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 Балакин Андрей Сергеевич

Москва, 2022

Содержание

Введение

Композиционный материал (композит, КМ) — неоднородный сплошной материал, состоящий из двух или более компонентов, среди которых можно выделить армирующие элементы, обеспечивающие необходимые механические характеристики материала, и матрицу (или связующее), обеспечивающую совместную работу армирующих элементов.

Механическое поведение композита определяется соотношением свойств армирующих элементов и матрицы, а также прочностью связи между ними. 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.

В результате совмещения армирующих элементов и матрицы образуется комплекс свойств композита, не только отражающий исходные характеристики его компонентов, но и включающий свойства, которыми изолированные компоненты не обладают. В частности, наличие границ раздела между армирующими элементами и матрицей существенно повышает трещиностойкость материала, и в композитах, в отличие от металлов, повышение статической прочности приводит не к снижению, а, как правило, к повышению характеристик вязкости разрушения.

Однако, 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).

Созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

1. Аналитическая часть

1.1 Постановка задачи

Необходимо спрогнозировать ряд конечных свойств получаемых композиционных материалов имея данные о начальных свойствах компонентов (количество связующего, наполнителя, температурный режим отверждения и т.д.).

Для этого необходимо:

1) провести разведочный анализ предложенных данных;
2) провести предобработку данных;
3) обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении;
4) написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель;
5) разработать приложение с графическим интерфейсом или интерфейсом командной строки, которое будет выдавать прогноз.

1.2 Описание используемых методов

Исследованием влияния одной или нескольких независимых переменных на зависимую переменную занимается регрессионный анализ. Независимые переменные иначе называют регрессорами или предикторами, а зависимые переменные — критериальными или регрессантами.

1.2.1 Линейная регрессия

Линейная регрессия — 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:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \cdots + w_p x_p$$

Метод LinearRegression использует метод наименьших квадратов и подгоняет линейную модель с коэффицентами $w = (w_1, \dots, w_p)$ к минимизации остаточной суммы квадрата между наблюдаемого целевого признака в наборе данных и предсказанного целевого признака по линейной аппроксимации. Математически это решение проблемы в следующем виде: $\min_w ||Xw — y||_2^2$

1.2.2 Градиентный бустинг

Градиентный бустинг – это продвинутый алгоритм машинного обучения для решения задач классификации и регрессии. Он строит предсказание в виде ансамбля слабых предсказывающих моделей, которыми в основном являются деревья решений. Из нескольких слабых моделей в итоге мы собираем одну, но уже эффективную. Общая идея алгоритма – последовательное применение предиктора (предсказателя) таким образом, что каждая последующая модель сводит ошибку предыдущей к минимуму.

1.2.3. Случайный лес

В случайных лесах (RandomForestRegressor) каждое дерево в ансамбле строится из выборки, взятой с заменой (то есть выборкой начальной загрузки) из обучающего набора.

Кроме того, при разбиении каждого узла во время построения дерева наилучшее разбиение находится либо по всем входным характеристикам, либо по случайному подмножеству размера max_features.

Назначение этих двух источников случайности — уменьшить дисперсию оценки леса. В самом деле, отдельные деревья решений обычно демонстрируют высокую дисперсию и имеют тенденцию переоснащаться. Внедренная случайность в лесах дает деревья решений с несколько несвязанными ошибками прогнозирования. Если взять среднее значение этих прогнозов, некоторые ошибки могут быть устранены. Случайные леса уменьшают дисперсию за счет комбинирования разных деревьев, иногда за счет небольшого увеличения

смещения. На практике уменьшение дисперсии часто бывает значительным, что дает в целом лучшую модель.

1.2.4 Полиномиальная регрессия

Простая линейная регрессия может быть расширена путем построения полиномиальных функций из коэффициентов. В случае стандартной линейной регрессии у вас может быть модель, которая выглядит следующим образом для двумерных данных:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + w_2 x_2$$

Если мы хотим подогнать к данным параболоид, а не плоскость, мы можем объединить функции в полиномы второго порядка, чтобы модель выглядела так:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2$$

Наблюдение заключается в том, что это все еще линейная модель. Чтобы убедиться в этом, представьте, что вы создаете новый набор функций:

$$z = [x_1, x_2, x_1 x_2, x_1^2, x_2^2]$$

С этой перемаркировкой данных наша проблема может быть записана:

$$\hat{y}(w, z) = w_0 + w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5$$

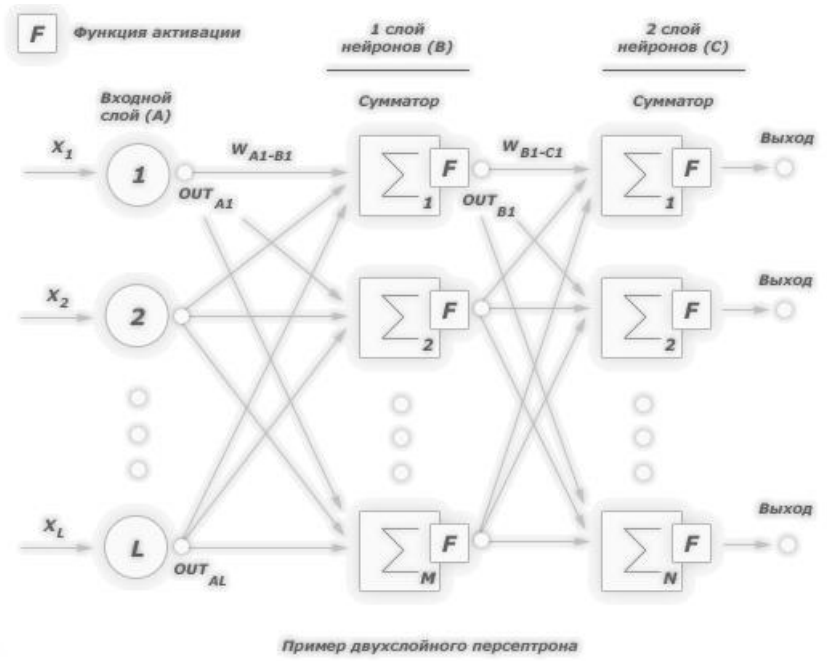
Мы видим, что полученная полиномиальная регрессия относится к тому же классу линейных моделей, который мы рассмотрели выше (т.е. модель линейна now) и могут быть решены теми же методами. Рассматривая линейные соответствия в многомерном пространстве, построенном с помощью этих базовых функций, модель обладает гибкостью, позволяющей соответствовать гораздо более широкому диапазону данных.

1.2.5 Многослойный персептрон

Многослойными персептронами называют нейронные сети прямого распространения. Входной сигнал в таких сетях распространяется в прямом направлении, от слоя к слою. Многослойный персептрон в общем представлении состоит из следующих элементов:

- множества входных узлов, которые образуют входной слой;
- одного или нескольких скрытых слоев вычислительных нейронов;
- одного выходного слоя нейронов.

Многослойный персептрон представляет собой обобщение однослойного персептрона Розенблатта. Примером многослойного персептрона является следующая модель нейронной сети:



Пример двухслойного персептрона

1.3 Разведочный анализ данных

Разведочный анализ — предварительный исследование данных с целью выявления наиболее общих зависимостей, закономерностей и тенденций, характера и свойств анализируемых данных, законов распределения анализируемых величин.

Методы разведочного анализа применяются для нахождения связей между переменными в ситуациях, когда отсутствуют (или недостаточны) априорные представления о природе этих связей.

Для проведения разведочного анализа предоставленные данные в двух дтаблицах Excel (X_bp.xlsx, X_nup.xlsx) были объединены по индексу, тип объединения INNER.

После загрузки и объединения данных получена краткая сводка данных о созданном датасете с помощью метода df.info():

```
<class 'pandas.core.frame.DataFrame'>
Float64Index: 1023 entries, 0.0 to 1022.0
Data columns (total 13 columns):
#   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
0   Соотношение матрица-наполнитель       1023 non-null   float64
1   Плотность, кг/м3                      1023 non-null   float64
2   модуль упругости, ГПа                 1023 non-null   float64
3   Количество отвердителя, м.%           1023 non-null   float64
4   Содержание эпоксидных групп,%_2       1023 non-null   float64
5   Температура вспышки, С_2              1023 non-null   float64
6   Поверхностная плотность, г/м2         1023 non-null   float64
7   Модуль упругости при растяжении, ГПа  1023 non-null   float64
8   Прочность при растяжении, МПа         1023 non-null   float64
9   Потребление смолы, г/м2               1023 non-null   float64
10  Угол нашивки, град                    1023 non-null   float64
11  Шаг нашивки                           1023 non-null   float64
12  Плотность нашивки                     1023 non-null   float64
dtypes: float64(13)
memory usage: 144.2 KB
```

Полученная таблица показывает количество не нулевых значений и тип данных. Объем выборки – 1023 строк. Пропуски в данных отсутствуют.

Для получения общих статистических сведений о датасете, включая кол-во значений, среднее значение, стандартное отклонение, минимальный элемент, 25% — первый квартиль, 50% — медиана, 75% — третий квартиль и максимальный элемент, использован метод df.describe(), который дает представление о распределении значений для каждого признака:

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660	5.591742
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375	2207.773481
модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	500.047452	739.664328	961.812526	1911.536477
Количество отвердителя, м.%	1023.0	110.570769	28.295911	17.740275	92.443497	110.564840	129.730366	198.953207
Содержание эпоксидных групп,%_2	1023.0	22.244390	2.406301	14.254985	20.608034	22.230744	23.961934	33.000000
Температура вспышки, С_2	1023.0	285.882151	40.943260	100.000000	259.066528	285.896812	313.002106	413.273418
Поверхностная плотность, г/м2	1023.0	482.731833	281.314690	0.603740	266.816645	451.864365	693.225017	1399.542362
Модуль упругости при растяжении, ГПа	1023.0	73.328571	3.118983	64.054061	71.245018	73.268805	75.356612	82.682051
Прочность при растяжении, МПа	1023.0	2466.922843	485.628006	1036.856605	2135.850448	2459.524526	2767.193119	3848.436732
Потребление смолы, г/м2	1023.0	218.423144	59.735931	33.803026	179.627520	219.198882	257.481724	414.590628
Угол нашивки, град	1023.0	44.252199	45.015793	0.000000	0.000000	0.000000	90.000000	90.000000
Шаг нашивки	1023.0	6.899222	2.563467	0.000000	5.080033	6.916144	8.586293	14.440522
Плотность нашивки	1023.0	57.153929	12.350969	0.000000	49.799212	57.341920	64.944961	103.988901

