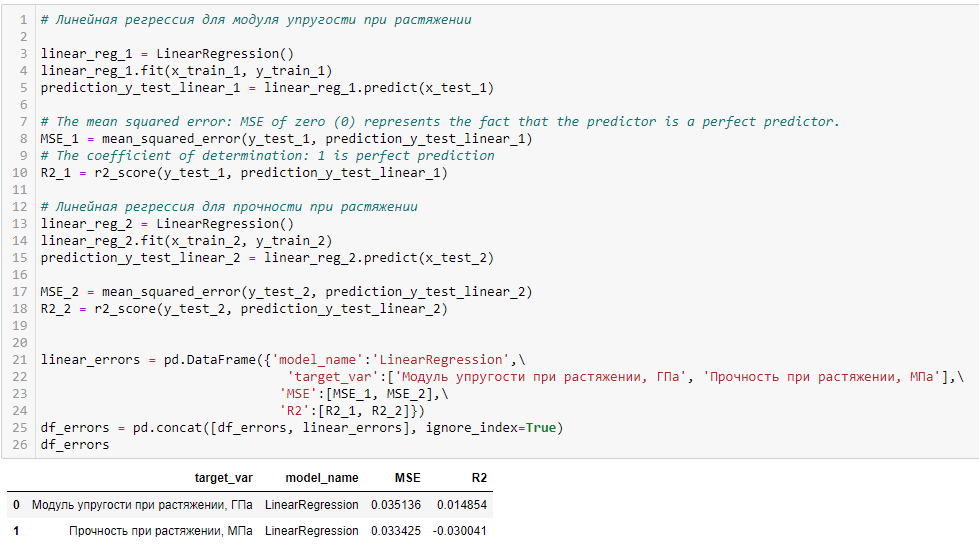
Общий алгоритм работы с моделями следующий:

1. Вызываем объект метода регрессии
2. Задаем возможные параметры оценки. Все модели будем прогонять по сетке GridSearchCV для получения наилучших параметров оценки целевых переменных.
3. Вызываем метод GridSearchCV, передавая в него сам объект метода регрессии и возможные параметры. Получаем лучшее возможное решение.
4. Считаем ошибки модели, записываем их в датасет ошибок.
5. Визуализируем результат работы модели.

## 2.3 Результаты работы моделей

Ниже представлены результаты работы моделей для двух целевых переменных в виде соотношения тест/прогноз, а также итоговый датасет ошибок.

