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 Залуцкая Виктория Олеговна

(ФИО)

Москва, 2022

# **Оглавление**

[**Оглавление** 2](#_Toc106190483)

[**Введение** 3](#_Toc106190484)

[**1. Анализ исходных данных и выбор методов решения** 5](#_Toc106190485)

[**1.1 Постановка задачи** 5](#_Toc106190486)

[**1.2 Разведочный анализ данных** 6](#_Toc106190487)

[**1.3 Визуализация данных** 8](#_Toc106190488)

[**2. Практическая часть** 10](#_Toc106190489)

[**2.1 Предобработка данных** 10](#_Toc106190490)

[**2.2 Разработка и обучение модели** 11](#_Toc106190491)

[**2.3 Описание используемых моделей** 13](#_Toc106190492)

[**2.4 Тестирование моделей** 18](#_Toc106190493)

[**2.5 Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель** 23](#_Toc106190494)

[**2.6 Разработка приложения** 27](#_Toc106190495)

[**2.7 Создание удаленного репозитория и загрузка результатов работы на него** 28](#_Toc106190496)

[**Заключение** 29](#_Toc106190497)

[**Список литературы** 30](#_Toc106190498)

# **Введение**

Развитие современной техники требует новых конструкционных материалов, превосходящих по своим прочностным, упругим и другим свойствам традиционные. Композиционный материал – это искусственно созданный неоднородный сплошной материал, состоящий из двух и более композитов с четкой границей, раздела между ними. В большинстве композитов (за исключением слоистых) компоненты можно разделить на матрицу и включённые в нее элементы.

Машинное обучение воспринимается как многообещающий инструмент для разработки и открытия новых материалов для широкого спектра приложений. Разработка новых материалов с превосходными индивидуальными свойствами является конечной целью современных инженерных приложений. За последние несколько десятилетий, благодаря быстрому развитию высокопроизводительных параллельных вычислений, материаловедения и численного моделирования, многие важные свойства материалов теперь могут быть рассчитаны с помощью моделирования с достаточной точностью. По сравнению с простым прогнозированием свойств известных материалов разработка новых материалов для достижения регулируемых свойств является более важной задачей для научных и инженерных целей. Эффективность и работоспособность материала зависят от правильного выбора исходных компонентов и технологии их совмещения, призванной обеспечить прочную связь между компонентами при сохранении их первоначальных характеристик. 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).

В последние годы искусственный интеллект развивается очень быстрыми темпами, даже сама технология нейронных сетей стала более доступной обычным пользователям. Также появилось множество технологий, использующих нейронные сети для ускорения или упрощения процессов. Машинное обучение являет собой самый простой вариант искусственного интеллекта. Оно предполагает, что с помощью различных методов на основе большого количества «тренировочных» данных можно классифицировать или  
предсказать любой объект, явление или событие.

# **1. Анализ исходных данных и выбор методов решения**

## **1.1 Постановка задачи**

Предметом выпускной квалификационной работы являются построение моделей для прогнозирования таких характеристик композиционных материалов, как модуль упругости при растяжении, прочность при растяжении и создание нейронной сети для рекомендации соотношения матрица-наполнитель.

Для решения поставленной задачи потребуется:

1) описать методы, которые используются для решений;

2) провести разведочный анализ предложенных датасетов;

а) построить гистограммы распределения каждой из переменных;

б) построить диаграммы ящиков с усами;

в) построить попарные графики рассеяния точек;

г) получить среднее и медианное значения;

д) исключить выбросы, проверить отсутствие пропусков;

3) провести предобработку данных: удаление шумов, нормализацию;

4) обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении;

5) написать нейронную сеть, предназначенную для рекомендаций соотношения матрица-наполнитель;

6) оценить точность модели на тренировочном и тестовом датасете;

7) разработать приложение с графическим интерфейсом, которое будет выдавать прогноз;

8) создать репозиторий в GitHub и разместить там код исследования.

## **1.2 Разведочный анализ данных**

Для проведения разведочного анализа используется среда разработки Jupyter Notebook язык программирования Python и библиотеки Numpy, Pandas, Matplotlib, Seaborn и Sklearn.

Исходные данные о свойствах композиционных материалов получены структурным подразделением МГТУ им. Н.Э. Баумана – Центр НТИ «Цифровое материаловедение: новые материалы и вещества» и основан на реальных производственных задачах.

Данные состоит из двух файлов первый из которых – это файл X\_bp.xlsx с данными о параметрах базальтопластика, а второй файл X\_nup.xlsx с данными о нашивках из углепластика.

Потребовалось объединить файлы X\_bp.xlsx и X\_nup.xlsx по индексу с типом объединения INNER.

Количество строк в файле X\_bp.xlsx было 1023, столбцов 11. А количество строк в файле X\_nup.xlsx – 1040, столбцов 4.

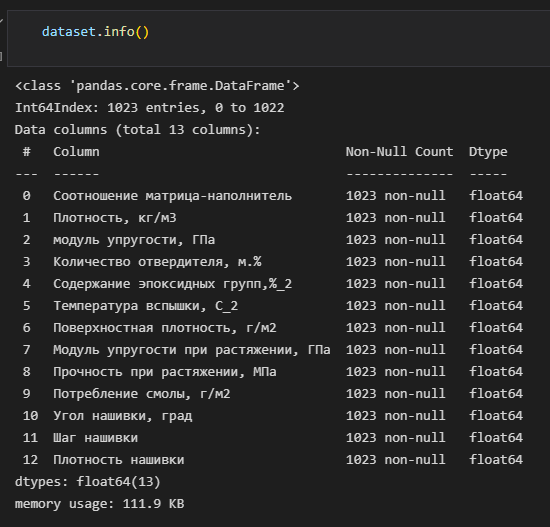
Полученные данные являются отправной точкой дальнейших исследований, требуют предварительный обработки.

Для получения представления о характере распределения исходных данных, формирования оценки качества исходных данных (наличия пропусков, выбросов) (Рисунок 1), выявления характера взаимосвязи между переменными с целью последующего выдвижения гипотез о наиболее подходящих для решения задачи моделях машинного обучения проведем разведочный анализ данных.

В качестве инструментов разведочного анализа используется оценка статистических характеристик данных (Рисунок 2), а также попарный график рассеяния (Рисунок 3), тепловая карта корреляции (Рисунок 4), гистограммы нормального распределения, поиск выбросов через ящик с усами.

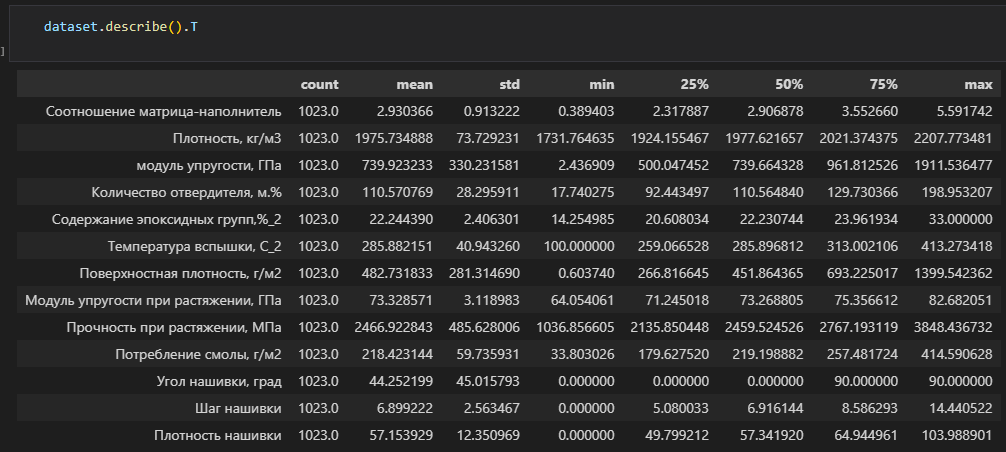
Проверку пропусков выполняли с помощью методов: dataset.info(), который показывает количество ненулевых значений и тип данных.

