

**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»

Прогнозирование конечных свойств новых материалов
(композиционных материалов)

Слушатель

Антонов Сергей Анатольевич

Москва, 2022

Оглавление

Срок реализации проекта	3
Введение	4
1. Анализ исходных данных и выбор методов решения	6
1.1 Описание задачи и исходных данных	6
1.2 Разведочный анализ и визуализация исходных данных	7
2. Разработка моделей машинного обучения	15
2.1 Предобработка данных	15
2.2 Разработка и обучение моделей	16
2.3 Тестирование моделей	18
2.4 Разработка нейронной сети	22
2.5 Разработка приложения	24
3. Создание удаленного репозитория и загрузка проекта в него	25
4. Заключение	26
5. Список используемой литературы	27

Срок реализации проекта

Период	Стадия
01.03.2022 – 07.03.2022	1. Изучить теоретические основы и методы решения поставленной задачи.
08.03.2022 – 11.03.2022	2. Провести разведочный анализ предложенных данных. (Необходимо нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек. Необходимо получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков.)
12.03.2022 – 16.03.2022	3. Провести предобработку данных (Необходимо удалить шумы, провести нормализацию).
17.03.2022 – 21.03.2022	4. Обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении. При построении модели необходимо 30% данных оставить на тестирование модели, на остальных происходит обучение моделей.
22.03.2022 – 25.03.2022	5. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель.
26.03.2022 – 30.03.2022	6. Разработать приложение с графическим интерфейсом или интерфейсом командной строки, которое будет выдавать прогноз, полученный на стадии 4 или 5 (один или два прогноза, на выбор).
31.03.2022 – 04.04.2022	7. Оценить точность модели на тренировочном и тестовом датасете.
05.04.2022 – 07.04.2022	8. Создать репозиторий в GitHub / GitLab и разместить код исследования.

Введение

Композитные материалы – многокомпонентные материалы, изготовленные из двух (или более) компонентов с существенно различающимися физическими и/или химическими свойствами, которые, в сочетании, приводят к появлению нового материала с характеристиками, отличными от характеристик отдельных компонентов и не являющимися простой их суперпозицией

В составе композита принято выделять матрицу/матрицы и наполнитель/наполнители. Варьируя состав матрицы и наполнителя, их соотношения, ориентацию наполнителя, можно получить материалы с требуемым сочетанием эксплуатационных и технологических свойств. Многие композиты превосходят традиционные материалы и сплавы по своим механическим свойствам и в то же время они легче. Использование композитов обычно позволяет уменьшить массу конструкции при сохранении или улучшении её механических характеристик.

Композиты, в которых матрицей служит полимерный материал, являются одним из самых многочисленных и разнообразных видов материалов. Их применение в различных областях даёт значительный экономический эффект.

При использовании феноменологического подхода при расчете и проектировании элементов конструкций из композитов даже одного вида материала, сопряжено с проведением трудоемких экспериментальных исследований, из этого следует, что полученные таким образом данные представляют высокую ценность, а их обработка требует дополнительного анализа.

Вопрос построения математической модели состоит в поиске достаточно надежного описания количественных взаимосвязей между свойствами компонентов композита и композитного материала при различных способах их сочетания.

Современным подходом к решению задач такого типа является применение технологий машинного обучения в целях исследования влияния одной или нескольких независимых переменных на зависимую переменную.

Актуальность решения задачи обусловлена широким использованием композитных материалов практически во всех областях производства.

Прогнозирование модели может существенно сократить количество проводимых испытаний, а также пополнить базу данных материалов новыми свойствами материалов, и цифровыми двойниками новых композитов.

1. Анализ исходных данных и выбор методов решения

1.1 Описание задачи и исходных данных

Предметом настоящей работы является построение при помощи методов машинного обучения моделей прогнозирования характеристик «модуль упругости при растяжении» и «прочность при растяжении», рекомендации «соотношение матрица-наполнитель».

Исходные данные о свойствах композиционных материалов и способах их компоновки получены структурным подразделением МГТУ им. Н.Э. Баумана – Центр НТИ «Цифровое материаловедение: новые материалы и вещества» в рамках решения производственных задач.

Соотношения и свойства используемых компонентов композитов (6 входных переменных вещественного типа), а также интересующие выходные характеристики композитов (3 выходных переменных вещественного типа и 7 входных переменных вещественного типа), представлены в виде excel-таблицы, которая содержит 1023 строки и 10 столбцов с данными.

Способы компоновки материалов композитов (3 входных переменных вещественного типа) представлены в виде в виде excel-таблицы, которая содержит 1040 строк и 3 столбца с данными.

Данные таблицы имеют колонку с целочисленным индексом, не являющимся входным или выходным переменным, служащим для сопоставления таблиц данных.

Поставленная задача в рамках классификации категорий машинного обучения относится к машинному обучению с учителем, задача регрессии.

Анализ, предобработка данных, построение моделей выполнены посредством языка программирования Python с использованием библиотек [Pandas, Matplotlib] и Sklearn.

1.2 Разведочный анализ и визуализация исходных данных

Целями разведочного анализа является получение первоначальных представлений о характерах распределений переменных исходного набора данных, формирование оценки качества исходных данных (наличие пропусков, выбросов), выявление характера взаимосвязи между переменными с целью последующего выдвижения гипотез о наиболее подходящих для решения задачи моделях машинного обучения.

Разведочный анализ данных в рамках данной задачи проведен над датафреймом Pandas, полученным путем импорта (и объединения по типу INNER по полю индекса) таблиц исходных данных.

Таблица 1 – Наименования параметров

№п/п	Наименование параметра	Входной/Выходной параметр
1	Соотношение матрица-наполнитель	Выходной
2	Модуль упругости при растяжении, ГПа	Выходной
3	Прочность при растяжении, МПа	Выходной
4	Плотность, кг/м ³	Входной
5	модуль упругости, ГПа	Входной
6	Количество отвердителя, м.%	Входной
7	Содержание эпоксидных групп, % ₂	Входной
8	Температура вспышки, С ₂	Входной
9	Поверхностная плотность, г/м ²	Входной
10	Потребление смолы, г/м ²	Входной
11	Угол нашивки, град	Входной
12	Шаг нашивки	Входной
13	Плотность нашивки	Входной

Сформированный исходный датафрейм содержит 1023 записи с 9 входными параметрами и 3 выходными параметрами вещественного типа, пропуски значений отсутствуют.

Показатели описательной статистики и визуализация гистограмм и/или диаграмм размаха («ящик с усами») позволяют получить наглядное представление о характерах распределений переменных.

Описательная статистика исходных данных описана на рисунке 1.

	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
count	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	44.252151
std	0.913222	73.729231	330.231581	28.295911	2.406301	40.943260	281.314690	3.118983	485.628006	59.735931	45.015751
min	0.389403	1731.764635	2.436909	17.740275	14.254985	100.000000	0.603740	64.054061	1036.856605	33.803026	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
50%	2.906878	1977.621657	739.664328	110.564840	22.230744	285.896812	451.864365	73.268805	2459.524526	219.198882	0.000000
75%	3.552660	2021.374375	961.812526	129.730366	23.961934	313.002106	693.225017	75.356612	2767.193119	257.481724	90.000000
max	5.591742	2207.773481	1911.536477	198.953207	33.000000	413.273418	1399.542362	82.682051	3848.436732	414.590628	90.000000

Рисунок 1 Описательная статистика исходных данных

Построим гистограммы распределения по каждой переменной для оценки повторяющихся значений в многомерном пространстве.

