Центр дополнительного образования

МГТУ им. Н.Э. Баумана

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по курсу
«Data Science»

Слушатель Парфенов А.А.

Москва 2023

Содержание:

Оглавление

Введение

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

Композиционными называются материалы, в которых имеет место сочетание двух (или более) химически разнородных компонентов (фаз) с чёткой границей раздела между ними. Это неоднородные по химическому составу и структуре материалы.

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

Композиционные материалы характеризуются совокупностью свойств, не присущих каждому в отдельности взятому компоненту. За счет выбора армирующих элементов, варьирования их объёмной доли в матричном материале, а также размеров, формы, ориентации и прочности связи по границе «матрица- наполнитель», свойства композиционных материалов можно регулировать в значительных пределах.

Композитные материалы применяются во многих областях жизни современного мира человека.

Учитывая такое широкое распространение и высокую потребность в новых материалах, тема данной работы является очень актуальной.

1. Аналитическая часть

1.1. Постановка задачи

В работе исследуется композит с нашивками из углепластика. Предоставлен датасет, содержащий данные о свойствах матрицы и наполнителя, производственных параметрах и свойствах готового композита. От слушателя требуется разработать модели, прогнозирующие значения некоторых свойств в зависимости от остальных. Так же требуется разработать приложение, делающее удобным использование данных моделей специалистом предметной области.

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

По заданию обе таблицы требуется объединить с типом INNER.

После объединения исследуем данные объединенного датасета.

Описание признаков объединенного датасета приведено в таблице. Все признаки имеют тип float64, то есть вещественный. Пропусков в данных нет. Все признаки, кроме «Угол нашивки», являются непрерывными, количественными. «Угол нашивки» принимает только два значения и будет рассматриваться как бинарный признак.

Сначала мы импортировали необходимые библиотеки и загрузи датасет. В данном случае я использовал Google Colab, версия Python 3.9.16, версия TensorFlow 2.11.0.