Рисунок 1 – Структура исходных данных



Оценка статистических характеристик данных с помощью метода dataset.describe().T

Рисунок 2 – Оценка статистических характеристик данных



## **1.3 Визуализация данных**

Для визуализации данных необходимо отобразить попарные графики рассеяния точек (Рисунок 3), гистограммы распределения каждой из переменной и ящика с усами (Рисунок 4), корреляционную тепловую карту Пирсона (Рисунок 5).

Рисунок 3 – Попарные графики рассеяния точек

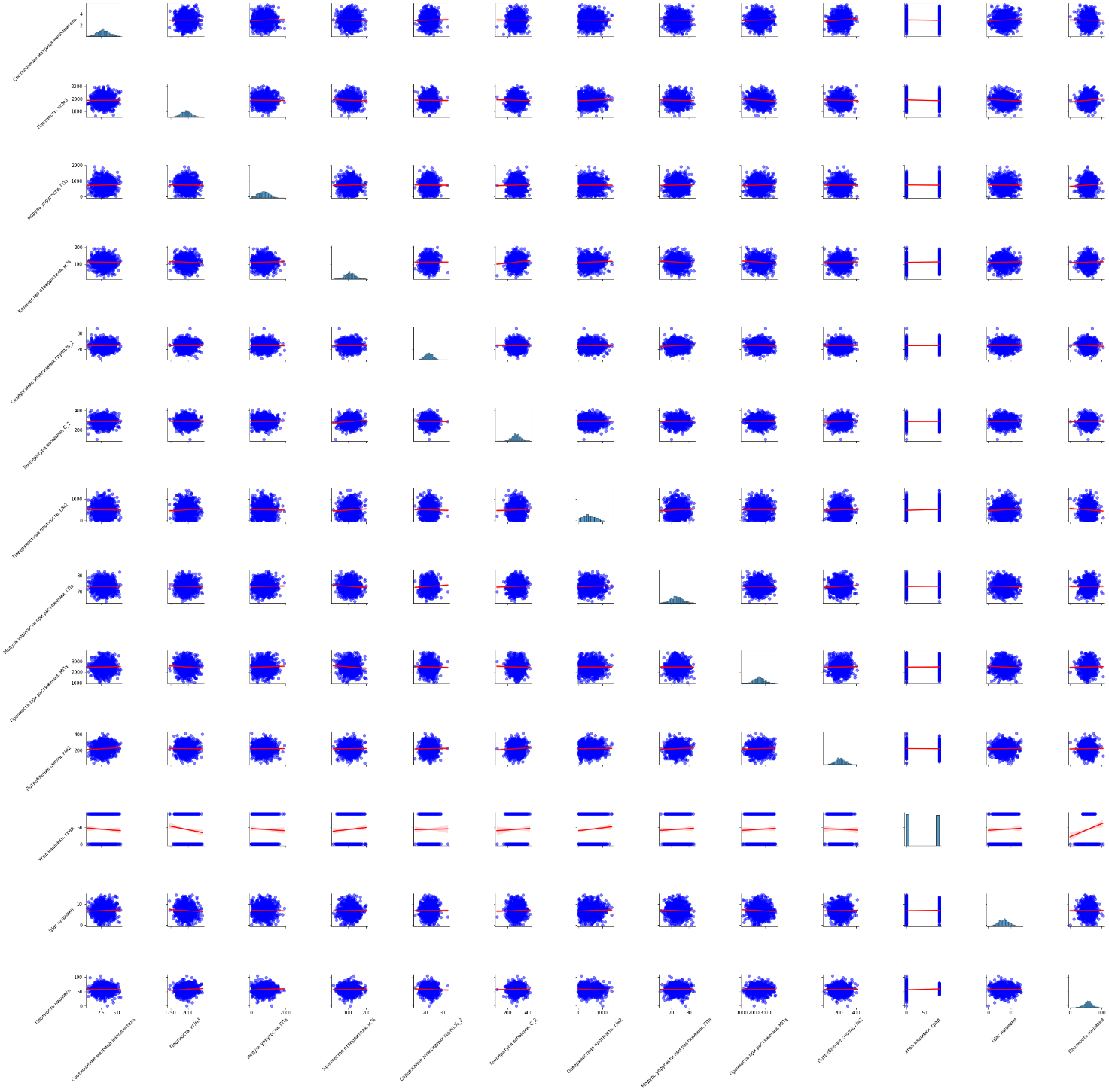
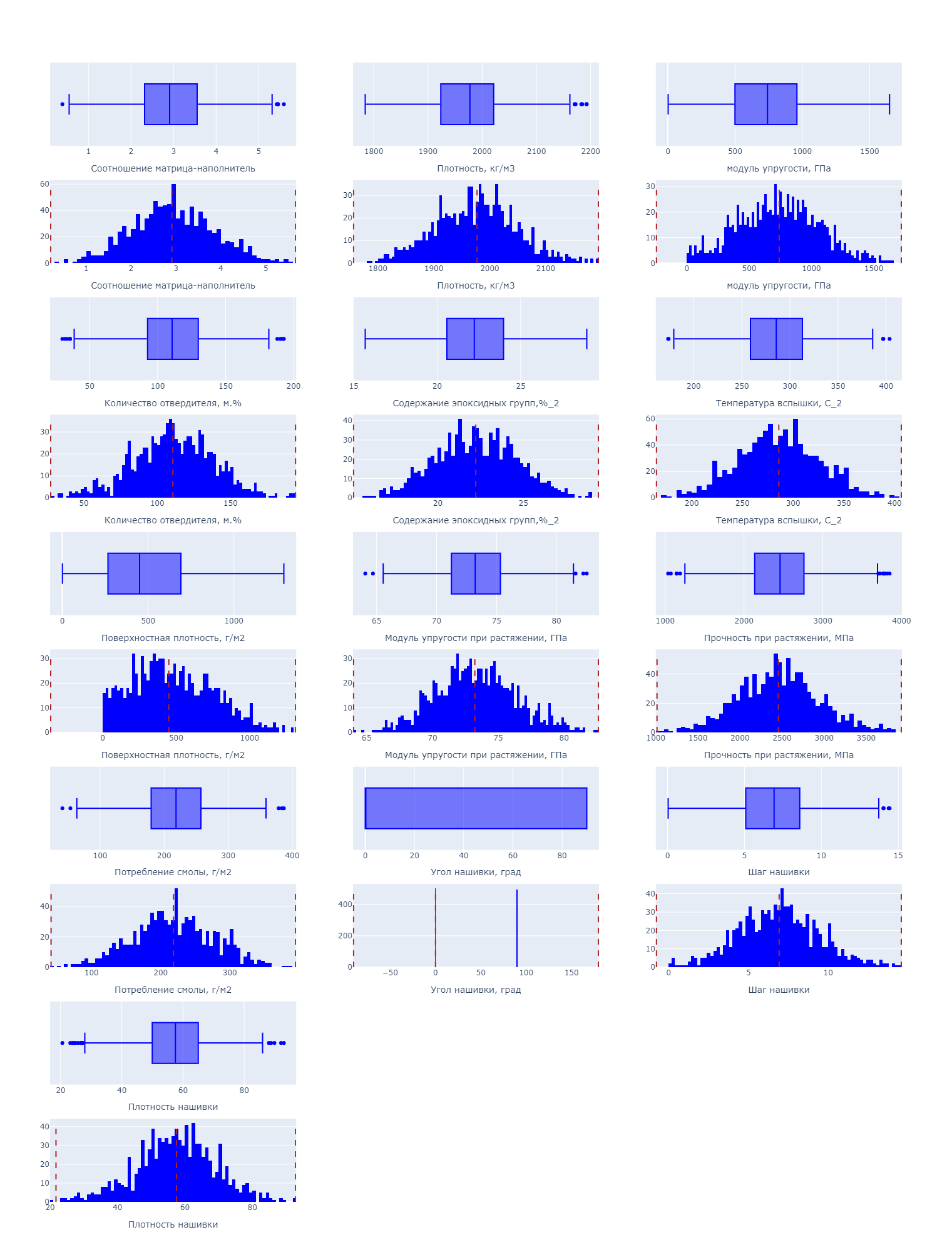


Рисунок 4 - Гистограммы распределения каждой из переменной и ящики с усами



По форме разброса точек попарного графика рассеяния видно, что зависимости между переменными, на которых будет основываться работа модели, не обнаруживаются.