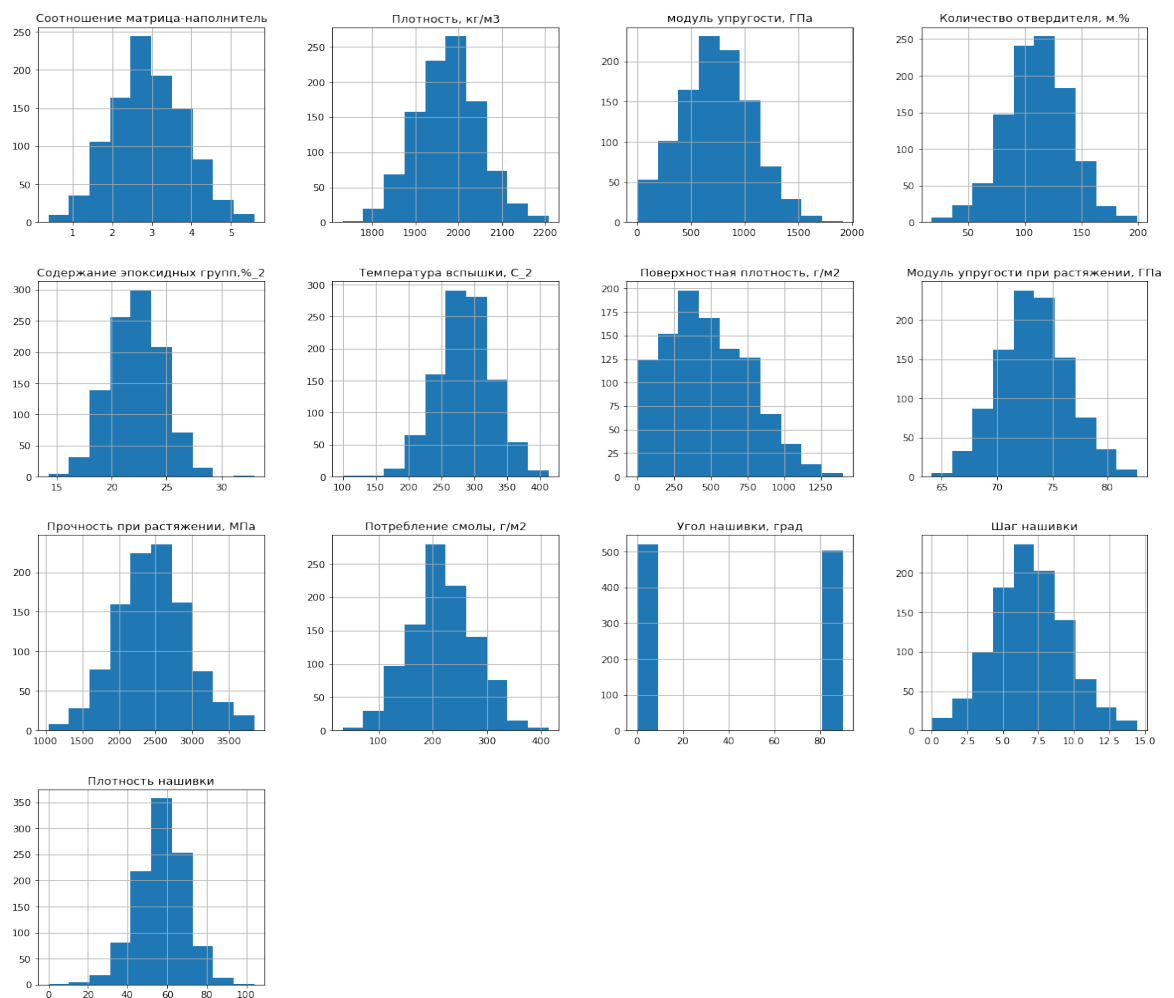


Рисунок 2 Гистограммы распределения

Построим для каждой переменной диаграммы размаха для определения наличия выбросов в данных. Шкалы приведем к величинам в диапазоне $[0,1]$, чтобы "ящики с усами" были одного масштаба. Это поможет нам определить все выбросы и избавиться от них в дальнейшем для того, чтобы набор данных имел более сглаженный вид с точки зрения нормализации.

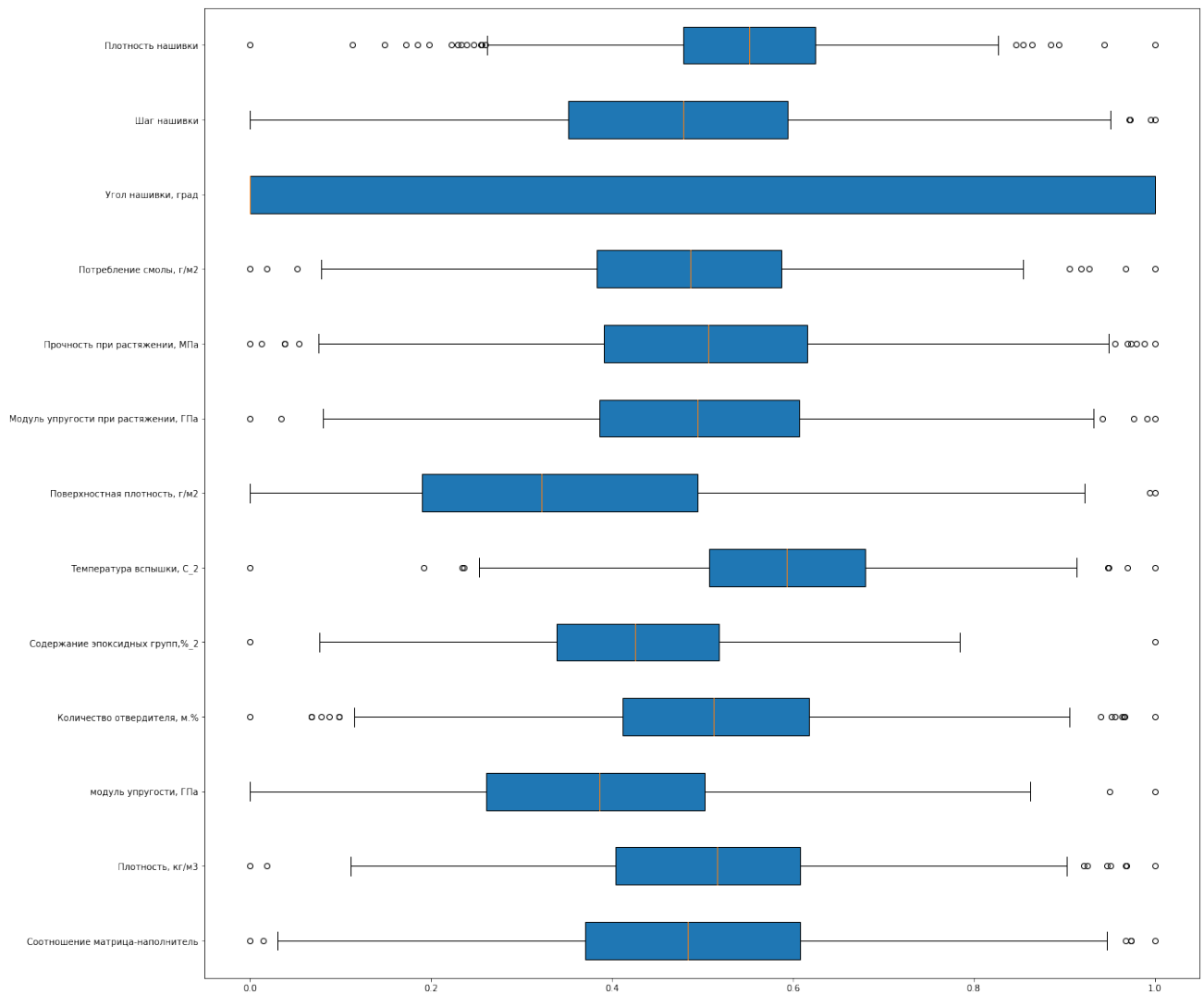


Рисунок 3 Диаграммы размаха

Как видим на рисунке 3 все параметры, кроме «Угла нашивки», представленного всего двумя значениями (0 градусов и 90 градусов), сильнее прочих также выделяется «Поверхностная плотность, г/м2», медиана которого

разнится с выборочным средним сильнее, чем у других параметров, а форма распределения менее других походит на нормальное.

Также «ящички с усами» по каждому из параметров, кроме «Угол нашивки, град», показывают наличие некоторого количества значений, находящихся за пределами полутора межквартильных расстояний от первого и третьего квартилей.

