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»

Тема: «Прогнозирование конечных свойств новых материалов (композиционных материалов)»

Слушатель Дьяченко Ю.А.

Москва, 2023

Содержание

Введение

Предметом исследования в данной работе выступают композитные материалы. Это искусственно созданные материалы, применяемые во многих сферах деятельности: строительстве, авиационной промышленности, медицине, автомобилестроении и многих других.

Композитные материалы – это многокомпонентные материалы, изготовленные из двух или более компонентов с существенно различными физическими или химическими свойствами, которые приводят к появлению нового материала с характеристиками, отличными от характеристик отдельных компонентов.

Сложность получения новых композитов заключается в прогнозировании свойств будущих материалов, поэтому для упрощения и удешевления исследований можно применять машинное обучение. Прогнозные модели помогут уменьшить количество проводимых испытаний.

В этой работе воспроизведено исследование с анализом данных, а также созданы модели и нейросеть с набором различных параметров для прогнозирования конечных свойств новых композитных материалов. На основе работы нейросети создано приложение, позволяющее пользователю получить вариант прогноза Соотношения «Матрица/Наполнитель».

1. Аналитическая часть

1.1 Постановка задачи

Цель прогнозирования заключается в симуляции представительного элемента объема композита на основе данных о характеристиках входящих компонентов (связующего и армирующего компонента). Необходимо создать и обучить модели и нейронные сети.

Входные данные состоят из двух датасетов. В файле с физическими характеристиками базальтопластика (X_bp.xlsx) содержится 1023 строки и 10 признаков:

- Соотношение матрица-наполнитель;
- Плотность;
- Модуль упругости;
- Количество отвердителя;
- Содержание эпоксидных групп;
- Температура вспышки;
- Поверхностная плотность;
- Модуль упругости при растяжении;
- Прочность при растяжении;
- Потребление смолы.

Таблица представлена на рисунке 1.

```
# просмотр данных,содержащихся в датасете
df_bp.sample(5)
```

	Unnamed: 0	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
490	490	2.438256	1832.019730	449.761027	89.995476	24.096229	254.541470	466.376124	75.891243	2456.393368	97.026947
35	35	3.247617	1813.234600	757.874479	81.379871	23.422465	279.080157	575.062857	69.341133	3188.136358	252.870569
208	208	4.706654	1938.949730	1014.173382	35.620904	24.829676	328.864659	379.246075	74.445896	3063.025225	164.547578
271	271	3.524311	1972.053816	677.704062	95.764072	26.544123	252.053613	112.459564	76.540169	2229.631180	225.957383
908	908	2.656357	2059.639384	914.508236	152.474608	24.429749	353.806349	710.848687	73.264735	2354.830599	208.378508

```
# размерность датасета
df_bp.shape
```

```
(1023, 11)
```

Рисунок 1 – X_bp, характеристики базальтопластика

Файл с геометрическими характеристиками нашивки углепластика (X_nup.xlsx) содержит 1040 строк и 3 параметра:

- Угол нашивки;
- Шаг нашивки;
- Плотность нашивки.

Таблица представлена на рисунке 2.

	Unnamed: 0	Угол нашивки, град	Шаг нашивки	Плотность нашивки
517	517	0	7.260030	57.094641
420	420	0	9.785548	56.094595
44	44	0	8.325699	46.045428
552	552	90	7.863236	53.551282
1001	1001	90	10.450893	71.574356

```
# размерность датасета
df_up.shape
```

```
(1040, 4)
```

Рисунок 2 – X_nup, характеристики углепластика

По заданию, таблицы были объединены в один датасет по индексу типом объединения INNER. Пример полученного датасета на рисунке 3.

	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	1.857143	2030.000000	738.736842	30.000000	22.267857	100.000000	210.000000	70.000000	3000.000000	220.000000	0	4.000000	57.000000
1	1.857143	2030.000000	738.736842	50.000000	23.750000	284.615385	210.000000	70.000000	3000.000000	220.000000	0	4.000000	60.000000
2	1.857143	2030.000000	738.736842	49.900000	33.000000	284.615385	210.000000	70.000000	3000.000000	220.000000	0	4.000000	70.000000
3	1.857143	2030.000000	738.736842	129.000000	21.250000	300.000000	210.000000	70.000000	3000.000000	220.000000	0	5.000000	47.000000
4	2.771331	2030.000000	753.000000	111.860000	22.267857	284.615385	210.000000	70.000000	3000.000000	220.000000	0	5.000000	57.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1018	2.271346	1952.087902	912.855545	86.992183	20.123249	324.774576	209.198700	73.090961	2387.292495	125.007669	90	9.076380	47.019770
1019	3.444022	2050.089171	444.732634	145.981978	19.599769	254.215401	350.660830	72.920827	2360.392784	117.730099	90	10.565614	53.750790
1020	3.280604	1972.372865	416.836524	110.533477	23.957502	248.423047	740.142791	74.734344	2662.906040	236.606764	90	4.161154	67.629684
1021	3.705351	2066.799773	741.475517	141.397963	19.246945	275.779840	641.468152	74.042708	2071.715856	197.126067	90	6.313201	58.261074
1022	3.808020	1890.413468	417.316232	129.183416	27.474763	300.952708	758.747882	74.309704	2856.328932	194.754342	90	6.078902	77.434468

1023 rows × 13 columns

Рисунок 3 – Объединённый датасет

Выходными переменными по объединённому датасету являются:

- Соотношение матрица-наполнитель;
- Модуль упругости при растяжении, ГПа;
- Прочность при растяжении, Мпа.

Для каждого параметра необходимо вычислить среднее и медианное значения, произвести анализ и исключение выбросов, проверить наличие пропусков и провести предварительную обработку данных, включая удаление выбросов. Так же следует применить нормализацию и стандартизацию, и приступить к обучению моделей для выходных переменных. После, написать нейронную сеть, которая будет рекомендовать соотношение «матрица/наполнитель» и разработать по ней приложение с графическим интерфейсом. Затем нужно оценить точность модели на тренировочном и тестовом датасете, а также создать репозиторий в GitHub и разместить код исследования.

1.2 Описание используемых методов

В рамках классификации категорий машинного обучения, задача данной работы относится к машинному обучению с учителем и, чаще всего, это задача регрессии.

Задача регрессии в машинном обучении – это предсказание параметра Y по известному параметру X, то есть, модель прогнозирует значение метки по набору связанных компонентов. Метка здесь может принимать любое значение, а не просто выбирается из конечного набора значений, как в задачах классификации.

Алгоритмы регрессии моделируют зависимость меток от связанных компонентов, чтобы определить закономерности изменения меток при разных значениях компонентов. На вход алгоритма регрессии подается набор примеров с метками известных значений. Результатом работы алгоритма регрессии

является функция, которая умеет прогнозировать значения метки для любого нового набора входных компонентов.

Для наилучшего решения в процессе исследования были применены следующие методы:

- Случайный лес;
- Линейная регрессия;
- Градиентный бустинг.

Случайный лес (RandomForest) — это множество решающих деревьев. Универсальный алгоритм машинного обучения с учителем, представитель ансамблевых методов. Если точность дерева решений оказалось недостаточной, мы можем множество моделей собрать в коллектив.

Достоинства метода: не переобучается; не требует предобработки входных данных; эффективно обрабатывает пропущенные данные, данные с большим числом классов и признаков; имеет высокую точность предсказания и внутреннюю оценку обобщающей способности модели, а также высокую параллелизуемость и масштабируемость.

Недостатки метода: построение занимает много времени; сложно интерпретируемый; не обладает возможностью экстраполяции; может недообучаться; трудоёмко прогнозируемый; иногда работает хуже, чем линейные методы.