## 2.3.1 Линейная регрессия



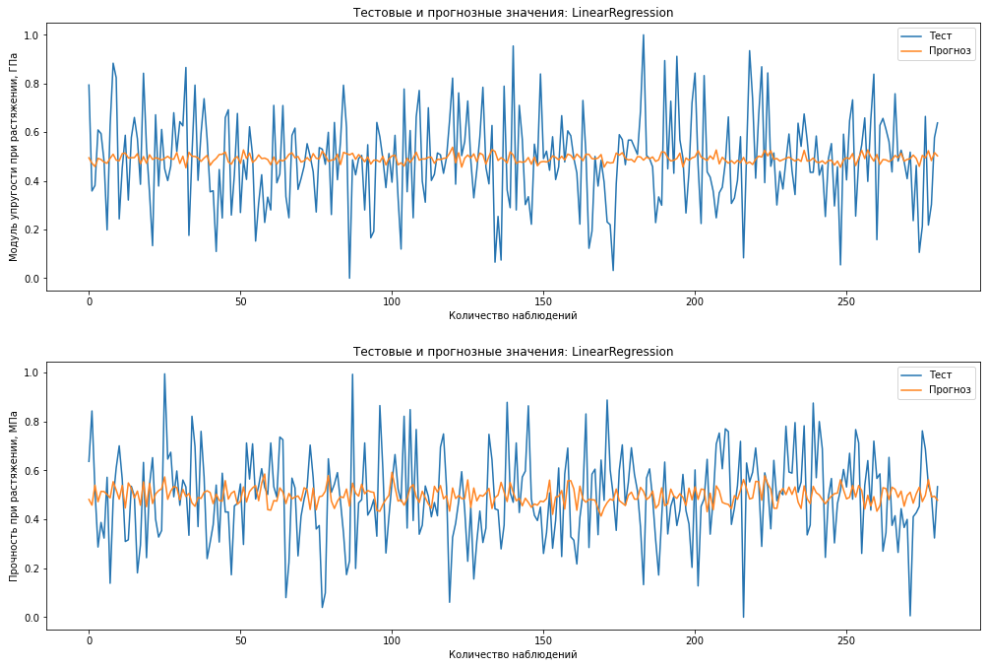


Рисунок 7 – Линейная регрессия

## 2.3.2 Регрессия k-ближайших соседей

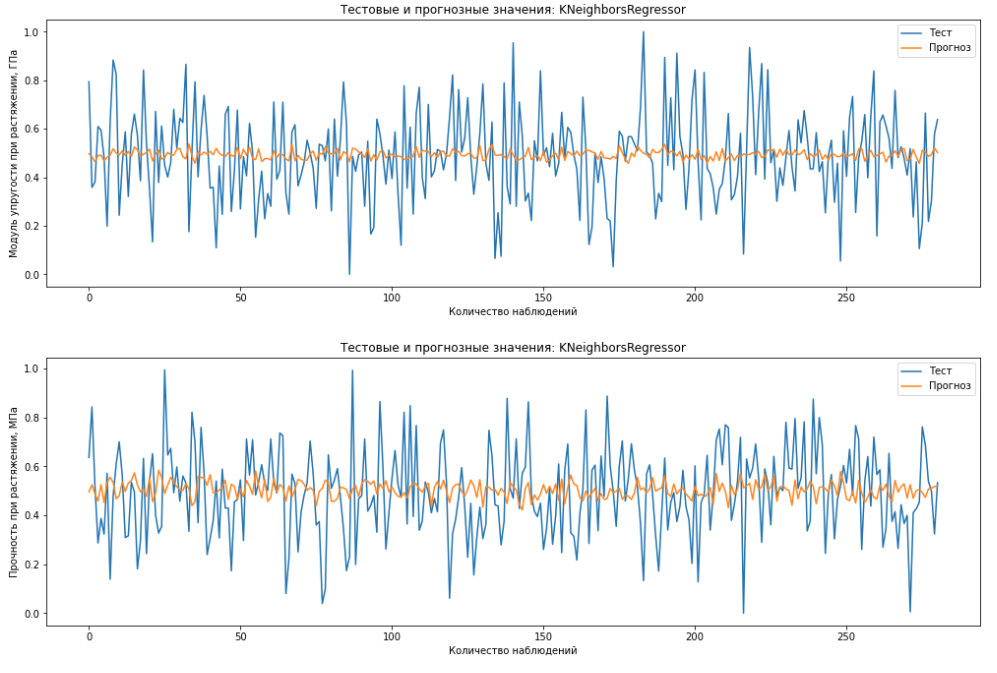
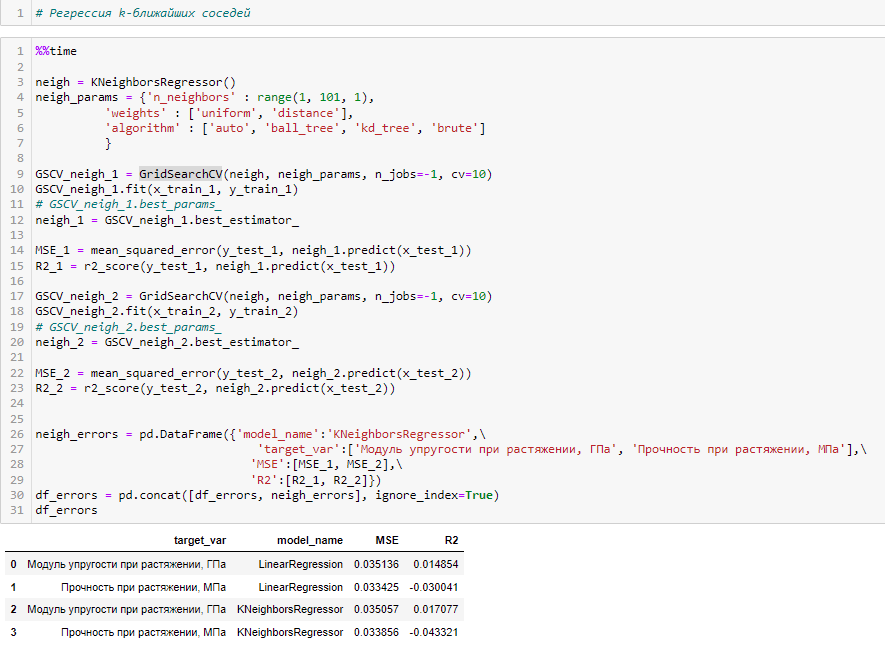