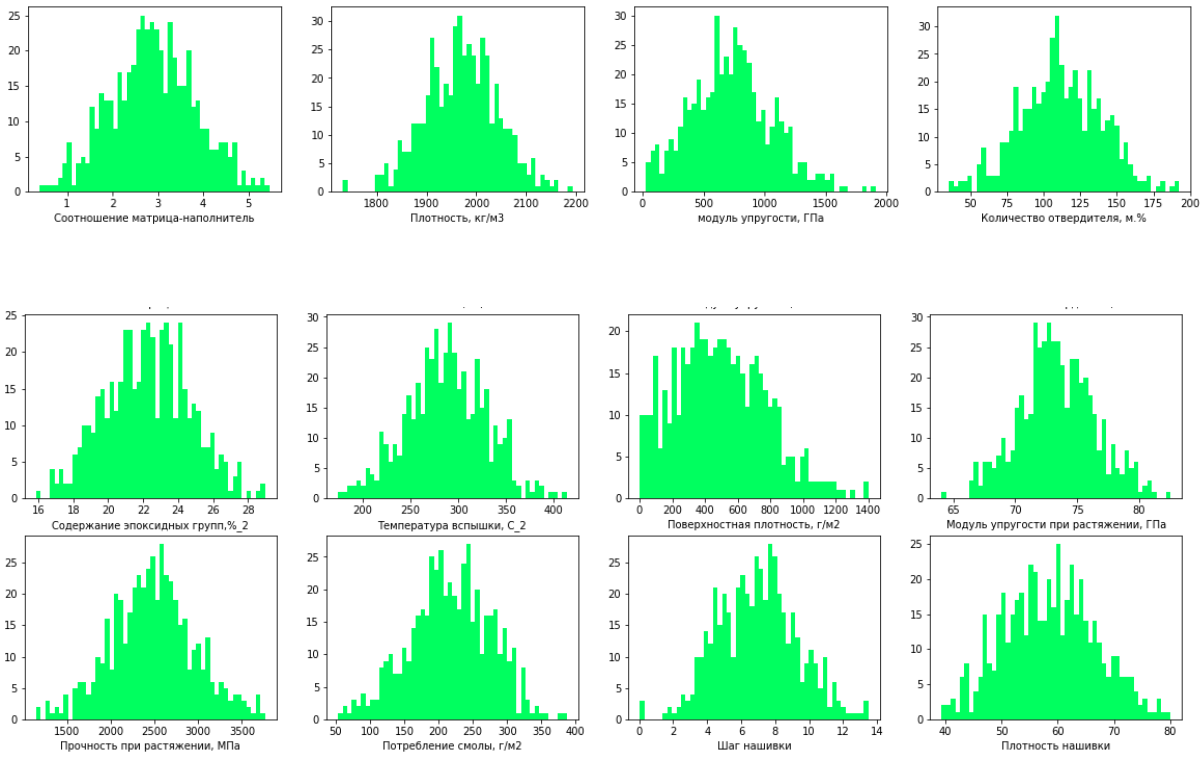
С помощью метода df.nunique() получаем кол-во уникальных значений в каждом признаке:

```
Соотношение матрица-наполнитель          1014
Плотность, кг/м3                         1013
модуль упругости, ГПа                    1020
Количество отвердителя, м.%              1005
Содержание эпоксидных групп,%_2          1004
Температура вспышки, С_2                 1003
Поверхностная плотность, г/м2            1004
Модуль упругости при растяжении, ГПа     1004
Прочность при растяжении, МПа            1004
Потребление смолы, г/м2                  1003
Угол нашивки, град                          2
Шаг нашивки                               989
Плотность нашивки                         988
dtype: int64
```

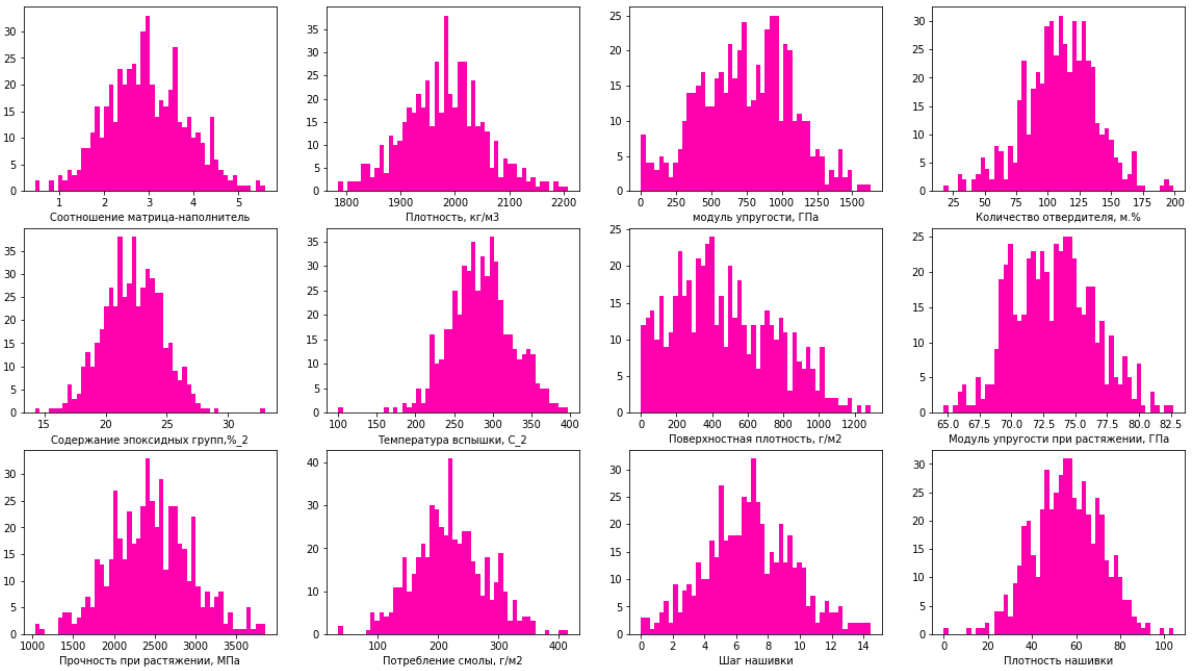
Признак «Угол нашивки, град» имеет всего 2 значения 0 и 90. В нашем случае данный признак является категориальным, поэтому дальнейшее рассмотрение данных будем проводить для каждой категории по отдельности.

С помощью методов визуализации данных построим гистограммы распределения, «ящики с усами» и попарную диаграмму рассеяния.

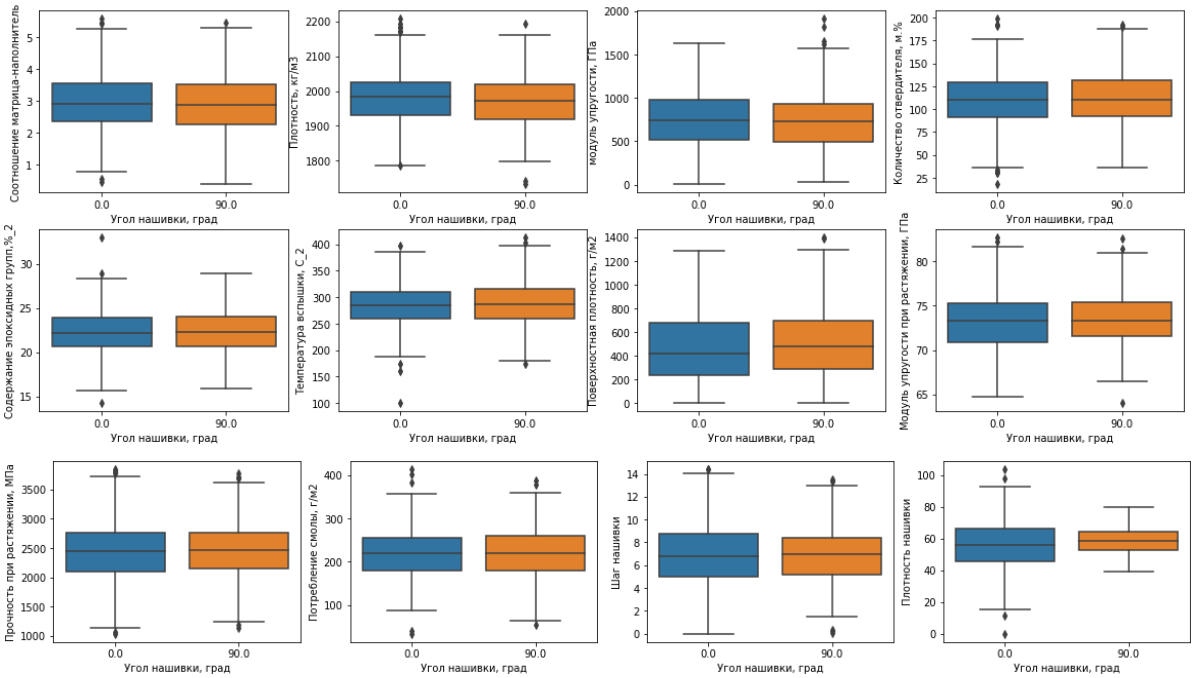
Гистограмма распределения данных при угле нашивки 90 градусов:



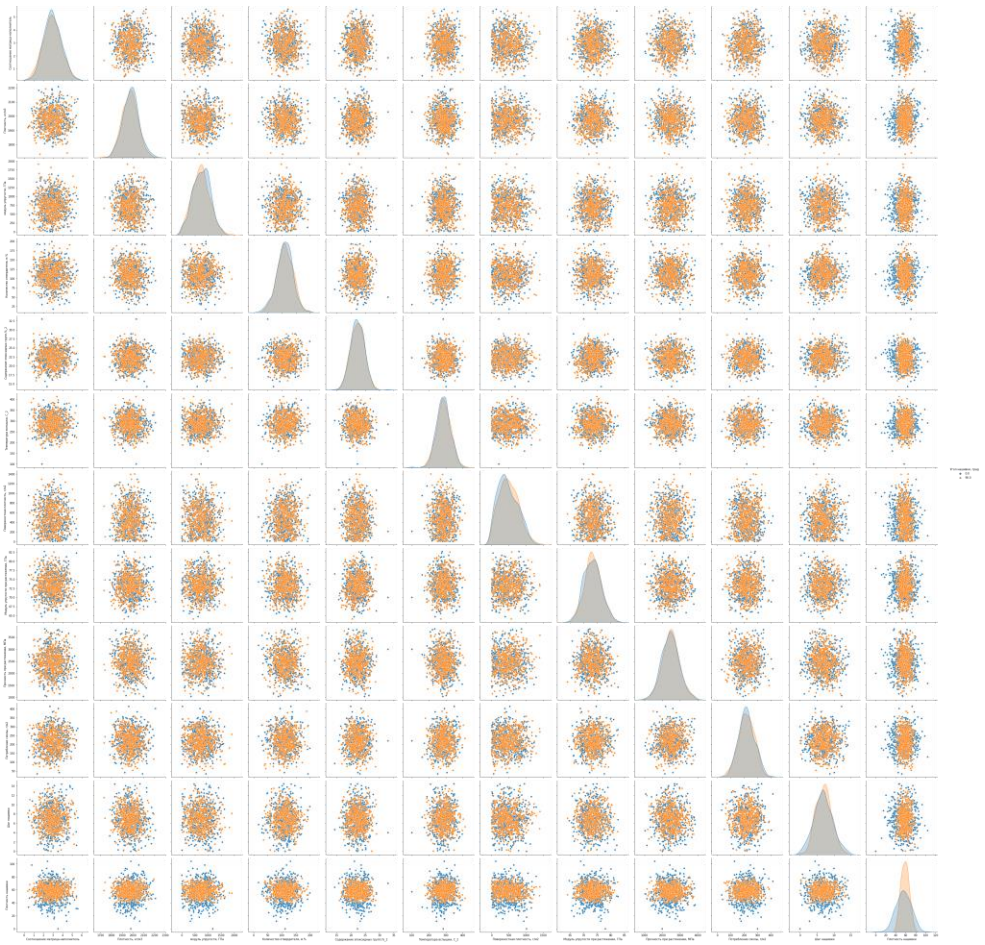
Гистограмма распределения данных при угле нашивки 0 градусов:



«Ящик с усами» для каждой категории:



Попарная диаграмма рассеяния для каждой категории:



Согласно построенным гистограммам распределения – все признаки близки к нормальному распределению. На основе графиков «ящик с усами» мы можем сказать о наличии некоторого количества выбросов. Попарная диаграмма рассеяния дает представление о корреляции между данными. По предоставленным графикам мы можем сделать вывод, что линейная зависимость между переменными выражена слабо.

2. Практическая часть

2.1 Предобработка данных

Предварительная обработка данных является важным шагом в процессе интеллектуального анализа данных целью которой является обеспечение корректной работы моделей машинного обучения.

После проведения разведочного анализа предоставленных данных было установлено:

1) наличие некоторого количества выбросов;
2) наличие категориального признака «Угол нашивки, град».

Для удаления выбросов принято решение использовать правило 3-х сигм: отклонение значения нормально распределённой случайной величины X от её математического ожидания M(x) не превосходит утроенного среднеквадратического отклонения σ с вероятностью около 0,9973. Иначе говоря, с вероятностью 0,9973 значение нормально распределённой случайной величины X находится в интервале [M(x) - 3σ ... M(x) + 3σ], где σ - среднеквадратическое отклонение случайной величины. В результате операции осталось 1 003 строки.

С помощью метода LabelEncoder() обрабатываем категориальный признак «Угол нашивки, град». Значения преобразуются в 0 и 1.

Поскольку значения остальных признаков изменяются в достаточно большом диапазоне, проведена нормализация данных с помощью метода MinMaxScaler() – приведение всех значений признаков к новому диапазону от 0 до 1. То есть проведено линейное преобразование данных в диапазоне от 0 до 1, где минимальное и максимальное масштабируемые значения соответствуют 0 и 1 соответственно.

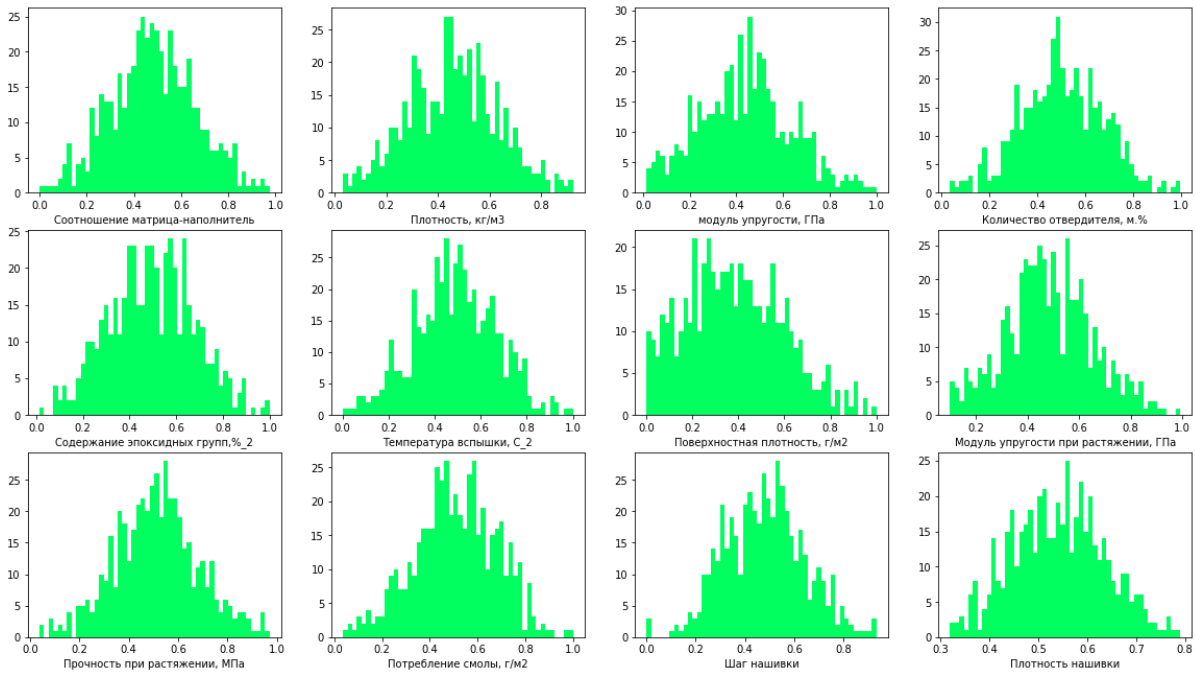
Необходимость нормализации выборок данных обусловлена природой используемых алгоритмов и моделей машинного обучения: разность между

значениями признаков может ухудшить работы модели, как следствие ухудшить результаты обучения и замедлить процесс моделирования.

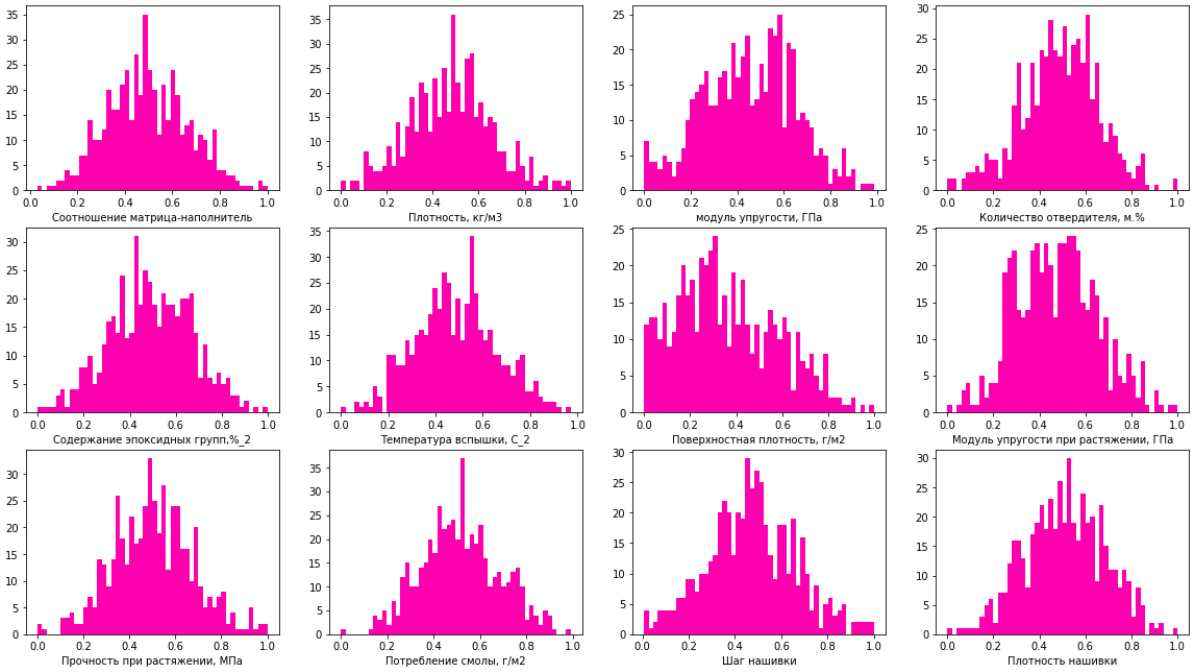
После нормализации и категоризации все числовые значения входных признаков приведены к одинаковой области их изменения – что позволяет свести обеспечит корректную работу вычислительных алгоритмов.