Рис. 1. Характеристика датасета

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1023 entries, 0 to 1022
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   int64  
 11  Шаг нашивки                           1023 non-null   float64
 12  Плотность нашивки                     1023 non-null   float64
dtypes: float64(12), int64(1)
memory usage: 111.9 KB
```

Количество уникальных значений (кроме «Угол нашивки») довольно велико, пропусков нет, дублей нет, что говорит о высоком качестве датасета.

1.2. Описание используемых методов

Метод k-ближайших соседей

Метод k-ближайших соседей (k-nearest neighbors algorithm) — метрический алгоритм для автоматической классификации объектов или регрессии. В случае использования метода для классификации объект присваивается тому классу, который является наиболее распространённым среди соседей данного элемента, классы которых уже известны. В случае использования метода для регрессии, объекту присваивается среднее значение по ближайшим к нему объектам, значения которых уже известны.

Линейная регрессия

Линейная регрессия (англ. Linear regression) — используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) x с линейной функцией зависимости. Проще говоря, линейная регрессия – это статистический тест, применяемый к набору данных для определения и количественной оценки

взаимосвязи между рассматриваемыми переменными. Он прост в использовании и до сих пор считается одним из самых мощных алгоритмов. Использование алгоритма линейной регрессии важно по следующим причинам:

- Описание: помогает проанализировать силу связи между результатом (зависимой переменной) и переменными-предикторами.
- Корректировка: регулирует влияние ковариата или искажающих факторов.
- Предикторы: помогает оценить важные факторы риска, влияющие на зависимую переменную.
- Степень прогноза: помогает проанализировать величину изменения независимой переменной «единицы», которое может повлиять на зависимую переменную.
- Прогнозирование: помогает количественно оценить новые случаи.

Случайный лес

Random forest (с англ. — «случайный лес») — алгоритм машинного обучения, заключающийся в использовании комитета (ансамбля) решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга, и метод случайных подпространств. Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.

Метод опорных векторов

Метод опорных векторов (с (англ. SVR, support vector regression) —— это контролируемый алгоритм обучения, который используется для прогнозирования дискретных значений. Регрессия опорных векторов использует тот же принцип, что и SVM. Основная идея SVR состоит в том,

чтобы найти наиболее подходящую линию. В SVR наилучшей линией соответствия является гиперплоскость с максимальным количеством точек.

Основная идея метода — перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с наибольшим зазором в этом пространстве. Две параллельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющей классы. Разделяющей гиперплоскостью будет гиперплоскость, создающая наибольшее расстояние до двух параллельных гиперплоскостей. Алгоритм основан на допущении, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора.

1.3. Разведочный анализ данных

Разведочный анализ данных (англ. *Exploratory data analysis, EDA*) — анализ основных свойств данных, нахождение в них общих закономерностей, распределений и аномалий, построение начальных моделей, зачастую с использованием инструментов визуализации.

В проекте были использованы следующие методы разведочного анализа данных:

- Визуальный анализ гистограмм
- Визуальный анализ диаграмм размаха («ящик с усами»)
- Проверка нормальности распределения по критерию Пирсона
- Анализ попарных графиков рассеяния переменных
- Корреляционный анализ с целью поиска коэффициентов

Цель разведочного анализа данных — выявить закономерности в данных.

Разведочный анализ данных (Exploratory Data Analysis, EDA) - это общий подход к исследованию наборов данных с помощью простой сводной статистики и графических визуализаций для более глубокого понимания данных. Он помогает в последующем более эффективно анализировать и моделировать данные.

Для получения статистики по набору данных использовались следующие команды библиотеки pandas для работы с данными:

1) df.info() - вывод информации о типах переменных;

2) df.isnull().sum() - вывод информации о количестве пропусков;

3) df.duplicated().sum() - количество полностью совпадающих строк;

4) df.shape - информация о количестве наблюдений и количестве переменных;

