

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

Боровиков Кирилл Андреевич

Москва, 2022

Содержание

Оглавление

Содержание.....	2
Введение.....	1
Описание исходных данных.....	2
Описание используемых методов.....	4
И используемые библиотеки.....	4
Описание моделей машинного обучения.....	8
Дополнительно используемые методы.....	9
Разведочный анализ.....	10
Корреляционный анализ.....	14
Предобработка данных.....	15
Обучение нескольких моделей.....	17
для прогноза модуля упругости при растяжении.....	17
и прочности при растяжении.....	17
Построение Модели MLP.....	21
для параметра Модуль упругости при растяжении, ГПа.....	21
Разработка приложения.....	24
Заключение.....	28
Список используемой литературы.....	29

Введение

Композиционные материалы (композиты), материалы, представляющие собой объёмное сочетание компонентов с чётко выраженной границей раздела. Состоят из матрицы (связующего) и равномерно распределённых в ней упрочнителей, или армирующих наполнителей.

Свойства композиционных материалов в основном зависят от физико-механических свойств компонентов и прочности связи между ними. Матрица придаёт изделию из композиционных материалов заданную форму и монолитность; свойства матрицы определяют эксплуатационные характеристики (рабочую темп-ру, плотность, удельную прочность, сопротивление усталостному разрушению и воздействию окружающей среды) и технологичные режимы получения композиционных материалов. Армирующие наполнители вводят в композиционные материалы с целью увеличения прочности, жёсткости и пластичности, а также изменения электрофизических, теплофизических характеристик в различных направлениях или отдельных местах изделия. композиционные материалы характеризуются свойствами, которыми не обладает ни один из компонентов в отдельности.

Именно поэтому очень сложно прогнозировать появление новых свойств композиционных материалов. Для определения новых свойств необходимо построить математическую модель.

Актуальность решения задачи обусловлена широким использованием композиционных материалов в различных отраслях.

Описание исходных данных

Цель исследования сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

Объём и характеристики датасета

№	Column	Количество записей	Non-Null Count (пустые значений)	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

На входе имеются данные о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.).

На выходе необходимо спрогнозировать ряд конечных свойств получаемых композиционных материалов. Кейс основан на реальных производственных задачах Центра НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана).

Изначально выборка состоит из двух датасетов. Первый датасет состоит из 1022 элементов, у каждого элемента присутствуют следующие характеристики: Соотношение матрица-наполнитель; Плотность, кг/м³; модуль упругости, ГПа; Количество отвердителя, м; Содержание эпоксидных групп, %₂; Температура вспышки, С₂; Поверхностная плотность, г/м²; Модуль упругости при растяжении, ГПа; Прочность при растяжении, МПа; Потребление смолы, г/м². Второй датасет состоит из 1039 элементов, у каждого элемента присутствуют следующие характеристики: Угол нашивки, град; Шаг нашивки; Плотность нашивки.

После объединения этих датасетов, мы получили итоговую выборку из 1022 элементов. Итоговый датасет был проверен на пропуски, по итогу проверки пропуски не обнаружены, выше представлен способ выявления пропусков в датасете.

```
data.info()
```

```
<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
```

Описание используемых методов

И используемые библиотеки

В рамках исследовательского проекта были использованы следующие библиотеки:

- - NumPy Данная библиотека (сокращенно от Numerical Python) – библиотека с открытым исходным кодом для языка программирования Python. Возможности: поддержка многомерных массивов (включая матрицы); поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами. NumPy предоставляет реализации вычислительных алгоритмов (в виде функций и операторов), оптимизированные для работы с многомерными массивами. В результате любой алгоритм, который может быть выражен в виде последовательности операций над массивами (матрицами) и реализованный с использованием NumPy, работает так же быстро, как эквивалентный код, выполняемый в MATLAB.
- Pandas Программная библиотека на языке Python для обработки и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами. Название библиотеки происходит от эконометрического термина «панельные данные», используемого для описания многомерных структурированных наборов информации. Pandas распространяется под новой лицензией BSD.

Основная область применения – обеспечение работы в рамках среды Python не только для сбора и очистки данных, но для задач анализа и моделирования данных, без переключения на более специфичные для статобработки языки (такие, как R и Octave). Пакет прежде всего предназначен для очистки и первичной оценки данных по общим показателям, например среднему значению, квантилям и так далее; статистическим пакетом он в полном смысле не является, однако наборы данных типов DataFrame и Series применяются в качестве входных в большинстве модулей анализа данных и машинного обучения.

- Matplotlib Содержит большой набор инструментов для двумерной графики. Она проста в использовании и позволяет получать графики высокого качества. В этом разделе мы рассмотрим наиболее распространенные типы диаграмм и различные настройки их отображения.
- Matplotlib.pyplot модуль matplotlib предоставляет процедурный интерфейс к (объектно-ориентированной) библиотеке matplotlib, который во многом копирует инструменты пакета MATLAB. Инструменты модуля pyplot фактически являются стандартным способом работы с библиотекой matplotlib, поэтому при разработке задания ВКР применялся именно этот пакет.
- Seaborn Это библиотека для создания статистических графиков на Python. Она основывается на matplotlib и тесно взаимодействует со структурами данных pandas. Архитектура Seaborn позволяет быстро изучить и понять свои данные.

Seaborn захватывает целые фреймы данных или массивы, в которых содержатся все ваши данные, и выполняет все внутренние функции, нужные для семантического маппинга и статистической агрегации для преобразования данных в информативные графики. Она абстрагирует сложность, позволяя проектировать графики в соответствии с нуждами.

- **PySpark** Установка идёт аналогично установке и обновлениям пакетов через `pip` – `pip install pyspark`. PySpark – это интерфейс для Apache Spark в Python. Он не только позволяет вам писать приложения Spark с использованием API-интерфейсов Python, но также предоставляет оболочку PySpark для интерактивного анализа ваших данных в распределенной среде. PySpark поддерживает большинство функций Spark, таких как Spark SQL, DataFrame, Streaming, MLlib (машинное обучение) и Spark Core.
- **Sklearn** Scikit-learn – один из наиболее широко используемых пакетов Python для Data Science и Machine Learning. Он позволяет выполнять множество операций и предоставляет множество алгоритмов. Scikit-learn также предлагает отличную документацию о своих классах, методах и функциях, а также описание используемых алгоритмов. Scikit-Learn поддерживает:
 - предварительную обработку данных;
 - уменьшение размерности;
 - выбор модели;
 - регрессии;
 - классификации;
 - кластерный анализ.