Линейная регрессия (Linear regression) — это алгоритм машинного обучения, основанный на контролируемом обучении, рассматривающий зависимость между одной входной и выходными переменными. Это один из самых простых и эффективных инструментов статистического моделирования. Она определяет зависимость переменных с помощью линии наилучшего соответствия. Модель регрессии создаёт несколько метрик. R^2, или коэффициент детерминации, позволяет измерить, насколько модель может объяснить дисперсию данных. Если R^2 равен 1, это значит, что модель описывает все данные. Если же R^2 равен 0,5, модель объясняет лишь 50 процентов дисперсии

данных. Оставшиеся отклонения не имеют объяснения. Чем ближе R^2 к единице, тем лучше.

Достоинства метода: быстрый и простой в реализации; легко интерпретируем; имеет меньшую сложность по сравнению с другими алгоритмами.

Недостатки метода: моделирует только прямые линейные зависимости; требует прямую связь между зависимыми и независимыми переменными; выбросы оказывают огромное влияние, а границы линейны.

Градиентный бустинг (Gradient Boosting) — это ансамбль деревьев решений, обученный с использованием градиентного бустинга. В основе данного алгоритма лежит итеративное обучение деревьев решений с целью минимизировать функцию потерь. Основная идея градиентного бустинга: строятся последовательно несколько базовых классификаторов, каждый из которых как можно лучше компенсирует недостатки предыдущих. Финальный классификатор является линейной композицией этих базовых классификаторов.

Достоинства метода: новые алгоритмы учатся на ошибках предыдущих; требуется меньше итераций, чтобы приблизиться к фактическим прогнозам; наблюдения выбираются на основе ошибки; прост в настройке темпа обучения и применения; легко интерпретируем.

Недостатки метода: необходимо тщательно выбирать критерии остановки, иначе это может привести к переобучению; наблюдения с наибольшей ошибкой появляются чаще; слабее и менее гибок, чем нейронные сети.

Для решения задачи создания рекомендательной системы был использован метод многослойного персептрона. Многослойный персептрон — частный случай персептрона Розенблатта, в котором один алгоритм обратного распространения ошибки обучает все слои.

Особенностью является наличие более чем одного обучаемого слоя (как правило — два или три). Необходимость в большом количестве обучаемых слоёв отпадает, так как теоретически единственного скрытого слоя достаточно, чтобы

перекодировать входное представление таким образом, чтобы получить линейную разделимость для выходного представления.

Все вышеперечисленные задачи в данной работе решены на языке Python с использованием библиотек Pandas, NumPy, Matplotlib, Seaborn и Tensorflow.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Pandas — это библиотека машинного обучения, представляющая структуры данных высокого уровня и большой диапазон инструментов для анализа. Отличительной чертой Pandas считается возможность переводить сложнейшие операции с информацией, используя всего одну либо две команды. Данная библиотека содержит массу способов для объединения данных, их группировки и фильтрации.

К особенностям Pandas относятся:

- возможность упростить манипуляции данными;

- поддержка сортировки, визуализации и прочих опций.

Pandas обеспечивает широкую гибкость, функциональность, если эксплуатировать ее с иными библиотеками.

NumPy — основная библиотека Python, которая упрощает работу с векторами и матрицами. Содержит готовые методы для разных математических операций: от создания, изменения формы, умножения и расчета детерминант матриц, до решения линейных уравнений и сингулярного разложения. Это значительно повышает производительность и, соответственно, ускоряет время выполнения работы.

Matplotlib — низкоуровневая библиотека для создания двумерных диаграмм и графиков. С ее помощью можно отображать широкий спектр визуализаций: линейные и точечные диаграммы, диаграммы с областями,

гистограммы, круговые диаграммы, диаграммы «стебель-листья», контурные графики, поля векторов и спектрограммы.

Seaborn — библиотека более высокого уровня, чем matplotlib. С ее помощью проще создавать специфическую визуализацию: тепловые карты, временные ряды и скрипичные диаграммы.

TensorFlow — это библиотека AI, которая помогает разработчикам создавать крупномасштабные нейронные сети со многими слоями, используя графики потоков данных. TensorFlow также облегчает построение моделей глубокого обучения, продвигает современную технологию ML / AI и позволяет легко развертывать приложения на базе ML. TensorFlow достаточно эффективен, когда дело доходит до классификации, восприятия, понимания, обнаружения, прогнозирования и создания данных.

Scikit-learn – библиотека, которая основана на NumPy и SciPy. В ней есть алгоритмы для машинного обучения и интеллектуального анализа данных: кластеризации, регрессии и классификации.

Особенностями Scikit-Learn являются: возможность извлечения элементов из текстов и картинок; перекрестная проверка – множество различных методов проверки точности контролируемой модели на невидимой информации; большое количество алгоритмов машинного обучения; возможность осуществления дорогостоящих задач.

1.3 Разведочный анализ данных

Прежде чем данные передать в работу моделей машинного обучения, необходимо обработать и очистить их. Необработанные данные могут содержать искажения и пропущенные значения, что может негативно отразиться на работе моделей.

Цель разведочного анализа – 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.

В качестве инструментов разведочного анализа используется: оценка статистических характеристик датасета; гистограммы распределения каждой из переменной; диаграммы ящика с усами; попарные графики рассеяния точек; график «квантиль-квантиль»; тепловая карта; описательная статистика для каждой переменной; анализ и полное исключение выбросов; проверка наличия пропусков и дубликатов; ранговая корреляция Пирсона.

Команда df.info() выводит общую информацию о датасете: количество строк и столбцов, количество значений, название переменных, тип данных. Результат команды представлен на рисунке 4.

```
df.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
```

Рисунок 4 – Общая информация о датасете

Чтобы получить развернутые сведения о данных можно воспользоваться возможностями Pandas Profiling.

Pandas Profiling – это библиотека с открытым исходным кодом, которая может создавать интерактивные отчеты для любого набора данных с помощью всего одной строчки кода. Pandas Profiling создает отчеты профиля из pandas DataFrame. Отчет состоит из следующих 6 блоков информации.

Первая часть отчёта, изображенная на рисунке 5, содержит раздел Overview (Обзор), дающий основные сведения о данных: количество наблюдений; количество переменных; тип данных; процент и количество пропущенных значений; процент и количество дубликатов. Кроме того, он будет содержать список предупреждений, уведомляющий аналитика о том, на что стоит обратить особое внимание, но в данном примере таких предупреждений не появилось.

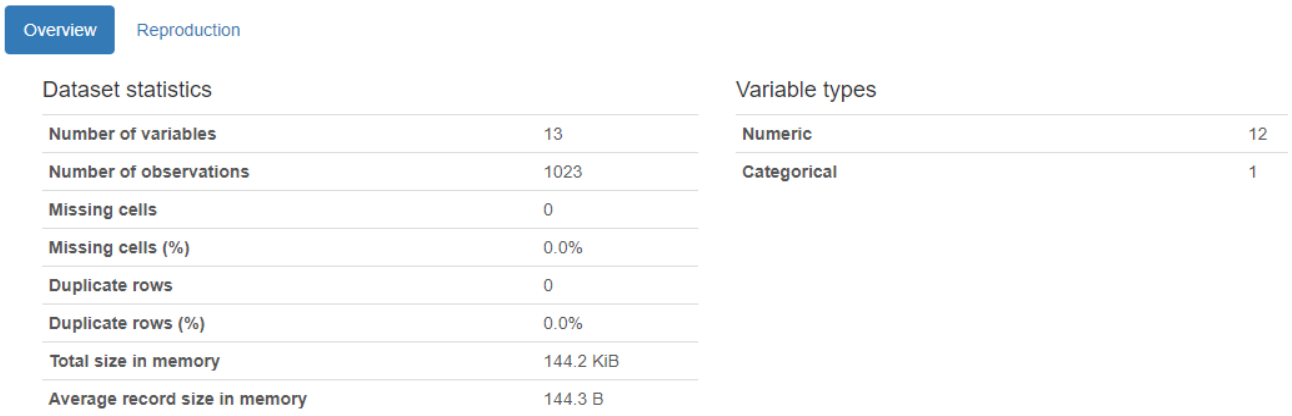

Рисунок 5 - Раздел Overview из отчета Pandas Profiling

Вторая вкладка Reproduction, представленная на рисунке 6, имеет техническое назначение и содержит системную информацию.