Принимая во внимание, во-первых, источник формирования данных – решение производственных задач (данные измерений), во-вторых то, что нетипичные значения параметров, хотя и находятся за пределами «усов», не демонстрируют экстремально больших отклонений, и в-третьих то, что такие значения присутствуют в том числе и у целевых параметров (а задача исследований в данной области – получение композитов с уникальными свойствами), такие значения вне дополнительных уточнений не следует трактовать как выбросы, по крайней мере, до тех пор, пока их наличие в обучающей и тестовых выборках не будет негативно сказываться на точности предсказаний модели.

Менее радикальным способом оценки качества исходных данных является применение «правила трех сигм».

Описательная статистика и диаграммы размаха датасета после удаления выбросов с применением «правила трех сигм» представлены на рисунках 4 - 6.

	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
count	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000
mean	0.000868	0.587569	0.216978	0.032887	0.006601	0.085021	0.142062	0.021821	0.742763	0.064855	0.013891
std	0.000275	0.056993	0.092124	0.008474	0.000926	0.014082	0.081065	0.002203	0.056655	0.017766	0.013515
min	0.000163	0.444650	0.000709	0.011339	0.004113	0.049402	0.001902	0.016105	0.590461	0.021630	0.000000
25%	0.000679	0.548948	0.151021	0.027292	0.005925	0.075135	0.078825	0.020292	0.706068	0.052063	0.000000
50%	0.000857	0.585227	0.219229	0.032910	0.006589	0.083934	0.138593	0.021720	0.747345	0.064468	0.022461
75%	0.001052	0.626059	0.280808	0.038817	0.007208	0.094452	0.199600	0.023319	0.782554	0.076808	0.026792
max	0.001593	0.743130	0.476145	0.055088	0.009122	0.123083	0.368343	0.027834	0.877580	0.114133	0.034285

Рисунок 4 Описательная статистика датасета после очистки выбросов



Рисунок 5 Гистограммы рассеяния после очистки выбросов

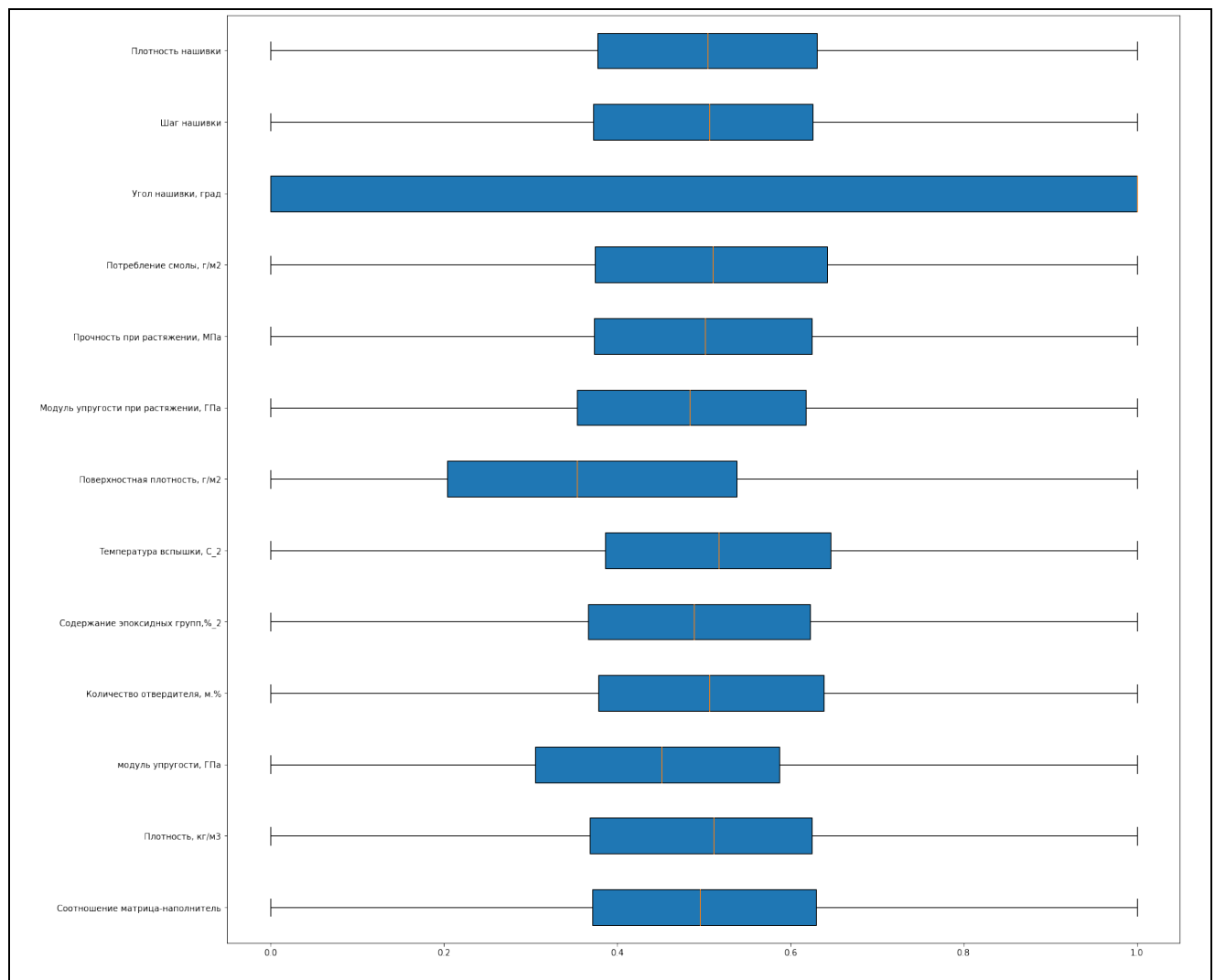


Рисунок 6 Диаграммы размаха после очистки выбросов

После очистки выбросов по «правилу трех сигм» медианы и выборочные средние параметров «подтянулись» ближе друг к другу, за исключением параметра «Поверхностная плотность, г/м2», чья форма распределения, отличная от нормального, также сохранилась.

Построение матрицы корреляции и/или визуализация матрицы рассеивания позволяют получить представление о том, как попарно связаны между собой те или иные параметры.

