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

Содержание

[Введение 3](#_Toc102054153)

[1. Аналитическая часть 4](#_Toc102054154)

[1.1. Постановка задачи 4](#_Toc102054155)

[1.2. Описание используемых методов 8](#_Toc102054156)

[1.2.1 Линейная регрессия 8](#_Toc102054157)

[1.2.2 Регрессия k-ближайших соседей 10](#_Toc102054158)

[1.2.3 Случайный лес 10](#_Toc102054159)

[1.2.4 Многослойный перцептрон 11](#_Toc102054160)

[1.2.5 Лассо регрессия 12](#_Toc102054161)

[1.2.6 Метрики качества моделей 13](#_Toc102054162)

[1.3. Разведочный анализ данных 13](#_Toc102054163)

[2. Практическая часть 17](#_Toc102054164)

[2.1. Предобработка данных 17](#_Toc102054165)

[2.2. Разработка и обучение моделей 30](#_Toc102054166)

[2.3. Тестирование моделей 31](#_Toc102054167)

[2.3.1 Линейная регрессия 31](#_Toc102054168)

[2.3.2 Регрессия k-ближайших соседей 33](#_Toc102054169)

[2.3.3 Случайный лес 34](#_Toc102054170)

[2.3.4 Многослойный перцептрон 36](#_Toc102054171)

[2.3.5 Лассо регрессия 38](#_Toc102054172)

[2.4. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель. 40](#_Toc102054173)

[2.5. Разработка приложения 42](#_Toc102054174)

[2.6. Создание удаленного репозитория и загрузка результатов работы на него. 43](#_Toc102054175)

[В процессе выполнения выпускной квалификационной работы был создан репозиторий на GitHub, который находится по адресу: 43](#_Toc102054176)

[Заключение 43](#_Toc102054177)

[Библиографический список 44](#_Toc102054178)

**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.

Исходные датасеты 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.

Датасет 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.

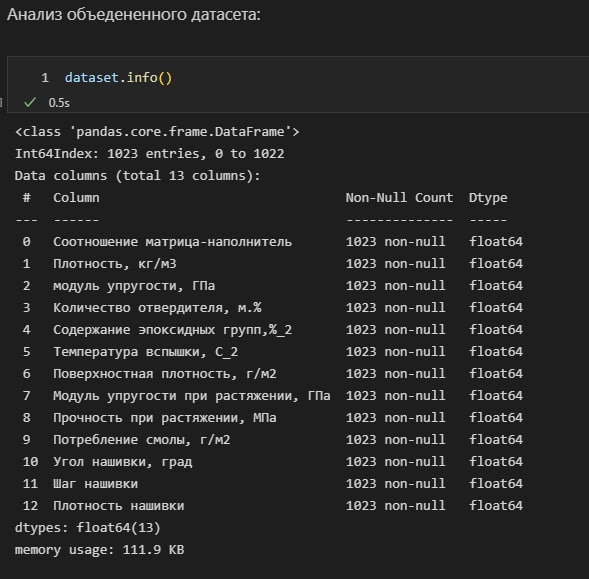
Количество строк в файле X\_bp.xlsx было 1023, столбцов 11. А количество строк в файле X\_nup.xlsx – 1040, столбцов 4.

После объединения таблиц удалим неинформационный столбец.

Дальнейшие исследования проводились с датасетом dataset, содержащий 1023 строк и 13 столбцов (Таблица 1).

Полученный dataset является отправной точкой дальнейших исследований, требует предобработки, которая будет ниже выполнена.