Рисунок 6 – Раздел Reproduction из отчета Pandas Profiling

Во втором блоке Variables находится информация по каждой переменной. В отчете указано количество уникальных записей и их процент; количество пропущенных значений и их процент; количество значений NaN и их процент; среднее, минимальное и максимальное значение; количество и процент нулевых значений. Так же строится график распределения значений, который можно увидеть на рисунке 7.

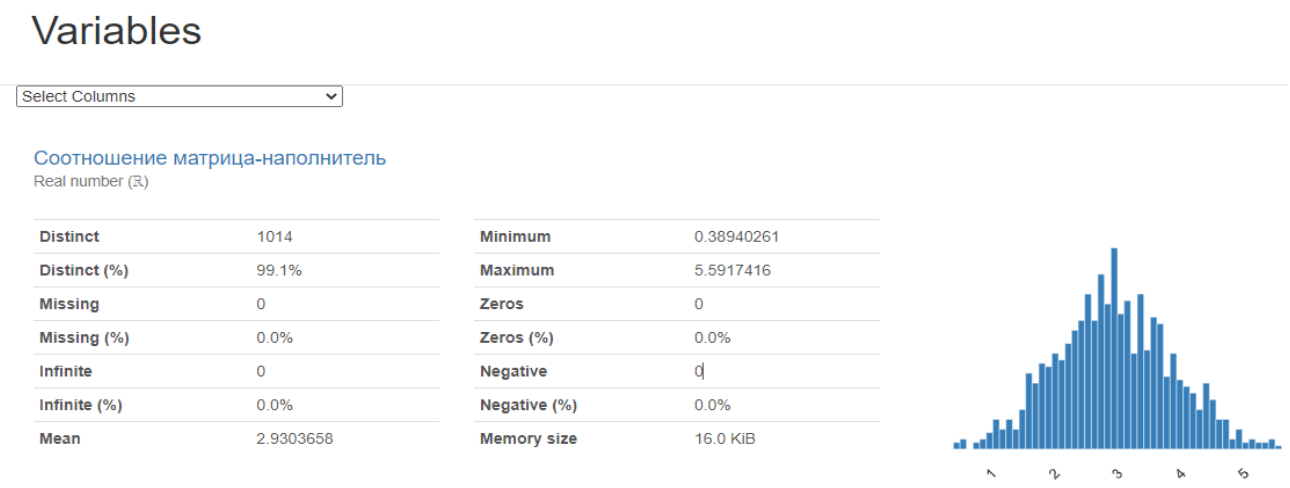

Рисунок 7 - Раздел Variables, сведения о признаке «Соотношение матрица-наполнитель»

При необходимости можно посмотреть более детальную информацию: квантильную и описательную статистику, пример на рисунке 8; полноразмерную диаграмму распределения значений; минимальные и максимальные значения.

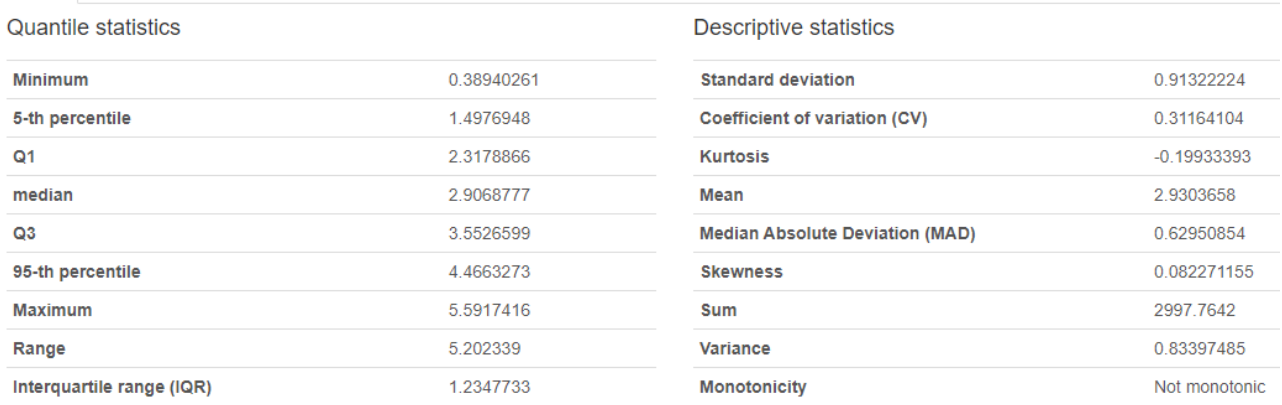

Рисунок 8 - Раздел Variables, квантильная и описательная статистика

В третьем блоке Interactions происходит автоматическая генерация графиков по парам переменных для визуализации зависимостей и распределения значений. График зависимости между переменными « Плотность нашивки» и «Соотношение матрица/наполнитель» и распределения значений представлен на рисунке 9.

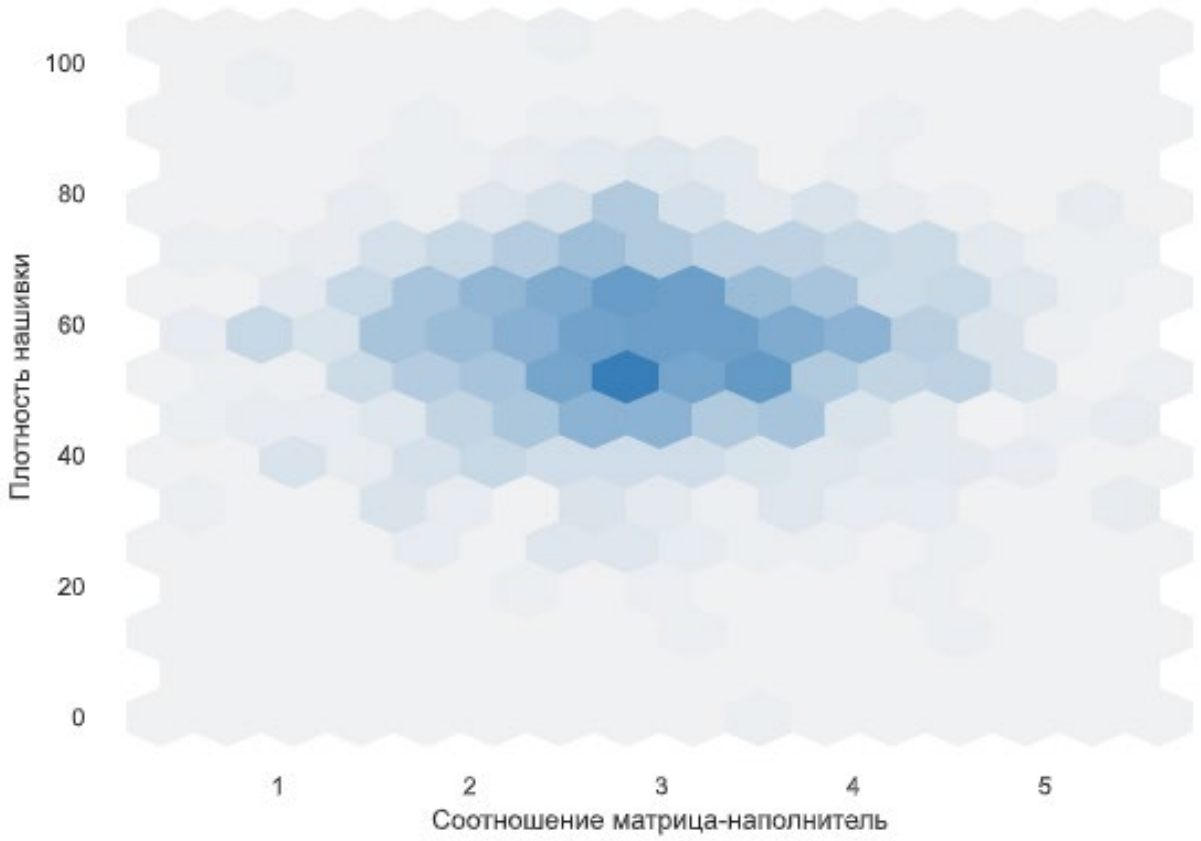

Рисунок 9 – Раздел Interactions, график зависимости между переменными «Плотность нашивки» и «Соотношение матрица/наполнитель»

В четвертом блоке Correlations на рисунке 10 отражены значения корреляции всех пар переменных. Позже будет построен график корреляции с другими настройками для более удобного считывания информации о взаимосвязи данных.