Рисунок 8 – Регрессия k-ближайших соседей

## 2.3.3 Случайный лес

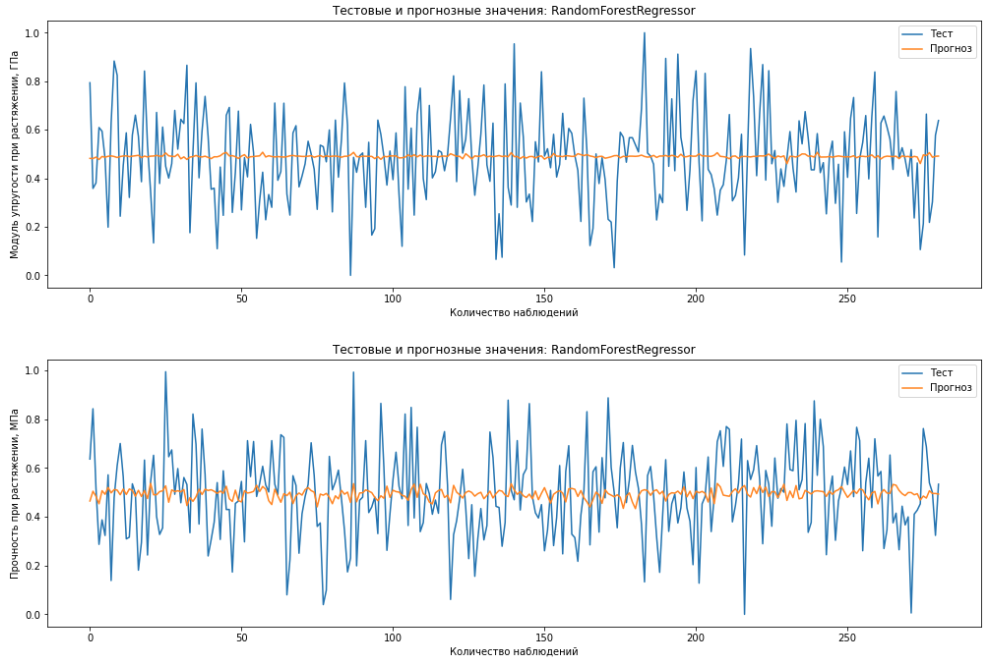
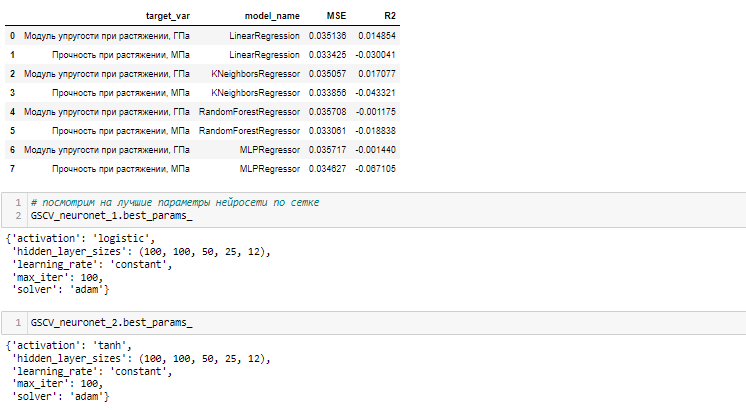