- TensorFlow Это бесплатная программная библиотека с открытым исходным кодом для машинного обучения и искусственного интеллекта . Его можно использовать для решения целого ряда задач, но особое внимание уделяется обучению и выводу глубоких нейронных сетей. TensorFlow можно использовать в самых разных языках программирования, в первую очередь в Python, а также в Javascript, C++ и Java. Такая гибкость позволяет использовать ее в различных областях.

Описание моделей машинного обучения

Моделью машинного обучения называется файл, который обучен распознаванию определенных типов закономерностей. Модель обучается на основе набора данных, предоставляя ей алгоритм, который она может использовать для анализа и обучения на основе этих данных.

Завершив обучение модели, можно применить ее для принятия решений и выполнения прогнозов по данным, которые ранее не встречались.

Краткая характеристика моделей:

- `sklearn.neighbors.KNeighborsRegressor` – регрессия на основе k ближайших соседей. Цель прогнозируется путем локальной интерполяции целей, связанных с ближайшими соседями в обучающем наборе.
- `sklearn.linear_model.LinearRegression` – обычная линейная регрессия методом наименьших квадратов. Соответствует линейной модели с коэффициентами $w = (w_1, \dots, w_p)$, чтобы минимизировать остаточную сумму квадратов между наблюдаемыми целями в наборе данных и целями, предсказанными линейным приближением.
- `sklearn.svm.SVC` – С-классификация опорных векторов. Реализация основана на `libsvm`. Время подгонки масштабируется квадратично с количеством выборок и может быть непрактичным за пределами десятков тысяч выборок. Поддержка мультиклассов осуществляется по схеме «один против одного».
- `sklearn.ensemble.RandomForestRegressor` – случайный лес – это метаоценка, которая соответствует ряду классифицирующих деревьев решений для различных подвыборок набора данных и использует усреднение для повышения точности прогнозирования и контроля переобучения.

- `sklearn.ensemble.GradientBoostingRegressor` – используется для повышения градиента для регрессии. `GradientBoostingRegressor` строит аддитивную модель поэтапно вперед; он позволяет оптимизировать произвольные дифференцируемые функции потерь. На каждом этапе дерево регрессии аппроксимируется отрицательным градиентом заданной функции потерь.

Дополнительно используемые методы

- `sklearn.metrics.r2_score` – коэффициент детерминации (R-квадрат) — это доля дисперсии зависимой переменной, объясняемая рассматриваемой моделью зависимости, то есть объясняющими переменными. Более точно – это единица минус доля необъяснённой дисперсии (дисперсии случайной ошибки модели, или условной по факторам дисперсии зависимой переменной) в дисперсии зависимой переменной. Его рассматривают как универсальную меру зависимости одной случайной величины от множества других. В частном случае линейной зависимости является квадратом так называемого множественного коэффициента корреляции между зависимой переменной и объясняющими переменными. В частности, для модели парной линейной регрессии коэффициент детерминации равен квадрату обычного коэффициента корреляции между y и x .
- `train_test_split` – для нашей задачи ВКР мы будем использовать модуль `train_test_split` библиотеки `Scikit-learn`, который очень полезен для разделения датасетов, независимо от того, будете ли вы применять `Scikit-learn` для выполнения других задач машинного обучения. Конечно, можно выполнить такие разбиения каким-либо другим способом (возможно, используя только `Numpy`). Библиотека `Scikit-learn` включает полезные функции, позволяющее сделать это

немного проще. Разделяет массивы или матрицы на случайные поезда и тестовые подмножества. Быстрая утилита, которая объединяет проверку ввода и приложение для ввода данных в один вызов для разделения (и, при необходимости, подвыборки).

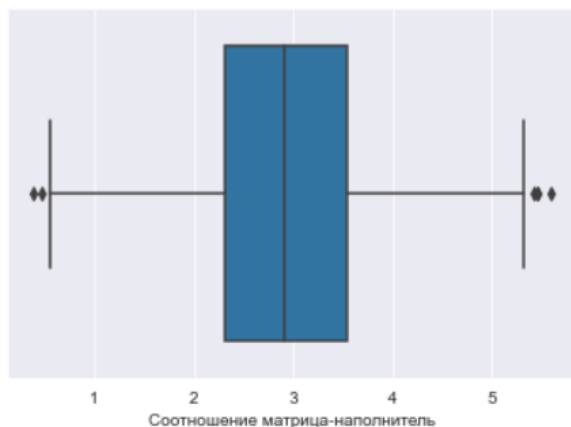
Разведочный анализ

Далее был проведен разведочный анализ данных для выявления выбросов в значениях, методов выявления был выбран plotbox (ящик с усами).

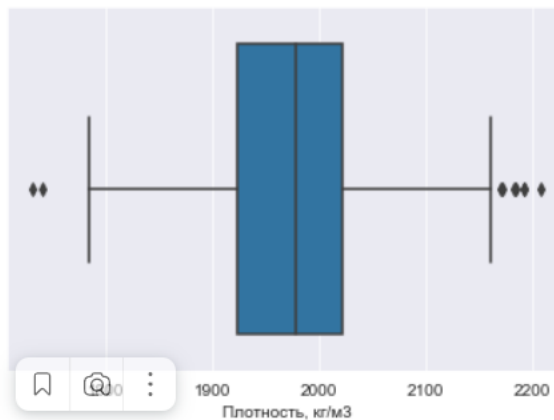
Ящик с усами, диаграмма размаха (англ. box-and-whiskers diagram or plot, box plot) – график, использующийся в описательной статистике, компактно изображающий одномерное распределение исследуемых величин.

Такой вид диаграммы в удобной форме показывает медиану (или, если нужно, среднее), нижний и верхний квартили, минимальное и максимальное значение выборки и выбросы.