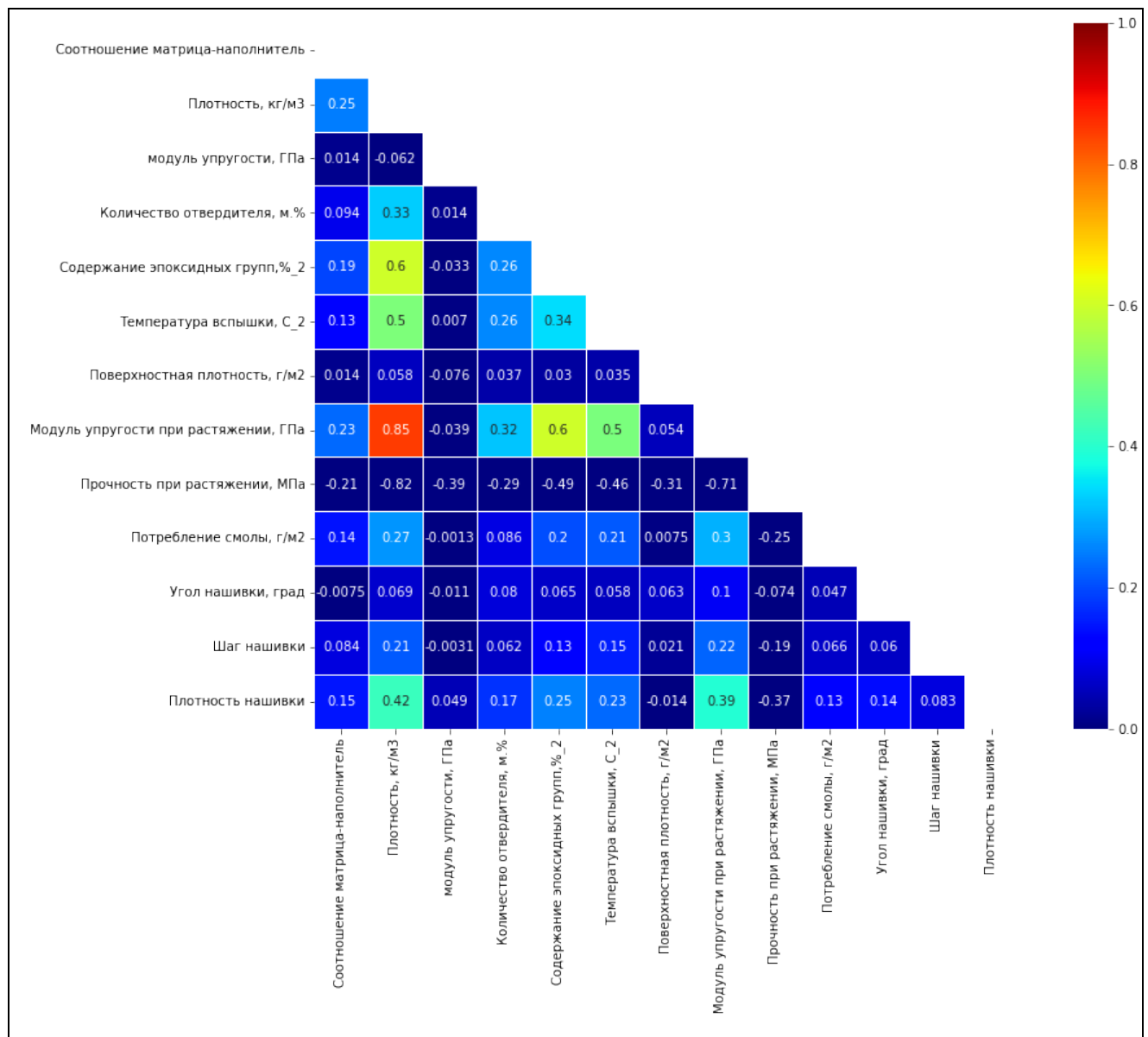


Рисунок 7 Матрица корреляции датасета

Визуализация матрицы корреляции и матрицы рассеяния исходных данных данной задачи, представленная на рисунке 8, показывает около нулевую попарную корреляцию между параметрами и, соответственно, указывают на нелинейный характер связей между ними.

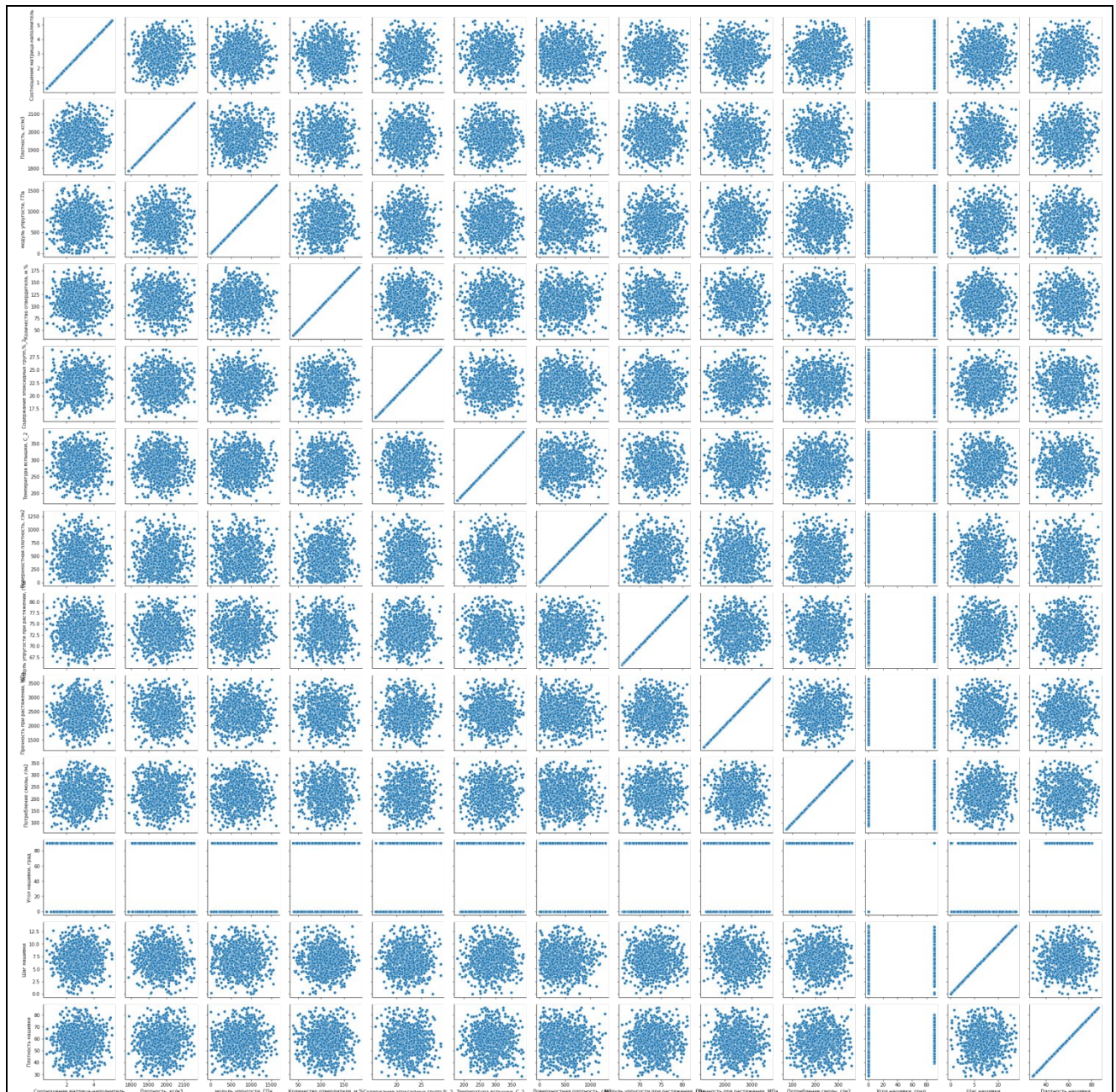


Рисунок 8 Матрица попарной зависимости датасета

В условиях нелинейных зависимостей, для возможности «взглянуть» на данные в совокупности признаков и попытаться проследить какие-то взаимосвязи, следует обратиться к методам обучения на базе многообразий – класс оценщиков без учителя, нацеленных на описание наборов данных, как низкоразмерных многообразий, вложенных в пространство большей размерности.

2 Разработка моделей машинного обучения

2.1 Предобработка данных

Для предобработки данных использовались следующие процедуры:

1. Анализ датасета на пропуски, дубликаты и удаление пропусков, с помощью методов `info()`, `uplicated()` и `describe()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 936 entries, 1 to 1022
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, C_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
dtypes: float64(13)
memory usage: 102.4 KB
```

Рисунок 9 Анализ датасета на пропуски

2. Удаление выбросов из датасета, замена данных, за пределами второго и третьего квантиля на пустые, затем удаление строк, содержащие пустые значения