Рисунок 9 – RandomForest регрессия

## 2.3.4 Многослойный перцептрон



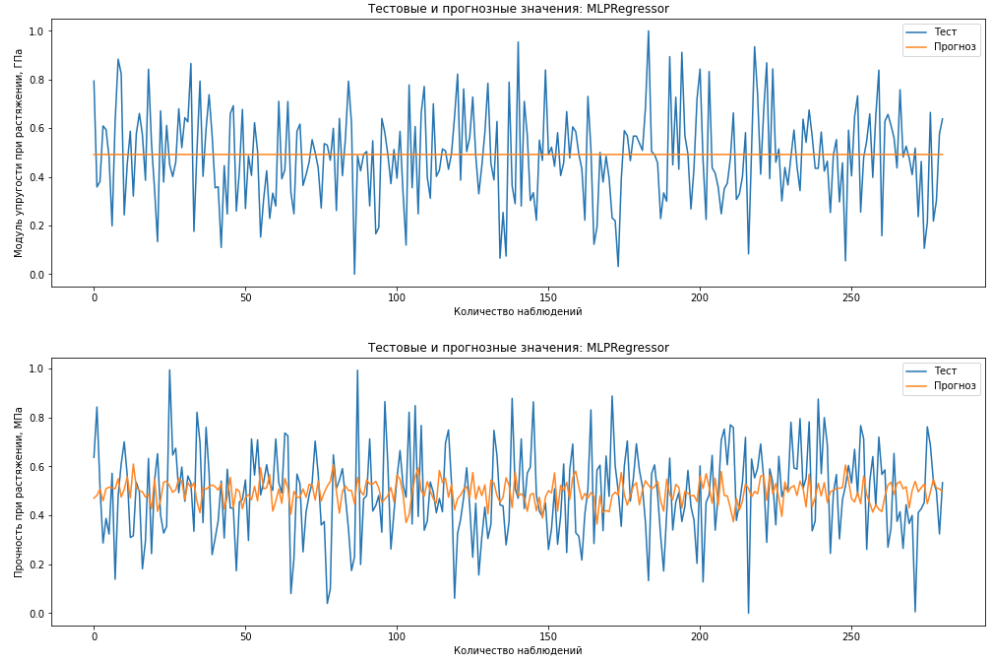


Рисунок 10 – Многослойный перцептрон

## 2.3.5 Лассо регрессия

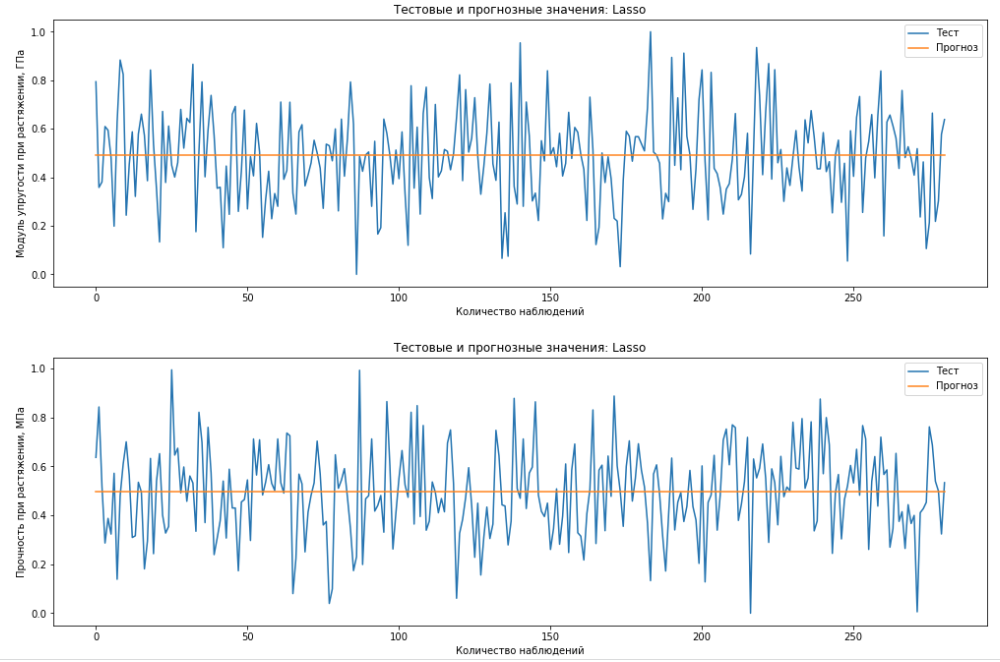
1. 