5) df.nunique() - количество уникальных значений по каждой переменной;

6) df.describe() - вывод статистик по количественным переменным:

- count - количество значений;
- mean - среднее арифметическое значение;
- std – среднее квадратическое (стандартное) отклонение;
- min - минимальное значение;
- max - максимальное значение;
- 25% - верхнее значение 1-го квартиля;
- 50% - верхнее значение 2-го квартиля (медиана);
- 75% - верхнее значение 3-го квартиля.

Рис. 2. Статистика по количественным переменным.

Метрика	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп, %_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
count	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000
mean	2.930366	1975.734888	739.923233	110.570769	22.244390	285.882151	482.731833	73.328571	2466.922843	218.423144	0.491691	6.899222	57.153929
std	0.913222	73.729231	330.231581	28.295911	2.406301	40.943260	281.314690	3.118983	485.628006	59.735931	0.500175	2.563467	12.350969
min	0.389403	1731.764635	2.436909	17.740275	14.254985	100.000000	0.603740	64.054061	1036.856605	33.803026	0.000000	0.000000	0.000000
25%	2.317887	1924.155467	500.047452	92.443497	20.608034	259.066528	266.816645	71.245018	2135.850448	179.627520	0.000000	5.080033	49.799212
50%	2.906878	1977.621657	739.664328	110.564840	22.230744	285.896812	451.864365	73.268805	2459.524526	219.198882	0.000000	6.916144	57.341920
75%	3.552660	2021.374375	961.812526	129.730366	23.961934	313.002106	693.225017	75.356612	2767.193119	257.481724	1.000000	8.586293	64.944961
max	5.591742	2207.773481	1911.536477	198.953207	33.000000	413.273418	1399.542362	82.682051	3848.436732	414.590628	1.000000	14.440522	103.988901

Сбор статистики показал, что в наборе данных все параметры имеют количественные значения (вещественные числа), параметров с качественными значениями нет. В наборе данных отсутствуют пропуски (нулевые значения) и строки-дубликаты. Также в нем нет бесполезных для анализа данных, т.е. таких параметров, у которых уникальных значений столько же, сколько и наблюдений, а также параметров только с одним уникальным значением (параметры-константы).

Для корректной работы большинства моделей желательна сильная зависимость выходных переменных от входных и отсутствие зависимости между входными переменными. По форме «облаков точек» (Рис. 2) мы не заметили зависимостей, которые станут основой работы моделей. Помочь выявить связь между признаками может матрица корреляции, приведенная на рисунке 3.

Рис. 3. Попарные графики рассеивания.

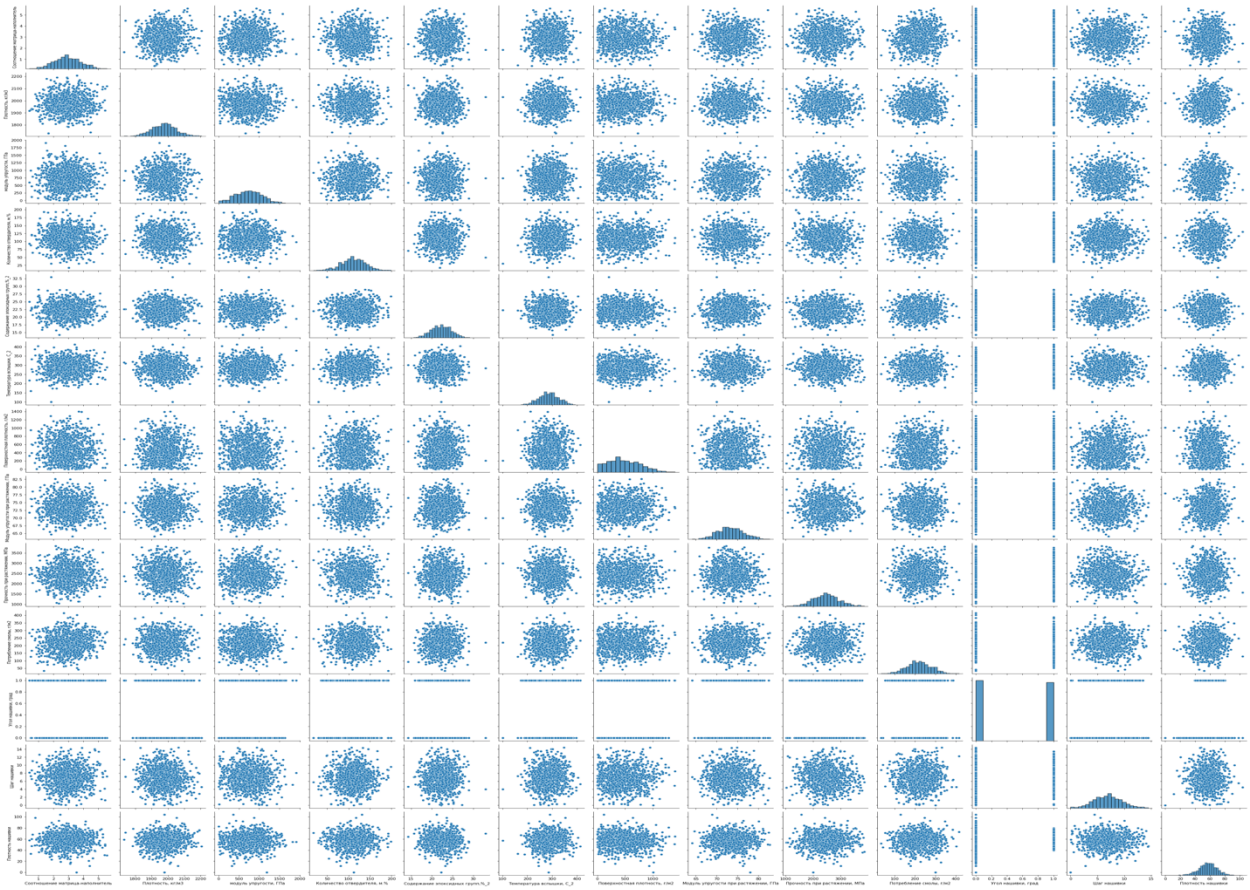
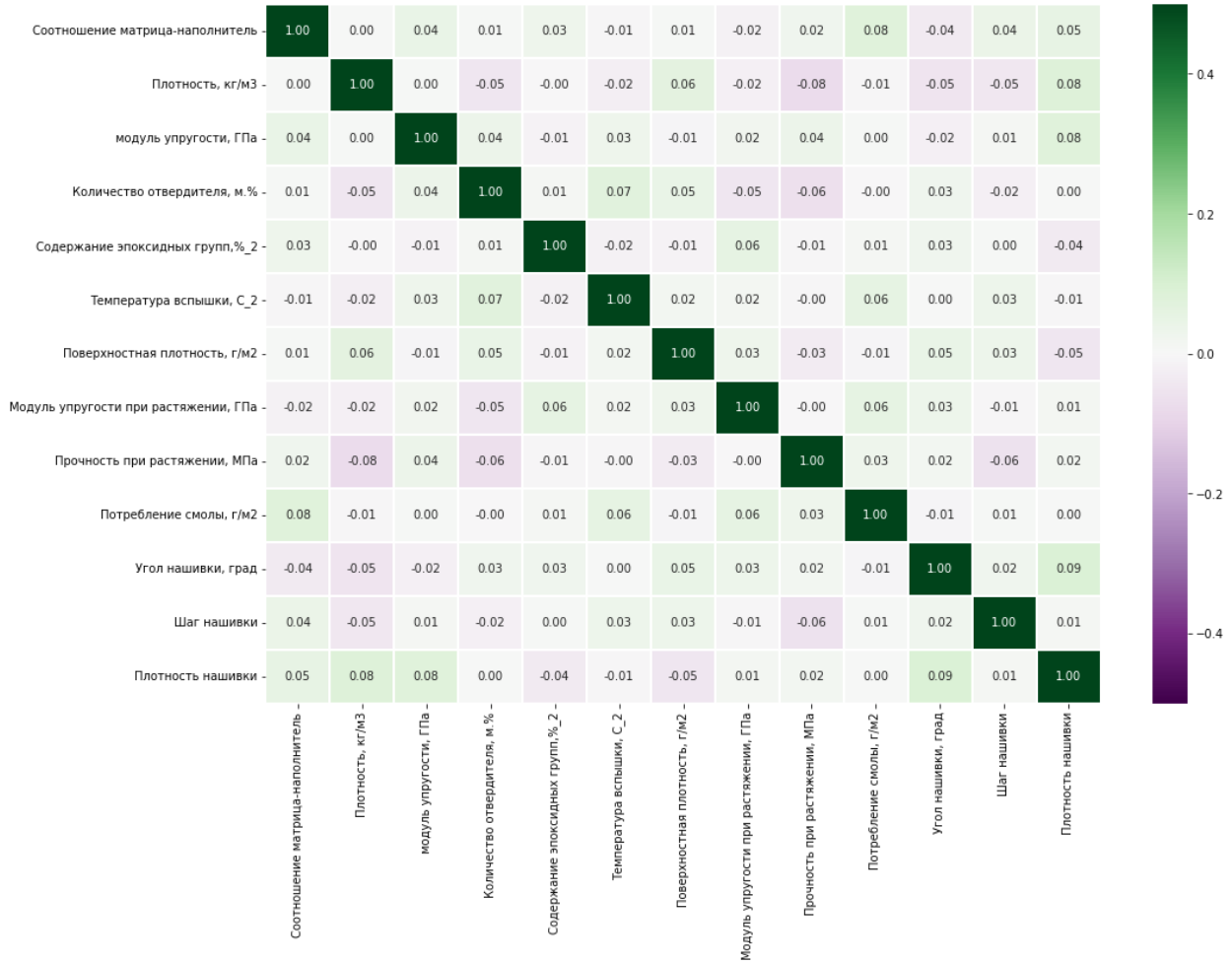


Рис. 4. Корреляционная матрица



По матрице корреляции мы видим, что все коэффициенты корреляции близки к нулю, что означает отсутствие линейной зависимости между признаками.

Также для разведочного анализа данных мы использовали:

- Визуальный анализ гистограмм

- Визуальный анализ boxplot («ящик с усами»)

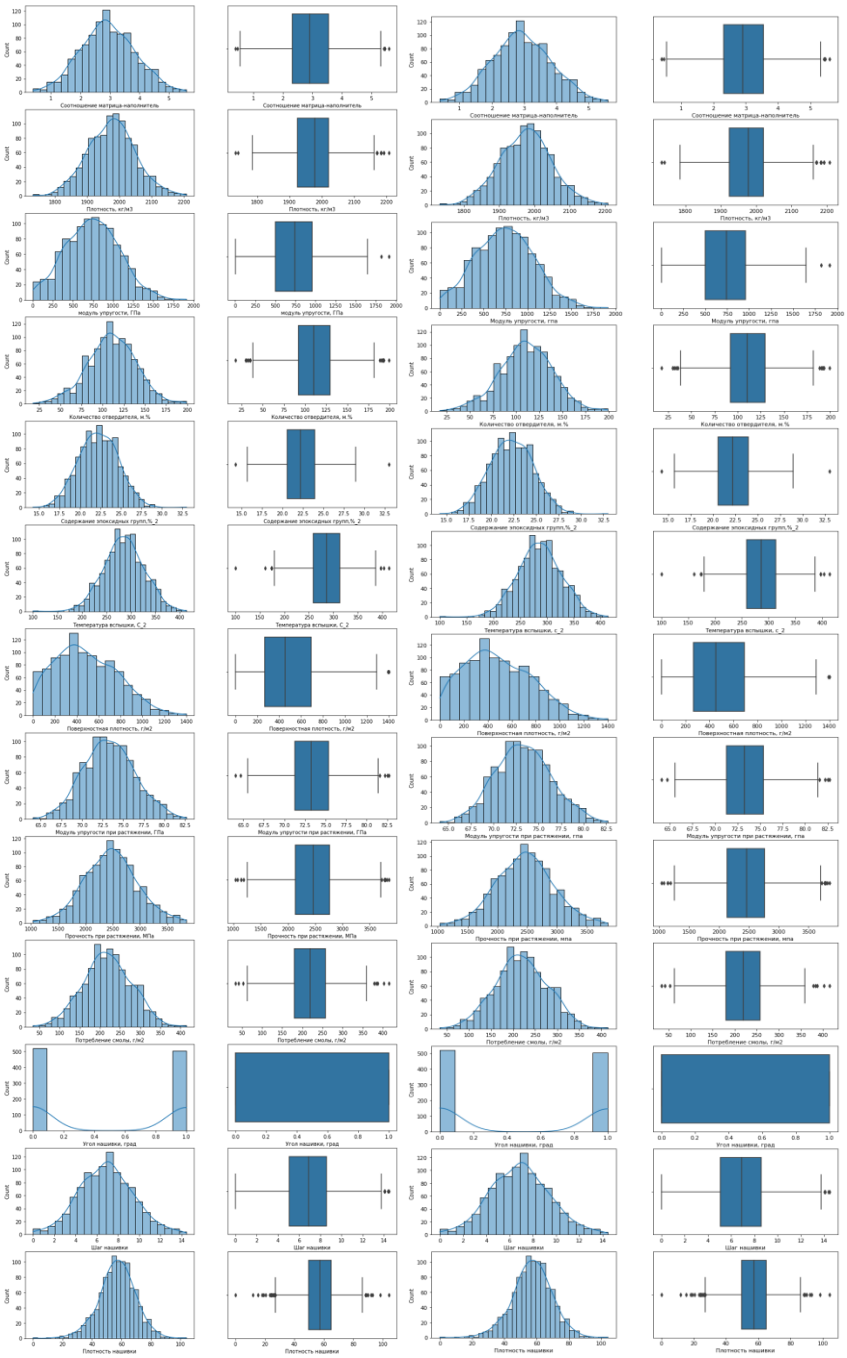
-Анализ попарных графиков рассеяния переменных в привязке к целевой переменной: модулю упругости и модулю прочности

2. Практическая часть

2.1. Предобработка данных

Для каждого параметра были построены графики распределния переменных, ящики с усами и попарные графики рассеивания.

Рис.5. Графики распределния и ящики с усами



На гистограммах можно увидеть, что распределения параметров являются нормальными или близкими к нормальному, кроме Угол нашивки и Поверхностная плотность.

В ходе проведения разведочного анализа с помощью диаграмм размаха было визуально установлено наличие в рабочем наборе данных выбросов. Для их нахождения был использован метод межквартильных интервалов (InterQuartile Range, IQR).

По графикам boxplot видно, что выбросы есть, т.к. некоторые точки стоят очень далеко от усов. Ящик у усами показывает точки, которые явноя являются выбросами. Согласно теоретической части выбросами считаются точки, превышающие 1,5 межквартильного расстояния. Межквартильное расстояние — это разница между 1-м и 3-м квартилями, т.е. между 25-м и 75-м процентилями.

Данные, выходящие за пределы 1,5 межквартильных расстояния, были заменены на пустые значения и посчитаны. С его помощью было установлено, что в рабочем наборе данных имеется 93 выброса, 9% от общего количества данных.

Их кол-во оказалось не большим и поэтому их удаление не повлияет существенно на построение моделей.

Все выбросы были помечены как NaN («не число») и удалены с помощью функции библиотеки pandas dropna().

Рис. 6. Количество выбросов