```
Соотношение матрица-наполнитель          6
Плотность, кг/м3                          9
модуль упругости, ГПа                     2
Количество отвердителя, м.%               14
Содержание эпоксидных групп,%_2          2
Температура вспышки, C_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                         21
dtype: int64
```

Рисунок 10 Количество выбросов по каждому из столбцов

3. Нормализация данных с помощью метода MinMaxScaler и Normalizer из библиотеки sklearn.

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	упругости при растяжении, ГПа	прочность при растяжении, МПа	Потребление смолы, г/м2	Уго нашивки гра
0	0.274768	0.651097	0.452951	0.079153	0.607435	0.509164	0.162230	0.272962	0.727777	0.514688	0.
1	0.274768	0.651097	0.452951	0.630983	0.418887	0.583596	0.162230	0.272962	0.727777	0.514688	0.
2	0.466552	0.651097	0.461725	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.
3	0.465836	0.571539	0.458649	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.
4	0.424236	0.332865	0.494944	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.
...
917	0.361662	0.444480	0.560064	0.337550	0.333908	0.703458	0.161609	0.473553	0.472912	0.183151	1.
918	0.607674	0.704373	0.272088	0.749605	0.294428	0.362087	0.271207	0.462512	0.461722	0.157752	1.
919	0.573391	0.498274	0.254927	0.501991	0.623085	0.334063	0.572959	0.580201	0.587558	0.572648	1.
920	0.662497	0.748688	0.454635	0.717585	0.267818	0.466417	0.496511	0.535317	0.341643	0.434855	1.
921	0.684036	0.280923	0.255222	0.632264	0.888354	0.588206	0.587373	0.552644	0.668015	0.426577	1.

922 rows x 13 columns

Рисунок 10 Нормализация данных с помощью метода MinMaxScaler

Итоговая выборка представляет собой датасет с 922 уникальными строками.

2.2 Разработка и обучение моделей

В данной работе разработка и обучение моделей машинного обучения осуществляется для двух выходных параметров: «Прочность при растяжении» и «Модуль упругости при растяжении». Для каждого признака построение моделей осуществляется отдельно.

Для признака «Прочность при растяжении» были разработаны и обучены следующие модели:

- модель k ближайших соседей (метод KNeighborsRegressor());
- модель на основе градиентного бустинга (метод GradientBoostingRegressor()).

Для признака «Модуль упругости при растяжении» были разработаны и обучены следующие модели:

- модель на основе линейной регрессии (метод LinearRegression);

- модель на основе опорных векторов (метод SVR).

Порядок разработки модели для каждого параметра и для каждого выбранного метода можно разделить на следующие этапы:

1) Разделение нормализованных данных на обучающую и тестовую выборки (в соотношении 70 на 30%, согласно поставленной задаче)

2) Задание сетки гиперпараметров, по которым будет происходить оптимизация модели. В качестве параметра оценки выбран коэффициент детерминации (R^2).

```
#Зададим сетку параметров, по которым будем оптимизировать модель
t_search_1 = {'weights': ['uniform', 'distance'],
              'n_neighbors': list(np.linspace(5, 100, 10, dtype = int)),
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'leaf_size': list(np.linspace(10, 100, 10, dtype = int))}
#В качестве первой модели будем использовать метод ближайших соседей
clf_1 = KNeighborsRegressor()
```

```
#Зададим сетку параметров, по которым будем оптимизировать модель
t_search_2 = {'weights': ['uniform', 'distance'],
              'n_neighbors': list(np.linspace(5, 100, 10, dtype = int)),
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'leaf_size': list(np.linspace(10, 100, 10, dtype = int))}
#В качестве первой модели будем использовать метод ближайших соседей
clf_2 = KNeighborsRegressor()
```

Рисунок 11 Пример определения сетки параметров для модели k ближайших соседей.

3) Оптимизация подбора гиперпараметров модели с помощью выбора по сетке и перекрестной проверки.

4) Подстановка оптимальных гиперпараметров в модель и обучение модели на тренировочных данных.

2.3 Тестирование моделей

После обучения моделей была проведена оценка точности этих моделей на обучающей и тестовых выборках. В качестве параметра оценки модели использовалась средняя абсолютная ошибка (MAE). Для большей наглядности результатов работы модели на тестовых данных, были построены диаграммы рассеяния тестовых данных (реальные данные) и значений, полученных в качестве прогноза.

```
#Подставляем оптимальные гиперпараметры в модель
model_base_1 = KNeighborsRegressor(algorithm='brute', leaf_size=10, n_neighbors=100, weights='distance')
#Обучаем модель
model_base_1.fit(Xtrain1_1,Ytrain1_1)
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate(model_base_1, Xtrain1_1,Ytrain1_1)
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate(model_base_1, Xtest1_1,Ytest1_1)

Средняя абсолютная ошибка: 0.0000
Средняя абсолютная ошибка: 0.1546
```

```
#Подставляем оптимальные гиперпараметры в модель
model_base_2 = KNeighborsRegressor(algorithm='brute', leaf_size=10, n_neighbors=100, weights='distance')
#Обучаем модель
model_base_2.fit(Xtrain2_1,Ytrain2_1)
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate(model_base_2, Xtrain2_1,Ytrain2_1)
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate(model_base_2, Xtest2_1,Ytest2_1)

Средняя абсолютная ошибка: 0.0000
Средняя абсолютная ошибка: 0.0202
```

Рисунок 12 Результаты модели k ближайших соседей для параметра «Прочность при растяжении»

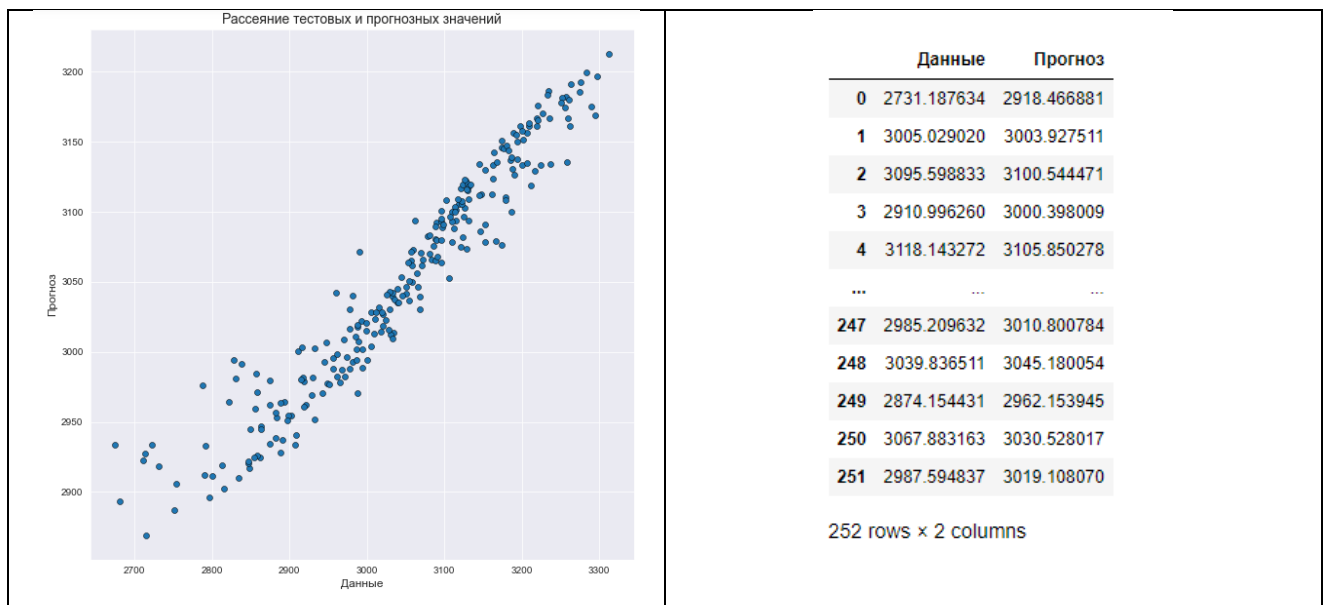


Рисунок 13 Результаты работы модели по оценке значений параметра на основе тестовых данных (Прочность при растяжении)

```
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate_2(model_base_11, Xtrain1_2, Ytrain1_2)
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate_2(model_base_11, Xtest1_2, Ytest1_2)
```

Средняя абсолютная ошибка: 0.1288
Средняя абсолютная ошибка: 0.1655

```
#Подставляем оптимальные гиперпараметры в модель
model_base_22 = GradientBoostingRegressor(loss='lad', max_depth=2)
#Обучаем модель
model_base_22.fit(Xtrain2_2, np.ravel(Ytrain2_2))
```

GradientBoostingRegressor(loss='lad', max_depth=2)