Рисунок 11 – Лассо регрессия

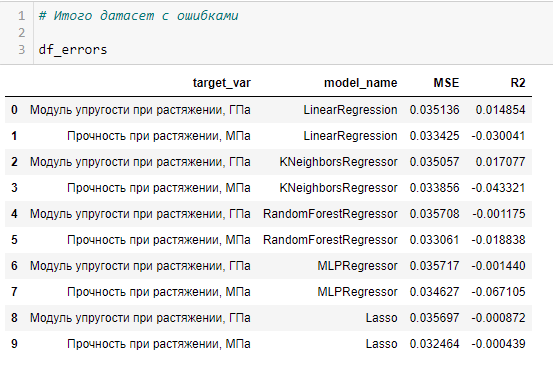


Рисунок 12 – Датасет ошибок

Все модели показали неудовлетворительный результат. Наилучшей моделью на этих данных для первой целевой переменной (Модуль упругости при растяжении, ГПа) является регрессия k-ближайших соседей, для второй целевой переменной (Прочность при растяжении, МПа) является Лассо-регрессия. Хотя в целом при таких результатах можно применять среднее значение переменной в качестве прогнозного.

## 2.4 Разработка нейронной сети для прогнозирования соотношения матрица-наполнитель

По заданию для соотношения матрица-наполнитель необходимо построить нейросеть модуля Keras библиотеки TensorFlow. Строю нейронную сеть с помощью класса keras.Sequential со следующими параметрами:

* входной слой нормализации 12 признаков;
* выходной слой для 1 признака;
* скрытых слоев: 5;
* нейронов в скрытом слое: 128, 128, 128, 64, 32, 16;
* активационная функция скрытых слоев: relu;
* оптимизатор: Adam;
* loss-функция: MeanSquaredError.

Архитектура нейросети приведена на рисунке 13.

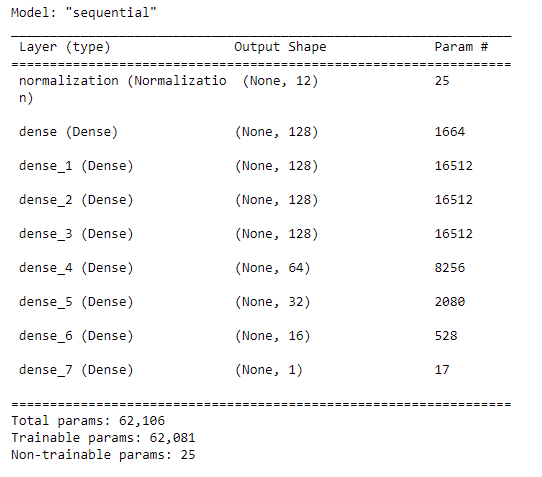


Рисунок 13 — Архитектура нейросети

Параметры нейросети следующие:

* пропорция разбиения данных на тестовые и валидационные: 30%;
* количество эпох: 100.

График потерь модели (MSE) а также сравнение тест/прогноз приведёны ниже на рисунке 19.