Гистограммы распределения переменных показали, что все признаки, кроме «Угол нашивки», имеют нормальное распределение и принимают неотрицательные значения. «Угол нашивки» принимает два значения: 0, 90 градусов.

Рисунок 5 – Корреляционная тепловая карта Пирсона



Из корреляционной тепловой карты Пирсона (Рисунок 5) видно, что все коэффициенты корреляции близки к нулю. Это означает отсутствие линейной зависимости между признаками.

При построении графика boxplot (ящик с усами) выявлено 25 выбросов методом 3-х сигм и 94 выброса методом межквартильных расстояний.

Для удаления выбросов используется метод 3х сигм так как при использовании этого метода исключается меньше полезной информации.

# **2. Практическая часть**

## **2.1 Предобработка данных**

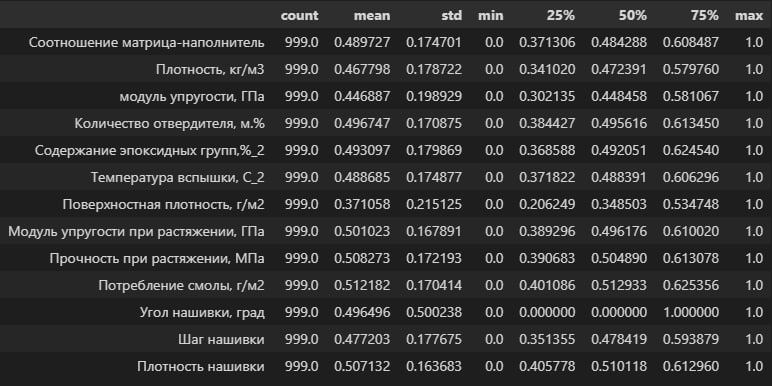
После удаления выбросов, проведя анализ данных видно, что значения находятся в разных диапазонах.

Для обучения моделей для прогноза модуля упругости при растяжении и прочности при растяжении необходимо данные нормализовать так как многие алгоритмы машинного обучения работают лучше, когда числовые входные переменные масштабируются до стандартного диапазона.

Для масштабирования данных используется метод MinMaxScaler который масштабирует все функции данных в диапазоне [0, 1].

Оценка статистических характеристик данных выпалена с помощью метода dataset.describe().T после проведения нормализации данных при помощи MinMaxScaler() (Рисунок 6).

Рисунок 6 – Оценка статистических характеристик данных с помощью метода после проведения нормализации данных



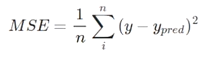
## **2.2 Разработка и обучение модели**

Для прогноза модуля упругости при растяжении и прочности при растяжении использованы следующие модели: LinearRegression, SGDRegressor, Ridge Regression, LassoRegressor, BayesianRidge, DecisionTreeRegressor, LinerSVR, SVR, KNeighborsRegressor, AdaBoostRegressor, BaggingRegressor, ExtraTreesRegressor, GradientBoostingRegressor, RandomForestRegressor.

Методом train\_test\_split данные разделены на тестовую и обучающую выборки: 30% данных оставлены на тестирование модели, на остальных данных происходит обучение моделей.

Для поиска наиболее подходящих критериев модели создана функция GridSearchCVCouner которая с помощью GridSearchCV выявляет наиболее подходящих критериев переданной модели. А также создана функция ErrorCouner которая считает метрики MAE, MSE, R2.

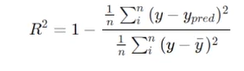
Средняя квадратичная ошибка (Mean Squared Error, MSE). MSE применяется в ситуациях, когда нам надо подчеркнуть большие ошибки и выбрать модель, которая дает меньше больших ошибок прогноза (1). Грубые ошибки становятся заметнее за счет того, что ошибку прогноза мы возводим в квадрат. И модель, которая дает нам меньшее значение среднеквадратической ошибки, можно сказать, что что у этой модели меньше грубых ошибок.

(1)

Средняя абсолютная ошибка (Mean Absolute Error, MAE). Среднеквадратичный функционал сильнее штрафует за большие отклонения по сравнению со среднеабсолютным, и поэтому более чувствителен к выбросам (2).

(2)

Коэффициент детерминациию. Коэффициент детерминации измеряет долю дисперсии, объясненную моделью, в общей дисперсии целевой переменной. Фактически, данная мера качества — это нормированная среднеквадратичная ошибка. Если она близка к единице, то модель хорошо объясняет данные, если же она близка к нулю значит связь между переменными регрессионной модели отсутствует (3).

(3)

## **2.3 Описание используемых моделей**

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.

Линейная регрессия — первый тщательно изученный метод регрессионного анализа. Его главное достоинство — простота. Такую модель можно построить и рассчитать даже без мощных вычислительных средств. Простота является и главным недостатком этого метода. Тем не менее, именно с линейной регрессии целесообразно начать подбор подходящей модели.

LassoRegressor - (LASSO, Least Absolute Shrinkage and Selection Operator) — это вариация линейной регрессии, специально адаптированная для данных, которые демонстрируют сильную мультиколлинеарность (то есть сильную корреляцию признаков друг с другом). Она автоматизирует части выбора модели, такие как выбор переменных или исключение параметров. LASSO использует сжатие коэффициентов (shrinkage), то есть процесс, в котором значения данных приближаются к центральной точке (например, среднему значению).

Ridge Regression (гребневая регрессия) очень похожа на регрессию LASSO в том, что она применяет сжатие. Оба алгоритма хорошо подходят для наборов данных с большим количеством признаков, которые не являются независимыми друг от друга (коллинеарность). Однако самое большое различие между ними в том, что гребневая регрессия использует регуляризацию L2, то есть ни один из коэффициентов не становится нулевым, как это происходит в регрессии LASSO. Вместо этого коэффициенты всё больше приближаются к нулю, но не имеют большого стимула достичь его из-за природы регуляризации L2.

SGDRegressor - линейная модель подобрана путем минимизации регуляризованных эмпирических потерь с помощью SGD. SGD означает стохастический градиентный спуск: градиент потерь оценивается для каждой выборки за раз, и модель обновляется по пути с уменьшением графика силы (он же скорость обучения). Регуляризатор — это штраф, добавленный к функции потерь, который сжимает параметры модели до нулевого вектора, используя либо квадрат евклидовой нормы L2, либо абсолютную норму L1, либо их комбинацию (эластичная сеть). Если обновление параметра пересекает значение 0.0 из-за регуляризатора, обновление усекается до 0.0, чтобы можно было изучить разреженные модели и добиться выбора онлайн-функций. Эта реализация работает с данными, представленными в виде плотных массивов значений с плавающей запятой для функций.

BayesianRidge - регрессия байесовского гребня. Байесовская регрессия позволяет естественному механизму выжить при недостатке данных или плохо распределенных данных, формулируя линейную регрессию с использованием распределителей вероятностей, а не точечных оценок. Предполагается, что выход или ответ получен из распределения вероятностей, а не оценивается как одно значение. Математически для получения полностью вероятностной модели предполагается, что отклик y имеет распределение по Гауссу.

KNeighborsRegressor - метод решения задач регрессии, основанный на поиске ближайших объектов с известными значения целевой переменной. 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.

DecisionTreeRegressor - дерево решений приходит к оценке, задавая ряд вопросов к данным, каждый вопрос сужает наши возможные значения до тех пор, пока модель не станет достаточно уверенной, чтобы сделать один прогноз. Порядок вопросов, а также их содержание определяются моделью. Кроме того, все вопросы заданы в форме Верно/Неверно. Для каждого ответа «Верно» и «Неверно» есть отдельные ветки. Независимо от ответов на вопросы, мы в конечном итоге получаем предсказание (листовой узел). Начало с корневого узла вверху и продвигайтесь по дереву, отвечая на вопросы по пути.

LSVR - Linear Support Vector Regression это один из самых популярных алгоритмов обучения с учителем, который используется как для задач классификации, так и для задач регрессии. Цель алгоритма SVM — создать наилучшую линию или границу решения, которая может разделить n-мерное пространство на классы, чтобы мы могли легко поместить новую точку данных в правильную категорию в будущем. Эта граница наилучшего решения называется гиперплоскостью.