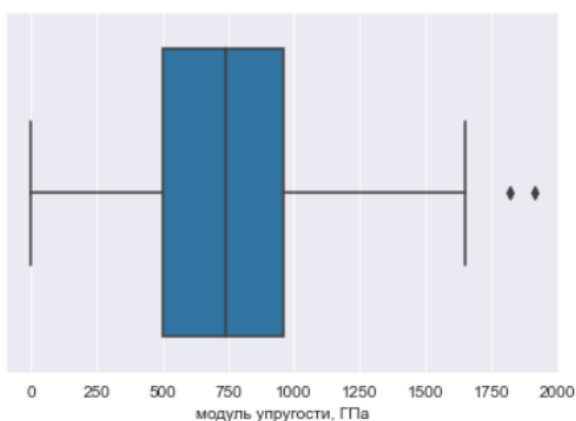
Результатом применения метода были выявлены выбросы в следующих столбцах: "Соотношение матрица-наполнитель", "Плотность, кг/м³", "модуль упругости, ГПа", "Количество отвердителя, м.%", "Содержание эпоксидных групп, %₂", "Температура вспышки, С₂", "Поверхностная плотность, г/м²", "Модуль упругости при растяжении, ГПа", "Прочность при растяжении, МПа", "Потребление смолы, г/м²", "Шаг нашивки", "Плотность нашивки". Ниже приведены графики с результатами.



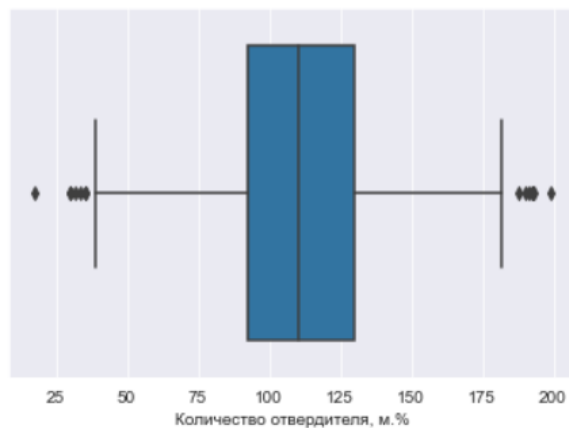
Минимальное значение: 0.389402605178414
 Максимальное значение: 5.59174159869754
 Среднее значение: 2.9303657734325506
 Медианное значение: 2.90687765033521



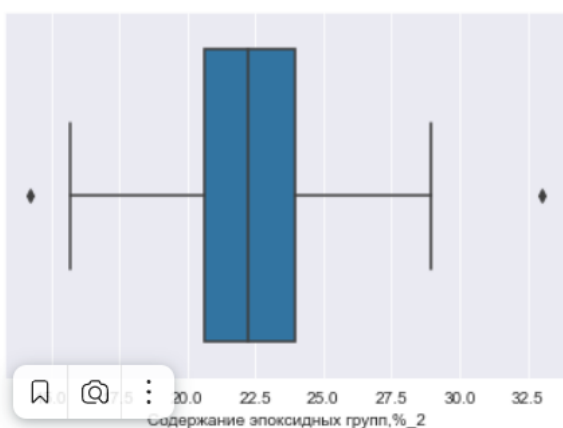
Минимальное значение: 1731.764635096
 Максимальное значение: 2207.77348061119
 Среднее значение: 1975.7348881101548
 Медианное значение: 1977.62165679058



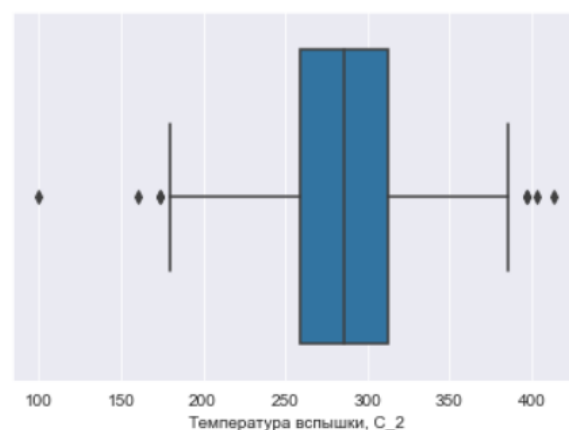
Минимальное значение: 2.4369087535075
 Максимальное значение: 1911.53647700054
 Среднее значение: 739.9232327560719
 Медианное значение: 739.664327697792



Минимальное значение: 17.7402745562519
 Максимальное значение: 198.953207190451
 Среднее значение: 110.57076864736258
 Медианное значение: 110.564839894065

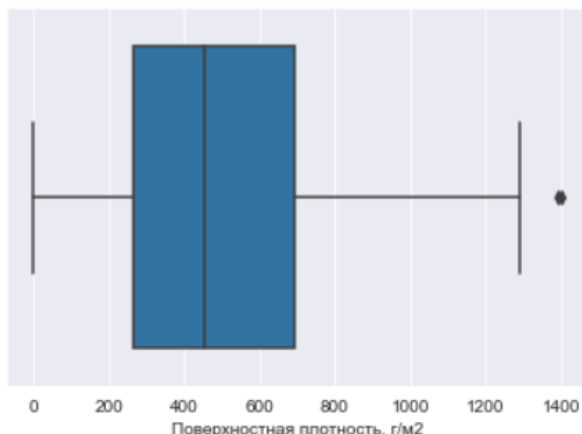


Минимальное значение: 14.2549854977161
 Максимальное значение: 33.0
 Среднее значение: 22.24438954776772
 Медианное значение: 22.2307437560244

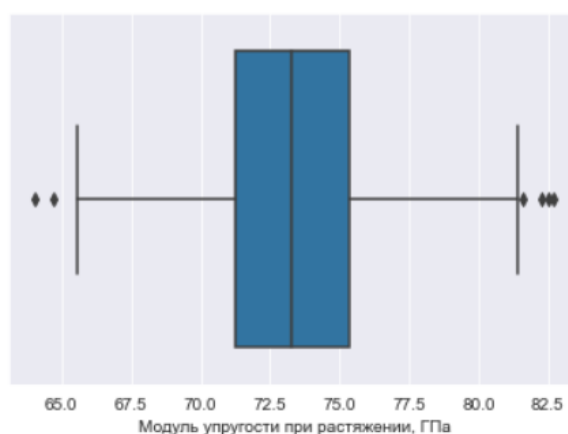


Минимальное значение: 100.0
 Максимальное значение: 413.273418243566
 Среднее значение: 285.88215135162227
 Медианное значение: 285.896812331237