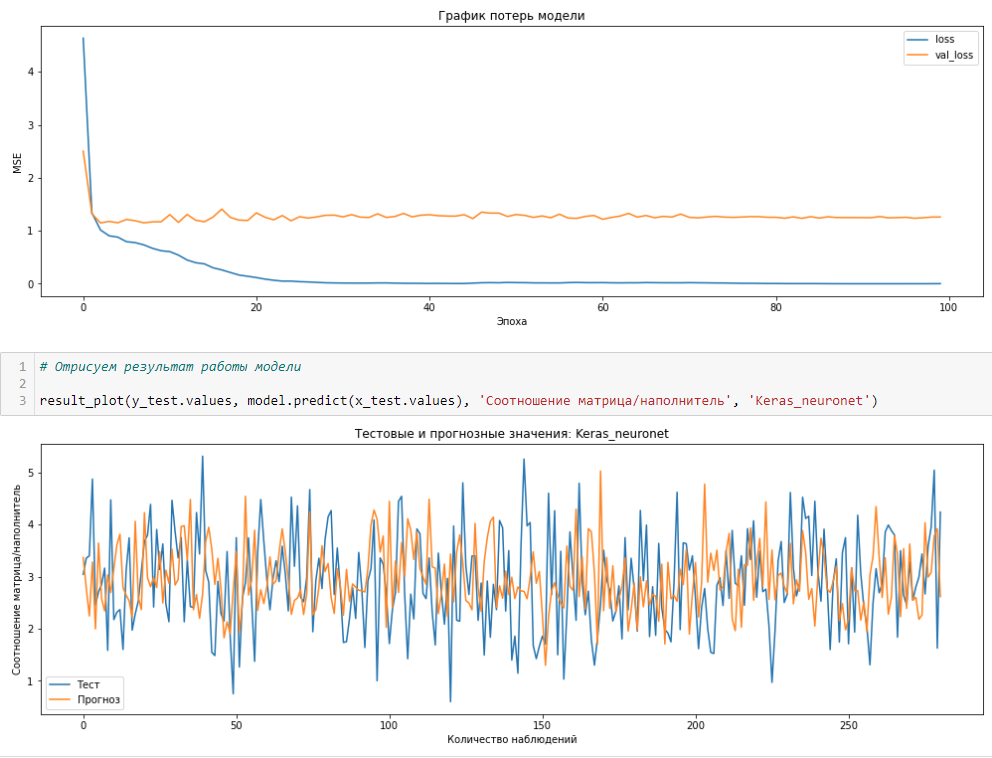


Рисунок 14 – Результаты работы модели и график потерь

По графику делаем вывод что для обучения спроектированной нейронной сети достаточно 30 эпох. Ошибки модели следующие: MSE=1.251727, R^2 = -0.488577. Результаты неудовлетворительны.

## 2.5. Разработка приложения

Несмотря на то, что пригодных к внедрению моделей получить не удалось, можно разработать тестовое веб-приложение на фреймворке Flask. Сохраняем модель keras в качестве логической составляющей приложения.

В приложении необходимо реализовать следующие функции:

* загрузка сохраненной модели;
* ввод входных параметров;
* получение и отображение прогноза выходных параметров.