Таблица 1 – Анализ объединённого датасета

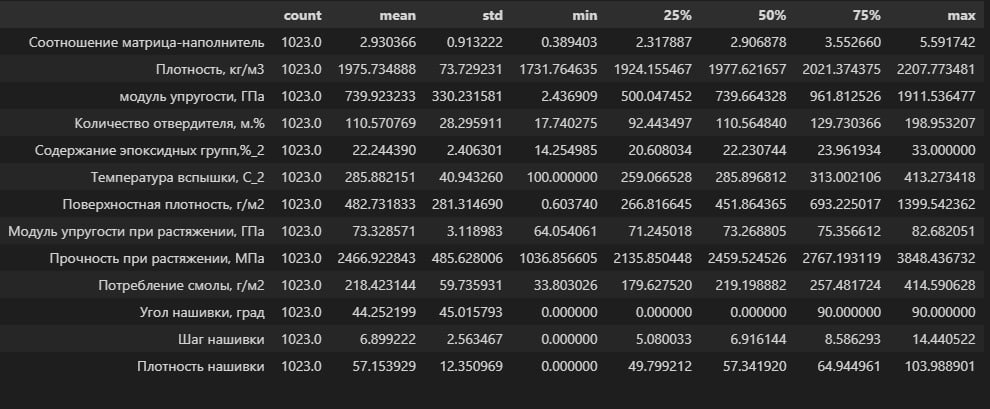


**1.2 Разведочный анализ данных**

Для получения представления о характере распределения переменных в датасете, формирования оценки качества исходных данных (наличия пропусков, выбросов), 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 (Таблица 2), а также матрица попарной корреляции, тепловая карта корреляции, гистограммы нормального распределения, поиск выбросов через ящик с усами.

Таблица 2 – Оценка статистических характеристик данных



Проверку пропусков выполняли с помощью методов: dataset.info(), который показывает количество ненулевых значений и тип данных, и dataset.insnull().sum(), который возвращает количество пропущенных значений в данных. Как видим пропуски в данных отсутствуют.

Метод dataset.nunique() возвращает количество уникальных значений для каждого столбца.

Для проведения разведочного анализа использовали язык программирования Python и библиотеки Numpy, Pandas, Matplotlib, Seaborn и Sklearn.