```
Соотношение матрица-наполнитель          6
Плотность, кг/м3                         9
Модуль упругости, гпа                    2
Количество отвердителя, м.%             14
Содержание эпоксидных групп,%_2          2
Температура вспышки, с_2                 8
Поверхностная плотность, г/м2            2
Модуль упругости при растяжении, гпа     6
Прочность при растяжении, мпа           11
Потребление смолы, г/м2                  8
Угол нашивки, град                       0
Шаг нашивки                              4
Плотность нашивки                        2
```

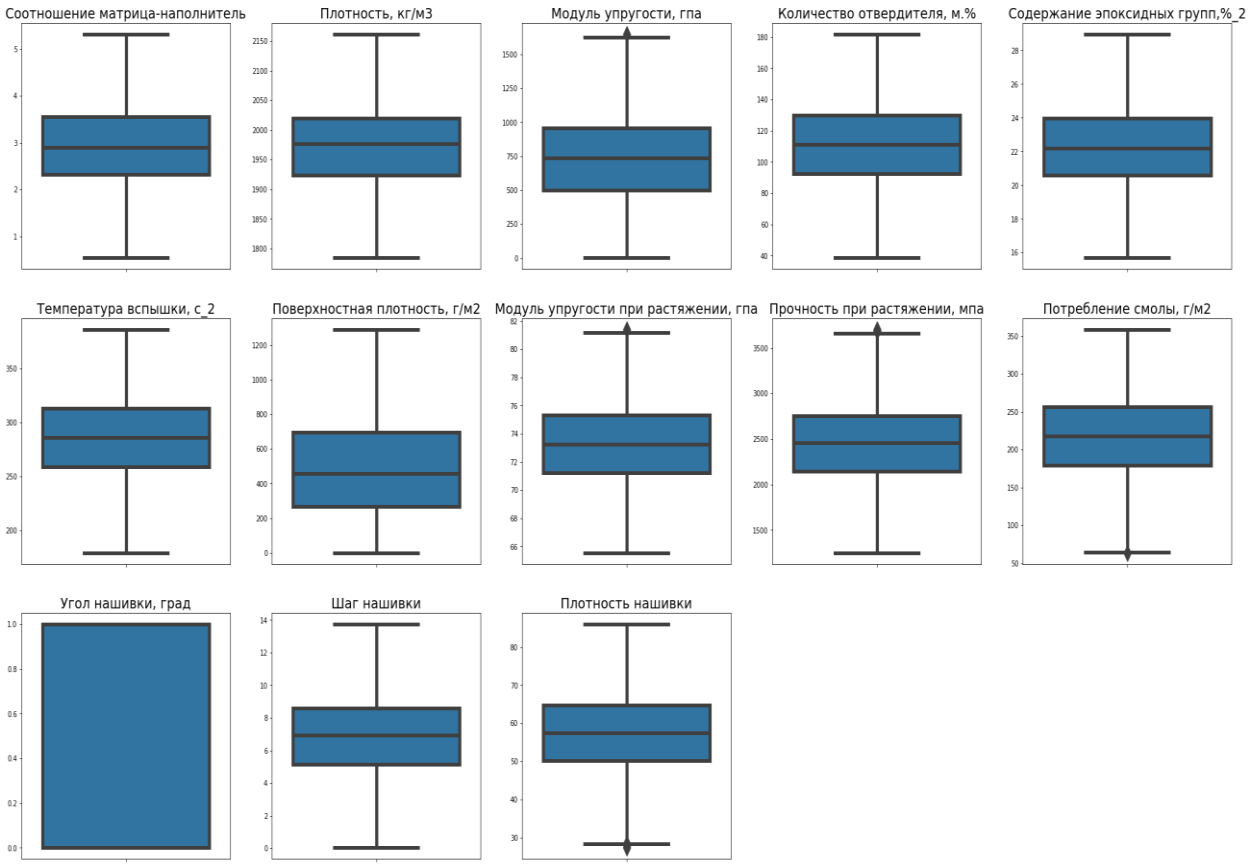
Рис. 7. Датасет после уделния выбросов.