Блок Missing values на рисунке 11 предназначен для анализа пропущенных значений в выборке.

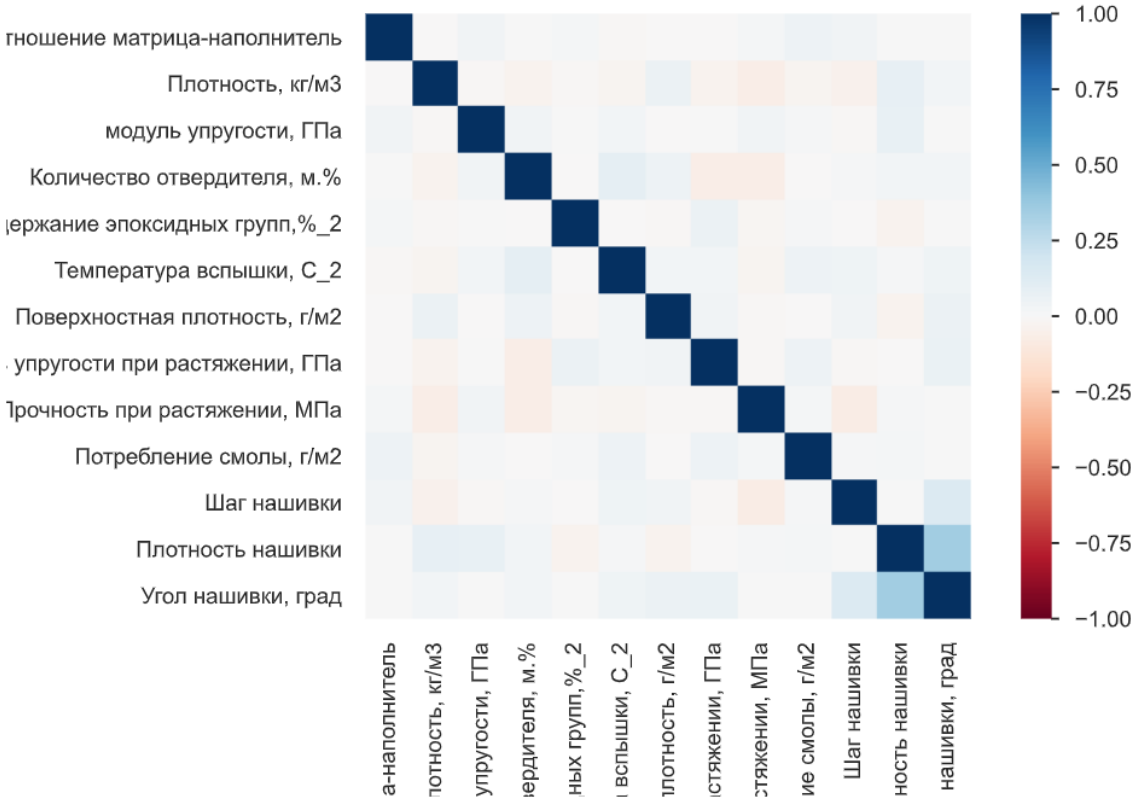

Рисунок 10 – Раздел Correlations, график корреляции всех пар переменных

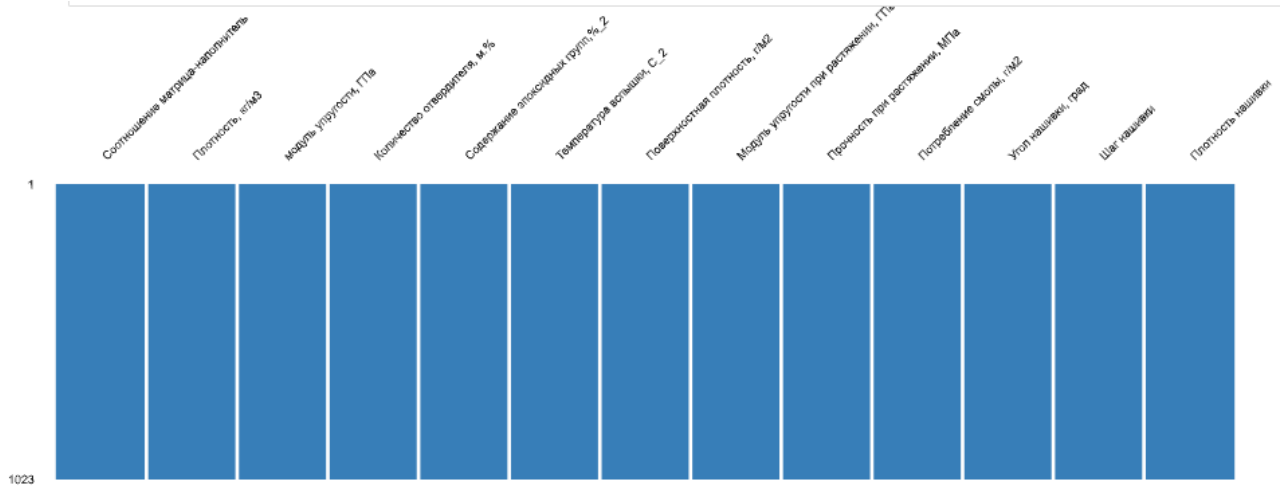

Рисунок 11 – Раздел Missing values, график пропущенных значений

Шестой блок Sample позволяет ознакомиться с выборкой первых и последних строк из таблицы. Отображение первых десяти строк представлены на рисунке 12.

First rows | Last rows

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2
0	1.857143	2030.0	738.736842	30.00	22.267857
1	1.857143	2030.0	738.736842	50.00	23.750000
2	1.857143	2030.0	738.736842	49.90	33.000000
3	1.857143	2030.0	738.736842	129.00	21.250000
4	2.771331	2030.0	753.000000	111.86	22.267857
5	2.767918	2000.0	748.000000	111.86	22.267857
6	2.569620	1910.0	807.000000	111.86	22.267857
7	2.561475	1900.0	535.000000	111.86	22.267857
8	3.557018	1930.0	889.000000	129.00	21.250000
9	3.532338	2100.0	1421.000000	129.00	21.250000

Рисунок 12 – Раздел Sample, выборка строк датасета

У метода разведочного анализа Pandas Profiling есть существенный недостаток – он медленно работает с большими массивами данных.

Так как в работе используется датасет небольшого объема, данный метод эффективно справился с задачей предоставления отчетов для разведочного анализа. С его помощью удалось установить, что у большинства признаков распределение стремится к нормальному, кроме признаков «Поверхностная плотность, г/м2» и «Угол нашивки, град». У признака «Поверхностная плотность, г/м2» на рисунке 13 прослеживается отрицательная асимметрия.