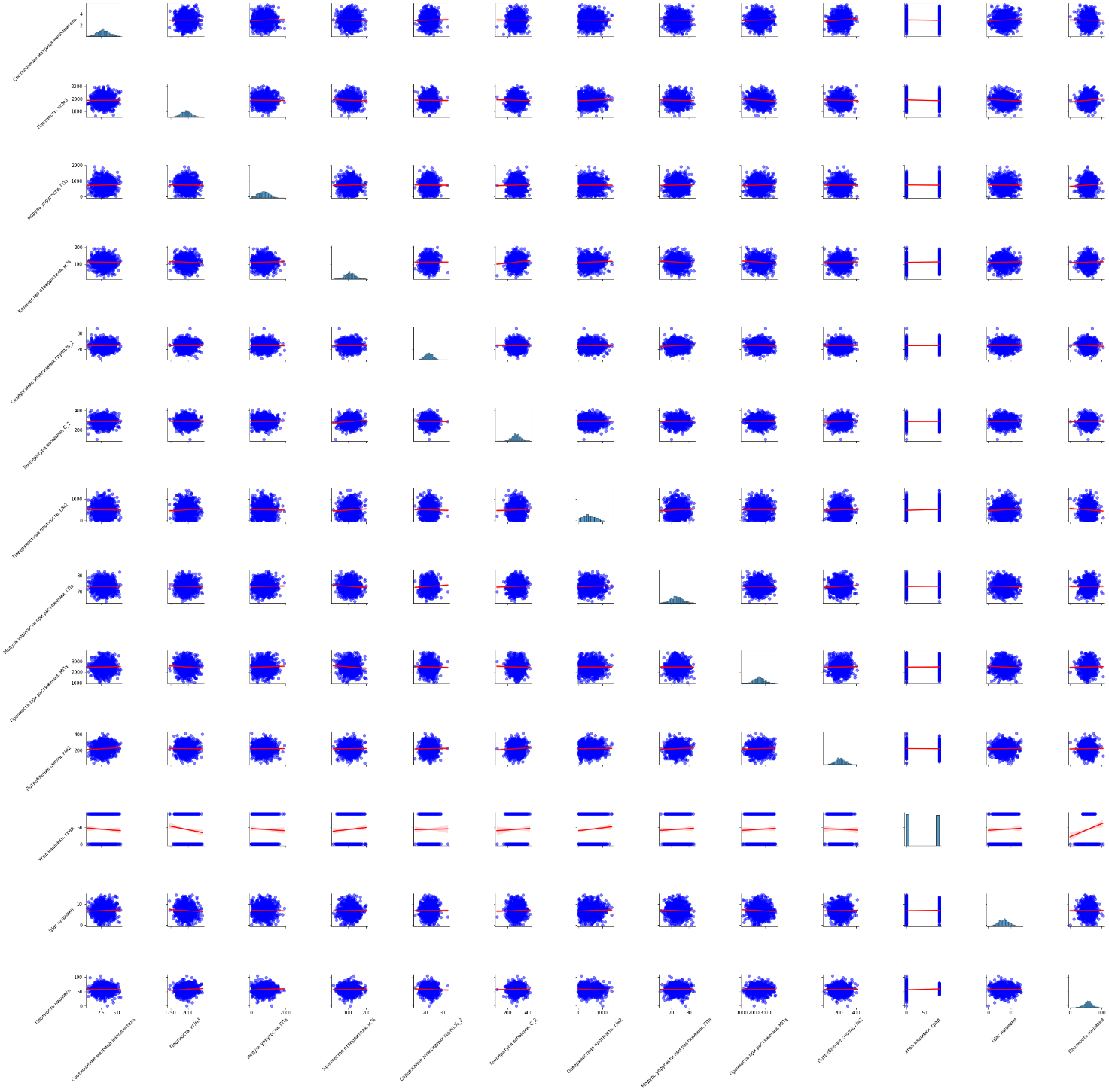
**1.3 Визуализация данных**

Для визуализации данных необходимо нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек, корреляционную тепловую карту Пирсона.

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.

Изучим графики попарного рассеяния (Рисунок 1).

Рисунок 1 – Попарные графики рассеяния точек



По форме разброса точек, в виде облаков, становится понятно, что зависимости между переменными, на которых будет основываться работа модели, не обнаруживаются.

Попробуем выявить связь между характеристиками с помощью корреляционной тепловой карты Пирсона (Рисунок 2).

Рисунок 2 – Корреляционная тепловая карта Пирсона



Видно, что все коэффициенты корреляции близки к нулю. Это означает отсутствие линейной зависимости между признаками.

Построим boxplot (ящик с усами) и посмотрим число выбросов.

Видим: 25 выброса методом 3-х сигм и 94 выброса методом межквартильных расстояний.

Метод 3-х сигм найдено меньше выбросов. График «ящик с усами» показывает не большой размах, поэтому удалим те данные, которые выявили, как выбросы методом 3-х сигм, тем самым, сохраним большее количество данных в датасете: 999 rows × 13 columns, количество строк стало на 25 меньше.

Построим тепловую матрицу корреляции после удаления выбросов чтобы посмотреть, как изменились зависимости. Как видим по графику, в результате удаления выбросов, корреляция выросла незначительно и кардинальных изменений нет, корреляция между признаками по-прежнему, визуально не выявляется.

**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.

Многие алгоритмы машинного обучения работают лучше, когда числовые входные переменные масштабируются до стандартного диапазона. Сюда входят алгоритмы, использующие взвешенную сумму входных данных, такие как линейная регрессия, и алгоритмы, использующие меры расстояния, такие как k ближайших соседей.