Поскольку нормализация и категоризация данных подразумевает изменение диапазонов, без изменения формы распределения, гистограммы распределения каждого из признаков не меняются:

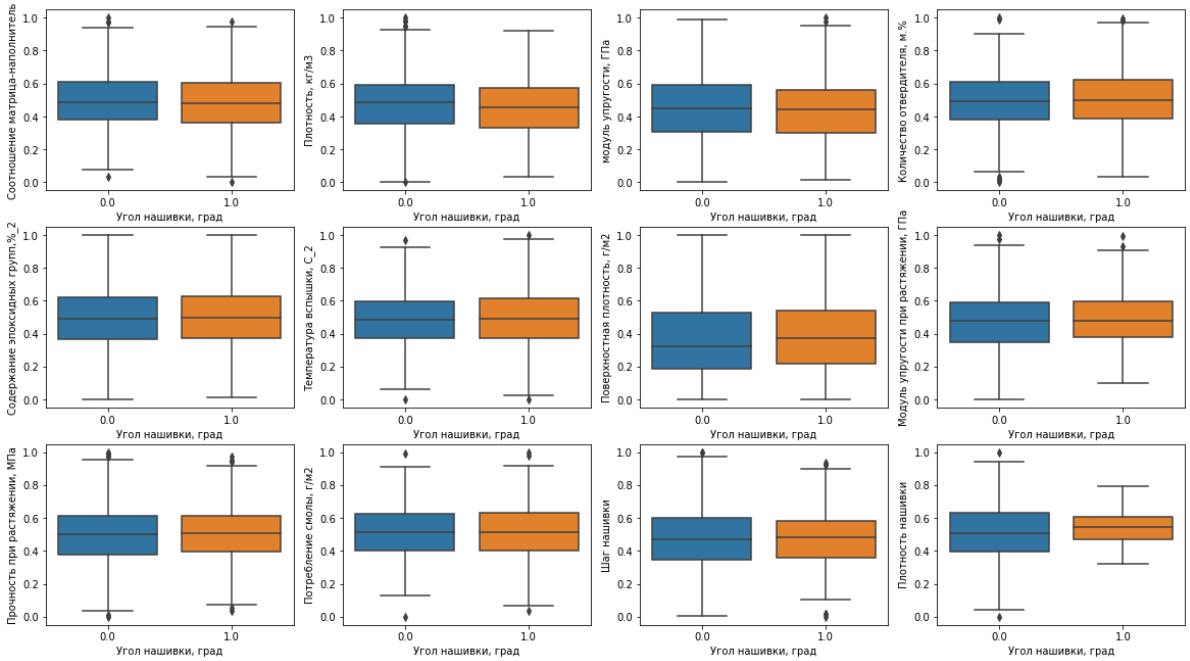
Гистограмма распределения нормализованных данных при угле нашивки 90 градусов:



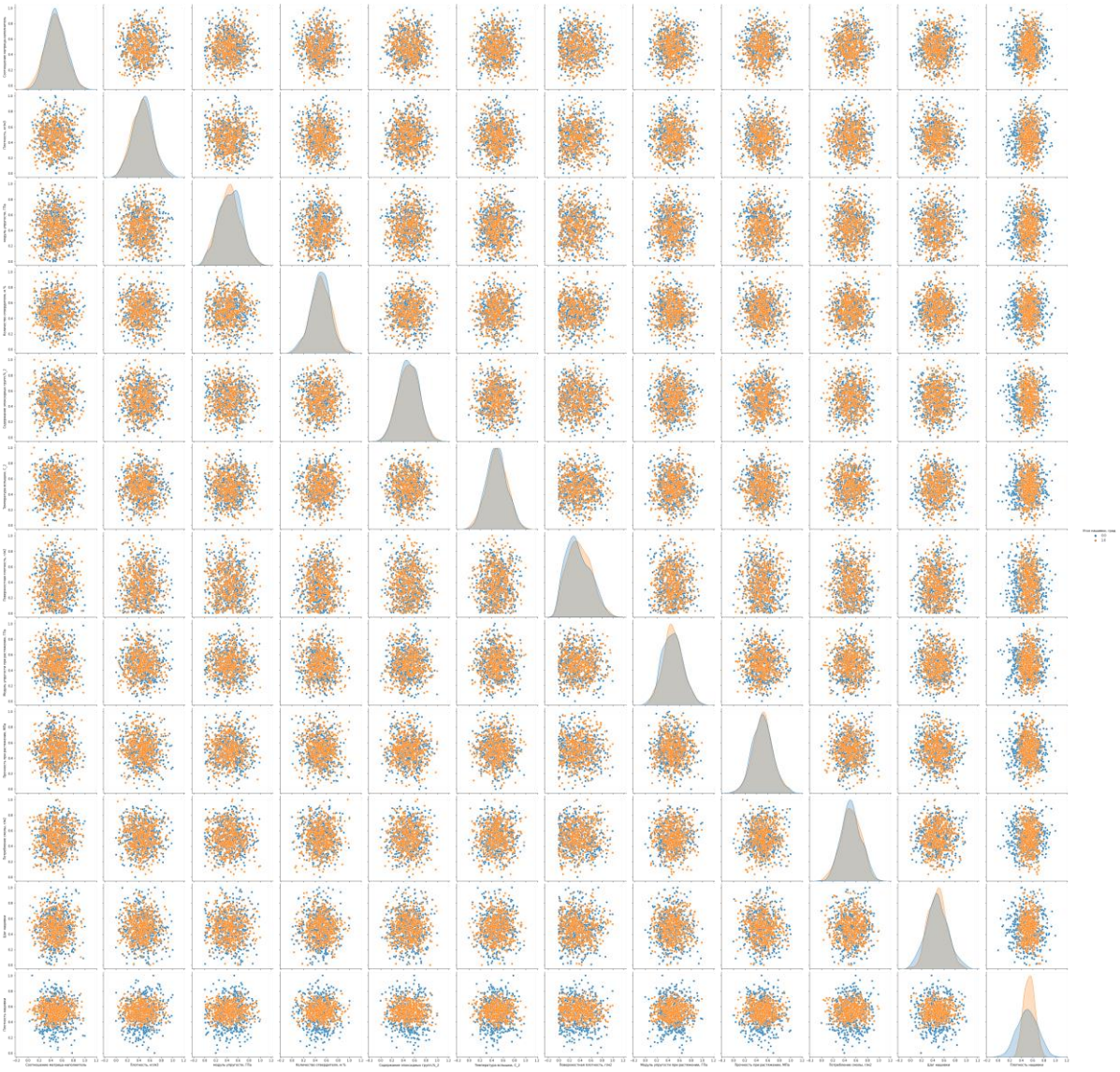
Гистограмма распределения нормализованных данных при угле нашивки 0 градусов:



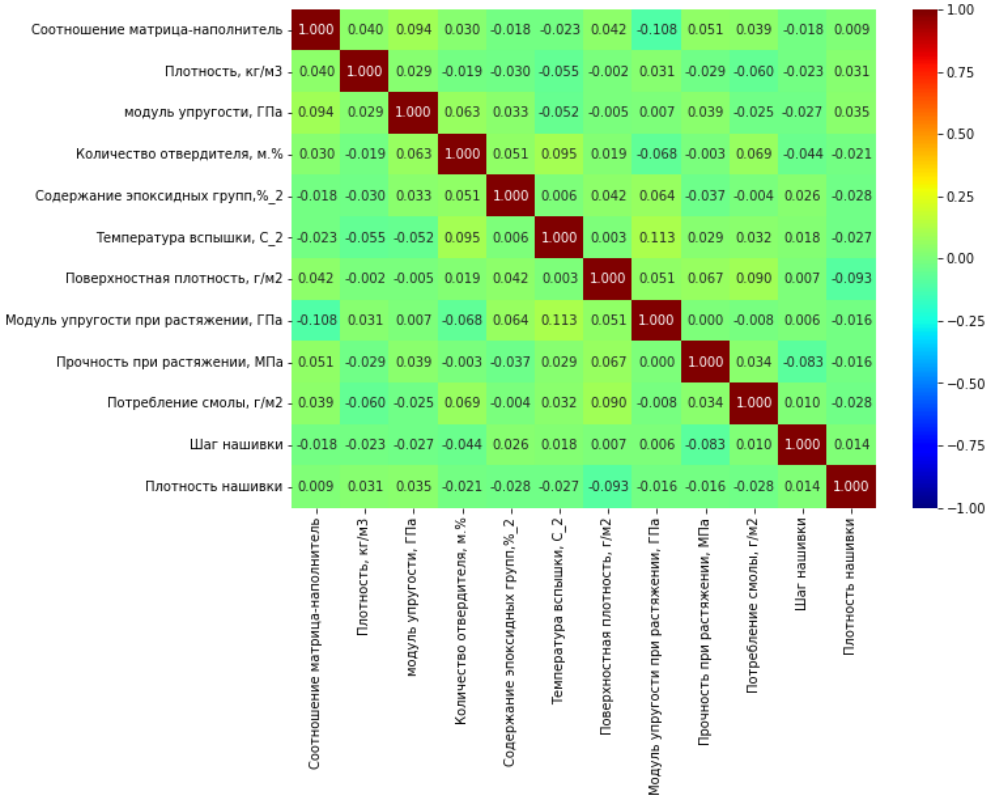
«Ящик с усами» нормализованных данных для каждой категории:



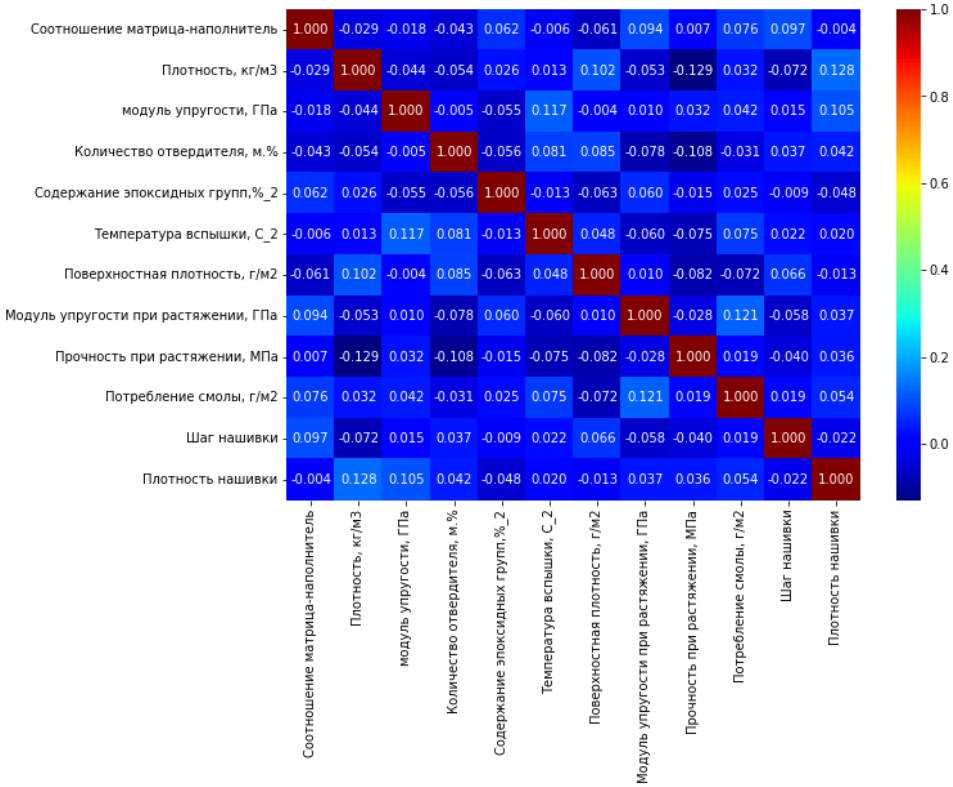
Попарная диаграмма рассеяния нормализованных данных для каждой категории:



Дополнительно для предобработанных данных построим тепловую карту корреляции. Корреляционная матрица для угла нашивки 90 градусов:



Корреляционная матрица для угла нашивки 0 градусов:



Согласно матрице корреляции, коэффициенты корреляции близки к нулю, что означает отсутствие линейной зависимости между признаками. Можно предположить, что качество прогноза линейных моделей будет невысоким.