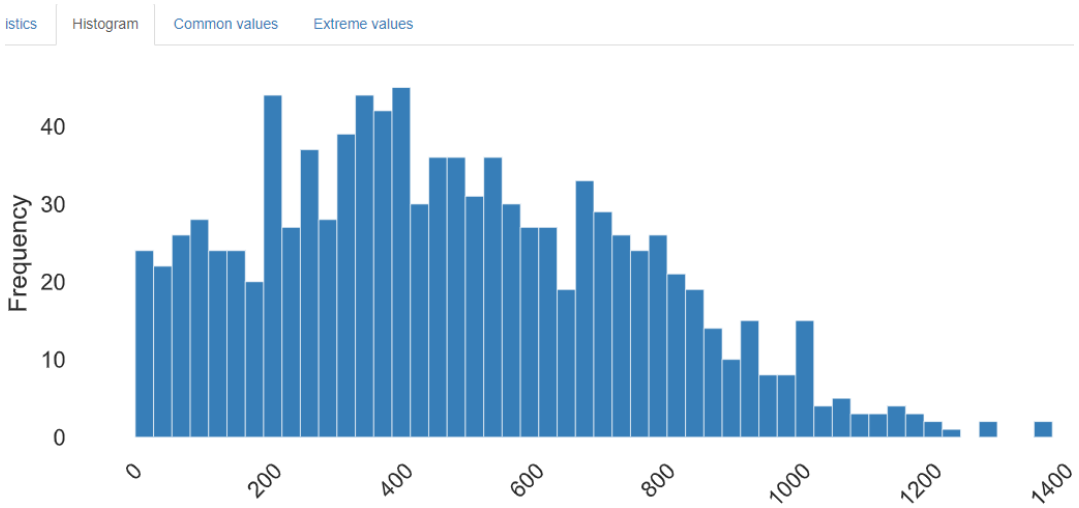

Рисунок 13 – Гистограмма распределения признака «Поверхностная плотность, г/м2»

Признак «Угол нашивки, град» на рисунке 14 показал себя, как категориальный, так как его записи имеют всего два уникальных значения.

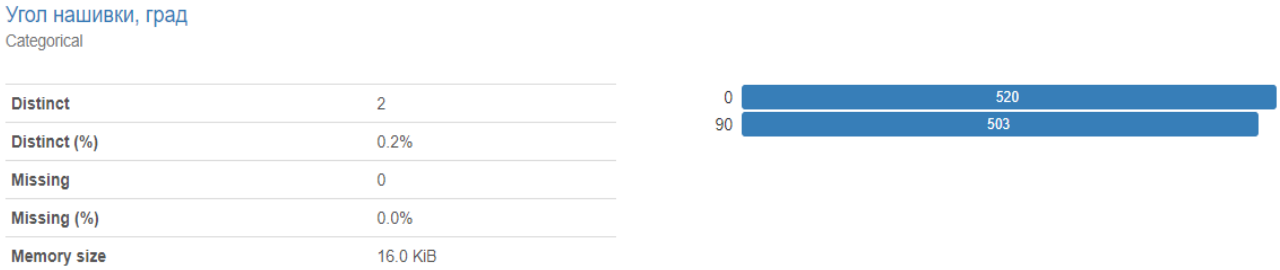

Рисунок 14 – Гистограмма распределения признака «Угол нашивки, град»

Для того, чтобы оценить взаимосвязи между признаками нужно изучить корреляционную матрицу, она представлена на рисунке 15.

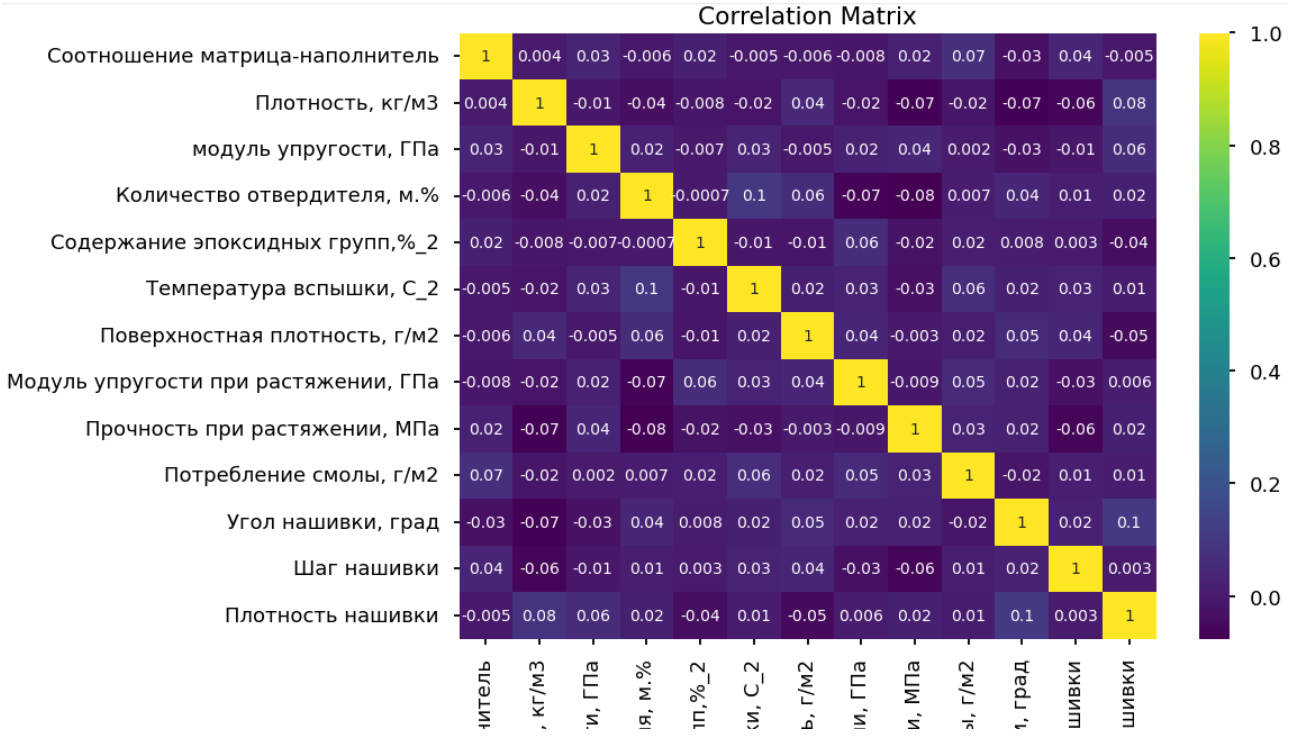

Рисунок 15 – Корреляционная матрица признаков

Проанализировав матрицу, можно сделать вывод, что признаки между собой не имеют линейной зависимости, так как между ними низкий коэффициент корреляции.

Важным этапом разведочного анализа является выявление выбросов в данных и их устранение. Один из наиболее эффективных методов знакомства с выбросами является диаграмма «ящик с усами» или Boxplot. Такой вид диаграммы в удобной форме показывает медиану, нижний и верхний квартили, минимальное и максимальное значение выборки и выбросы.

Нижний и верхний концы ящика соответствуют первому и третьему квартилям (25% и 75% квантилям соответственно), а горизонтальная линия внутри ящика – медиане. Верхний «ус» продолжается вверх вплоть до максимального значения, но не выше полуторного межквартильного расстояния от верхней кромки ящика. Аналогично нижний «ус» продолжается вниз до минимального значения, но не ниже полуторного межквартильного расстояния от нижней кромки ящика. Концы «усов» обозначаются небольшими горизонтальными линиями. А за пределами «усов» значения изображаются в виде отдельных точек – эти значения можно считать выбросами. Диаграмма Boxplot представлена на рисунке 16.