SVM выбирает крайние точки/векторы, которые помогают в создании гиперплоскости. Эти крайние случаи называются опорными векторами. Учитывая набор обучающих примеров, каждый из которых помечен как принадлежащий к одной из двух категорий, обучающий алгоритм SVM строит модель, которая относит новые примеры к той или иной категории, превращая его в невероятностный двоичный линейный классификатор.

AdaBoostRegressor – это метаоценщик, относящийся к ансамблевым алгоритмам, который начинает с подгонки регрессора к исходному набору данных, а затем подбирает дополнительные копии регрессора к тому же набору данных, но где веса экземпляров корректируются в соответствии с ошибкой текущего прогноза. Таким образом, последующие регрессоры больше фокусируются на сложных случаях.

BaggingRegressor — это ансамблевая метаоценка, которая подбирает базовые регрессоры для каждого из случайных подмножеств исходного набора данных, а затем объединяет их индивидуальные прогнозы (путем голосования или усреднения) для формирования окончательного прогноза. Такая метаоценка обычно может использоваться как способ уменьшить дисперсию оценки black-box (например, дерева решений) путем введения рандомизации в процедуру ее построения и последующего создания из нее ансамбля.

ExtraTreesRegressor - этот класс реализует метаоценку, которая соответствует ряду рандомизированных деревьев решений (также называемых дополнительными деревьями) для различных подвыборок набора данных и использует усреднение для повышения точности прогнозирования и контроля над подбором.

GradientBoostingRegressor строит аддитивную модель поэтапно вперед, он позволяет оптимизировать произвольные дифференцируемые функции потерь. На каждом этапе дерево регрессии аппроксимируется отрицательным градиентом заданной функции потерь. Это способ объединения нескольких простых моделей в единую составную модель. Простые модели (также известные как слабые ученики) добавляются по одной, сохраняя при этом существующие деревья в модели неизменными. По мере того, как мы комбинируем все больше и больше простых моделей, полная окончательная модель становится более сильным предиктором.

RandomForestRegressor — это метаоценка, которая соответствует ряду классифицирующих деревьев решений для различных подвыборок набора данных и использует усреднение для повышения точности прогнозирования и контроля переобучения. Представитель ансамблевых методов.

Если точность дерева решений оказалось недостаточной, мы можем множество моделей собрать в коллектив. Формула итогового решателя (6) — это усреднение предсказаний отдельных деревьев.

 (6)

где

N – количество деревьев;

i – счетчик для деревьев;

b – решающее дерево;

x – сгенерированная нами на основе данных выборка.

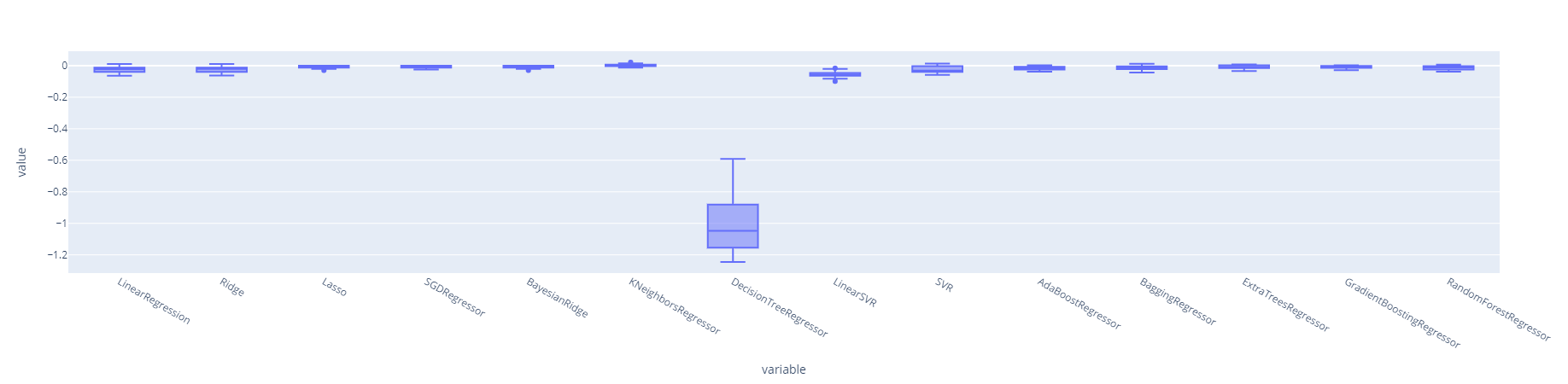
Для определения входных данных каждому дереву используется метод случайных подпространств. Базовые алгоритмы обучаются на различных подмножествах признаков, которые выделяются случайным образом.

Преимущества: 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.

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

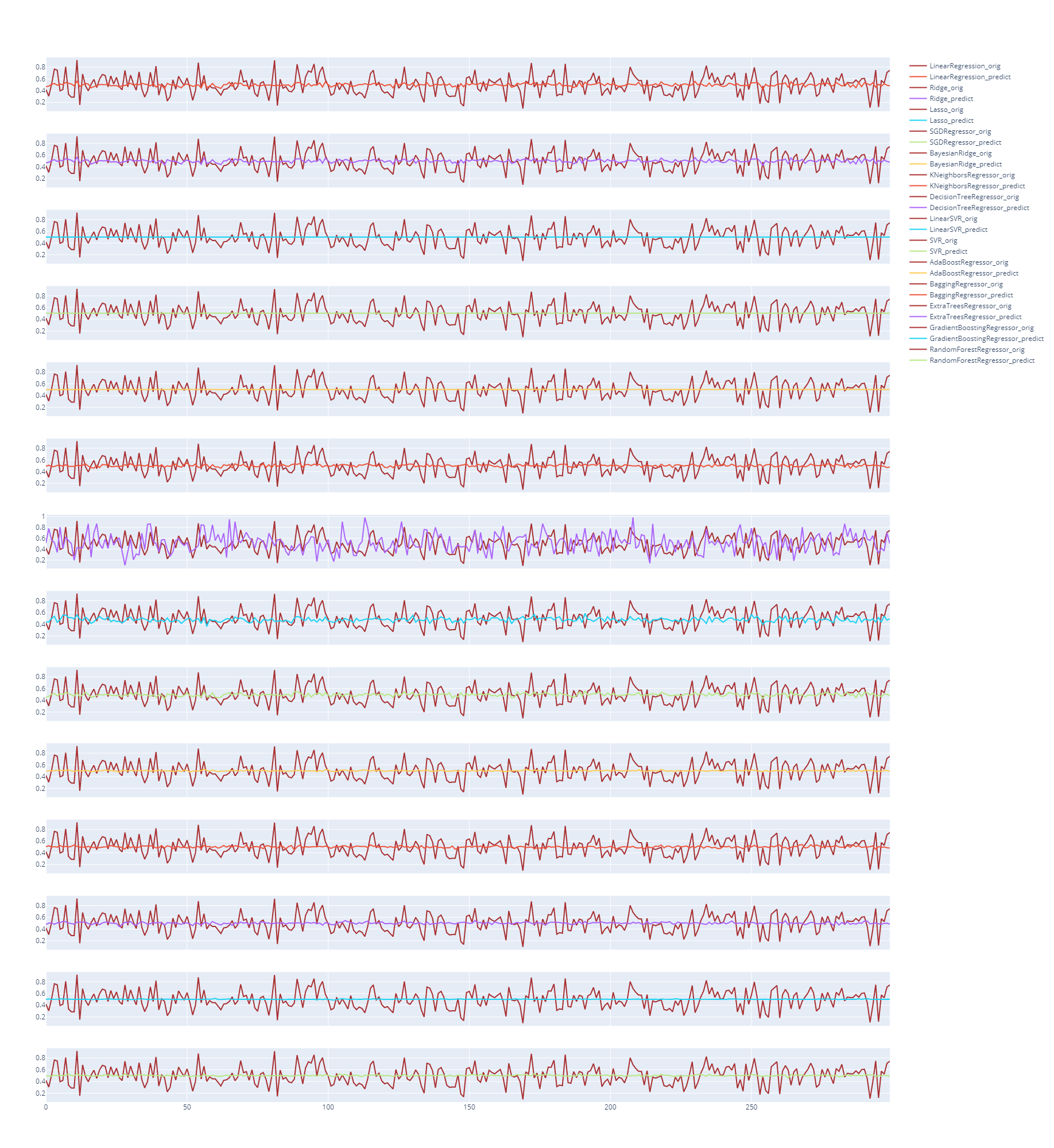
## **2.4 Тестирование моделей**

На графике видно, что все модели с использованием кросс-валидации показали неудовлетворительный результат. Для оценки качества регрессии использовался коэффициент детерминации.