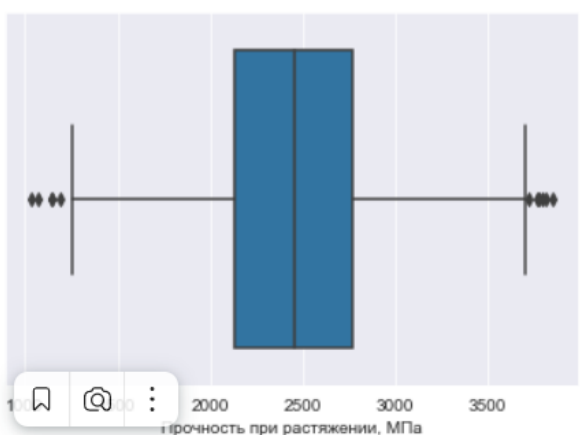
Графики «plotbox»



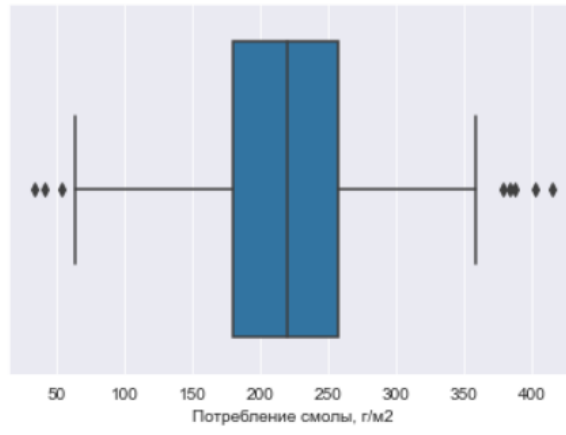
Минимальное значение: 0.603739925153945
 Максимальное значение: 1399.54236233989
 Среднее значение: 482.7318330384181
 Медианное значение: 451.86436518306



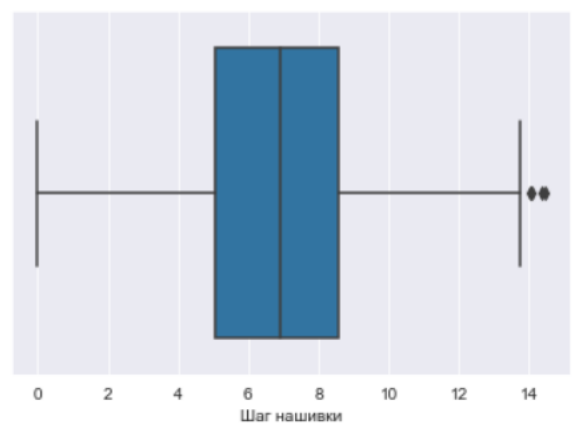
Минимальное значение: 64.0540605597917
 Максимальное значение: 82.682051035271
 Среднее значение: 73.32857125009073
 Медианное значение: 73.2688045943481



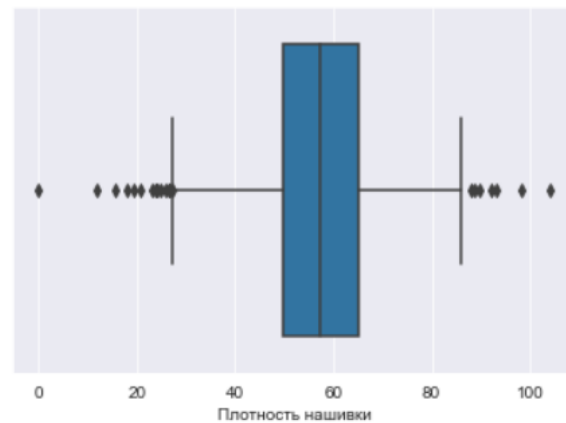
Минимальное значение: 1036.85660535
 Максимальное значение: 3848.43673187618
 Среднее значение: 2466.9228426979025
 Медианное значение: 2459.52452600309



Минимальное значение: 33.8030255329625
 Максимальное значение: 414.590628361534
 Среднее значение: 218.42314367654265
 Медианное значение: 219.198882195134



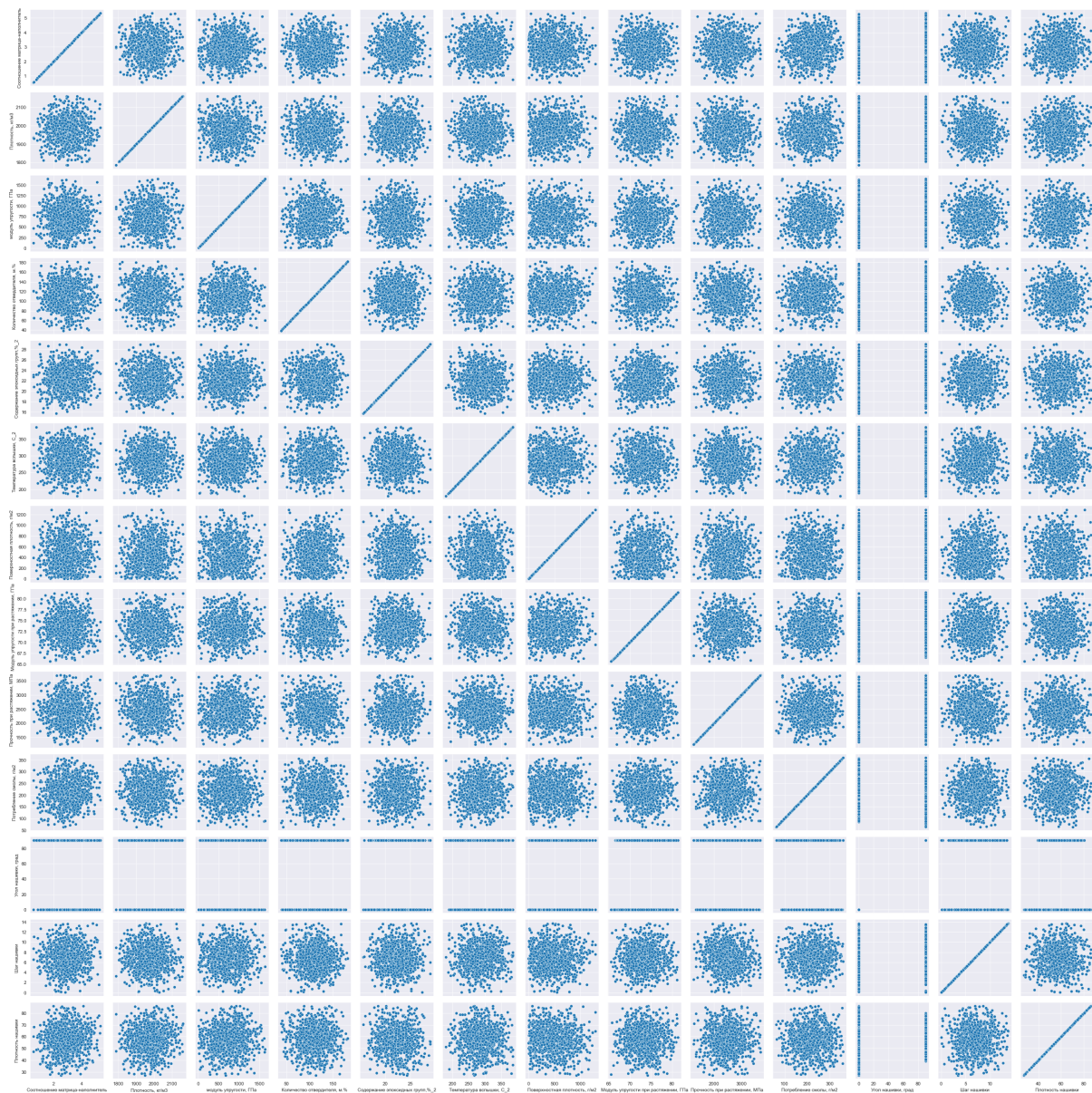
Минимальное значение: 0.0
 Максимальное значение: 14.4405218753969
 Среднее значение: 6.899222077675024
 Медианное значение: 6.9161438559491