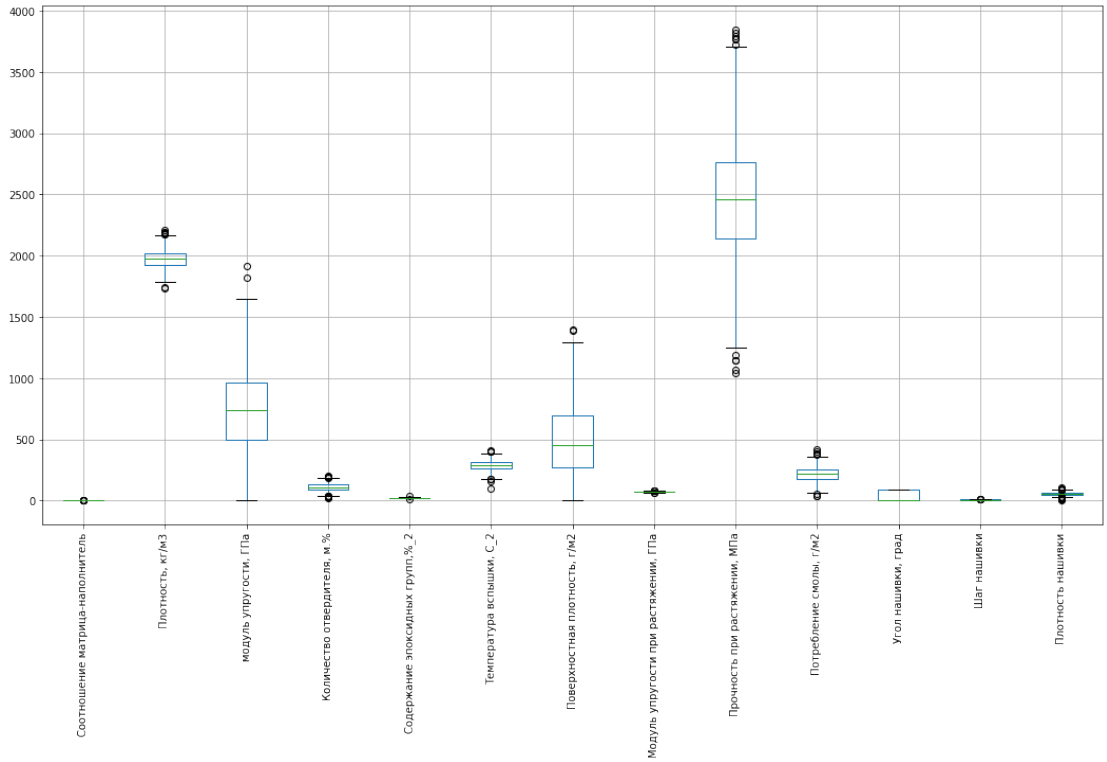

Рисунок 16 – Диаграмма Boxplot или «ящик с усами»

На диаграмме видно, что выбросы есть по всем характеристикам, кроме «Угол нашивки, град».

Следующим шагом в работе с данными было построение попарных графиков рассеяния точек. Присвоив каждой оси переменную, можно определить, существуют ли отношения или корреляция между этими двумя переменными на рисунке 17. Отображаемые на диаграммах рассеяния паттерны позволяют увидеть разные типы корреляции. Среди них могут быть: положительная (оба значения увеличиваются), отрицательная (одно значение увеличивается, в то время как второе уменьшается), нулевая (отсутствие корреляции), линейная, экспоненциальная и подковообразная. Сила корреляции определяется по тому, насколько близко расположены друг от друга точки на графике. Данный график показал очень слабую зависимость между переменными датасета. Также имеем возможность еще раз увидеть наличие некоторого количества выбросов – точки на графике, которые значительно удалены от общего кластера.

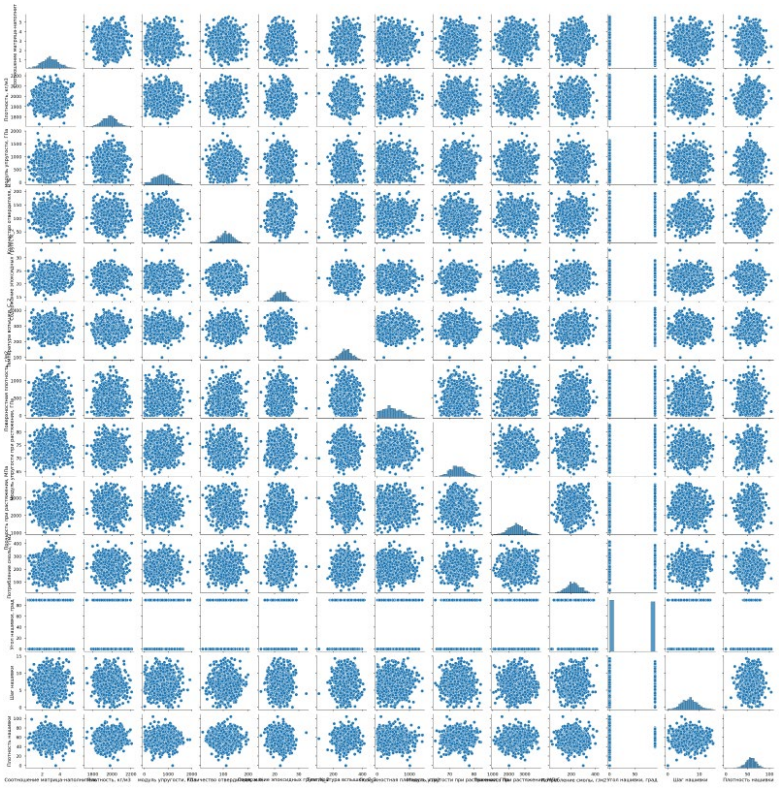

Рисунок 17 – Попарные графики рассеяния точек

2 Практическая часть

2.1 Предобработка данных

К задачам предварительной обработки данных относятся:

- Очистка данных;
- Редактирование данных;
- Заполнение пропусков.

При очистке данных удаляют устаревшие данные, дубликаты, аномалии, пропуски и ошибки. В данном датасете мы проведем очистку от выбросов методом трех сигм. Этот метод основан на стандартном отклонении данных от среднего значения. Согласно правилу трех сигм, большинство данных должны находиться в пределах трех стандартных отклонений от среднего значения. Используя эту концепцию, можно определить, какие данные являются выбросами.

После удаления выбросов на рисунке 18 можно увидеть, что размерность датасета уменьшилась.

```
count_3S = 0
for column in df:
  d = df.loc[:, [column]]
  zscore = (df[column] - df[column].mean()) / df[column].std()
  d['3S'] = zscore.abs() > 3
  count_3S += d['3S'].sum()
print('Количество выбросов методом трех сигм:', count_3S)
```

```
Количество выбросов методом трех сигм: 24
```

```
#удаление выбросов:
df_clean = df[(np.abs(stats.zscore(df)) <= 3).all(axis = 1)]
```

```
df_clean.shape
```

```
(999, 13)
```

Рисунок 18 – Обнаружение и удаление выбросов методом трех сигм

Для того, чтобы удостовериться в «чистоте» данных, была произведена повторная проверка тем же методом и обнаружено еще 3 выброса. Они были так же удалены.

Следующей задачей предварительной обработки данных является редактирование. Данные могут быть записаны с ошибками или в разных форматах, поэтому их нужно корректировать.

Датасет с характеристиками композитных материалов содержит числовые типы данных. Их минимальные и максимальные значения можно оценить на рисунке 19.