```
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate_2(model_base_22, Xtrain2_2, Ytrain2_2)
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate_2(model_base_22, Xtest2_2, Ytest2_2)
```

Средняя абсолютная ошибка: 0.0050
Средняя абсолютная ошибка: 0.0085

Рисунок 14 Результаты модели повышения градиента для параметра «Прочность при растяжении»

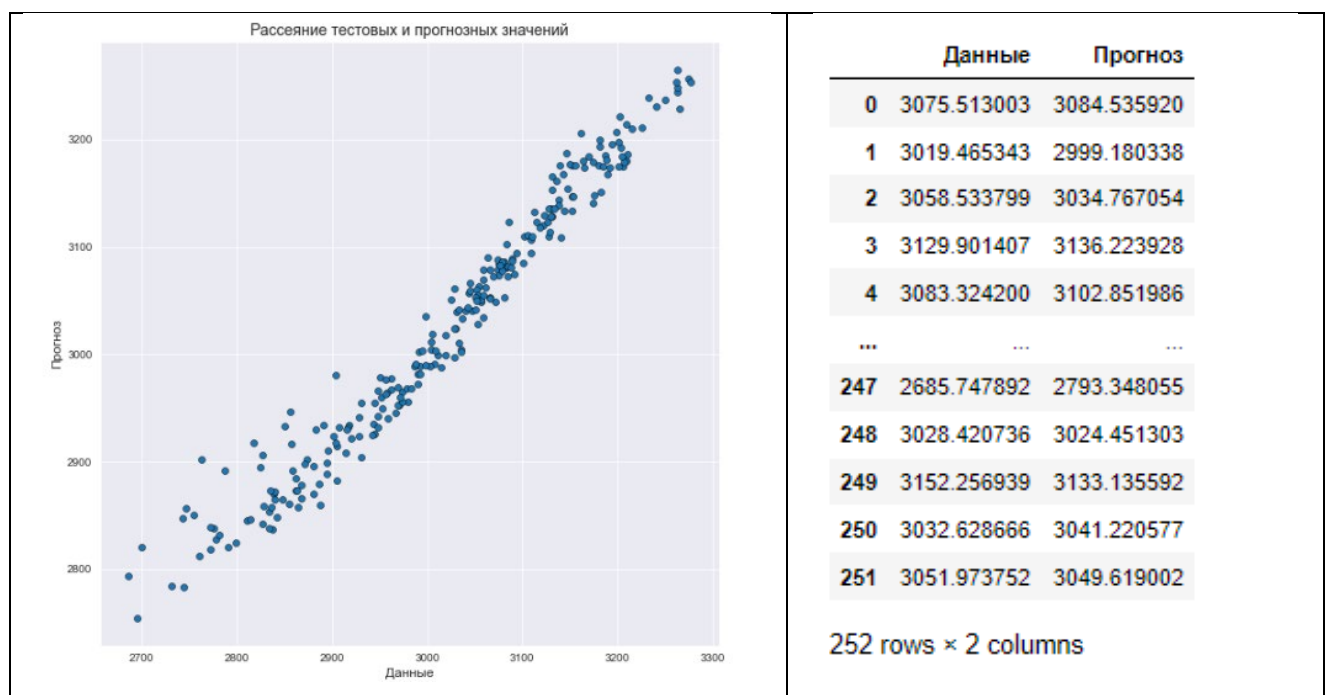


Рисунок 15 Результаты работы модели повышения градиента для параметра «Прочность при растяжении»

```
best_estimator = model11.best_estimator_
#Выводим гиперпараметры для оптимальной модели
print(best_estimator)
#Выводим точность оптимального трейнера
print(model11.best_score_)
```

```
LinearRegression()
-0.026111116626460085
```

```
#Подставляем оптимальные гиперпараметры в модель
model_base11 = LinearRegression()
#Обучаем модель
model_base11.fit(Xtrain11,Ytrain11)
```

```
LinearRegression()
```

```
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate(model_base11, Xtrain11,Ytrain11)
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate(model_base11, Xtest11, Ytest11)
```

Средняя абсолютная ошибка: 0.1608

Средняя абсолютная ошибка: 0.1509

Рисунок 16 Результаты модели LinearRegression для параметра «Модуль упругости при растяжении»

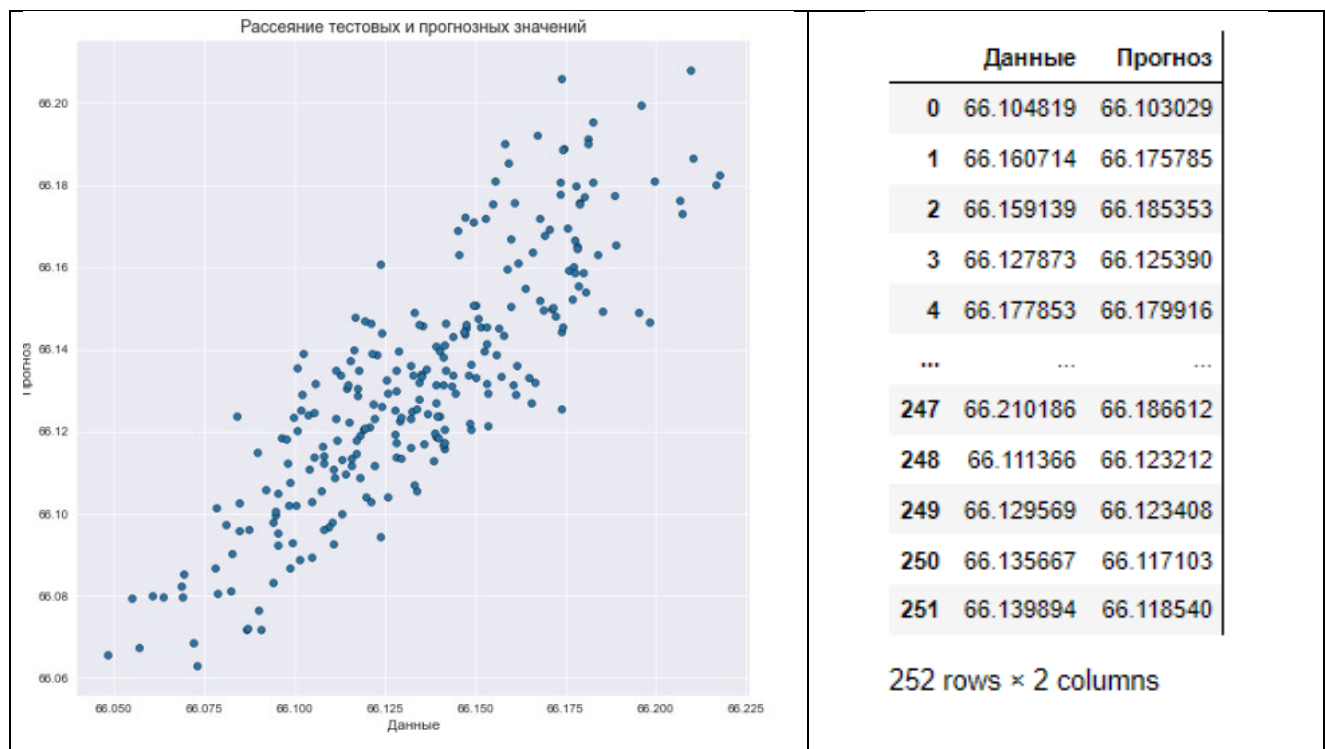


Рисунок 17 Результаты работы модели LinearRegression для параметра «Модуль упругости при растяжении»

```
best_estimator = model111.best_estimator_
#Выводим гиперпараметры для оптимальной модели
print(best_estimator)
#Выводим точность оптимального трейнера
print(model111.best_score_)
```

```
SVR(kernel='linear')
-0.0304111012748157
```

```
#Подставляем оптимальные гиперпараметры в модель
model_base111 = SVR(kernel='linear')
#Обучаем модель
model_base111.fit(Xtrain111,np.ravel(Ytrain111))
```

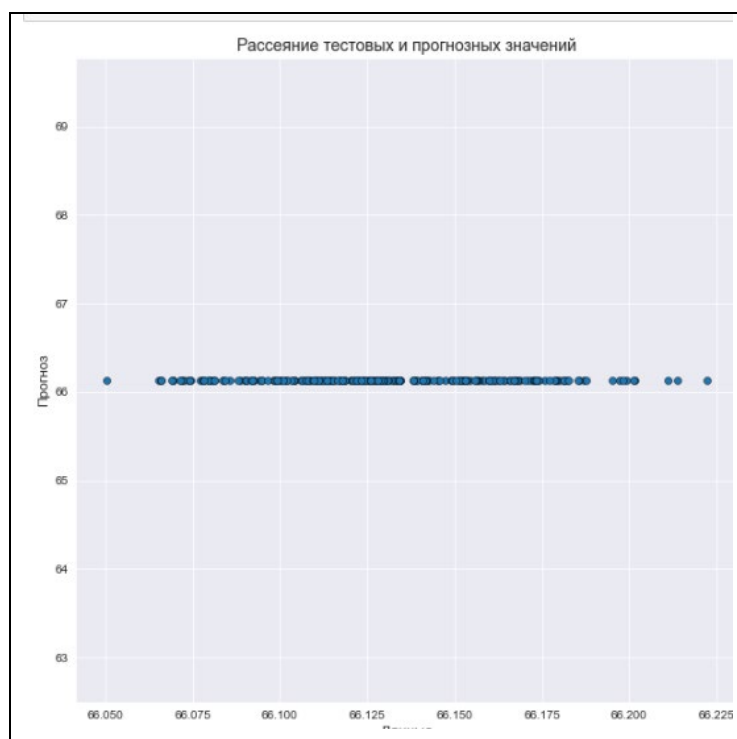
```
SVR(kernel='linear')
```