Минимальное значение: 0.0
 Максимальное значение: 103.988901301494
 Среднее значение: 57.15392943285763
 Медианное значение: 57.3419198469929

Найденные выбросы были заменены на пустые значения, после чего строки с пустыми значениями были удалены из датасета. После удаления выбросов осталось 935 строк со значениями.

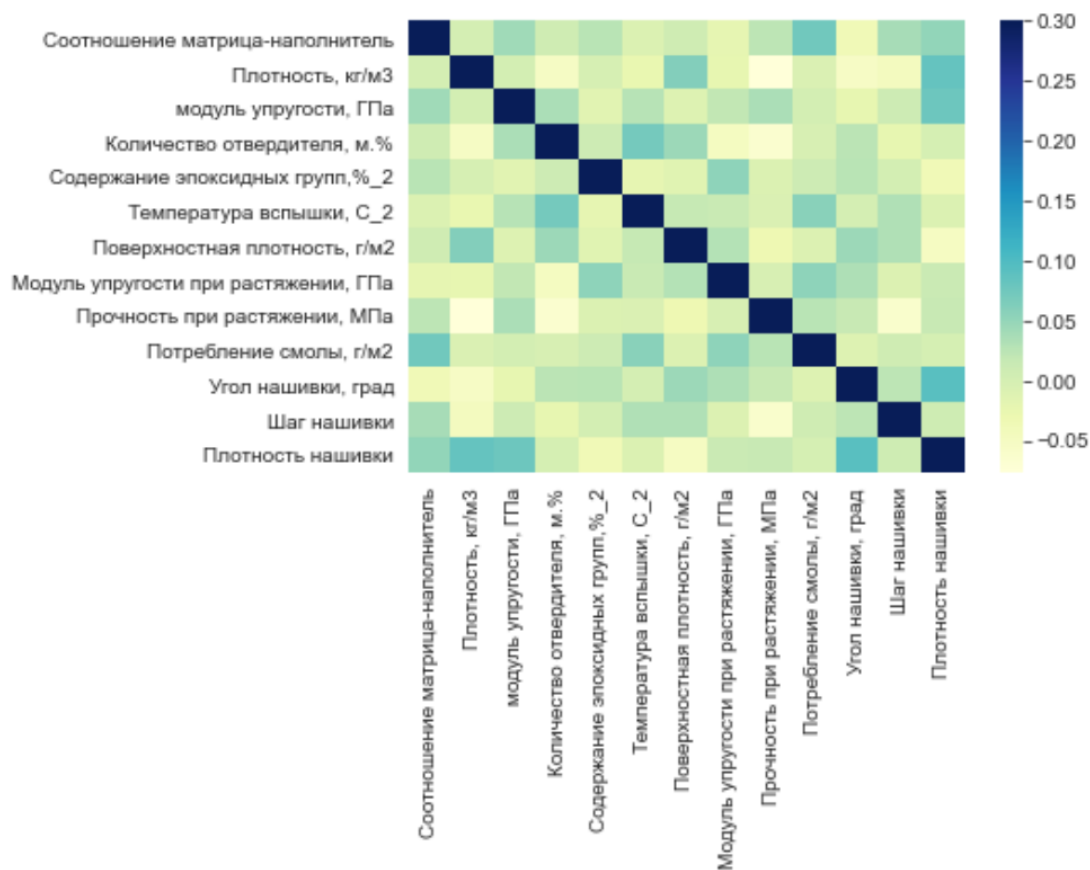
После удаления выбросов были построены попарные графики зависимости, таким образом можно найти переменные с высокой корреляцией, и удалить их сократив размерность выборки. Ниже представлен график.



Изучив график можно сделать вывод что никакой видимой зависимости между переменными нет.

Корреляционный анализ

Метод обработки статистических данных, заключающийся в изучении коэффициентов корреляции между переменными.



В настоящем исследовании для каждой пары параметров определяются коэффициенты корреляции, которые представляют собой количественные меры взаимосвязи (совместной изменчивости) двух переменных.

Хитмап карта также показывает очень низкую корреляцию. График представлен выше.

Предобработка данных

Нормализация данных.

В машинном обучении нормализацией называют метод предобработки числовых признаков в обучающих наборах данных с целью приведения их к некоторой общей шкале без потери информации о различии диапазонов.

Необходимость нормализации вызвана тем, что разные признаки обучающего набора данных могут быть представлены в разных масштабах и изменяться в разных диапазонах. Например, модуль упругости, который изменяется от 2 до 2000, и содержание эпоксидных групп, изменяющийся от 14 до 33. То есть диапазоны изменения признаков различаются в десятки раз.

После нормализации все числовые значения входных признаков будут приведены к одинаковой области их изменения – некоторому узкому диапазону. Это позволит свести их вместе в одной модели Machine Learning и обеспечит корректную работу вычислительных алгоритмов.