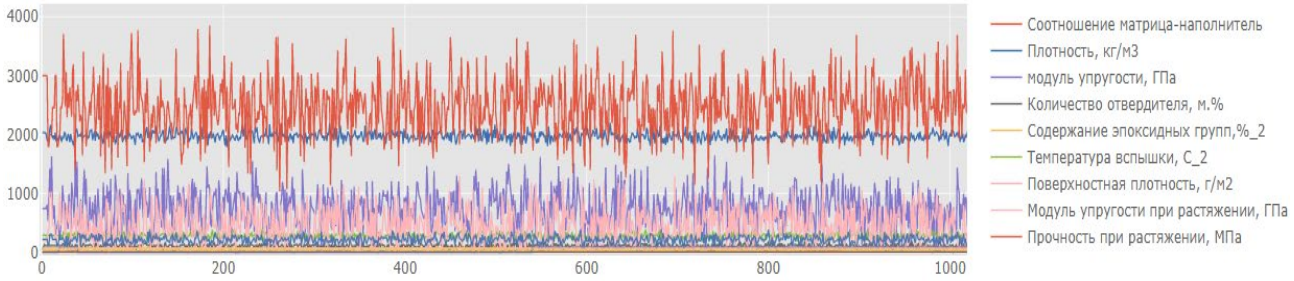

Рисунок 19 – График распределения значений до нормализации данных

Что касается числовых данных, то, чтобы привести их к единому формату, можно преобразовать значения в диапазон от 0 до 1. Такое преобразование возможно сделать при помощи методов нормализации данных.

Одним из методов нормализации данных является MinMaxScaler. Этот метод используется для масштабирования признаков в диапазоне от 0 до 1. Он работает путем пересчета значений признаков на основе их минимального и максимального значений в наборе данных.

Формула для MinMax-нормализации следующая:

x_scaled = (x - x_min) / (x_max - x_min)

где:

x_scaled - отмасштабированное значение признака;

x - оригинальное значение признака;

x_min - минимальное значение признака в наборе данных;

x_max - максимальное значение признака в наборе данных.

Увидеть изменение диапазона данных позволяет график на рисунке 20, построенный с помощью команды df_norm.iplot().

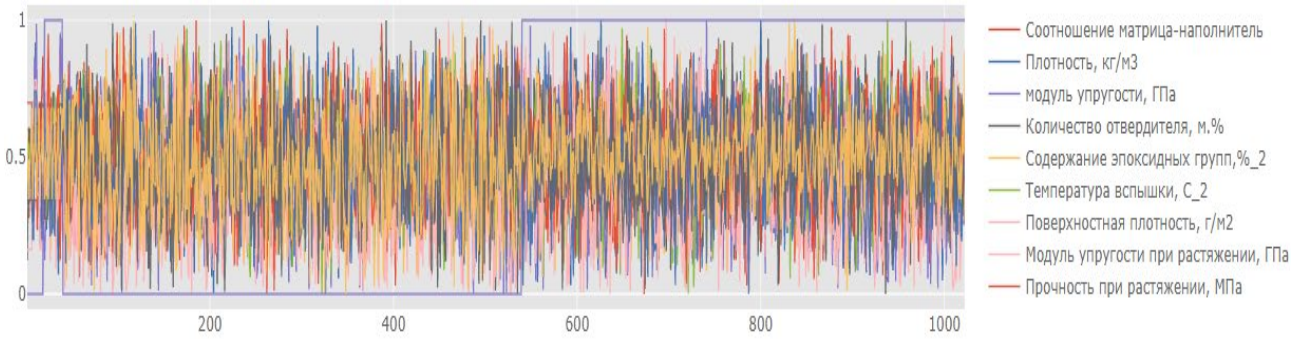

Рисунок 20 – График распределения значений после нормализации данных

Третьей задачей предобработки датасета является заполнение пропусков, но в данной работе нет такой необходимости.

2.2 Разработка и обучение модели

Для решения поставленной задачи данные были разделены на обучающую и тестовую выборки в соотношении 70/30. Разработка и обучение моделей машинного обучения осуществлялась для двух выходных параметров: «Прочность при растяжении» и «Модуль упругости при растяжении» отдельно.

Для каждой модели был создан словарь с гиперпараметрами и с помощью поиска по сетке с перекрестной проверкой были найдены лучшие гиперпараметры для каждой модели.

Для решения задачи предсказания модуля упругости при растяжении и прочности при растяжении были использованы следующие методы:

- Линейная регрессия;
- Случайный лес;
- Градиентный бустинг.

2.3 Тестирование модели

Оценка качества работы каждой из моделей выполнялась с помощью вычисления коэффициента детерминации (R^2), среднеквадратичной ошибки (RMSE) и средней абсолютной ошибка (MAE).

Показатели точности работы моделей для параметра «Прочность при растяжении» представлены на рисунке 21.

	Model	MAE	MSE	R2 score
Прочность при растяжении	LinearRegression()	0.13712	0.029412	-0.017691

	Model	MAE	MSE	R2 score
Прочность при растяжении	RandomForestRegressor(random_state=42)	0.135316	0.028723	0.006151

	Model	MAE	MSE	R2 score
Прочность при растяжении	GradientBoostingRegressor()	0.13497	0.028699	0.006997

Рисунок 21 – Ошибки модели предсказания для параметра «Прочность при растяжении»

Показатели точности работы моделей для параметра «Модуль упругости при растяжении» представлены на рисунке 22.

	Model	MAE	MSE	R2 score
Модуль упругости при растяжении	LinearRegression()	0.138812	0.029498	0.00418

	Model	MAE	MSE	R2 score
Модуль упругости при растяжении	RandomForestRegressor(random_state=42)	0.138678	0.029595	0.000933

	Model	MAE	MSE	R2 score
Модуль упругости при растяжении	GradientBoostingRegressor()	0.13886	0.029657	-0.001164

Рисунок 22 – Ошибки модели предсказания для параметра «Модуль упругости при растяжении»

Результаты построения и обучения моделей, к сожалению, не дали значительного положительного результата. Наименьшая ошибка в предсказании для признака «Модуль упругости при растяжении» получилась у модели линейной регрессии, в предсказании для признака «Прочность при растяжении» - у модели «случайный лес». Но и эти показатели не сильно отличаются от остальных моделей.

2.4 Нейронная сеть

Для построения рекомендательной системы признака «Соотношение матрица-наполнитель» использовали многослойный персептрон.

Первым шагом необходимо разделить очищенный от выбросов датасет на выходные данные в виде колонки «Соотношение матрица-наполнитель» и входные данные, которые включают все остальные колонки. Разделить входные и выходные данные на тренировочную и тестовую части в соотношении 80 и 20% с помощью train_test_split, после нормализовать данные используя TensorFlow.layers.Normalization.

После этого можно создавать нейронную сеть с помощью Sequential – это модель в библиотеке Keras, позволяющая создать нейронную сеть прямого распространения путем последовательного добавления слоев. Результат на рисунке 23.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 normalization (Normalizatio  (None, 12)               3
 n)

 dense (Dense)               (None, 1024)              13312

 dense_1 (Dense)             (None, 1024)              1049600

 dense_2 (Dense)             (None, 1)                 1025

=================================================================
Total params: 1,063,940
Trainable params: 1,063,937
Non-trainable params: 3
```

Рисунок 23 – Информация о модели нейронной сети

Модель нейронной сети имеет следующие настраиваемые гиперпараметры:

- входной слой нормализации 12 признаков;
- скрытые слои - 2;
- активационная функция скрытых слоев; relu - выполняет простое нелинейное преобразование поданных на вход данных (x). Возвращает x, если x > 0 и 0 в противном случае. Отличается высокой скоростью вычисления;
- нейронов в каждом скрытом слое: по 1024;
- выходной слой с 1 нейроном (т.е. для одного признака), так как на выходе выводится одно значение для введенных данных;
- метод Adam (adaptive moment estimation) – оптимизационный алгоритм, используемый для обучения сети, основная функция которого – изменение весов для уменьшения ошибки сети в процессе обучения. Для каждого нейрона алгоритм изменяет веса индивидуально;
- оценка качества модели при помощи loss-функция: MeanSquaredError (MSE).

Далее было проведено обучение модели на тренировочных данных при помощи метода fit со следующими параметрами:

- аргумент validation_split позволяет автоматически зарезервировать часть тренировочных данных для валидации. Это необходимо для того, чтобы иметь возможность обучить модель и оценить результаты работу с данными параметрами, не затрагивая тестовую выборку. Значением аргумента является доля данных, которые должны быть зарезервированы, в нашем случае это 20% тренировочных данных;
- verbose – режим вывода информации о процессе обучения нейронной сети;
- epoch – количество повторений циклов обучения для всей выборки данных. В данном случае их 70, так как большее количество эпох не дало лучшего результата. Итог обучения модели представлен на рисунке 24.