```
Data columns (total 13 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   Соотношение матрица-наполнитель        936 non-null    float64
 1   Плотность, кг/м3                       936 non-null    float64
 2   Модуль упругости, гпа                  936 non-null    float64
 3   Количество отвердителя, м.%            936 non-null    float64
 4   Содержание эпоксидных групп,%_2        936 non-null    float64
 5   Температура вспышки, с_2               936 non-null    float64
 6   Поверхностная плотность, г/м2          936 non-null    float64
 7   Модуль упругости при растяжении, гпа   936 non-null    float64
 8   Прочность при растяжении, мпа          936 non-null    float64
 9   Потребление смолы, г/м2                936 non-null    float64
 10  Угол нашивки, град                     936 non-null    float64
 11  Шаг нашивки                            936 non-null    float64
 12  Плотность нашивки                      936 non-null    float64
```

Видим, что после уделния выбросов количество значений в датасете уменьшилось с 1023 до 936. Приянто решение, что это количество выбросов явдяется незначительным и удаление этих данных не повлияет на дальнейший анализ.

Снова строим ящики с усами и видим, что теперь выбросов нет.

Рис. 8. Boxplot после удаления выбросов.



Также при выполнении разведочного анализа данных было замечено, что значения данных изменяются в очень больших диапазонах и также у разных параметров отличаются на порядки. Это может приводить к некорректной работе моделей машинного обучения – большой дисбаланс между значениями признаков может ухудшать результаты обучения и замедлять сам процесс моделирования. Поэтому данные были нормализованы с использованием метода MinMaxScaler из библиотеки Sklearn. Т.к. в нашем наборе данных нет отрицательных значений, то этот метод отмасштабировал все данные от 0 до 1.

Нормализация данных необходима, т.к. модели машинного обучения не работают с естественными значениями.

Рис. 9. Датасет после нормализации данных.

Индекс	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0	0.274768	0.651097	0.447061	0.079153	0.607435	0.509164	0.16223	0.280303	0.71259	0.529221	0.0	0.289334	0.557156
1	0.274768	0.651097	0.447061	0.630983	0.418887	0.583596	0.16223	0.280303	0.71259	0.529221	0.0	0.362355	0.335840
2	0.466552	0.651097	0.455721	0.511257	0.495653	0.509164	0.16223	0.280303	0.71259	0.529221	0.0	0.362355	0.506083
3	0.465836	0.571539	0.452685	0.511257	0.495653	0.509164	0.16223	0.280303	0.71259	0.529221	0.0	0.362355	0.557156
4	0.424236	0.332865	0.488508	0.511257	0.495653	0.509164	0.16223	0.280303	0.71259	0.529221	0.0	0.362355	0.727399

Рис. 10. Описательная статистика после нормализации

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	936.0	0.50	0.19	0.0	0.37	0.49	0.63	1.0
Плотность, кг/м3	936.0	0.50	0.19	0.0	0.37	0.51	0.62	1.0
модуль упругости, ГПа	936.0	0.45	0.20	0.0	0.30	0.45	0.58	1.0
Количество отвердителя, м.%	936.0	0.50	0.19	0.0	0.38	0.51	0.64	1.0
Содержание эпоксидных групп,%_2	936.0	0.49	0.18	0.0	0.37	0.49	0.62	1.0
Температура вспышки, С_2	936.0	0.52	0.19	0.0	0.39	0.52	0.65	1.0
Поверхностная плотность, г/м2	936.0	0.37	0.22	0.0	0.21	0.35	0.54	1.0
Модуль упругости при растяжении, ГПа	936.0	0.49	0.19	0.0	0.36	0.49	0.62	1.0
Прочность при растяжении, МПа	936.0	0.50	0.19	0.0	0.37	0.49	0.61	1.0
Потребление смолы, г/м2	936.0	0.52	0.20	0.0	0.39	0.52	0.65	1.0
Угол нашивки, град	936.0	0.51	0.50	0.0	0.00	1.00	1.00	1.0