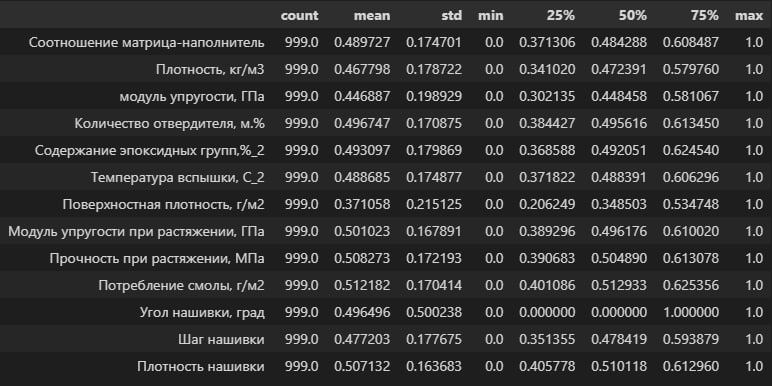
Двумя наиболее популярными методами масштабирования числовых данных перед моделированием являются нормализация и стандартизация. Нормализация масштабирует каждую входную переменную отдельно до диапазона от 0 до 1, который является диапазоном для значений с плавающей запятой, где мы имеем наибольшую точность.

Проводим нормализацию датасета при помощи MinMaxScaler().

После нормализации проводим анализ нового датасета dataset\_norm: рисуем гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек.

Смотрим распределения для каждого признака после нормализации, максимальные и минимальные значения (Таблица 3).

Таблица 3 – Оценка статистических характеристик данных после нормализации



**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.

Делим методом train\_test\_split данные на тестовую и обучающую выборки согласно заданию: 30% данных оставим на тестирование модели, на остальных происходит обучение моделей. Зерно генератора случайных чисел зададим постоянным для воспроизводимости результатов обучения.

Объявляю функцию, которая имеет следующий алгоритм работы с моделями:

1) Вызываем модель регрессии, передаем параметры

2) Все модели будем прогонять по сетке GridSearchCV для получения наилучших параметров оценки целевых переменных.

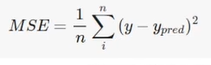
3) Вызываем метод GridSearchCV, передавая в него модель регрессии и возможные параметры. Получаем лучшее возможное решение, выводим результат лучшего параметра.

4) Обучаем модель с учетом лучших параметров.

5) Считаем ошибки модели, записываем их в созданный пустой словарь ошибок.

Будем использовать следующие метрики качества регрессии.

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.

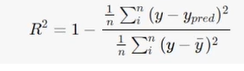


(1)

### Cредняя 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.

Простая линейная регрессия имеет место, если рассматривается зависимость между одной входной и одной выходной переменными. Для этого определяется уравнение регрессии (4) и строится соответствующая прямая, известная как линия регрессии.

y=ax+b (4)

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 (5).

Y=b0+b1\*x1+b2\*x2+⋯+bn\*xn, (5)

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.

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.

На языке python линейная регрессия реализована в sklearn.linear\_model.LinearRegression.

Ridge Regression (гребневая регрессия) решает некоторые проблемы обычных методов наименьших квадратов, налагая штраф на размер коэффициентов. Параметр сложности управляет степенью сжатия: чем больше значение, тем больше степень сжатия и, таким образом, коэффициенты становятся более устойчивыми к коллинеарности. Член штрафа (лямбда) упорядочивает коэффициенты таким образом, что если коэффициенты принимают большие значения, функция оптимизации подвергается штрафу. Таким образом, гребневая регрессия уменьшает коэффициенты и помогает уменьшить сложность модели. Эта модель решает модель регрессии, в которой функция потерь является линейной функцией наименьших квадратов, а регуляризация задается нормой L2.

L2-регуляризация предотвращает переобучения модели путём запрета на непропорционально большие весовые коэффициенты. Суть состоит в том, что мы изменяем нашу первоначальную функцию, добавляя «штраф» на большие весовые коэффициенты. Для этого мы добавляем постоянную , умноженный на квадрат w.