Рисунок 7 – Результат кросс-валидация для каждой модели прогноза модуля упругости при растяжении 

Данный показатель является статистической мерой согласия, с помощью которой можно определить, насколько уравнение регрессии соответствует реальным данным. Коэффициент детерминации изменяется в диапазоне от 0 до 1. На графике видно, что коэффициент детерминации приближен к 0, это означает, что связь между переменными регрессионной модели отсутствует.

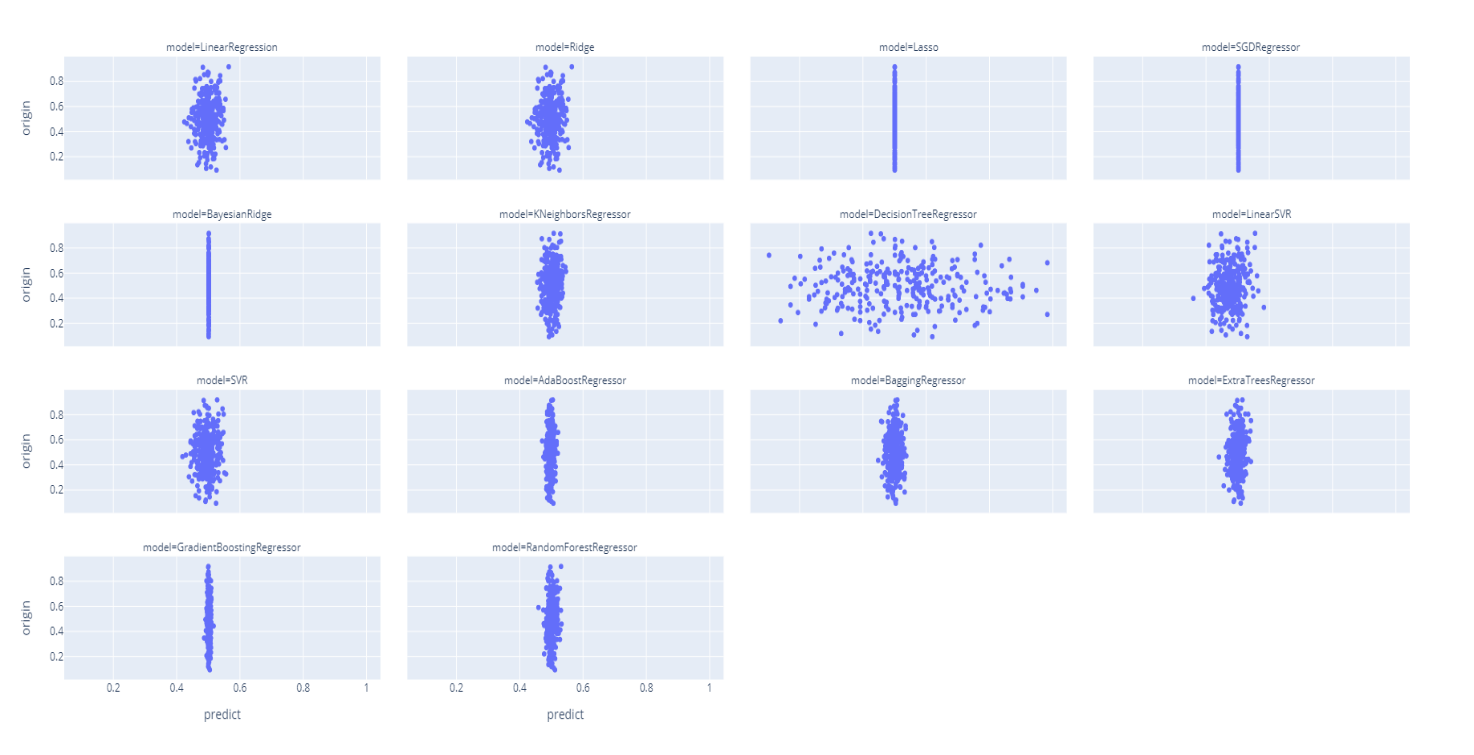
Модель DecisionTreeRegressor показала большой разброс и отрицательные значения результатов кросс-валидации. Это означает, что предсказания, сделанные данной моделью хуже, чем оценки на основе среднего значения (Рисунок 7).

Рисунок 8 – Оригинальные и предсказанные значения по каждой модели прогноза модуля упругости при растяжении 

По графику видно, что модели показали близкое значение к усредненному значению по выборке.

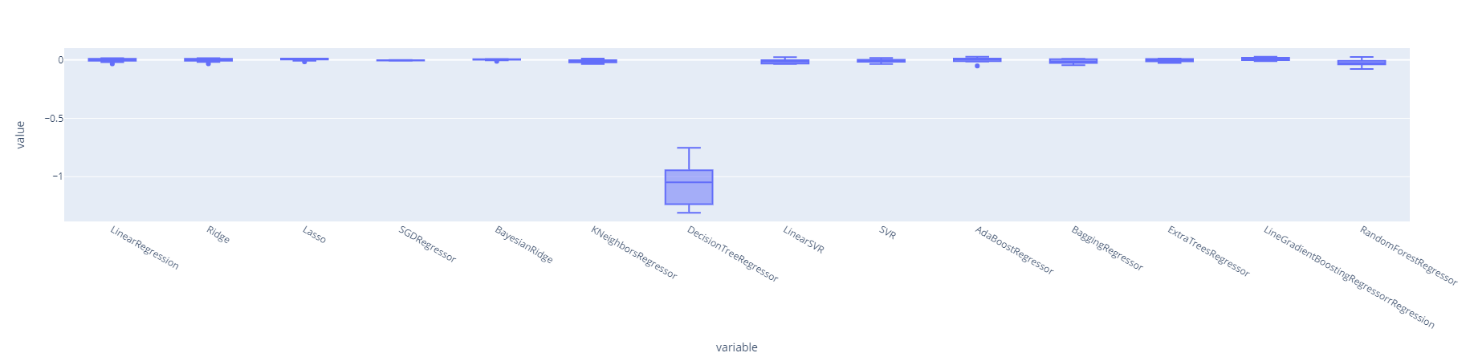
Разработанные модели не дают достоверный прогноз (Рисунок 8).

Рисунок 9 – Диаграмма рассеяния предсказанных значений модуля упругости при растяжении



