ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ МГТУ им. Н.Э. Баумана

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по курсу

«Data Science Pro»

Слушатель Тимошевский Александр Станиславович

Москва, 2025

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Содержание

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Введение

Композиционные материалы — это искусственно созданные материалы, состоящие из нескольких других с четкой границей между ними. Композиты обладают теми свойствами, которые не наблюдаются у компонентов по отдельности. При этом композиты являются монолитным материалом, т.е. компоненты материала неотделимы друг от друга без разрушения конструкции в целом. Яркий пример композита — железобетон. Железобетон прекрасно сопротивляется сжатию, но плохо растяжению. Стальная арматура внутри железобетона компенсирует его неспособность сопротивляться сжатию, формируя тем самым новые, уникальные свойства. Современные композиты изготавливаются из других материалов: полимеры, керамика, стеклянные и углеродные волокна, но данный принцип сохраняется. У такого подхода есть и недостаток: даже если мы знаем характеристики исходных компонентов, определить характеристики композита, состоящего из этих компонентов, достаточно проблематично. Для решения этой проблемы есть два пути: физические испытания образцов материалов или прогнозирование характеристик.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

1 Аналитическая часть

1.1 Постановка задачи

Тема выпускной квалификационной работы:

Прогнозирование конечных свойств новых материалов (композиционных материалов).

Актуальность:

Развитие науки и технологий предоставляет возможность создавать новое оборудование или техники либо совершенствовать уже имеющиеся оборудование и/или технику. Зачастую фактором, останавливающим возможность реализации идеи, является отсутствие материала, который позволит создать задуманное оборудование и/или технику.

Ввиду чего, разработка новых материалов – ключевой фактор, определяющий технологический прогресс, который приводит к созданию новых устройств и техники, которые недавно еще были невозможными. Однако, создание новых материалов – сложный процесс, который требует больших затрат времени и ресурсов. В связи с этим возникает необходимость в разработке методов и технологий, которые позволят сократить время и средства, необходимые для создания новых материалов. Одним из основных методов является прогнозирование конечных свойств новых материалов. Это позволяет ускорить процесс исследований, уменьшить затраты на эксперименты и сделать процесс создания новых материалов более эффективным и безопасным.

Расширение разнообразия материалов, используемых при проектировании нового композиционного материала, увеличивает необходимость определения свойств нового композита при минимальных финансовых затратах. Для решения этой проблемы обычно используются два способа: физические тесты образцов материалов или оценка свойств, в том числе на основе физико-математических

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

моделей. Традиционно разработка композитных материалов является долгосрочным процессом, так как из свойств отдельных компонентов невозможно рассчитать конечные свойства композита. Для достижения определенных характеристик требуется большое количество различных комбинированных тестов, что делает насущной задачу прогнозирования успешного решения, снижающего затраты на разработку новых материалов и затраты на рабочую силу.

В выпускной квалификационной работе будет использован путь прогнозирования итоговых характеристик композитного материала.

Прогнозирование характеристик композитного материала будет осуществлено посредством построения моделей машинного обучения и построение модели нейронной сети и их обучения. Модели будут выдавать прогнозные значения зависимых переменных (для моделей машинного обучения это «Модуль упругости при растяжении, ГПа», «Прочность при растяжении, МПа» и для модели нейронной сети это «Соотношение матрица-наполнитель») на основании значений независимых переменных.

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

Поставленные в выпускной квалификационной работе задачи:

1) Изучение теоретических основ и методов решения поставленной задачи;

2) Проведение разведочного анализа предложенных данных;

3) Проведение предварительной обработки данных;

4) Построение и обучение моделей машинного обучения для прогноза значений переменных «Модуль упругости при растяжении, ГПа» и «Прочность при растяжении, МПа». Оценка точности моделей машинного обучения;

5) Построение и обучение модели нейронной сети, которая будет прогнозировать значение переменной «Соотношение матрица-наполнитель». Оценка точности нейронной сети;

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

6) Разработка приложения, которое будет выдавать прогноз, полученный в задании 5;

7) Создание репозитория в GitHub и размещения там кода исследования. Оформления файла README.

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.