2.2 Разработка и обучение модели

Для прогноза модуля упругости при растяжении и прочности при растяжении использованы модели:

1) линейная регрессия (метод наименьших квадратов)
2) градиентный бустинг
3) случайный лес
4) полиномиальная регрессия

В соответствии с постановкой задачи при построении модели необходимо разделить данные в соотношении 70/30 на тренировочные/тестовые соответственно. Разделение выборки произведено с помощью метода train_test_split().

2.3 Тестирование модели

В ходе проведенного тестирования моделей, построенных и примененных для прогнозирования модуля упругости при растяжении и модуля прочности при растяжении, рассчитаны и сопоставлены показатели средней квадратической ошибки (MSE) и коэффициент детерминации (R^2).

Средняя квадратическая ошибка (MSE) определяется уравнением:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2,$$

где y_i фактический ожидаемый результат и $\hat{y}_i$ это прогноз модели. MSE в основном измеряет среднеквадратичную ошибку наших прогнозов. Для каждой точки вычисляется квадратная разница между прогнозами и целью, а затем усредняются эти значения. Чем выше это значение, тем хуже модель. Он никогда не бывает отрицательным, поскольку мы возводим в квадрат отдельные ошибки

прогнозирования, прежде чем их суммировать, но для идеальной модели это будет ноль.

Коэффициент детерминации (R^2) является еще одним показателем, который мы можем использовать для оценки модели, и он тесно связан с MSE, но имеет преимущество в том, что безмасштабное – не имеет значения, являются ли выходные значения очень большими или очень маленькими, R^2 всегда будет между -∞ и 1. Когда R^2 отрицательно, это означает, что модель хуже, чем предсказание среднего значения.

$$R^2 = 1 - \frac{\text{MSE(model)}}{\text{MSE(baseline)}}$$

MSE модели рассчитывается, как указано выше, в то время как MSE базовой линии определяется как:

$$\text{MSE(baseline)} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2,$$

где у с чертой означает среднее из наблюдаемого y_i. R^2 - это соотношение между тем, насколько хороша наша модель, и тем, насколько хороша модель наивного среднего.

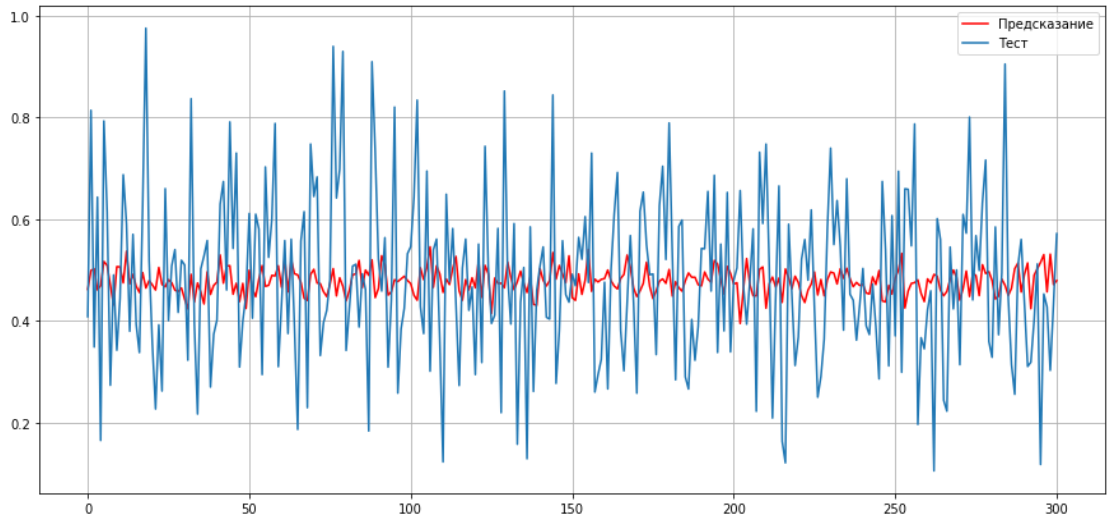
При построении моделей был проведен поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой – GridSearchCV(), с количеством блоков равным 10.

Результаты подбора гиперпараметров модели линейной регрессии для прогноза модуля упругости при растяжении:

{'copy_X': True,

'fit_intercept': True}

Результаты применения модели линейной регрессии для прогноза модуля упругости при растяжении:



Оценка эффективности модели:

1) Средняя квадратичная ошибка (MSE) - 0.027345286816019446

2) Коэффициент детерминации (R2) - 0.00031270815893347237

Результаты подбора гиперпараметров модели градиентного бустинга для прогноза модуля упругости при растяжении:

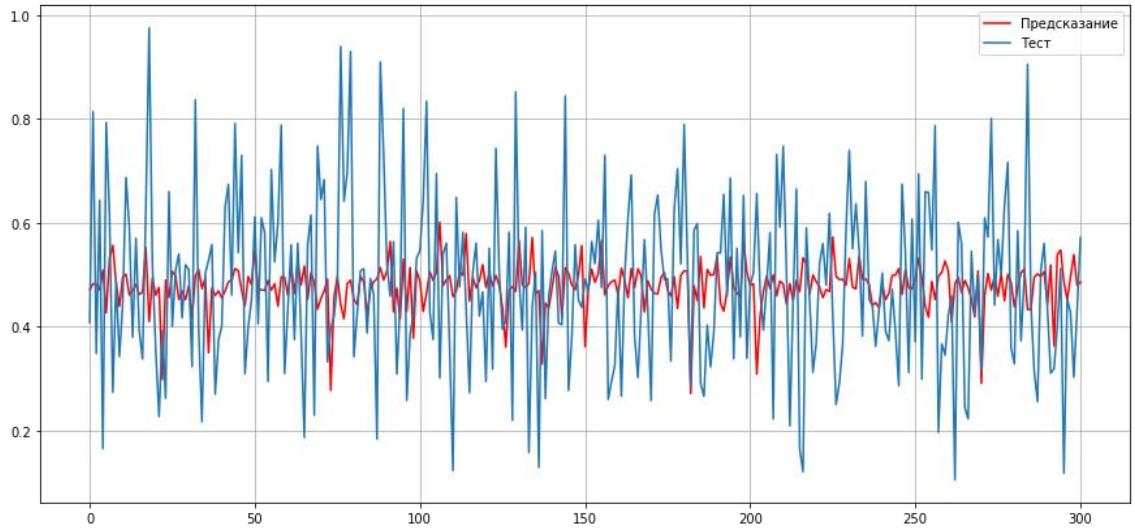
{'criterion': 'squared_error',

'loss': 'absolute_error',

'max_features': 'sqrt',

'n_estimators': 100}

Результаты применения модели градиентного бустинга для прогноза модуля упругости при растяжении:



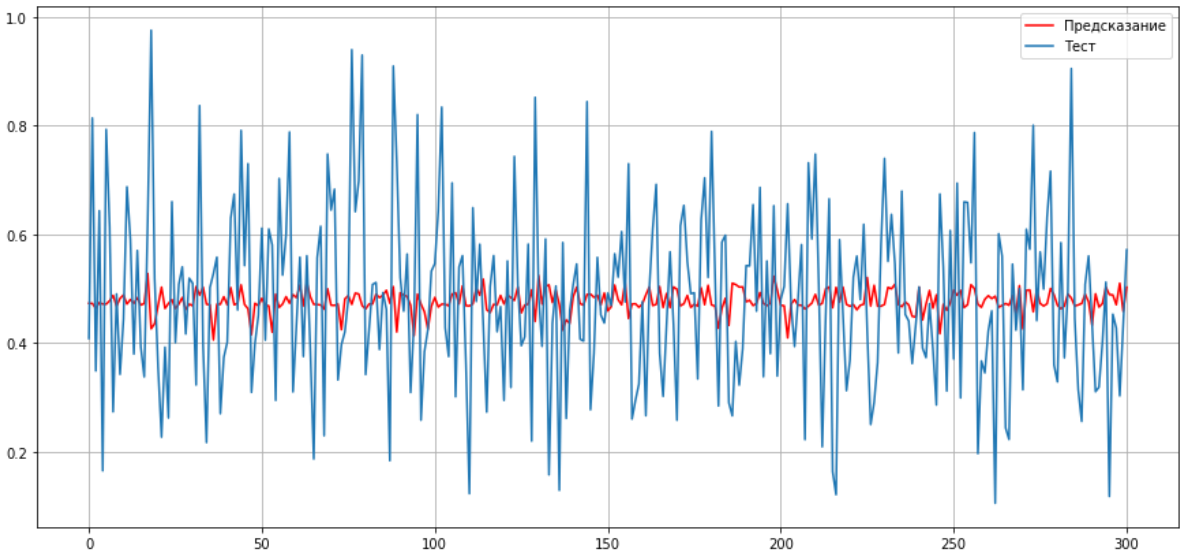
Оценка эффективности модели:

1) Средняя квадратичная ошибка (MSE) - 0.029467297660744712
2) Коэффициент детерминации (R2) - -0.07726363210377118

Результаты подбора гиперпараметров модели случайного леса для прогноза модуля упругости при растяжении:

{'bootstrap': False,
'criterion': 'squared_error',
'max_depth': 3,
'max_features': 'sqrt',
'n_estimators': 150}

Результаты применения модели случайного леса для прогноза модуля упругости при растяжении:



Оценка эффективности модели:

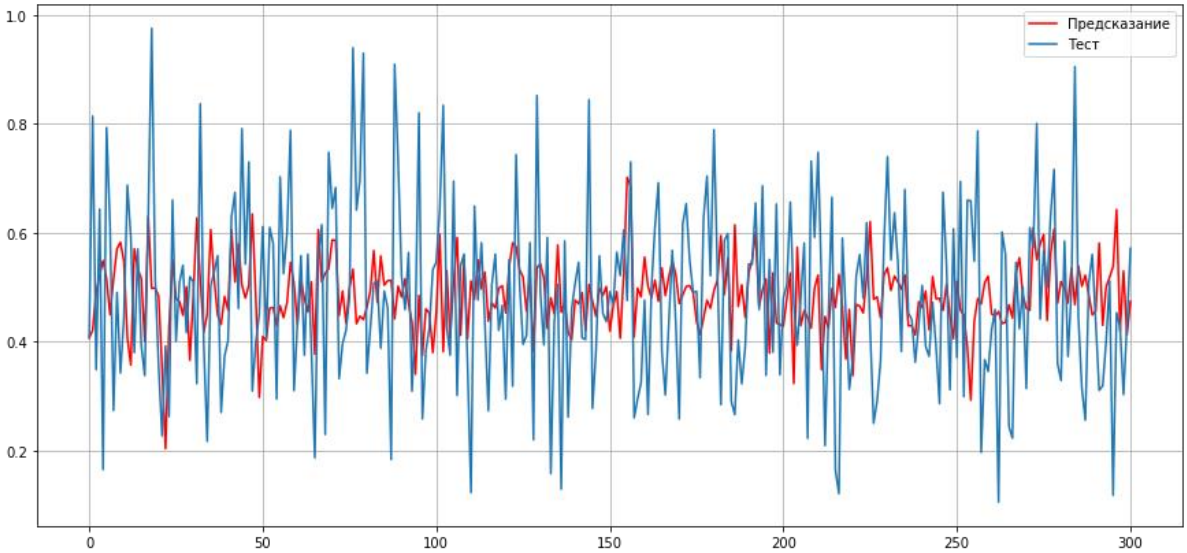
1) Средняя квадратичная ошибка (MSE) - 0.027802607282353512
2) Коэффициент детерминации (R2) - -0.016405984958784092

Результаты подбора гиперпараметров модели полиномиальной регрессии для прогноза модуля упругости при растяжении:

{'copy_X': True,

'fit_intercept': True}

Результаты применения модели полиномиальной регрессии (преобразование данных в функции взаимодействия) для прогноза модуля упругости при растяжении:



Оценка эффективности модели:

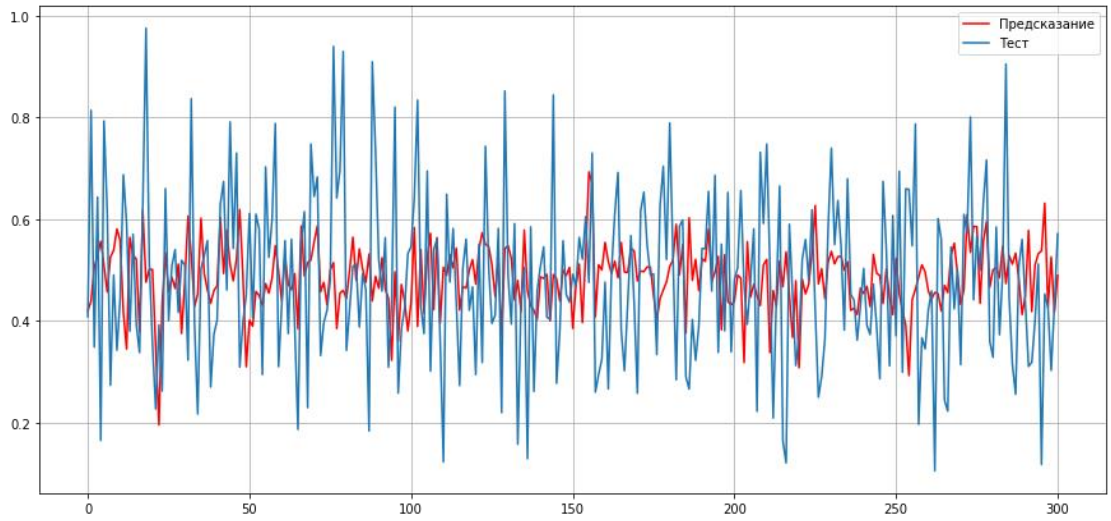
1) Средняя квадратичная ошибка (MSE) - 0.029053128732401397
2) Коэффициент детерминации (R2) - -0.06212247022362982

Результаты подбора гиперпараметров модели полиномиальной регрессии для прогноза модуля упругости при растяжении:

{'copy_X': True,

'fit_intercept': True}

Результаты применения модели полиномиальной регрессии (преобразование данный во 2-ю степень) для прогноза модуля упругости при растяжении:



Оценка эффективности модели:

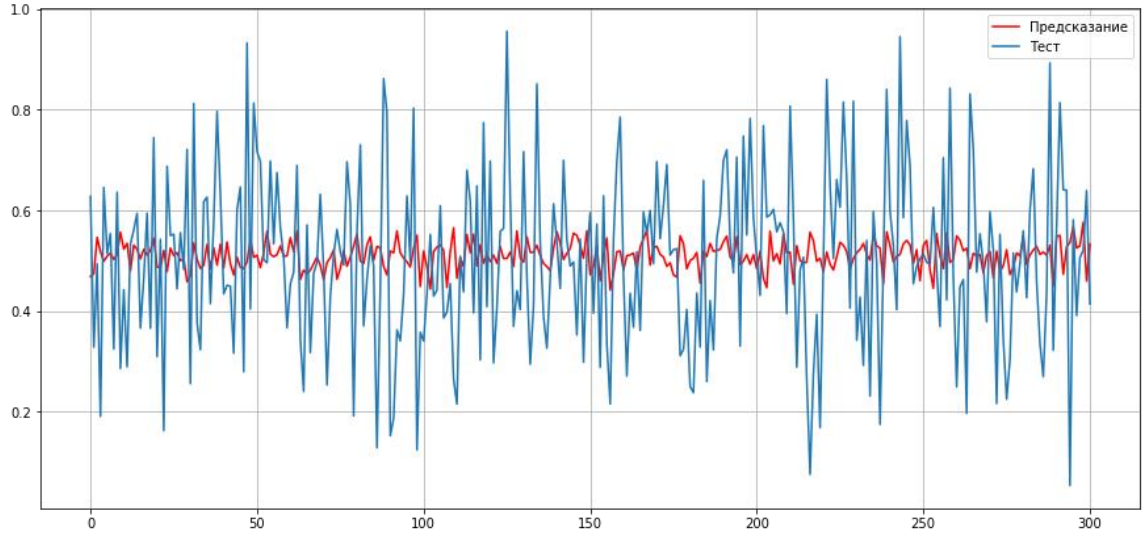
1) Средняя квадратичная ошибка (MSE) - 0.029346061183560812
2) Коэффициент детерминации (R2) - -0.07283147652377098

Результаты подбора гиперпараметров модели полиномиальной регрессии для прогноза прочности при растяжении:

{'copy_X': True,

'fit_intercept': True}

Результаты применения модели линейной регрессии для прогноза прочности при растяжении:



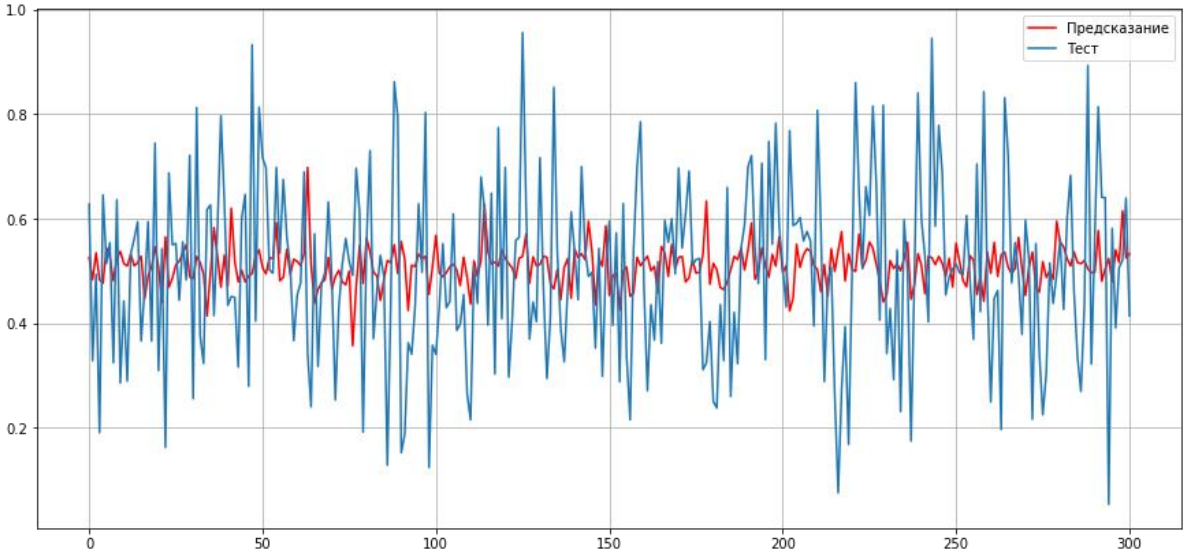
Оценка эффективности модели:

1) Средняя квадратичная ошибка (MSE) - 0.029649401268432553
2) Коэффициент детерминации (R2) - -0.029467652046270665

Результаты подбора гиперпараметров модели градиентного бустинга для прогноза прочности при растяжении:

{'criterion': 'friedman_mse',
'loss': 'absolute_error',
'max_features': 'log2',
'n_estimators': 100}

Результаты применения модели градиентного бустинга для прогноза прочности при растяжении:



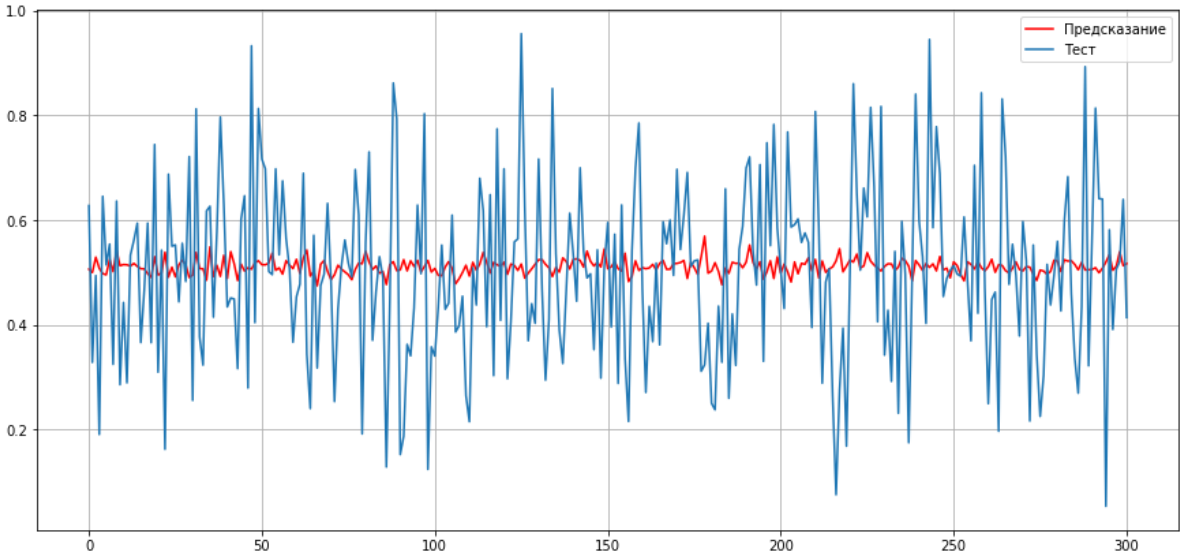
Оценка эффективности модели:

1) Средняя квадратичная ошибка (MSE) - 0.02983422117720099
2) Коэффициент детерминации (R2) - -0.035884851361989156

Результаты подбора гиперпараметров модели случайного леса для прогноза прочности при растяжении:

{'bootstrap': True,
'max_depth': 3,
'max_features': 'log2',
'n_estimators': 150}

Результаты применения модели случайного леса для прогноза прочности при растяжении:



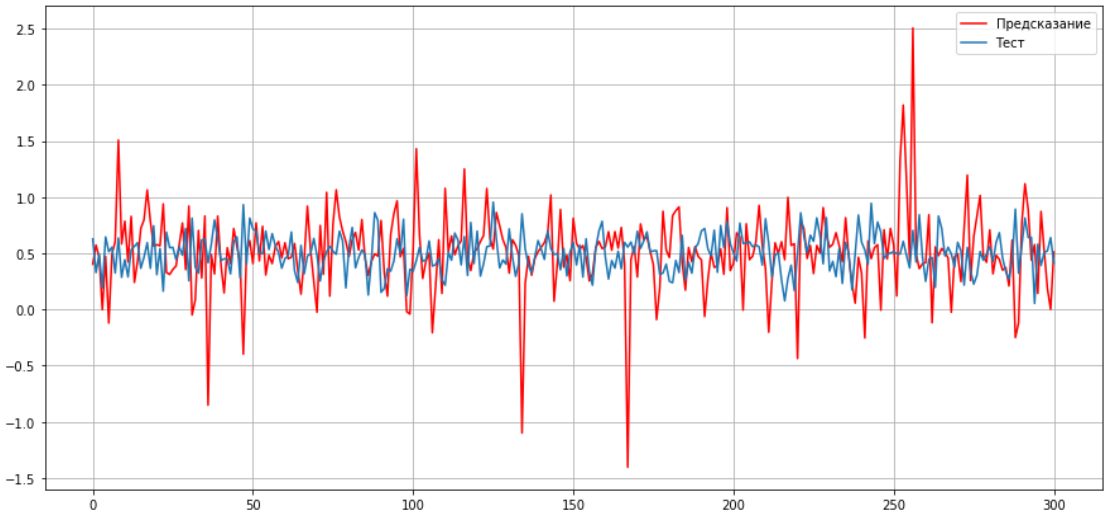
Оценка эффективности модели:

1) Средняя квадратичная ошибка (MSE) - 0.028720337795920776
2) Коэффициент детерминации (R2) - 0.0027906989062314036

Результаты подбора гиперпараметров модели полиномиальной регрессии для прогноза прочности при растяжении:

{'copy_X': True,
'fit_intercept': True}

Результаты применения модели полиномиальной регрессии (преобразование данных в 3-ю степень) для прогноза прочности при растяжении:



Оценка эффективности модели:

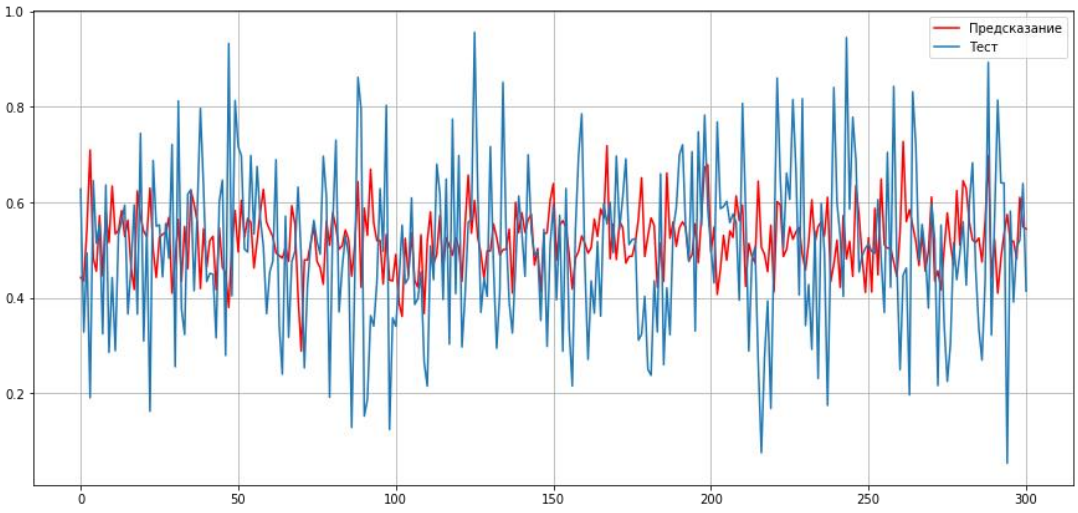
1) Средняя квадратичная ошибка (MSE) - 0.15914794481839756
2) Коэфициент детерминации (R2) - -4.525833712352958

Результаты подбора гиперпараметров модели полиномиальной регрессии для прогноза прочности при растяжении:

{'copy_X': True,

'fit_intercept': True}

Результаты применения модели полиномиальной регрессии (преобразование данных во 2-ю степень) для прогноза прочности при растяжении:



Оценка эффективности модели:

1) Средняя квадратичная ошибка (MSE) - 0.03298482318652948
2) Коэфициент детерминации (R2) - -0.14527805035818364

Результаты тестирования моделей представлен в таблице:

Модель	Прогноз модуля упругости		Прогноз прочности при растяжении	
	MSE	R^2	MSE	R^2
1	2	3	4	5
Линейная регрессия	0.027345286816019446	0.00031270815893347237	0.029649401268432553	-0.029467652046270665
Градиентный бустинг	0.029467297660744712	-0.07726363210377118	0.02983422117720099	-0.035884851361989156
Случайный лес	0.027802607282353512	-0.016405984958784092	0.028720337795920776	0.0027906989062314036
Полиномиальная регрессия (преобразование в функции взаимодействия)	0.029053128732401397	-0.06212247022362982		
Полиномиальная регрессия (преобразование во 2-ю степень)	0.029346061183560812	-0.07283147652377098	0.03298482318652948	-0.14527805035818364
Полиномиальная регрессия (преобразование в 3-ю степень)			0.15914794481839756	-4.525833712352958

Как видно из таблицы, построенные модели неэффективны для решения поставленных задач. А также оказались хуже, чем предсказание среднего значения. Также хочу заметить, что подбор параметров для каждой модели улучшил каждую из них, хоть и предсказания далеки от идеала.

2.4 Построение нейронной сети

В соответствии с заданием, с помощью модели keras.Sequential, построена нейронная сеть для расчета рекомендованных соотношений «матрица-наполнитель» со следующими параметрами:

1) входной слой нормализации 12 признаков;
2) выходной слой для 1 признака;
3) скрытых слоев: 2;
4) нейронов в скрытых слоях: 64 в каждом слое;
5) активационная функция скрытых слоев: relu;
6) оптимизатор: Adam;
7) loss-функция: MeanAbsoluteError.

Структура этой модели, выведенная с помощью метода summary():

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 normalization (Normalizatio  (None, 12)               25
 n)