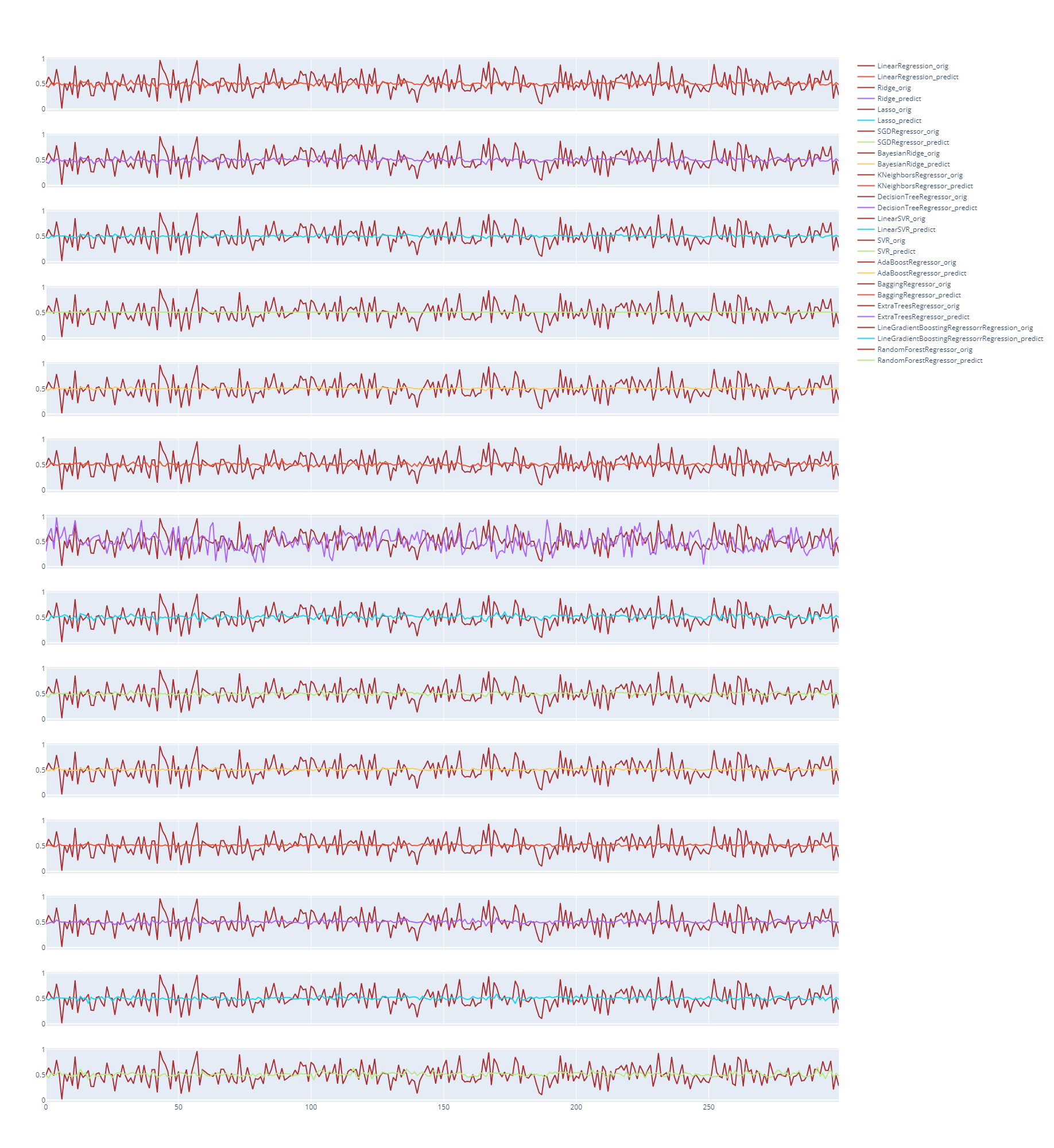
По диаграмме рассеяния видно, что корреляция между оригинальными значениями и предсказанными отсутствует (Рисунок 9).

Рисунок 10 – Результат кросс-валидация для каждой модели прогноза модуля прочности при растяжении



На графике видно, что все модели показали неудовлетворительный результат (Рисунок 10).

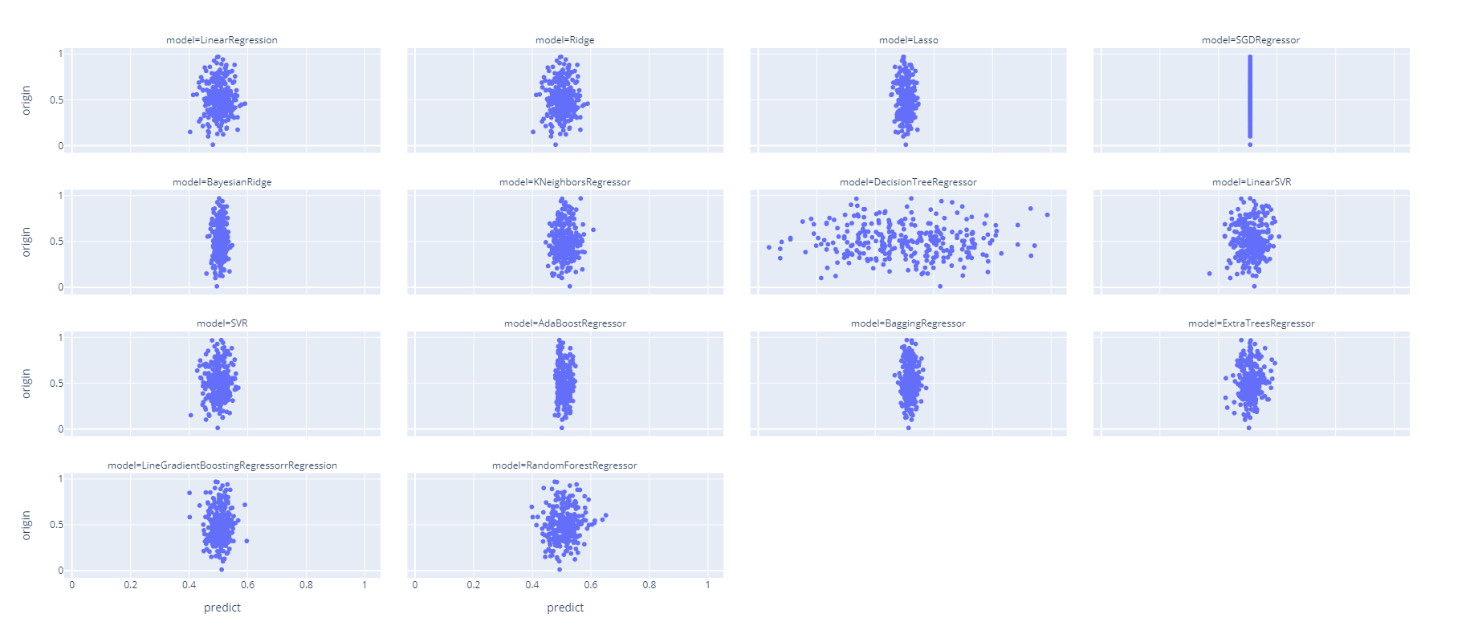
Рисунок 11 – Оригинальные и предсказанные значения по каждой модели прогноза модуля прочности при растяжении



По графику видно, что модели показали близкое значение к усредненному значению по выборке.

Разработанные модели не дают достоверный прогноз (Рисунок 11).

Рисунок 12 – Диаграмма рассеяния предсказанных значений модуля прочности при растяжении



По диаграмме рассеяния видно, что корреляция между оригинальными значениями и предсказанными отсутствует (Рисунок 12).

## **2.5** **Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель**

Методом train\_test\_split данные разделены на тестовую и обучающую выборки: 10% данных оставлены на тестирование модели, на остальных данных происходит обучение моделей.

Нейронная сеть построена с помощью класса keras.Sequential со следующими параметрами:

1. Normalization – слой нормализации
2. Dense(120, activation='relu') – входной плотно связанный слой с 120 нейронами, активационная функция RELU
3. BatchNormalization() - слой пакетной нормализации
4. Dense(60, activation='relu') – скрытый плотно связанный слой с 60 нейронами, активационная функция RELU
5. BatchNormalization()- слой пакетной нормализации
6. Dense(30, activation='relu') - скрытый плотно связанный слой с 30 нейронами, активационная функция RELU
7. BatchNormalization() - слой пакетной нормализации
8. Dense(15, activation='relu') - скрытый плотно связанный слой с 15 нейронами, активационная функция RELU
9. BatchNormalization() - слой пакетной нормализации
10. Dense(1) - выходной плотно связанный слой с 1 нейроном, активационная функция RELU

Для обучения нейронной сети используется 100 эпох. Используя функцию EarlyStopping обучение остановится когда val\_loss перестанет улучшаться. Нейронная сеть остановилась на 7 эпохе, так как с 3 эпохи val\_loss перестал улучшаться.