Одним из способов избежать переусложнения модели и её переобученности является L2-регуляризация, также известная под названием регрессии Риджа. Работает она так. Мы добавляем квадрат величины весовых коэффициентов, умноженный на константу λ, к нашей прежней квадратической функции погрешностей. Мы делаем это потому, что большое значение весовых коэффициентов являются признаком переобученности. Далее мы решаем уравнение относительно w, беря производную и приравнивая её к нулю. Это и есть метод максимизации апостериора (MAP), так как мы находим максимум апостериора w для данных.

LassoRegressor - регрессия по методу наименьших квадратов часто может стать неустойчивой, то есть сильно зависящей от обучающих данных, что обычно является проявлением тенденции к переобучению. Избежать такого переобучения помогает регуляризация L1 - общий метод, заключающийся в наложении дополнительных ограничений на искомые параметры, которые могут предотвратить излишнюю сложность модели. L1-регуляризация реализует это путём отбора наиболее важных факторов, которые сильнее всего влияют на результат. 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.

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 «y» 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.

Преимущества: высокая точность, нечувствительность к выбросам и отсутствие допущений при вводе данных.

Недостатки: высокая временная сложность и большая сложность пространства.

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. Это немного сложно понять, потому что это не то, как люди думают естественным образом, и, возможно, лучший способ показать эту разницу — создать настоящее дерево решений. 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. SVM сопоставляет обучающие примеры с точками в пространстве, чтобы максимизировать ширину разрыва между двумя категориями. Затем новые примеры сопоставляются с тем же пространством, и их принадлежность к категории определяется в зависимости от того, на какую сторону разрыва они попадают. В дополнение к выполнению линейной классификации SVM могут эффективно выполнять нелинейную классификацию, используя так называемый трюк ядра, неявно отображая свои входные данные в многомерные пространства признаков.

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.