```
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate_2(model_base111, Xtrain111, Ytrain111)
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate_2(model_base111,Xtest111, Ytest111)
```

Средняя абсолютная ошибка: 0.1581

Средняя абсолютная ошибка: 0.1585

Рисунок 18 Результаты модели регрессии опорных векторов (SVR) для параметра «Модуль упругости при растяжении»



	Данные	Прогноз
0	66.123800	66.132373
1	66.151416	66.132373
2	66.167024	66.132373
3	66.161214	66.132373
4	66.173399	66.132373
...
247	66.126000	66.132373
248	66.150265	66.132373
249	66.081114	66.132373
250	66.201315	66.132373
251	66.123064	66.132373

252 rows × 2 columns

Рисунок 19 Результаты работы регрессии опорных векторов (SVR) для параметра «Модуль упругости при растяжении»

2.5 Разработка нейронной сети

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

Модель состоит из трех скрытых уровней. Первый уровень содержит 64 нейрона, что немногим более чем в три раза превышает объем входных данных (10 входных переменных). Последующие скрытые уровни содержат 64 и 1 нейрон. Снижение числа нейронов на каждом уровне сжимает информацию, которую сеть обработала на предыдущих уровнях.

Скрытые уровни нейронной сети трансформируются функциями активации. Эти функции являются важными элементами сетевой инфраструктуры, так как они вносят в систему нелинейность.

Для эксперимента были выбраны три функции активации:

1. \tanh (арктангенс),
2. relu (выпрямленная линейная единица),
3. sigmoid (сигмоида $1/(1+\exp(-x))$)