Рисунок 13 – График потерь нейронной сети

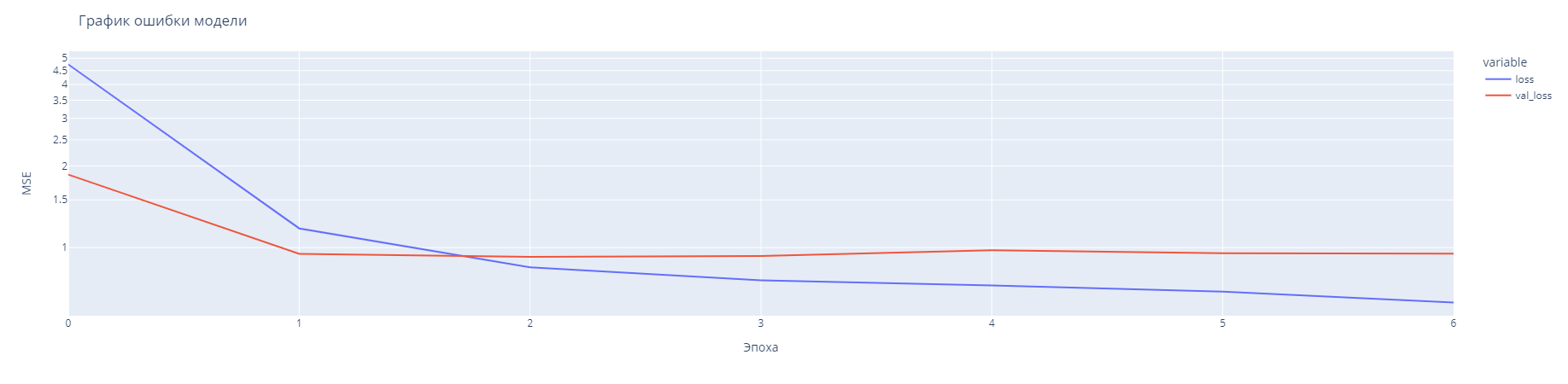


Рисунок 14 – Оригинальные и предсказанные значения нейронной сети соотношение матрица-наполнитель

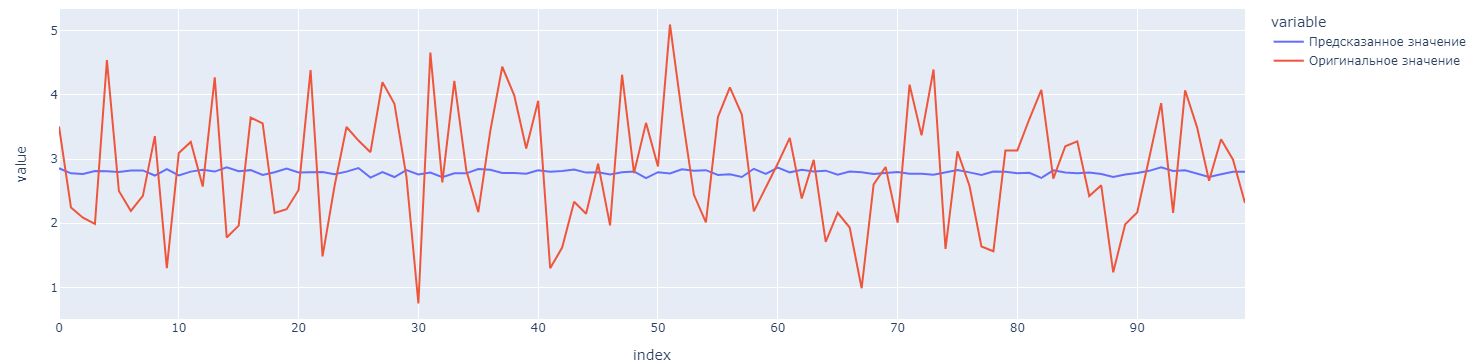
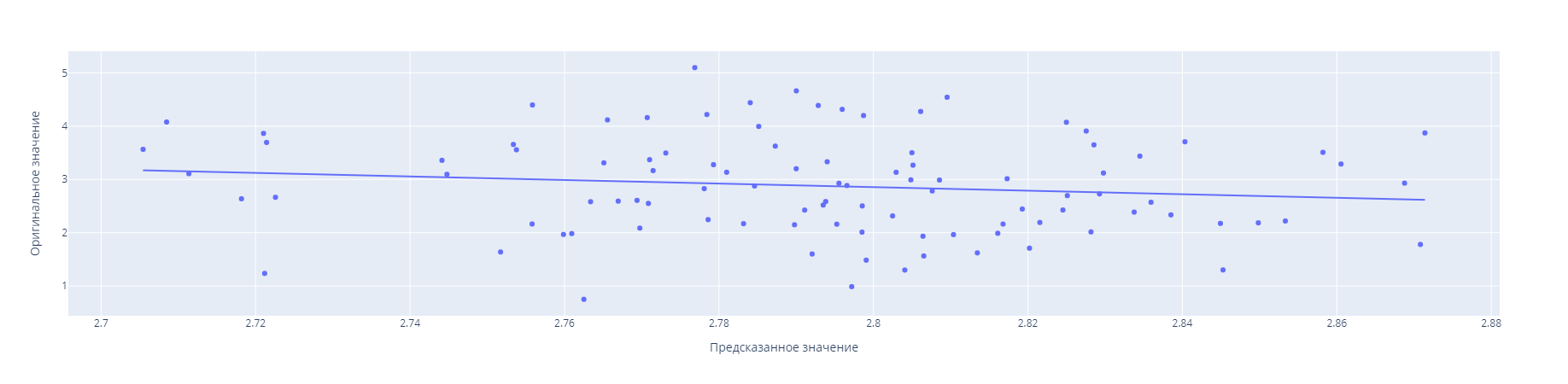


График оригинальные и предсказанные значения нейронной сети показывают близкое значение к усредненному значению по выборке.

Рисунок 15 – Диаграмма рассеяния предсказанных значений соотношение матрица-наполнитель

По диаграмме рассеяния видно, что корреляция между оригинальными значениями и предсказанными отсутствует.

Демонстрация более простой модели

Нейронная сеть построена с помощью класса keras.Sequential со следующими параметрами:

1. Normalization – слой нормализации
2. Dense(12, activation='relu') – входной плотно связанный слой с 120 нейронами, активационная функция RELU
3. BatchNormalization() - слой пакетной нормализации
4. Dense(1) - выходной плотно связанный слой с 1 нейроном, активационная функция RELU