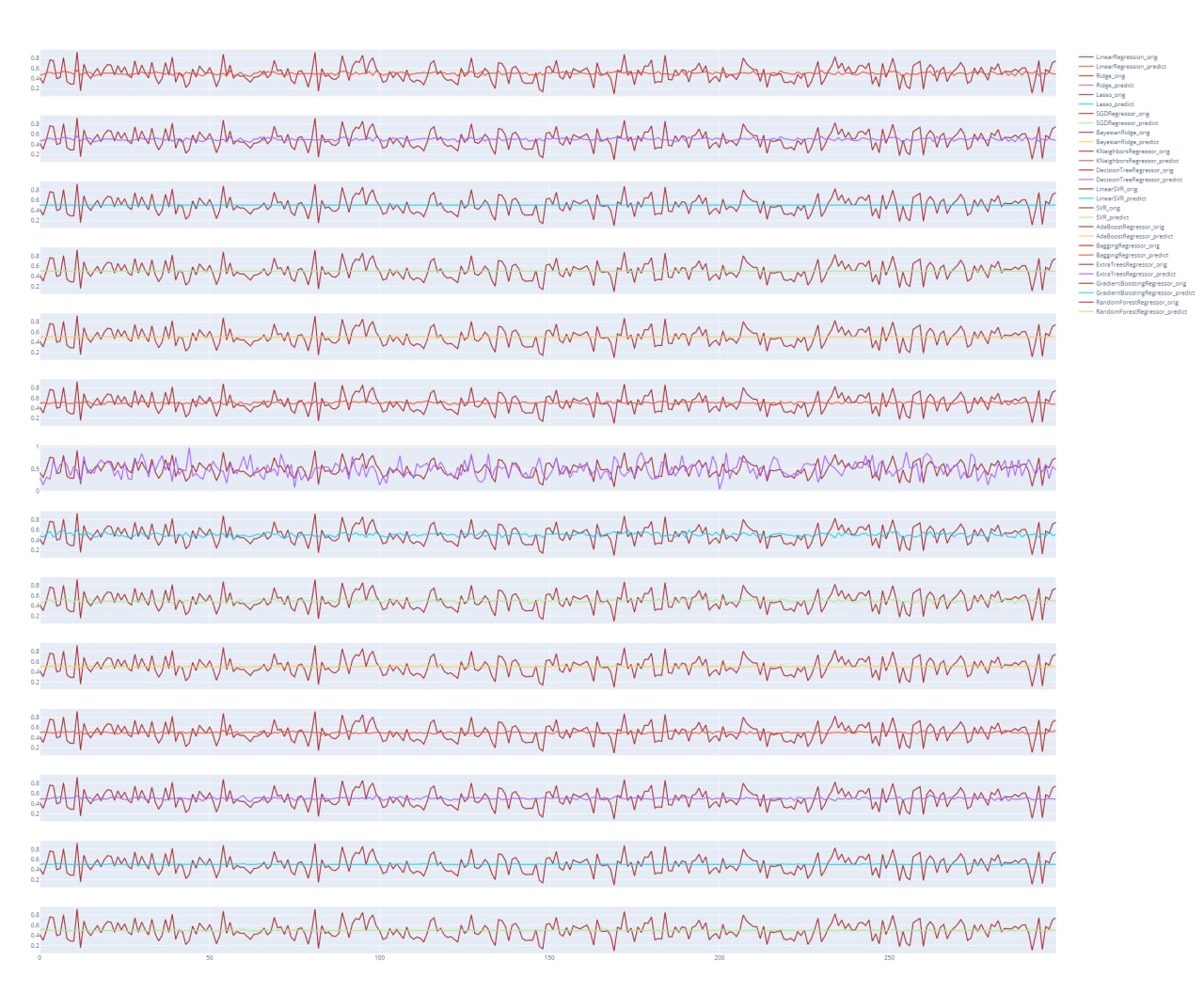
**2.4 Тестирование моделей**

# Рисунок 3 – Результат кросс-валидация для каждой модели

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

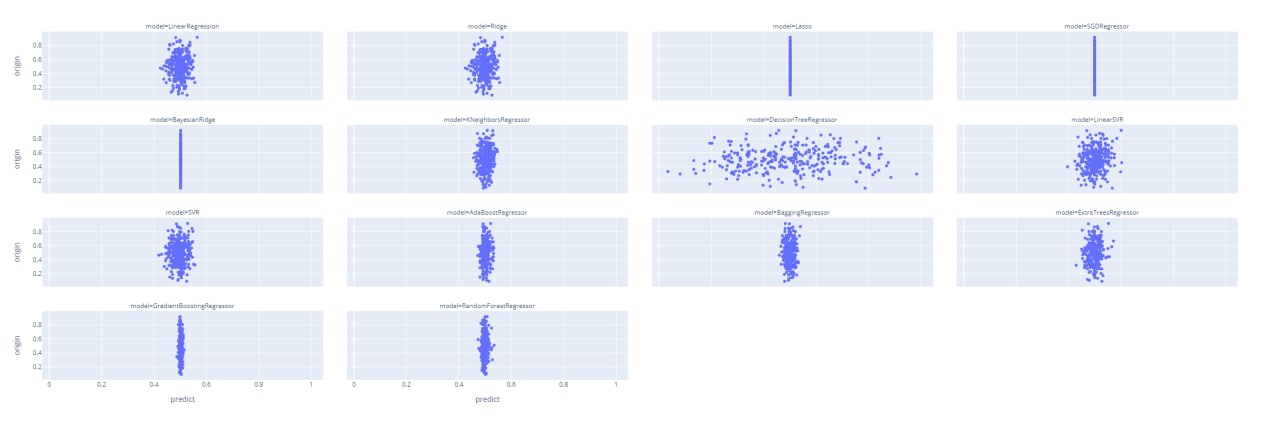
Данный показатель является статистической мерой согласия, с помощью которой можно определить, насколько уравнение регрессии соответствует реальным данным. Коэффициент детерминации изменяется в диапазоне от 0 до 1. Как мы видим, на графике он приближен к 0, это означает, что связь между переменными регрессионной 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 4 – 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 5 – Диаграмма рассеяния предсказанных значений модуля упругости при растяжении 