```
def build_model1():  
    model1=models.Sequential()  
    model1.add(layers.Dense(64, activation='tanh', input_shape=(X1trn1.shape[1],)))  
    model1.add(layers.Dense(64, activation='tanh'))  
    model1.add(layers.Dense(1))  
    model1.compile(optimizer='rmsprop', loss='mse', metrics=['mae'])  
    return model1
```

Рисунок 20 Архитектура нейронной сети

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). Данная функция вычисляет среднее квадратичное отклонение между предсказаниями и целями.

В качестве оптимизатора использовался RMSprop-оптимизатор, алгоритм которого похож на метод градиентного спуска с импульсом. Оптимизатор RMSprop ограничивает колебания в вертикальном направлении

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

Средняя абсолютная ошибка: 0.171.

На рисунках 21-23 представлены результаты прогноза модели на тестовых данных, по аналогии с результатами для моделей машинного обучения, описанных в предыдущем разделе.



Рисунок 21 Прогнозные данные для модели с функцией tanh



Рисунок 22 Прогнозные данные для модели с функцией relu



Рисунок 23 Прогнозные данные для модели с функцией sigmoid

2.6 Разработка приложения

Приложение разработано с Web-интерфейсом и позволяет решать задачи прогнозирования целевой переменной на основе входных данных.

Прогнозное значение параметра «Соотношение матрица-наполнитель»

Результат

Плотность, кг/м3	<input type="text"/>
модуль упругости, ГПа	<input type="text"/>
Количество отвердителя, м. %	<input type="text"/>
Содержание эпоксидных групп, %_2	<input type="text"/>
Температура вспышки, С_2	<input type="text"/>
Поверхностная плотность, г/м2	<input type="text"/>
Потребление смолы, г/м2	<input type="text"/>
Угол нашивки, град	<input type="text"/>
Шаг нашивки	<input type="text"/>
Плотность нашивки	<input type="text"/>
<input type="button" value="Отправить"/>	

Рисунок 24 Web-интерфейс приложения

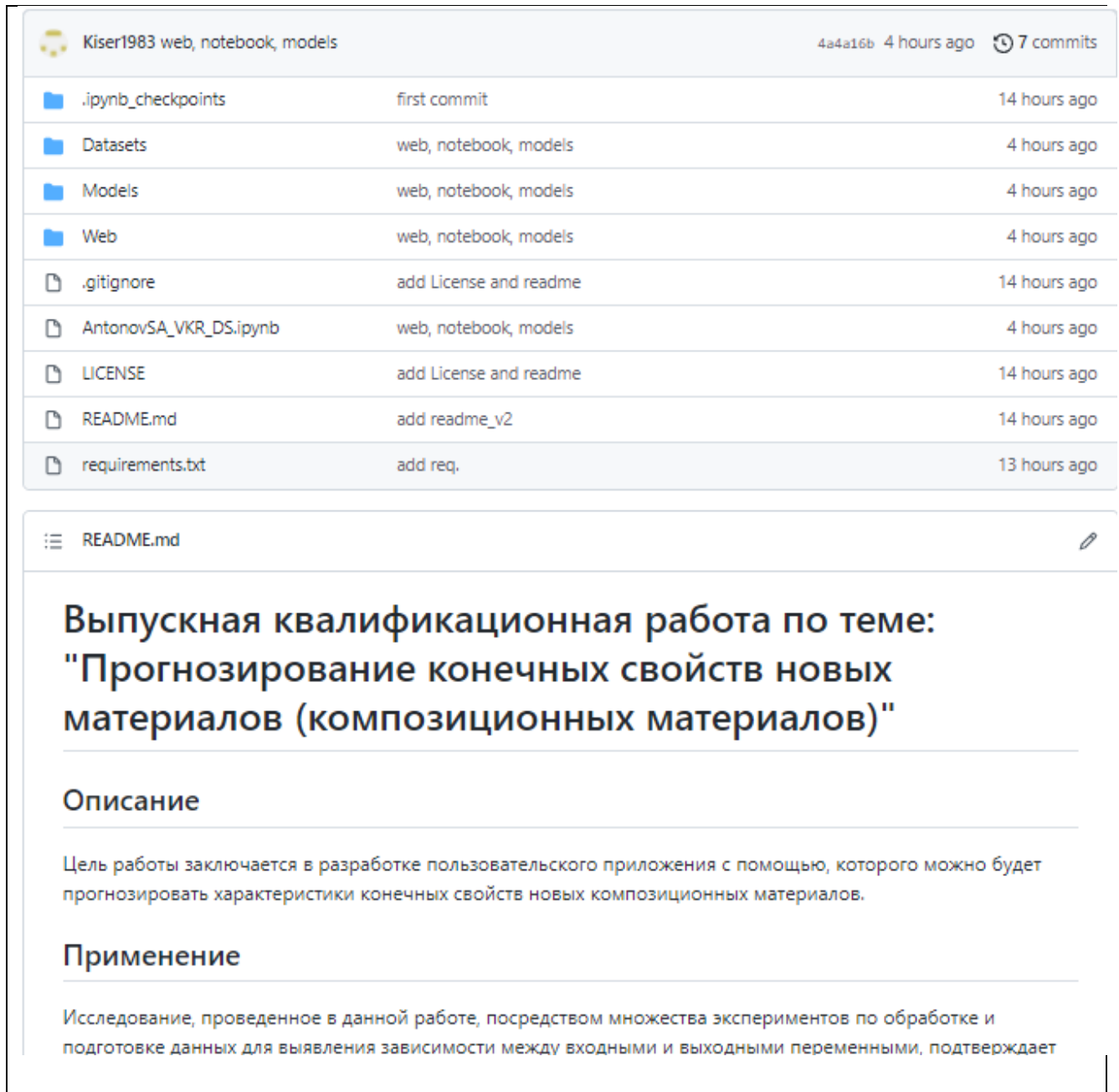
После ввода всех переменных выдается прогнозное значение параметра «Соотношение матрица-наполнитель», сформированного моделью нейронной сети.

3. Создание удаленного репозитория и загрузка файлов в него

Страница создана на GitHub.

Адрес страницы https://github.com/Kiser1983/Antonov_VKR_DS

В репозитории находятся: файлы тетрадок Юпитера, наборы данных, модели, приложение, ВКР в текстовом формате.



The screenshot shows a GitHub repository page for 'Kiser1983 web, notebook, models'. The repository has 4a4a16b commit, 4 hours ago, and 7 commits. The file list includes:

File	Commit	Time
.ipynb_checkpoints	first commit	14 hours ago
Datasets	web, notebook, models	4 hours ago
Models	web, notebook, models	4 hours ago
Web	web, notebook, models	4 hours ago
.gitignore	add License and readme	14 hours ago
AntonovSA_VKR_DS.ipynb	web, notebook, models	4 hours ago
LICENSE	add License and readme	14 hours ago
README.md	add readme_v2	14 hours ago
requirements.txt	add req.	13 hours ago

The README.md file is selected, showing the following content:

Выпускная квалификационная работа по теме: "Прогнозирование конечных свойств новых материалов (композиционных материалов)"

Описание

Цель работы заключается в разработке пользовательского приложения с помощью, которого можно будет прогнозировать характеристики конечных свойств новых композиционных материалов.

Применение

Исследование, проведенное в данной работе, посредством множества экспериментов по обработке и подготовке данных для выявления зависимости между входными и выходными переменными, подтверждает

Рисунок 25 Страница репозитория

4. 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.

5. Список используемой литературы

1. Труды ИСА РАН: Математические модели социально-экономических процессов. Динамические системы. Управление рисками и безопасностью. Оптимизация, идентификация, теория игр. Обработка и анализ изображений и сигналов. Интеллектуальный анализ данных и распознавание / Под ред. С.В. Емельянова. - М.: Красанд, 2013. - 128 с.
2. Искусственный интеллект и принятие решений: Интеллектуальный анализ данных. Моделирование поведения. Когнитивное моделирование. Моделирование и управление / Под ред. С.В. Емельянова. - М.: Ленанд, 2012. - 108 с.
3. Информационные технологии и вычислительные системы: Обработка информации и анализ данных. Программная инженерия. Математическое моделирование. Прикладные аспекты информатики / Под ред. С.В. Емельянова. - М.: Ленанд, 2015. - 104 с.
4. Комплексный анализ электромагнитных и других геофизических данных. (С цветной вкладкой) / Под ред. В.В. Спичака. - М.: Красанд, 2011. - 192 с.
5. Айзек, М.П. Графика, формулы, анализ данных в Excel. Пошаговые примеры / М.П. Айзек. - СПб.: Наука и техника, 2019. - 384 с.
6. Айзек, М.П. Вычисления, графики и анализ данных в Excel 2010: Самоучитель / М.П. Айзек, В.В. Серогодский, М.В. Финков. - СПб.: НиТ, 2013. - 352 с.
7. Айзек, М.П. Вычисления, графики и анализ данных в Excel 2013. Самоучитель / М.П. Айзек. - СПб.: Наука и техника, 2015. - 416 с.
8. Айзек, М.П. Вычисления, графики и анализ данных в Excel 2010. Самоучитель / М.П. Айзек. - СПб.: Наука и техника, 2013. - 352 с.
9. Бенгфорт, Б. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка / Б. Бенгфорт. - СПб.: Питер, 2016. - 400 с.

10. Бенгфорт, Б. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка / Б. Бенгфорт. - СПб.: Питер, 2019. - 368 с.
11. Бергер, А. Microsoft SQL Server 2005 Analysis Services. OLAP и многомерный анализ данных / А. Бергер. - СПб.: BHV, 2007. - 928 с.
12. Боровиков, В.П. Популярное введение в современный анализ данных в системе STATISTICA: Учебное пособие для вузов / В.П. Боровиков. - М.: ГЛТ, 2013. - 288 с.
13. Боровиков, В.П. Популярное введение в современный анализ данных в системе Statistica: Учебное пособие / В.П. Боровиков. - М.: ГЛТ, 2013. - 288 с.
14. Боровиков, В.П. Популярное введение в современный анализ данных в системе STATISTICA: Учебное пособие для вузов / В.П. Боровиков. - М.: РиС, 2015. - 288 с.
15. Крянев, А.В. Метрический анализ и обработка данных / А.В. Крянев, Г.В. Лукин, Д.К. Удумян. - М.: Физматлит, 2012. - 308 с.
16. Винстон, У. Бизнес-моделирование и анализ данных. Решение актуальных задач с помощью Microsoft Excel / У. Винстон. - СПб.: Питер, 2006. - 320 с.
17. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD / Ю.Е. Воскобойников. - СПб.: Лань, 2011. - 224 с.
18. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете Mathcad: Учебное пособие / Ю.Е. Воскобойников. - СПб.: Лань, 2011. - 224 с.
19. Горяинова, Е.Р. Прикладные методы анализа статистических данных: Учебное пособие / Е.Р. Горяинова, А.Р. Панков, Е.Н. Платонов. - М.: ИД ГУ ВШЭ, 2012. - 310 с.
20. Дайитбегов, Д.М. Компьютерные технологии анализа данных в эконометрике: Монография / Д.М. Дайитбегов. - М.: Вузовский учебник, НИЦ Инфра-М, 2013. - 587 с.

21. Есаулов, И.Г. Регрессионный анализ данных в пакете Mathcad: Учебное пособие / И.Г. Есаулов. - СПб.: Лань П, 2016. - 224 с.
22. Кабаков, Р. R в действии. Анализ и визуализация данных в программе R / Р. Кабаков. - М.: ДМК, 2016. - 588 с.
23. Калинина, В.Н. Анализ данных. компьютерный практикум (для бакалавров) / В.Н. Калинина, В.И. Соловьев. - М.: КноРус, 2017. - 240 с.
24. Кацко, И.А. Практикум по анализу данных на компьютере / И.А. Кацко, Н.Б. Паклин. - М.: КолосС, 2009. - 278 с.
25. Козлов, А. Статистический анализ данных в MS Excel: Учебное пособие / А. Козлов. - М.: Инфра-М, 2012. - 320 с.
26. Козлов, А.Ю. Статистический анализ данных в MS Excel: Учебное пособие / А.Ю. Козлов, В.С. Мхитарян, В.Ф. Шишов. - М.: Инфра-М, 2018. - 80 с.
27. Корячко, В.П. Анализ и проектирование маршрутов передачи данных в корпоративных сетях / В.П. Корячко, Д.А. Перепелкин. - М.: ГЛТ , 2012. - 236 с.
28. Лесковец, Ю. Анализ больших наборов данных / Ю. Лесковец, А. Раджараман. - М.: ДМК, 2016. - 498 с.
29. В.В. Васильев, В.Д. Протасов, В.В. Болотин и др.: Композитные материалы: справочник. Москва: Машиностроение, 1990, 510 с.
30. Плас Дж. Вандер, Python для сложных задач: наука о данных и машинное обучение. Санкт-Петербург: Питер, 2018, 576 с.
31. Язык программирования Python: - Режим доступа: <https://www.python.org/> (дата обращения 01.04.2022)
32. Библиотека Pandas – Режим доступа: <https://pandas.pydata.org/> (дата обращения 01.04.2022)
33. Библиотека Matplotlib – Режим доступа: [https:// matplotlib.org/](https://matplotlib.org/) (дата обращения 01.04.2022)

34. Библиотека Sklearn – Режим доступа: <https://scikit-learn.org/stable/>
(дата обращения 01.04.2022)

35. К. Андерсон, Аналитическая культура. От сбора данных до бизнес-результатов: монография. Москва: O'Reilly, 2017, 392 с.

36. How to choose a machine learning model in Python? – Режим доступа: <https://www.codestar.com/choose-machine-learning-models-python/>(дата обращения 03.04.2022)