Шаг нашивки	936.0	0.50	0.18	0.0	0.37	0.50	0.62	1.0
Плотность нашивки	936.0	0.51	0.19	0.0	0.39	0.52	0.64	1.0

Рис. 11. Выводим корреляцию после нормализации

Нормализация изменяет только диапазон величин, в пределах которого лежат данные, и не меняет форму распределения внутри этого диапазона.

2.2. Модели

Признаки датасета были разделены на входные и выходные, а строки - на тренировочное и тестовое множество.

Ни одна из выбранных моделей не справилась с задачей для наших данных.

Коэффициент детерминации R2 близок к нулю, а это значит что модель справляется не лучше, чем обычное усреднение. А когда показатель R2 меньше нуля, то значит модель работает хуже базовых моделей.

Средняя абсолютная ошибка MAE также примерно оинакова у каждой модели. Чем ближе MAE к нулю, тем точнее модель. Но MAE возвращается в том же масштабе значений, что и исходные данные.

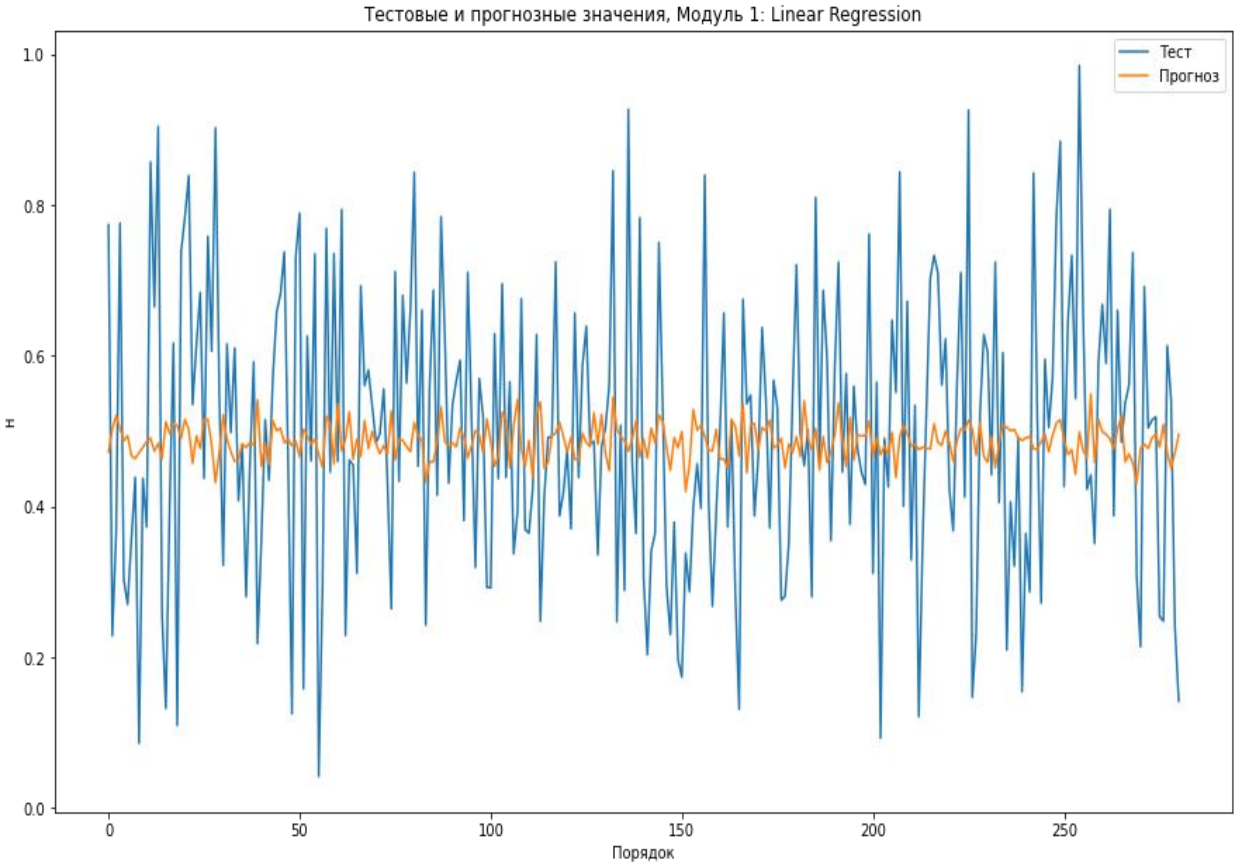
Рис 12. Показатели каждой модели.

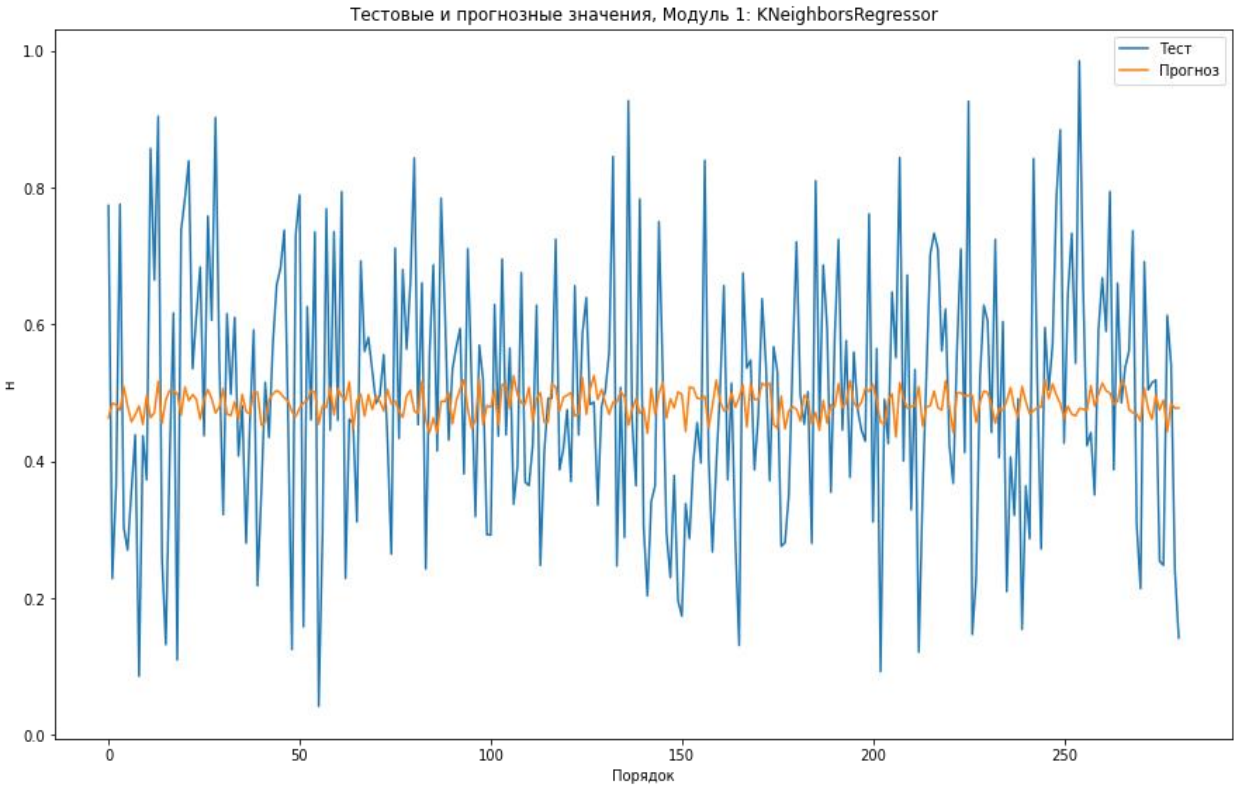
model	model	MSE общий	MSE_упругости	MSE_прочности	R2 общий	R2_упругости	R2_прочности
K0	Linear Regression	0.035074	0.034752	0.035396	-0.024522	-0.003601	-0.045443
1	KNeighborsRegressor	0.034788	0.034654	0.034921	-0.016102	-0.000792	-0.031412
2	SVR	0.034403	0.034629	0.034177	-0.004750	-0.000051	-0.009448
3	Random Forest Regressor	0.034899	0.034837	0.034962	-0.019342	-0.006064	-0.032620

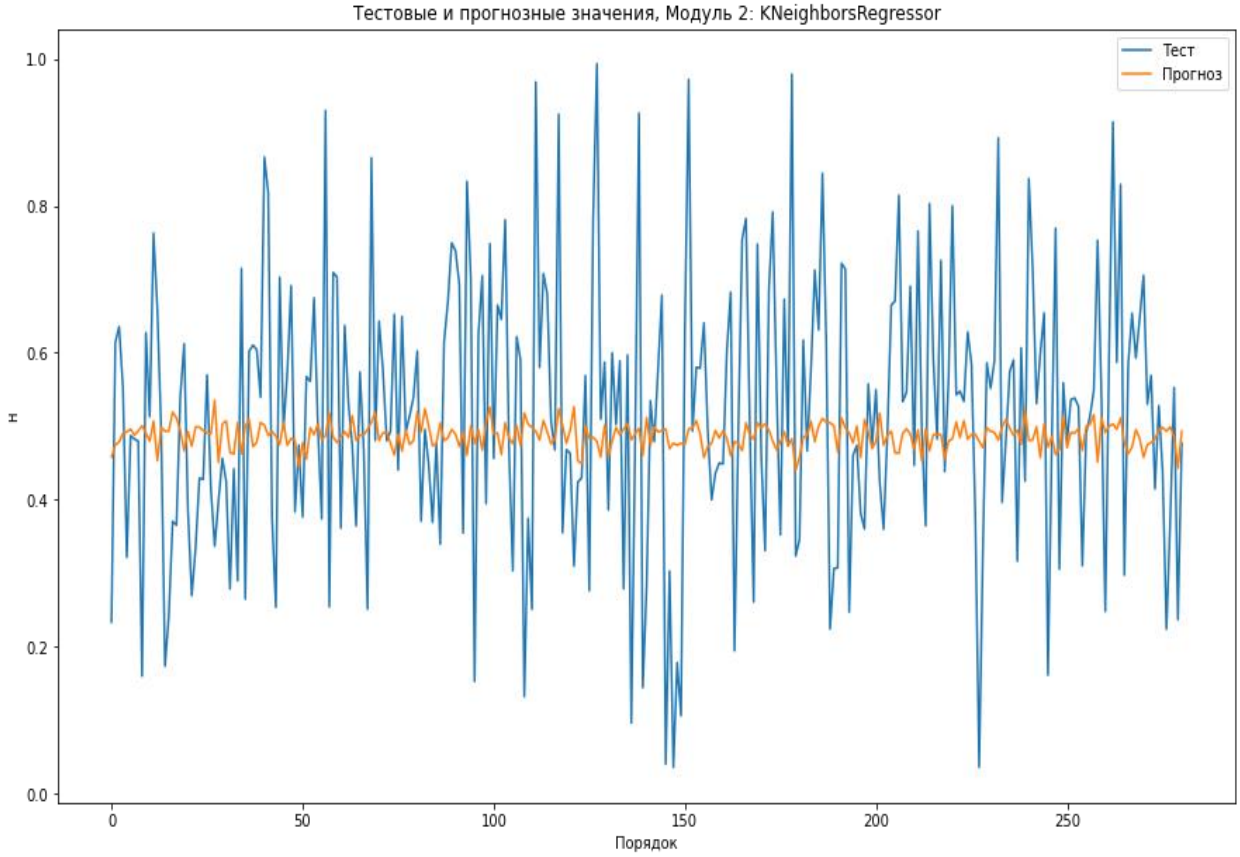
У всех моделей коэффициент детерминации имеет отрицательные значения. Т.о., модели не дают прогнозов, которые были бы лучше простого расчета среднего значения. Также зафиксированы большие значения ошибок MSE, что также свидетельствуют о низком качестве моделей.

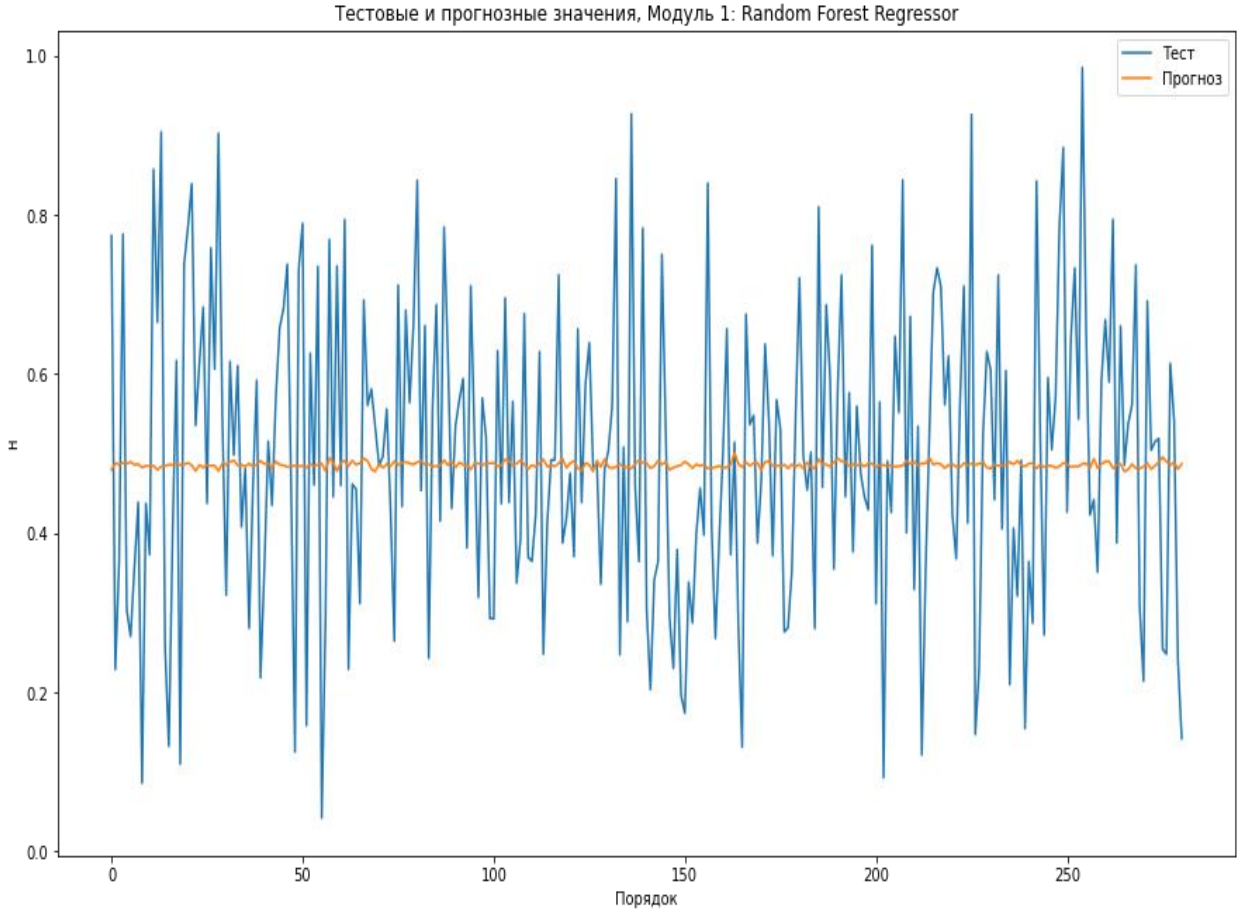
На Рис 11 визуально видно, что лучше всего отработал метод линейной регрессии. Но его показатели не намного лучше, чем у других моделей.

Рис.13. Визуализация тестовых и прогнозных значений.






Тестовые и прогнозные значения, Модуль 2: KNeighborsRegressor
Тест
Прогноз
н
Порядок
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
250


Тестовые и прогнозные значения, Модуль 1: Random Forest Regressor
Тест
Прогноз
н
Порядок
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
250

2.3. Нейронная сеть для соотношения матрица – наполнитель

Нейронный сети

Нейронная сеть — это последовательность нейронов, соединенных между собой связями. Структура нейронной сети пришла в мир программирования из биологии. Вычислительная единица нейронной сети — нейрон или персептрон.

У каждого нейрона есть определённое количество входов, куда поступают сигналы, которые суммируются с учётом значимости (веса) каждого входа.

Смещение – это дополнительный вход для нейрона, который всегда равен 1 и, следовательно, имеет собственный вес соединения.

Так же у нейрона есть функция активации, которая определяет выходное значение нейрона. Она используется для того, чтобы ввести нелинейность в нейронную сеть. Примеры активационных функций: relu, сигмоида, тангенс.

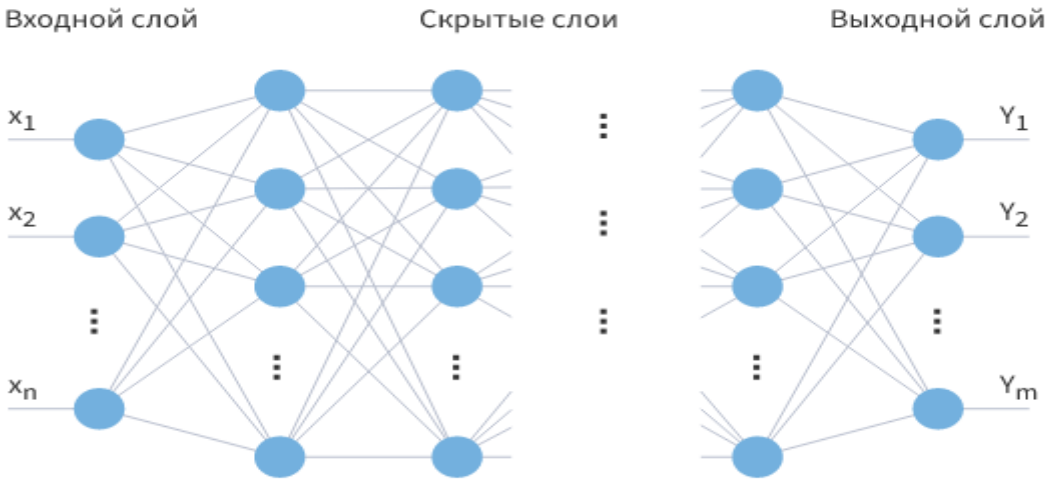
У полносвязной нейросети выход каждого нейрона подается на вход всем нейронам следующего слоя. У нейросети имеется:

входной слой - его размер соответствует входным параметрам;

скрытые слои - их количество и размерность определяем специалист;

выходной слой - его размер соответствует выходным параметрам.

Рис. 14. Нейронная сеть



Прямое распространение – это процесс передачи входных значений в нейронную сеть и получения выходных данных, которые называются прогнозируемым значением.