В настоящей работе для нормализации данных используется MinMaxScaler. Для каждого значения в параметре MinMaxScaler вычитает минимальное значение в параметре и затем делит на диапазон. Диапазон – это разница между исходным максимумом и исходным минимумом.

MinMaxScaler сохраняет форму исходного дистрибутива. Это не приводит к значительному изменению информации, встроенной в исходные данные. Однако, MinMaxScaler не уменьшает важность выбросов. Диапазон по умолчанию для функции, возвращаемой MinMaxScaler, составляет от 0 до 1.

Нормализация по данному методу позволяет достаточно просто провести и обратные преобразование, которые будет необходимо выполнить, поскольку обученная модель на выходе будет иметь нормализованные значения параметра.

Обучение нескольких моделей

для прогноза модуля упругости при растяжении

и прочности при растяжении

1. Определим входные и выходные переменные.

I. Переменные на вход: "Соотношение матрица-наполнитель", "Плотность, кг/м³", "модуль упругости, ГПа", "Количество отвердителя, м.%", "Содержание эпоксидных групп, %₂", "Температура вспышки, С₂", "Поверхностная плотность, г/м²", "Потребление смолы, г/м²", "Угол нашивки, град", "Шаг нашивки", "Плотность нашивки".

II. Переменные на выход: "Модуль упругости при растяжении, ГПа", "Прочность при растяжении, МПа"

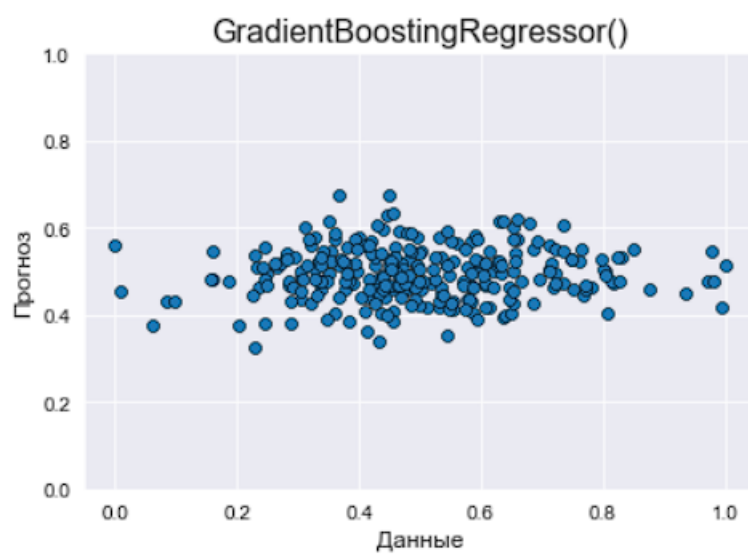
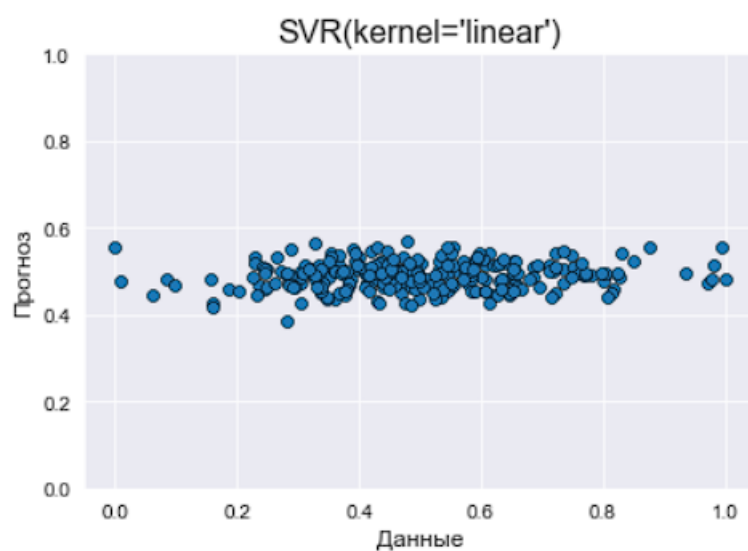
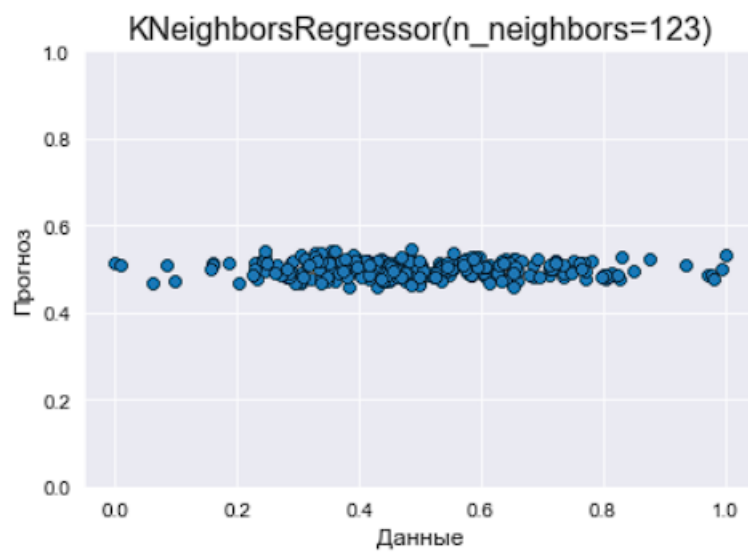
2. Дальше с помощью функции `train_test_split` разделим выборки на обучающую и тестовую.

3. Создаем массив с моделями для того, чтобы было удобно прогнать их через цикл, для нашей выборки мы будем использовать следующие модели: `LinearRegression` (метод наименьших квадратов), `RandomForestRegressor` (случайный лес), `KNeighborsRegressor` (метод К ближайших соседей), `SVR(kernel='linear')` (метод опорных векторов с линейным ядром), и `GradientBoostingRegressor` (Метод градиентного бустинга).

4. С помощью цикла `for` мы прогоняем данные через выше указанные модели. Для того чтобы понять какая модель нам лучше подойдет мы найдем коэффициент детерминации и построим графики рассеивания (правильный ответ/предсказанный ответ).

Ниже указаны графики рассеивания для каждой модели и коэффициенты детерминации.