 dense (Dense)               (None, 64)                832

 dense_1 (Dense)             (None, 64)                4160

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 5,082
Trainable params: 5,057
Non-trainable params: 25
_________________________________________________________________
```

Обучение нейросети происходит со следующими параметрами:

- разбиения данных на тестовые и валидационные: 30/70;
- количество эпох: 100.

График ошибки обучения сети:

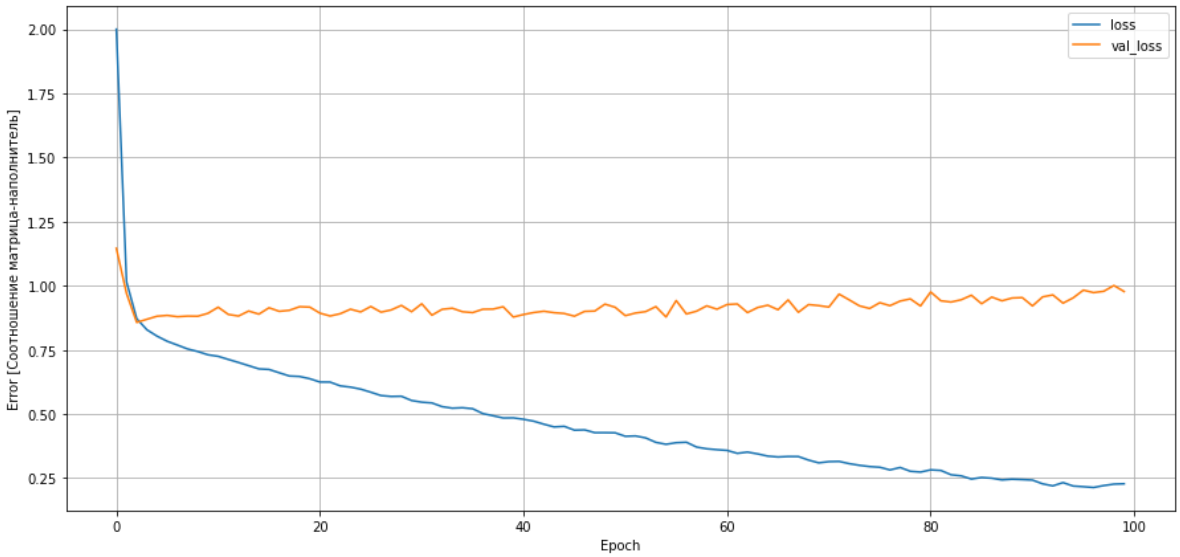
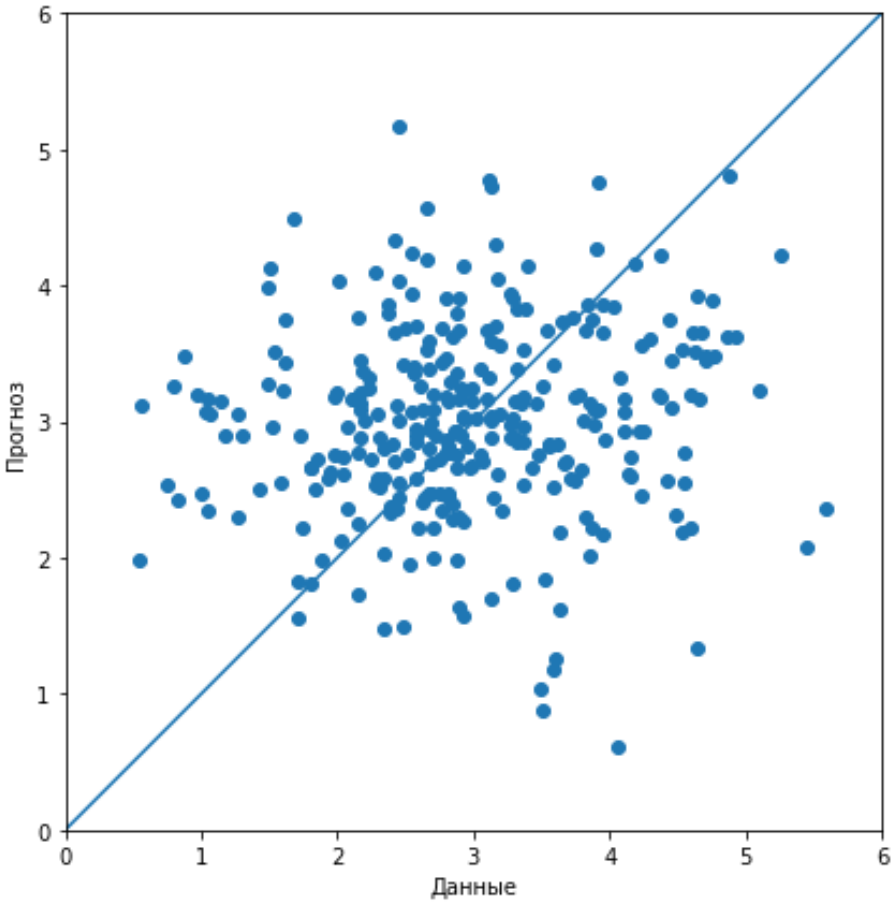


График сравнения прогноза с данными валидации:



Визуализация результатов показывает, что построенная нейросеть из библиотеки tensorflow неэффективна, и не решает поставленную задачу.

2.5 Разработка приложения

Построенные на предыдущих этапах работы с данными модели не позволяют решить поставленные задачи, поэтому разработка приложения нецелесообразна.

Однако, несмотря на то, что пригодных к внедрению моделей получить не удалось, можно разработать тестовое веб-приложение.

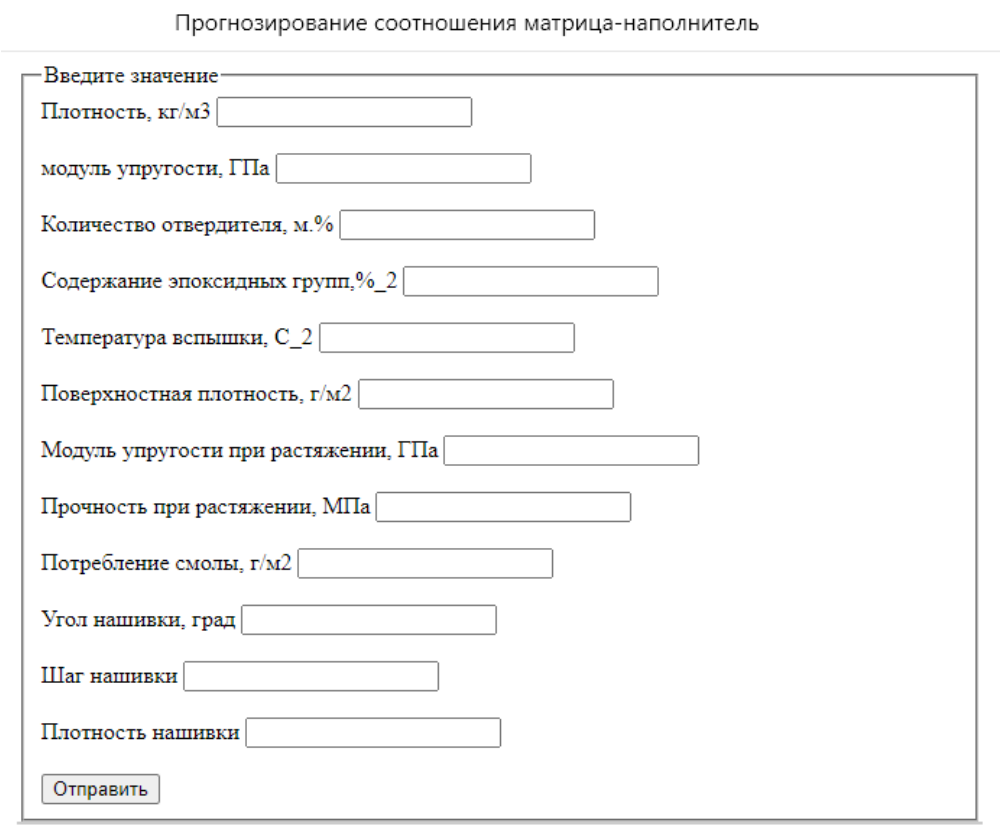
В приложении необходимо реализовать следующие функции:

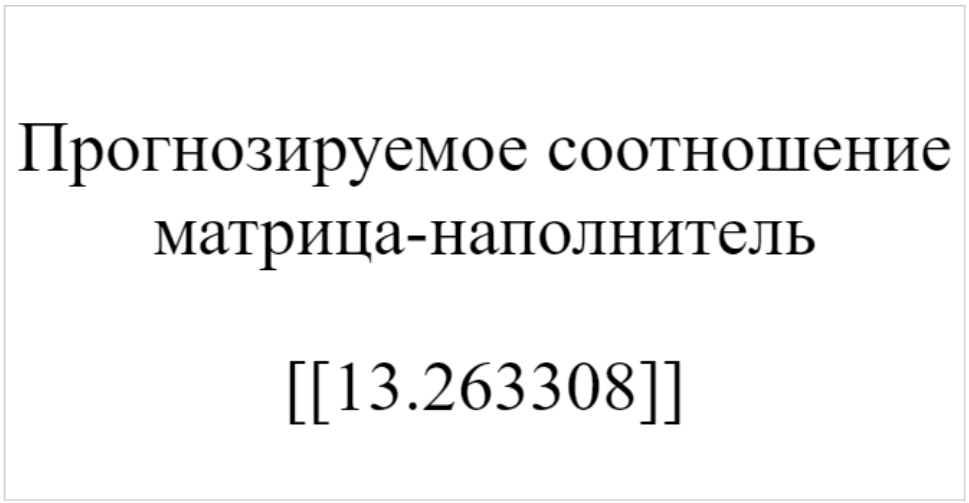
- ввод входных параметров;

- получение и отображение прогноза выходных параметров на основе построенной нейронной сети.

Для разработки веб-приложения использовался микрофреймворк Flask.

Скриншот веб-приложения для прогноза соотношения матрица-наполнитель:





2.6 Создание удаленного репозитория

В соответствии с поставленной задачей был создан удаленный репозиторий на GitHub, который находится по адресу https://github.com/usdocs/bmstu_ds. В него загружены результаты работы: исследовательский notebook, код веб-приложения, пояснительная записка и презентация.

Заключение

Задача работы основана на реальных производственных задачах Центра НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана).

Композиционные материалы все более востребованы в различных сегментах рынка поэтому решение задач прогнозирования характеристик композита на основе известных характеристик исходных компонентов, является актуальным.

Создание качественной прогнозной модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

В ходе выполнения данной работы было:

- изучены теоретические методы анализа данных и машинного обучения;
- проведен разведочный анализ данных;
- произведена предобработка данных;
- построены регрессионные модели;
- подобраны гиперпараметры к каждой модели;
- визуализированы модели и оценки качества прогноза;
- сохранена прогнозная модель;
- разработано и протестировано веб-приложение.

В рамках проведенного в настоящей работе исследования создание качественной прогнозной модели на основе предоставленных данных не выполнено.

На основании проведенного исследования можно сделать следующие основные выводы:

- распределение полученных данных близко к нормальному;

- коэффициенты корреляции между парами признаков стремятся к нулю;
- примененные модели линейной регрессии, градиентного бустинга, случайного леса, полиномиальной регрессии и нейронной сети не показали высокой эффективности в прогнозировании свойств композитов;
- все модели менее эффективны, чем предсказание среднего значения;
- необходимы дополнительные вводные данные для улучшения моделей.

Исходя из основных выводов, требуется вернуться на этап сбора данных, более тщательно оценить точность и качество входных данных, кол-во входных параметров и исследовать их содержательно, в том числе выявив наиболее важные переменные, после чего разработать новые гипотезы, выбрать и построить оптимальные модели.

Список литературы

1. Андерсон, Карл Аналитическая культура. От сбора данных до бизнес-результатов / Карл Андерсон ; пер. с англ. Юлии Константиновой ; [науч. ред. Руслан Салахиев]. — М. : Манн, Иванов и Фербер, 2017. — 336 с;
2. Билл Любанович. Простой Python. Современный стиль программирования. — СПб.: Питер, 2016. — 480 с.: ил. — (Серия «Бестселлеры O’Reilly»);
3. Аллен Б. Дауни – Основы Python. Научитесь думать как программист / Аллен Б. Дауни ; пер. с англ. С. Черникова ; [науч. ред. А. Родионов]. — Москва : Манн, Иванов и Фербер, 2021. — 304 с.;
4. Грас Д. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2021. - 416 с.: ил.;
5. Жерон, Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. Пер. с англ. - СпБ.: ООО ""Альфа-книга': 2018. - 688 с.: ил;
6. Документация по библиотеке numpy: – Режим доступа: https://numpy.org/doc/1.22/user/index.html#user;
7. Документация по библиотеке pandas: – Режим доступа: https://pandas.pydata.org/docs/user_guide/index.html#user-guide;
8. Документация по библиотеке matplotlib: – Режим доступа: https://matplotlib.org/stable/users/index.html;
9. Документация по библиотеке seaborn: – Режим доступа: https://seaborn.pydata.org/tutorial.html;
10. Документация по библиотеке sklearn: – Режим доступа: https://scikit-learn.org/stable/user_guide.html;
11. Документация по библиотеке keras: – Режим доступа: https://keras.io/api/.
12. Руководство по быстрому старту в flask: – Режим доступа: https://flask-russian-docs.readthedocs.io/ru/latest/quickstart.html.