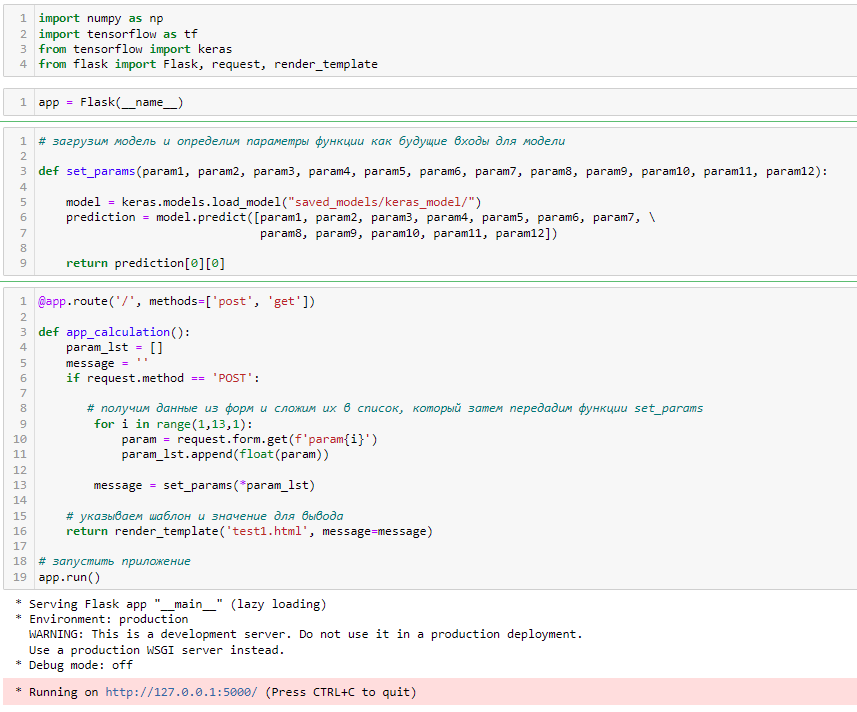
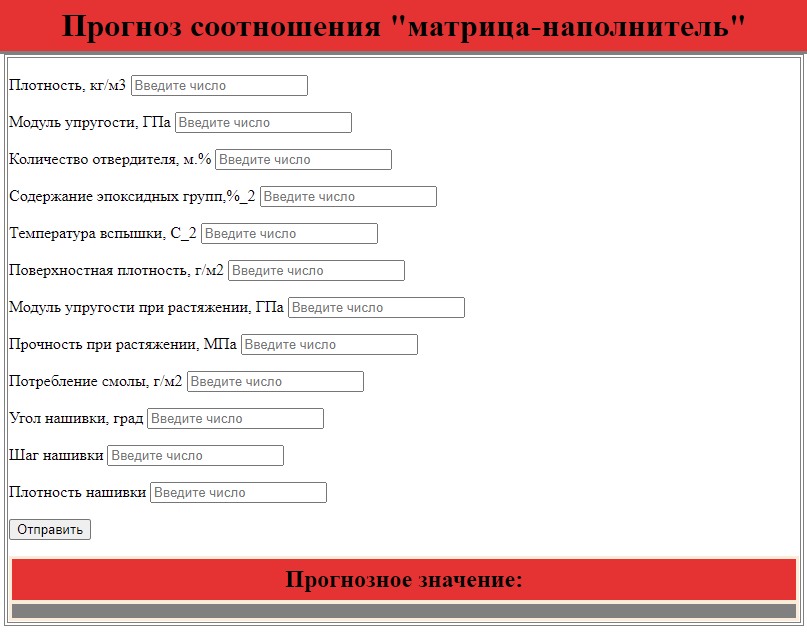
Ввод входных параметров пользователем осуществляется через формы на веб странице.

Рисунок 15 – Написание приложения

Ниже приведены скрины приложения, доступные пользователю:



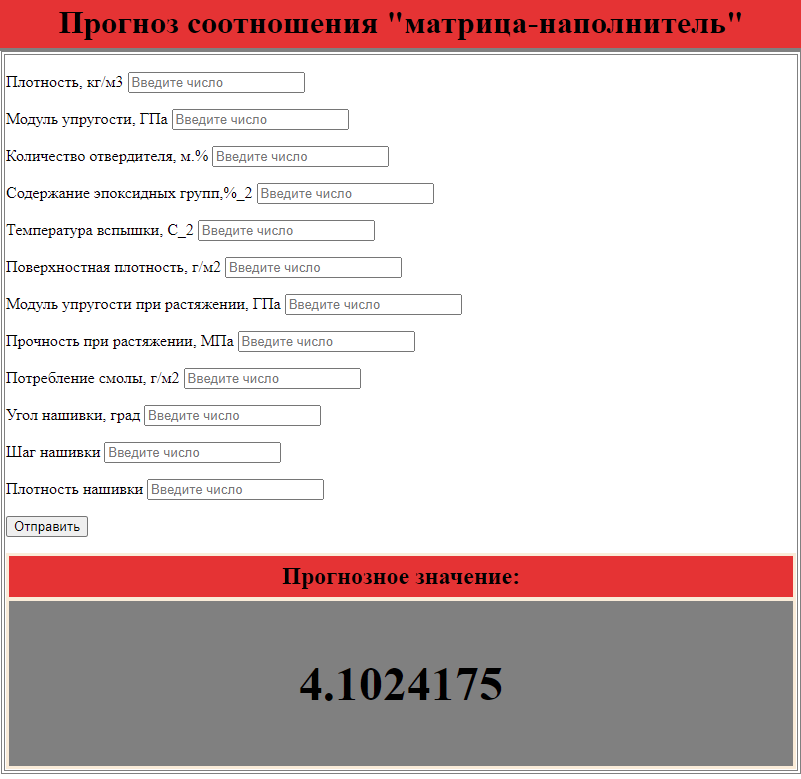


Рисунок 16 – Юзер-сторона веб-приложения

## 2.6. Создание удаленного репозитория

Для данного исследования был создан удаленный репозиторий на GitHub, который находится по адресу:

https://github.com/antonyalta/VKR\_Baumana

В него загружены результаты работы: код анализа данных, обучения моделей, код веб-приложения, пояснительная записка и презентация.

# Заключение

На основании проведенного исследования можно сделать следующие основные выводы по теме:

* распределение полученных данных близко к нормальному;
* коэффициенты корреляции между парами признаков стремятся к нулю;
* Примененные модели регрессии не показали высокой эффективности в прогнозировании свойств композитов, необходимы дополнительные вводные данные, получение новых результирующих признаков в результате математических преобразований, релевантных доменной области, консультации экспертов предметной области, новые исследования.
* лучшие метрики для первой целевой переменной (Модуль упругости при растяжении, ГПа) является регрессия k-ближайших соседей, для второй целевой переменной (Прочность при растяжении, МПа) является Лассо-регрессия.

# Список использованных источников

1. Силен Дэви, Мейсман Арно, Али Мохамед. Основы Data Science и Big Data. Python и наука о данных. – СПб.: Питер, 2017. – 336 с.: ил.
2. ГрасД. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербурr, 2021. - 416 с.: ил.
3. Гафаров, Ф.М., Галимянов А.Ф. Искусственные нейронные сети и приложения: учеб. пособие /Ф.М. Гафаров, А.Ф. Галимянов. – Казань: Издательство Казанского университета, 2018. – 121 с.
4. Библиотека scikit-learn [Электронный ресурс]: – Режим доступа: https://scikit-learn.org (дата обращения: 21.04.2022).
5. Библиотека Keras [Электронный ресурс]: – Режим доступа: https://keras.io/ (дата обращения: 21.04.2022).
6. Библиотека Tensorflow [Электронный ресурс]: – Режим доступа: https://www.tensorflow.org/overview (дата обращения: 21.04.2022).
7. Документация Pandas [Электронный ресурс]: – Режим доступа: https://pandas.pydata.org/docs/user\_guide/index.html#user-guide (дата обращения: 21.04.2022).
8. Документация Matplotlib [Электронный ресурс]: – Режим доступа: https://matplotlib.org/ (дата обращения: 21.04.2022).
9. Документация Seaborn [Электронный ресурс]: – Режим доступа: https://seaborn.pydata.org/ (дата обращения: 21.04.2022).
10. Документация Flask [Электронный ресурс]: – Режим доступа: https://pypi.org/project/Flask/ (дата обращения: 21.04.2022).
11. Документация Plotly [Электронный ресурс]: – Режим доступа: https://pypi.org/project/Plotly/ (дата обращения: 21.04.2022).