	R2_Y1	R2_Y2
Model		
LinearRegression	0.011664	-0.003653
RandomForestRegressor	0.010152	-0.003241
KNeighborsRegressor	0.021587	-0.023876
SVR	0.004842	0.009410
GradientBoostingRegressor	-0.023532	-0.077426





При идеальном совпадении предсказанных данных с исходными данными для тестовой выборки, мы бы увидели ровную линию из точек начинающуюся в левом нижнем углу и заканчивающуюся в правом верхнем. Так как приведенные выше графики сильно отличаются от этих показателей, мы можем предположить, что используемые модели не подходят для решения нашей задачи.

Построение Модели MLP

для параметра Модуль упругости при растяжении, Гпа

Для рекомендации соотношения «матрица-наполнитель» была разработана простая модель глубокого обучения с помощью TensorFlow.

Архитектура нейронной сети может быть описана следующим образом.

Заданы два плейсхолдера. Входные данные представлены в виде двумерной матрицы, а выходные данные – одномерного вектора. Важно понимать, какая форма входных и выходных данных нужна нейросети и соответственным образом их организовать.

Помимо плейсхолдеров, есть и другой важнейший элемент – это переменные. Если плейсхолдеры используются для хранения входных и целевых данных в графе, то переменные служат гибкими контейнерами внутри графа. Им позволено изменяться в процессе выполнения графа. Веса и смещения представлены переменными для того, чтобы облегчить адаптацию во время обучения. Переменные необходимо инициализировать перед началом обучения.

Приступим к созданию плейсхолдеров

1. Разделим данные на входные и выходные.

I. На входе: "Плотность, кг/м³", "модуль упругости, Гпа", "Количество отвердителя, м.%", "Содержание эпоксидных групп, %_2", "Температура вспышки, С_2", "Поверхностная плотность, г/м²", "Потребление смолы, г/м²", "Угол нашивки, град", "Шаг нашивки", "Плотность нашивки", "Соотношение матрица-наполнитель", "Прочность при растяжении, МПа".

II. На выходе: "Модуль упругости при растяжении, Гпа"

2. Подготовим обучающую и тестовую выборку с разделением 70 и 30 процентов. Для этого воспользуемся функцией `train_test_split()`
3. Далее создадим модель, для этого мы используем функцию `Sequential`. В этой функции после долгого перебора возможных параметров оптимальными параметрами указываем: десять скрытых слоев по двадцать четыре нейрона, три скрытых слоя по двенадцать нейронов, один скрытый слой десять нейронов и один скрытый слой 5 нейронов.
4. Далее нам необходимо выбрать активатор. Существуют десятки функций активации, одна из самых распространенных – выпрямленная линейная единица (rectified linear unit, ReLU). В данном руководстве используется именно она.

Ниже представлен рисунок с кодом MLP модели.

```
model3 = Sequential([
    Dense(12, activation = 'relu', input_dim=12),
    Dense(24, activation = 'relu'),
    Dense(24, activation = 'relu'),
    Dense(24, activation = 'relu'),
    Dense(24, activation = 'relu'),
    Dense(24, activation = 'relu'),
    Dense(24, activation = 'relu'),
    Dense(24, activation = 'relu'),
    Dense(24, activation = 'relu'),
    Dense(24, activation = 'relu'),
    Dense(24, activation = 'relu'),
    Dense(12, activation = 'relu'),
    Dense(12, activation = 'relu'),
    Dense(12, activation = 'relu'),
    Dense(10, activation = 'relu'),
    Dense(5, activation = 'relu'),
    Dense(1)])

model3.compile(optimizer='adam',
               loss='mse',
               metrics=['mae'])

model3.summary()
```


5. Далее была определена функция стоимости сети, которая используется для генерации оценки отклонения между прогнозами сети и реальными результатами наблюдений в ходе обучения. Для решения проблем с регрессией используют функцию средней квадратичной ошибки (mean squared error, MSE). Данная функция вычисляет среднее квадратичное отклонение между предсказаниями и целями.

6. В качестве оптимизатора использовался один из наиболее распространенных оптимизаторов в сфере машинного обучения Adam Optimizer. Adam – это аббревиатура для фразы “Adaptive Moment Estimation” (адаптивная оценка моментов), он представляет собой нечто среднее между двумя другими популярными оптимизаторами AdaGrad и RMSProp.

Обучение сети останавливается по достижению максимального числа эпох или при срабатывании другого определенного заранее критерия остановки.

После обучения модели для была определена средняя абсолютная ошибка на тестовом наборе данных.

Средняя абсолютная ошибка: 0,1534

Разработка приложения

Приложение разработано с интерфейсом командной строки и позволяет решать задачи прогнозирования целевой переменной на основе входных данных. Среди особенностей работы приложения можно выделить следующие:

- приложение позволяет выбрать любую переменную для прогнозирования из загруженных данных;
- приложение позволяет также выбрать параметры, которые будут считаться входными переменными (есть возможность исключить некоторые переменные из общего числа);
- приложение позволяет неоднократно производить выбор параметров, выбор модели или прогнозирования целевой функции.

Приложение может быть запущено из командной строки следующим образом:

1. Программа и датасет находятся в единой папке \VKR. В данном случае (на конкретном ПК) папка \VKR расположена по полному адресу: C:\Users\KiBor\projects\VKR.
2. В нашем случае используется операционная система Windows 11¹ и PowerShell² 7.2.2. Используя командную строку, запускаем команду `cd`³ и прописываем путь до папки: `cd projects\VKR`. Затем даём команду на запуск консольного приложения: `app_test.py`.

1 Проприетарная операционная система для персональных компьютеров, разработанная компанией Microsoft в рамках семейства Windows NT. При разработке ВКР использовалась версия Windows 11 Pro, версии 22H2, сборка 22581.200.

2 Расширяемое средство автоматизации от Microsoft с открытым исходным кодом, состоящее из оболочки с интерфейсом командной строки и сопутствующего языка сценариев. Новая версия PowerShell 7.2.2 устанавливается в новый каталог и работает параллельно с Windows PowerShell 5.1.

3 Отображает имя текущего каталога или изменяет текущий. Если используется только с буквой диска (например, `cd C:`), на `cd C:` диске отображаются имена текущего каталога на указанном диске.

3. Появляется запрос: «Введите путь до файла с данными X_br:».