Прогнозируемое значение сравниваем с фактическим с помощью функции потери. В методе обратного распространения ошибки градиенты (производные значений ошибок) вычисляются по значениям весов в направлении, обратном прямому распространению сигналов. Значение градиента вычитают из значения веса, чтобы уменьшить значение ошибки. Таким образом происходит процесс обучения. Обновляются веса каждого соединения, чтобы функция потерь минимизировалась.

Для обновления весов в модели используются различные оптимизаторы.

Количество эпох показывает, сколько раз выполнялся проход для всех примеров обучения.

Нейронные сети применяются для решения задач регрессии, классификации, распознавания образов и речи, компьютерного зрения и других. На настоящий момент это самый мощный, гибкий и широко применяемый инструмент в машинном обучении.

Обучение нейронной сети — это такой процесс, при котором происходит подбор оптимальных параметров модели, с точки зрения минимизации функционала ошибки.

Принято решение создавать нейронную сеть с помощью Sequential - модель в библиотеке Keras, позволяющая создать нейронную сеть прямого распространения путем последовательного добавления слоев.

Строю нейронную сеть с помощью класса Sequential. Модель состоит из двух скрытых Dense слоев, количество нейронов в которых равно 128 и 64 и выходного слоя с одним нейроном. Функция активации слоев – relu. Rectified Linear Unit — это наиболее часто используемая функция активации при глубоком обучении. Данная функция возвращает 0, если принимает отрицательный аргумент, в случае же положительного аргумента, функция возвращает само число. То

есть она может быть записана как $f(z)=max(0,z)$. и ее можно использовать в нейронных сетях с множеством слоев.

На выходе используеем функцию активации tanh. Её природа нелинейна, она хорошо подходит для комбинации слоёв, а диапазон значений функции - (-1, 1). Поэтому нет смысла беспокоиться, что активационная функция перегрузится от больших значений.

В качестве оптимизатора используем adam. Adam (adaptive moment estimation) - это алгоритм оптимизации, совмещающий принципы инерции MomentumSGD и адаптивного обновления параметров AdaGrad и его модификаций. mt - оценка первого момента (среднее градиентов); υt - оценка второго момента (средняя нецентрированная дисперсия градиентов).

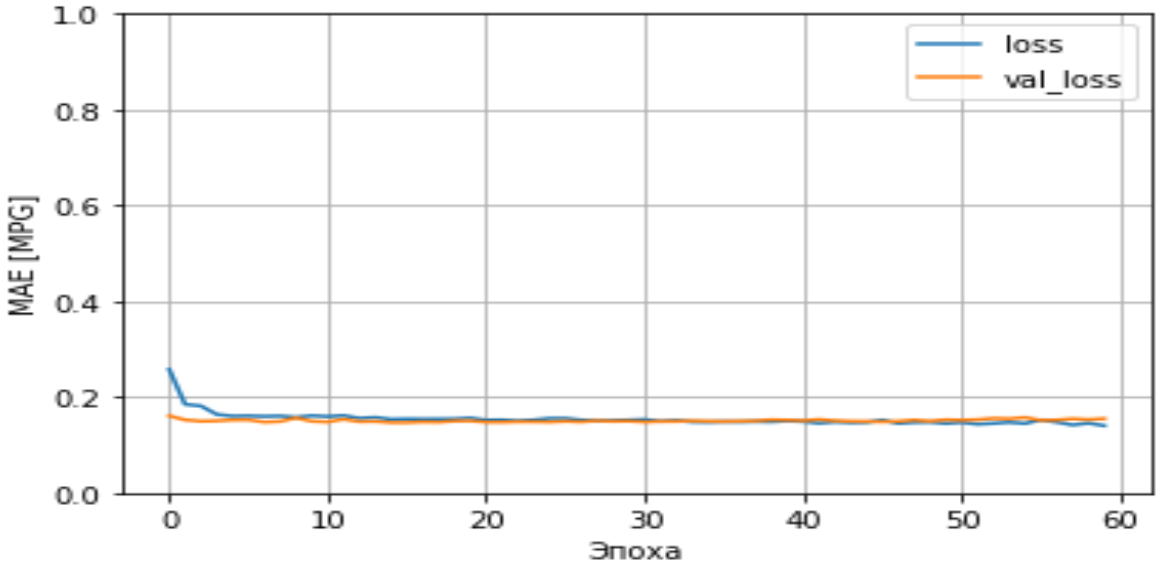
Количество жпох – 80. Для данного рода задачи это достаточное количество.

Для борьбы с переобучением были добавлены добавлены Dropout-слои. Использование ранней остановки сокращает время на обучение модели, а использование Dropout увеличивает, но уменьшается риск, что мы остановились слишком рано.

Рис. 15. Структура нейронной сети

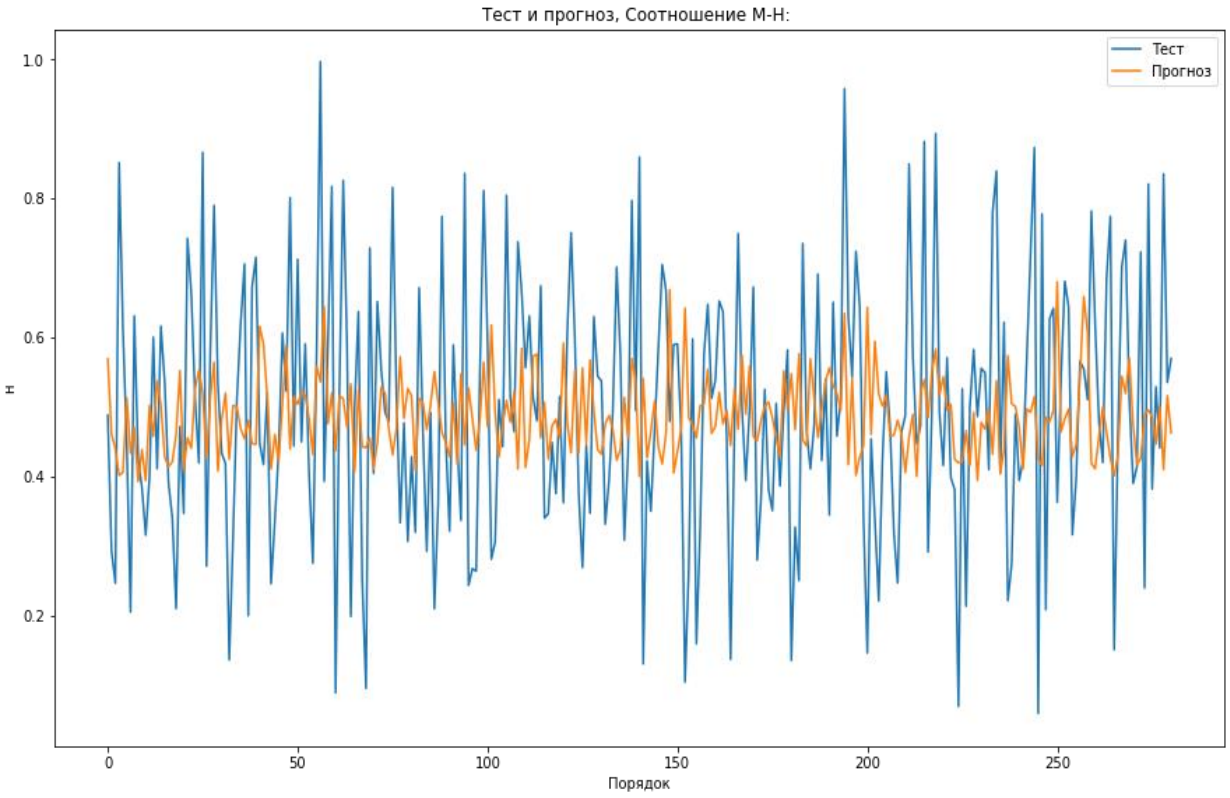
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_22 (Dense)            (None, 50)                650

 dropout_12 (Dropout)        (None, 50)                0

 dense_23 (Dense)            (None, 128)               6528

 dropout_13 (Dropout)        (None, 128)               0

 dense_24 (Dense)            (None, 19)                2451

 dropout_14 (Dropout)        (None, 19)                0

 dense_25 (Dense)            (None, 64)                1280

 dropout_15 (Dropout)        (None, 64)                0

 dense_26 (Dense)            (None, 32)                2080

 dense_27 (Dense)            (None, 1)                 33
```