Рисунок 16 – График потерь нейронной сети

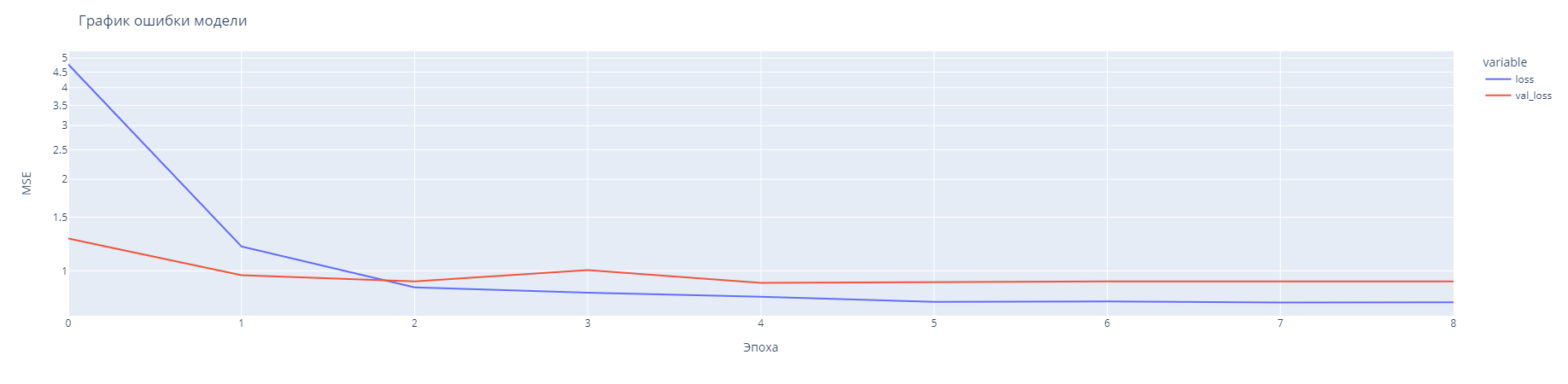


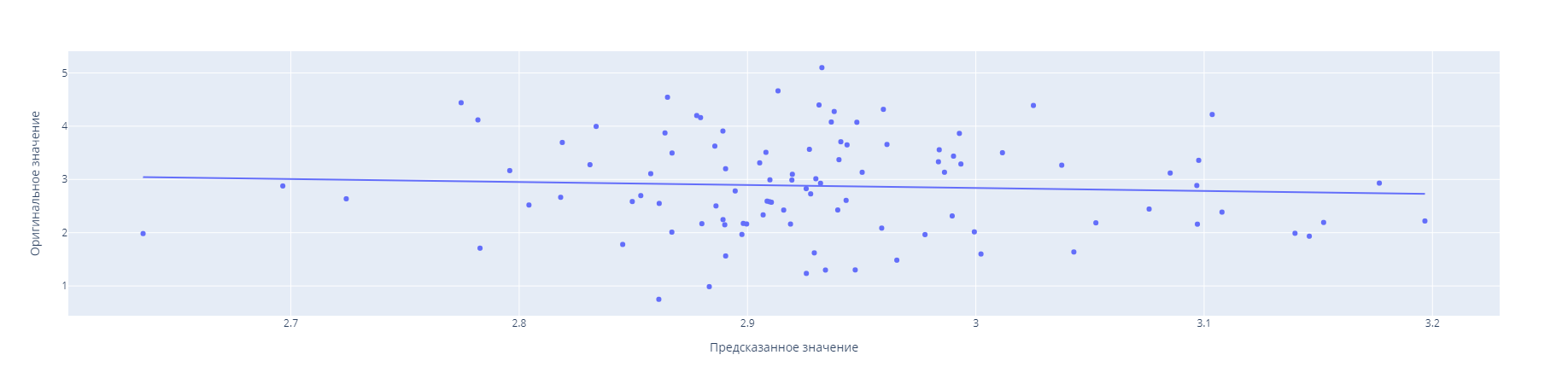
График оригинальные и предсказанные значения нейронной сети показывают близкое значение к усредненному значению по выборке (рисунок 16).

Рисунок 17 – Оригинальные и предсказанные значения нейронной сети соотношение матрица-наполнитель



Посмотрим на диаграмму рассеяния предсказанных значений соотношение матрица-наполнитель (рисунок 17).

Рисунок 18 – Диаграмма рассеяния предсказанных значений соотношение матрица-наполнитель



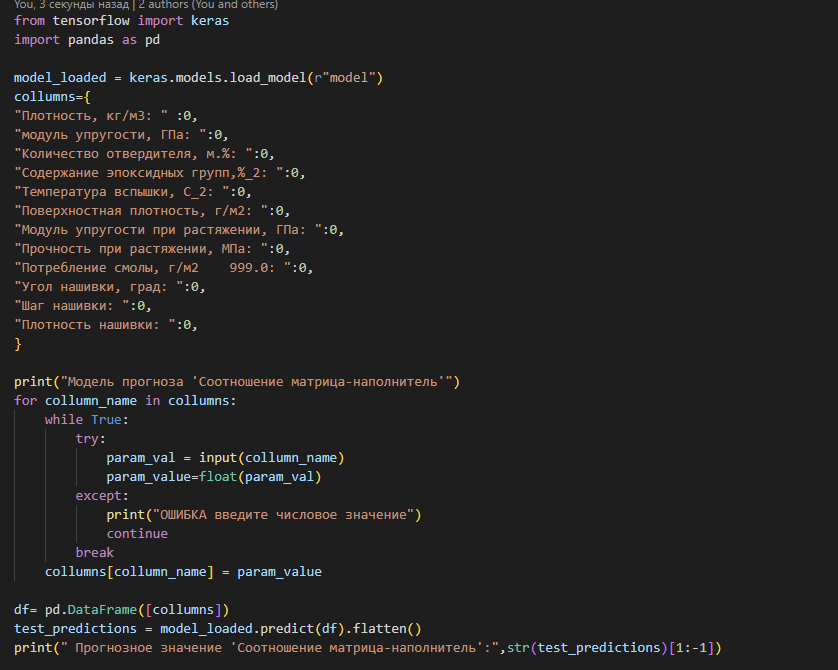
По диаграмме рассеяния видно, что корреляция между оригинальными значениями и предсказанными отсутствует.

Нейронная сеть с 2 слоями показала эффективность не хуже, чем нейронная сеть с большим количеством слоев.

## **2.6 Разработка приложения**

Разработаем приложение с интерфейсом командной строки, которое будет выдавать прогноз значения «Соотношение матрица-накопитель» (рисунок 18).

Рисунок 19 – Приложение прогноза Соотношение матрица-накопитель



* С помощью keras.models.load\_model() загружена сохраненная модель.
* В цикле программа запрашивает входные данные для нейронной сети.
* После получения необходимых входных данных модель предсказывает искомое значение.

## **2.7 Создание удаленного репозитория и загрузка результатов работы на него**

В процессе выполнения выпускной квалификационной работы был создан репозиторий на GitHub, который находится по адресу:

<https://github.com/ZViktoriya/VCR>

Ссылка на проект в Colab Notebook:

<https://colab.research.google.com/drive/1xesXzrpc7VQtECSdtvczeeDECV7SJeO8?usp=sharing>

В репозитории доступны следующие результаты: код анализа данных, обучения моделей, код веб-приложения, презентация, пояснительная записка к выпускной квалификационной работе.

# **Заключение**

Для прогноза модуля упругости при растяжении и прочности при растяжении использованы следующие модели: LinearRegression, SGDRegressor, Ridge Regression, LassoRegressor, BayesianRidge, DecisionTreeRegressor, LinerSVR, SVR, KNeighborsRegressor, AdaBoostRegressor, BaggingRegressor, ExtraTreesRegressor, GradientBoostingRegressor, RandomForestRegressor. Для прогноза значения соотношение матрица-накопитель использовалась нейронная сеть.

Использованные при разработке модели не дали достоверный прогноз.

Корреляция между оригинальными значениями и предсказанными отсутствует. Оригинальные и предсказанные значения моделей показывают близкое значение к усредненному значению по выборке.

Все модели с использованием кросс-валидации показали неудовлетворительный результат. Для оценки качества регрессии использовался коэффициент детерминации.

Для решения поставленной задачи предоставленный набор данных не достаточен.

# **Список литературы**

1 Д.А. Иванов А.И., Ситников, С.Д Шляпин – Композиционные материалы: учебное пособие для вузов, 2019. 13 с.

2 Чун-Те Чен и Грейс Х. Гу. Машинное обучение для композитных материалов (март 2019г.) – Режим доступа: https://www.cambridge.org/core/journals/mrs-communications/article/machine- learning-for-composite-materials/F54F60AC0048291BA47E0B671733ED15. (дата обращения 01.06.2022)

3 Язык программирования Python- Режим доступа:

https://www.python.org/. (дата обращения 01.06.2022)

4. Библиотека Pandas- Режим доступа: https://pandas.pydata.org/. (дата обращения 02.06.2022)

5 Библиотека Matplotlib- Режим доступа: https://matplotlib.org/. (дата обращения 02.06.2022)

6 Библиотека Seaborn- Режим доступа: https://seaborn.pydata.org/. (дата обращения 01.06.2022)

7 Библиотека Sklearn- Режим доступа: https://scikit-learn.org/stable/. (дата обращения 01.06.2022)

8 Среда разработки Jupyter Notebook- Режим доступа: https://jupyter.org/. (дата обращения 03.06.2022)

9 Библиотека Tensorflow: Режим доступа: https://www.tensorflow.org/. (дата обращения 02.06.2022)

10 Andre Ye. 5 алгоритмов регрессии в машинном обучении, о которых вам следует знать: – Режим доступа:https://habr.com/ru/company/vk/blog/513842/(дата обращения 02.06.2022)

11 Alex Maszański. Метод k-ближайших соседей (k-nearest neighbour): – Режим доступа: https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19. /(дата обращения 02.06.2022)

12 Миркес Е. М., Нейроинформатика. Учебное пособие с программами для выполнения лабораторных работ. 2003. ISBN 5-7636-0477-6