Вводим путь к нашему датасету:

C:/Users/KiBor/projects/VKR/X_br_vkrdataset.xlsx

4. Далее появляется запрос: «Введите путь до файла с данными

X_nup:». Вводим путь к нашему датасету:

C:/Users/KiBor/projects/VKR/X_nup_vkrdataset.xlsx

В ходе загрузки приложения, появятся сообщения о настройке библиотек программы, анализе выбросов, нормализации данных.

5. Появляется запрос: «Выберите целевую переменную для прогнозирования из предложенного списка». Вводим вручную по курсору, либо копированием, далее – нажимаем Enter (см. Рисунок ниже)

Выберите целевую переменную для прогнозирования из предложенного списка:

Соотношение матрица-наполнитель

Плотность, кг/м3

модуль упругости, ГПа

Количество отвердителя, м.%

Содержание эпоксидных групп,%_2

Температура вспышки, C_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

Плотность нашивки

6. Появляется запрос: «Выберите переменную для исключения из списка параметров, либо введите N (необходимо соблюдать регистр) для завершения исключения параметров». Вводим вручную по курсору, либо копированием, далее – нажимаем Enter (см. рисунок ниже).

Выберите переменную для исключения из списка параметров, либо введите N для завершения исключения параметров:
Соотношение матрица-наполнитель

Плотность, кг/м³

модуль упругости, ГПа

Количество отвердителя, м.%

Содержание эпоксидных групп, %₂

Температура вспышки, С₂

Поверхностная плотность, г/м²

Модуль упругости при растяжении, ГПа

Прочность при растяжении, МПа

Потребление смолы, г/м²

Угол нашивки, град

Шаг нашивки

7. Появляется запрос: «Для продолжения нажмите любую клавишу, для переподбора параметров нажмите Y». Так как для примера нам ничего переподбирать не требуется, вводим ENTER – запускаем. При необходимости ввода Y – соблюдать регистр.

8. Проводится расчёт и появляется отчёт: «Средняя абсолютная ошибка: 0,1534». Также появляется приглашение для ввода параметров «Введите значение параметра (Плотность, кг/м³):» (см. рисунок).

Температура вспышки, С₂

Поверхностная плотность, г/м²

Модуль упругости при растяжении, ГПа

Прочность при растяжении, МПа

Потребление смолы, г/м²

Угол нашивки, град

Шаг нашивки

N

Для продолжения нажмите любую клавишу, для переподбора параметров нажмите Y

Выберите прогнозную модель из списка:

Идет обучение модели

Средняя абсолютная ошибка: 0.1534

9. Теперь необходимо ввести данные для расчёта. В данном примере вводятся случайные данные для тестирования. Вводим данные (см. рисунок).

Введите значение параметра (Соотношение матрица-наполнитель): 1
Введите значение параметра (Плотность, кг/м³): 1
Введите значение параметра (модуль упругости, ГПа): 1
Введите значение параметра (Количество отвердителя, м.%): 1
Введите значение параметра (Содержание эпоксидных групп, %_2): 1
Введите значение параметра (Температура вспышки, С_2): 1
Введите значение параметра (Поверхностная плотность, г/м²): 1
Введите значение параметра (Модуль упругости при растяжении, ГПа): 1
Введите значение параметра (Прочность при растяжении, МПа): 1
Введите значение параметра (Потребление смолы, г/м²): 1
Введите значение параметра (Угол нашивки, град): 1
Введите значение параметра (Шаг нашивки): 1
Прогнозное значение параметра Плотность нашивки составляет -2.808001422096492
Для продолжения нажмите любую кнопку, для завершения введите END
END
Для возвращения к выбору параметров нажмите любую клавишу. Для завершения введите N
N

На этом текущий расчёт завершён. При необходимости проведения новых расчётов в рамках заданных параметров – нажимаем ENTER, повторяем ввод других переменных. Для смены исходных установок – ввести команду END (соблюдая регистр), для новых расчётов. Программа завершена.

Заключение

Теоретически разработанный метод определения надёжности изделий из композиционных материалов, основанный на использовании статистически достоверных характеристик материалов, полученных физическим и вычислительным экспериментом, позволяет оценивать уровень надёжности изделий как в отдельных точках, так и по всему объёму в целом.

Список используемой литературы

1. Нишант Шакла. Машинное обучение и TensorFlow [текст]/ Правообладатель: Питер Дата перевода: 2019 Дата написания: 2018
2. Г. М. Гуняев, Композиционные материалы, Большая российская энциклопедия, [Электронный ресурс] — режим доступа https://bigenc.ru/technology_and_technique/text/2087399 (дата обращения 01.04.2022)
3. Sequential model: руководство Сделано в Университете Искусственного Интеллекта [Электронный ресурс] — режим доступа https://ru-keras.com/guide-sequential/#P3_5 (дата обращения 25.03.22)
4. Layer activation functions, Keras documentation [Электронный ресурс] - режим доступа <https://keras.io/api/layers/activations/> (дата обращения 09.03.22)
5. Stack Overflow, question and answer website for professional and enthusiast programmers. [Электронный ресурс] – режим доступа <https://stackoverflow.com> (дата обращения 26.02.22)
6. Stack Overflow ru, Русскоязычное сообщество Stack Overflow [Электронный ресурс] - режим доступа <https://ru.stackoverflow.com> (дата обращения 20.02.22)
7. Pandas, Skill factory blog [Электронный ресурс] - режим доступа <https://blog.skillfactory.ru/glossary/pandas/> (дата обращения 18.02.22)
8. matplotlib.pyplot.ylabel matplotlib documentation [Электронный ресурс] - режим доступа https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.ylabel.html (дата обращения 18.02.22)
9. Tensorflow – полезное иллюстрированное руководство, Автор: Chris Дата записи 26.09.2021 [Электронный ресурс] - режим доступа

- <https://pythobyte.com/tensorflow-overview-83c3bc4c/> (дата обращения 30.02.22)
10. Train_test_split [Электронный ресурс]: – Режим доступа: https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation (дата обращения: 19.03.2022).
11. Sklearn.linear_model.LogisticRegression [Электронный ресурс]: – Режим доступа: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (дата обращения: 15.03.2022).
12. seaborn.heatmap, seaborn [Электронный ресурс]: – Режим доступа: documentation, <http://seaborn.pydata.org/generated/seaborn.heatmap.html> (дата обращения: 16.03.2022).
13. Flask, документация flask (русский перевод) [Электронный ресурс]: – Режим доступа <https://flask-russian-docs.readthedocs.io/ru/latest/> (дата обращения: 21.03.2022).