Рис. 16. Процесс обучения модели. График потерь



На рисунке 13 видно, что модель не претерпевает сущетсвенных изменений после 10 эпохи. В то же время модель не начала переобучаться даже после 50 эпохи.

Рис. 17. График прогнозных значений нейронной сети.



На графике прогнозных значений нейронной сети можно увидеть, что построенная модель нейронной сети не так хорошо прогнозирует значения на тестовой выборке данных, хотя местами правильно предсказывает направление тренда и величину прогнозного значения.

Датасет с ошибками показывает следующие ошибки нейронной сети:

```
MSE = 0.03773305824483605
R2_score = -0.1153229014325372
```

Рис. 18. Таблица потерь нейронной сети.

index	model	target	MSE общ.	MSE_упругости	MSE_прочности	R2 общ	R2_упругости	R2_прочности
0	Linear Regression	Модуль упругости и Прочность	0.035074	0.034752	0.035396	-0.024522	-0.003601	-0.045443
1	KNeighborsRegressor	Модуль упругости и Прочность	0.034788	0.034654	0.034921	-0.016102	-0.000792	-0.031412
2	SVR	Модуль упругости и Прочность	0.034403	0.034629	0.034177	-0.004750	-0.000051	-0.009448
3	Random Forest Regressor	Модуль упругости и Прочность	0.034899	0.034837	0.034962	-0.019342	-0.006064	-0.032620

Для оценки качества моделей регрессии использовались специальные показатели.

1) R2_score (коэффициент детерминации) принимает значение от 0 до 1 и показывает долю объяснённой дисперсии объясняемого рода. Чем ближе R2 к 1, тем меньше доля необъяснённого.

2) Среднеквадратическая ошибка (MSE) — это распространенный способ измерения точности предсказания модели. Он рассчитывается как:

MSE = (1/n) * Σ(фактическое – прогноз) 2

куда:

Σ — причудливый символ, означающий «сумма».

n – размер выборки

фактический – фактическое значение данных

прогноз – прогнозируемое значение данных

Чем ниже значение MSE, тем лучше модель способна точно предсказывать значения.

В качестве лучшей модели выбрана модель линейной регрессии с наименьшим значение ошибки. Но это значение не намного лучше, чем у других моделей.

3. Выводы

Данное исследоване позволяет сделать основные выводы по теме. Распределение полученных данных в объединённом датасете близко к нормальному, но коэффициенты корреляции между парами признаков стремятся к нулю. Использованные при разработке моделей подходы не позволили получить сколько-нибудь достоверных прогнозов. Применённые модели регрессии не показали высокой эффективности в прогнозировании свойств композитов.

Был сделан вывод, что невозможно определить из свойств материалов соотношение «матрица – наполнитель». Данный факт не указывает на то, что прогнозирование характеристик композитных материалов на основании предоставленного набора данных невозможно, но может указывать на недостатки базы данных, подходов, использованных при прогнозе, необходимости пересмотра инструментов для прогнозирования.

Необходимы дополнительные вводные данные, получение новых физико-химических свойств материалов учёными. Поскольку мы не являемся специалистами в области свойств композитных материалов, то можем опираться только на данные, полученные посредством машинного обучения.

Вместе с тем, датасет очень хорош для обучающихся на специалистов по машинному обучению, поскольку в данном случае невозможно сделать вывод о показателях модели исходя из логики или жизненного опыта, а можно лишь полагаться на модели и выдаваемые ими параметры.

Приложение

В разработанном приложении можно спрогнозировать с помощью обученной нейронной сетеи конечные свойства композиционных материалов, на основе введенных пользователем значений.

Приложение Visaul Studio Code оказалось чувствительным к обновлению операционных систем. Ни на одном из двух, доступных нам ноутбуков MacBook Pro с Mac OS Mojave 10.14.6 и Lenovo с Windows 7 приложение не заработало. Поэтому было принято решение интегрировать приложение в Google Colab, в котром был написан основная работа.

Приложение поэтапно просит ввести значение всех 11 параметров, после чего выдает значение.

Рис. 18. Работа приложения

```
Введите данные
Введите значение переменной Соотношение матрица-наполнитель: 1
Введите значение переменной Плотность: 2
Введите значение переменной Модуль упругости: 3
Введите значение переменной Количество отвердителя: 4
Введите значение переменной Содержание эпоксидных групп: 5
Введите значение переменной Температура вспышки: 6
Введите значение переменной Поверхностная плотность: 7
Введите значение переменной Потребление смолы: 8
Введите значение переменной Угол нашивки: 9
Введите значение переменной Шаг нашивки: 1
Введите значение переменной Плотность нашивки: 2
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:439: UserWarning: X
does not have valid feature names, but MinMaxScaler was fitted with feature
names
  warnings.warn(
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:439: UserWarning: X
does not have valid feature names, but LinearRegression was fitted with
feature names
  warnings.warn(
['Модуль упругости при растяжении, ГПа', 'Прочность при растяжении, МПа']
вызов модели
[[  74.05342054 3604.56651904]]
введите 1 для прогноза, 2 для выхода
```

Список используемой литературы

1. Грас, Джоэл. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2021. - 416 с.
2. 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.
3. Джулли, Пал: Библиотека Keras - инструмент глубокого обучения / пер. с англ. А. А. Слинкин.- ДМК Пресс, 2017. – 249 с.
4. https://ru.wikipedia.org/wiki/
5. https://neerc.ifmo.ru/wiki/
6. https://wiki.loginom.ru/articles/multilayered-perceptron.html
7. https://neerc.ifmo.ru/wiki/index.php?title=Batch-normalization
8. https://scikit-learn.org/stable/user_guide.html
9. https://keras.io/guides/
10. https://www.tensorflow.org/guide
11. https://e-plastic.ru/specialistam/composite/kompozicionnye-materialy
12. https://statpsy.ru/correlation/correlation/
13. https://www.machinelearningmastery.ru/5-ways-to-detect-outliers-that-every

Создание репозитория

Нами создан рипозитория на GitHub, куда был интегрирован код в формате Google Colaboratory. Также офрлмен файл README.

Рипозиторий доступен по ссылке:

https://github.com/Paabel/VKR_BMSTU