```
%%time
history = sqntl_model.fit(X_train, y_train, validation_split = 0.2, verbose = 1, epochs = 70)
```

```
Epoch 65/70
20/20 [==============================] - 0s 13ms/step - loss: 0.8143 - val_loss: 0.7762
Epoch 66/70
20/20 [==============================] - 0s 13ms/step - loss: 0.7953 - val_loss: 0.7893
Epoch 67/70
20/20 [==============================] - 0s 13ms/step - loss: 0.8099 - val_loss: 0.7736
Epoch 68/70
20/20 [==============================] - 0s 13ms/step - loss: 0.8050 - val_loss: 0.7643
Epoch 69/70
20/20 [==============================] - 0s 13ms/step - loss: 0.8112 - val_loss: 0.8699
Epoch 70/70
20/20 [==============================] - 0s 12ms/step - loss: 0.8268 - val_loss: 0.7783
Wall time: 19.4 s
```

Рисунок 24 – Обучение модели нейронной сети

По результатам обучения необходимо построить график, на котором две кривых: отображение среднеквадратической ошибки модели на тестовых (голубая линия) и валидационных данных (красная линия) относительно числа итераций. На рисунке 25 можно увидеть, что линии идут рядом, ошибка постепенно снижается и выходит на плато, где остается приблизительно на одном уровне до конца обучения.

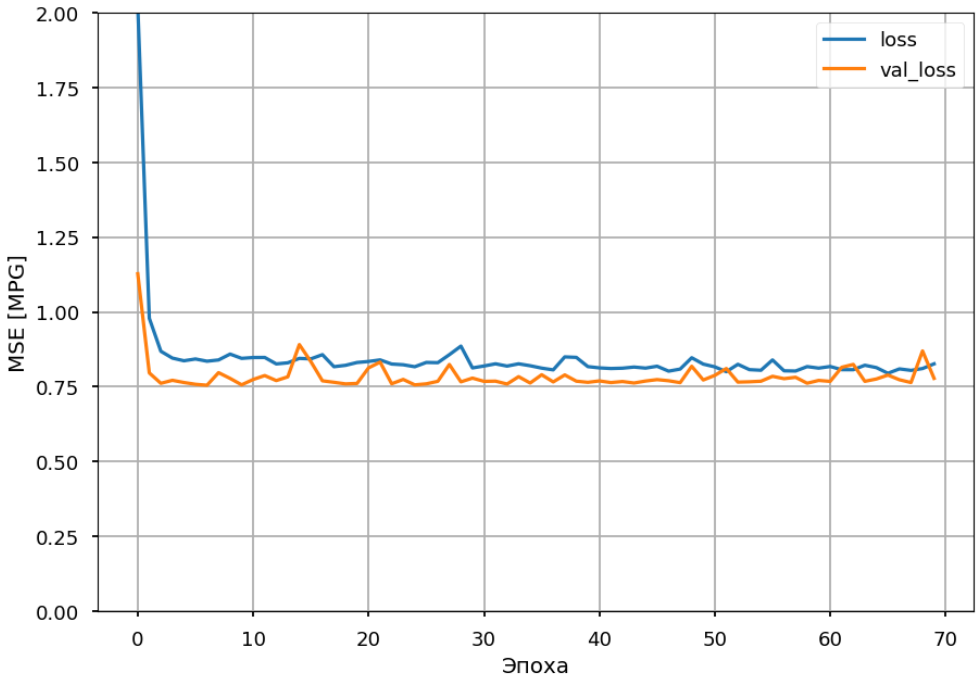

Рисунок 25 – Визуализация ошибки модели нейронной сети

Теперь нужно проверить модель на тестовых данных. По результатам работы модели получен график, представленный на рисунке 26, для сравнения оригинальных значений выборки и значениями, предсказанными моделью.

Рисунок 26 – Визуализация работы модели на тестовых данных

2.5 Разработка приложения

Веб-приложение для прогноза соотношения «Матрица-наполнитель» написано на языке программирования Python использованием библиотеки Flask, для написания шаблонов страниц был использован язык разметки HTML. На рисунке 27 представлена первая страница приложения.

1) Для запуска приложения необходимо:
2) Запустить среду разработки Visual Studio Code.
3) Запустить приложение командой "python app.py".
4) Перейти по сгенерированной ссылке.
5) Нажать на кнопку "Спрогнозировать значение матрица-наполнитель"
6) Ввести значение для каждого параметра.
7) Нажать кнопку "Рассчитать"

Рисунок 27 – Первая страница приложения

Далее на рисунке 28 представлена страница с полями для ввода параметров.

Рисунок 28 – страница приложения с расчетом параметров

2.6 Создание удаленного репозитория и загрузка результатов работы на него

Ссылка на репозиторий в GitHub:

https://github.com/shish27/VKR_Dyachenko

Заключение

В результате проделанной выпускной квалификационной работы по курсу «Data Science» были изучены теоретические основы методов машинного обучения и основные библиотеки высокоуровневого языка программирования Python, как один из основных инструментов для анализа данных. В процессе выполнения практической части были использованы изученные на курсе методы машинного обучения и построения моделей на реальных данных.

В данной работе не удалось разработать эффективную модель прогнозирования конечных свойств новых композиционных материалов на основе данных об их составе и структуре, точность предсказанных значений практически не превосходило среднее значения.

Тем не менее получен существенный практический опыт по анализу, визуализации и предобработке данных, созданию нейронных сетей и моделей машинного обучения, а также навык по созданию веб-приложения на основе этих прогнозных моделей.

Список использованной литературы

1) Композиционные материалы: Справочник /Под. ред. В.В. Васильева, Ю.М.Тарнопольского. –М.: Машиностроение, 1990. –512 с.

2) Библиотека Keras - инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow / пер. с англ. Слинкин А. А. - М.: ДМК Пресс, 2018. - 294 с.

3) Силен Дэви, Мейсман Арно, Али Мохамед. Основы Data Science и Big Data. Python и наука о данных. – СПб.: Питер, 2017. – 336 с.: ил.

4) Платформа scikit-learn [Электронный ресурс]: – Режим доступа: https://scikit-learn.org/stable/ (дата обращения: 21.03.2023).

5) Библиотека Seaborn- Режим доступа: https://seaborn.pydata.org/. (дата обращения 22.03.2023)

6) Язык программирования Python- Режим доступа: https://www.python.org/. (дата обращения 12.03.2023)

7) Библиотека Pandas – Режим доступа: https://pandas.pydata.org/ (дата обращения 17.03.2023)

8) Библиотека Sklearn – Режим доступа: https://scikit-learn.org/stable/ (дата обращения 23.03.2023)

9) Библиотека Pandas- Режим доступа: https://pandas.pydata.org/. (дата обращения 22.03.2023)

10) Библиотека Matplotlib- Режим доступа: https://matplotlib.org/. (дата обращения 18.03.2023)

11) Библиотека Tensorflow: Режим доступа: https://www.tensorflow.org/. (дата обращения 24.03.2023)