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

Рисунок 6 – Оригинальные и предсказанные значения по каждой модели

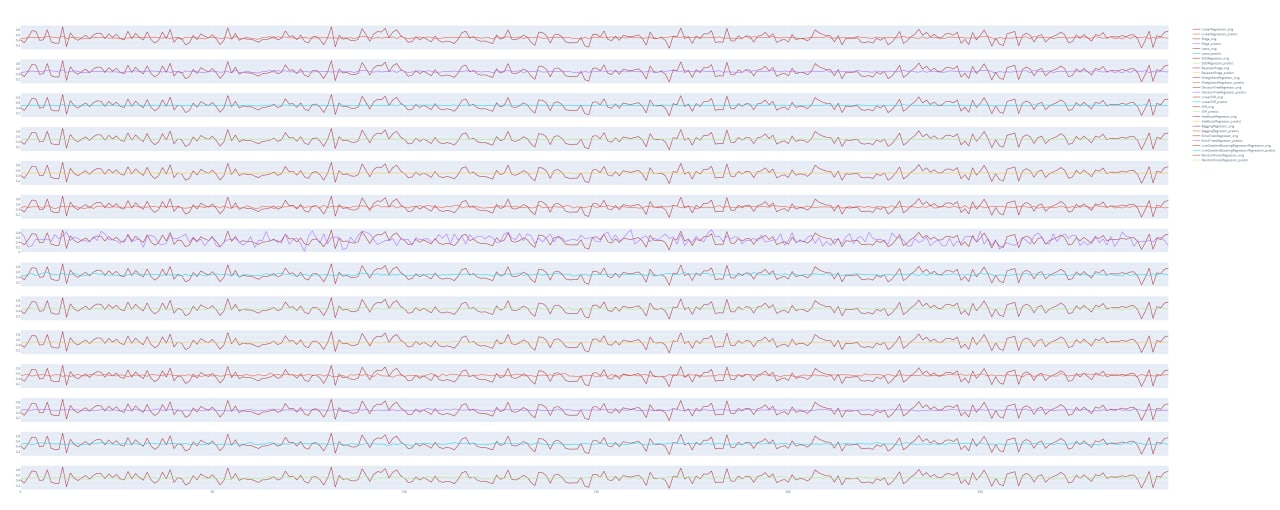
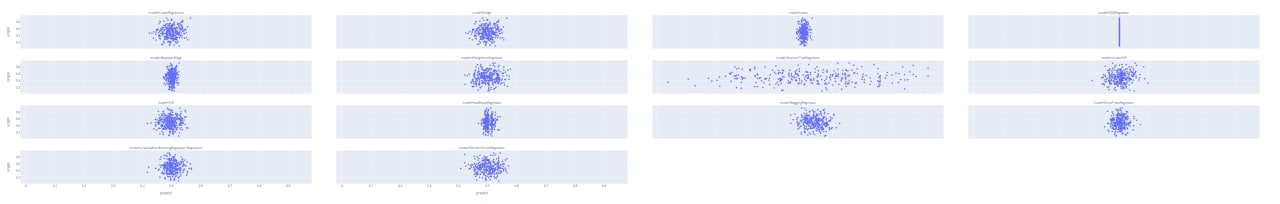


Рисунок 7 – Оригинальные и предсказанные значения по каждой модели



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

Разделим исходный датасет на тренировочную и тестовую части с помощью train\_test\_split. Затем нормализуем данные TensorFlow.layers.Normalization.

Постоим нейронную сеть с помощью класса keras.Sequential со следующими параметрами:

* входной слой нормализации 12 признаков;
* выходной слой для 1 признака;
* скрытых слоев: 4;
* нейронов в скрытом слое:120, 60, 30, 15;
* активационная функция скрытых слоев: relu;
* loss-функция: MeanSquaredError.