Данные для исследования (таблицы со свойствами композитов) разделены на два файла:

1) файл "X_bp.xlsx". Таблица содержит 1023 строки, индекс (столбец «Unnamed: 0») и 10 признаков;

2) файл "X_nup.xlsx". Таблица содержит 1040 строк, индекс (столбец «Unnamed: 0») и 3 признака.

В соответствии с поставленной задачей необходимы выполнить объединение таблиц в одну общую таблицу. Перед объединением из каждой таблицы исключаем столбец с нумерацией столбцов (индексами) – «Unnamed: 0». С помощью метода pandas.DataFrame.merge() произведено объединение таблиц. Объединение произведено по индексам, методом INNER.

```
df = data_frame1.merge(data_frame2, left_index = True, right_index = True, how = 'inner')
```

Рисунок 1 – код, объединяющий 2 таблицы в одну методом INNER

Итоговая таблица с данными имеет размер 13 столбцов и 1023 строки. Отмечаем, что 17 строк из таблицы № 2 не вошли в объединенную таблицу, так как за основу объединения бралась таблица № 1, в которой отсутствовала информация по этим 17 строкам.

	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

Рисунок 2 – информация о первых трех строках объединенной таблицы с входными данными

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

1.2 Описание используемых методов

Данная задача в рамках классификации категорий машинного обучения относится к машинному обучению с учителем и традиционно это задача регрессии. Цель любого алгоритма обучения с учителем — определить функцию потерь и минимизировать её.

В процессе исследования были применены следующие методы:

– Линейная регрессия (Linear regression);

– K-ближайших соседей (kNN - k Nearest Neighbours);

– Случайный лес (RandomForest);

– Лассо (Lasso);

– Байесовская регрессия (BayesianRidge).

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-квадрат равен 1, это значит, что модель описывает все данные. Если же R-квадрат 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: быстр и прост 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.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Метод ближайших соседей - К-ближайших соседей (kNN - k Nearest Neighbours) ищет ближайшие объекты с известными значения целевой переменной и основывается на хранении данных в памяти для сравнения с новыми элементами. Алгоритм находит расстояния между запросом и всеми примерами в данных, выбирая определенное количество примеров (k), наиболее близких к запросу, затем голосует за наиболее часто встречающуюся метку (в случае задачи классификации) или усредняет метки (в случае задачи регрессии).

Достоинства метода: прост в реализации и понимании полученных результатов; имеет низкую чувствительность к выбросам; не требует построения модели; допускает настройку нескольких параметров; позволяет делать дополнительные допущения; универсален; находит лучшее решение из возможных; решает задачи небольшой размерности.

Недостатки метода: замедляется с ростом объёма данных; не создаёт правил; не обобщает предыдущий опыт; основывается на всем массиве доступных исторических данных; невозможно сказать, на каком основании строятся ответы; сложно выбрать близость метрики; имеет высокую зависимость результатов классификации от выбранной метрики; полностью перебирает всю обучающую выборку при распознавании; имеет вычислительную трудоёмкость.

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.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

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

Лассо регрессия (Lasso) — это линейная модель, которая оценивает разреженные коэффициенты. Это простой метод, позволяющий уменьшить сложность модели и предотвратить переопределение, которое может возникнуть в результате простой линейной регрессии. Данный метод вводит дополнительное слагаемое регуляризации в оптимизацию модели. Это даёт более устойчивое решение. В регрессии лассо добавляется условие смещения в функцию оптимизации для того, чтобы уменьшить коллинеарность и, следовательно, дисперсию модели. Но вместо квадратичного смещения, используется смещение абсолютного значения. Лассо регрессия хорошо прогнозирует модели временных рядов на основе регрессии, таким как авторегрессии.

Достоинства метода: легко полностью избавляется от шумов в данных; быстро работает; не очень энергоёмко; способно полностью убрать признак из датасета; доступно обнуляет значения коэффициентов.

Недостатки метода: выбор модели не помогает и обычно вредит; часто страдает качество прогнозирования; выдаёт ложное срабатывание результата; случайным образом выбирает одну из коллинеарных переменных; не оценивает правильность формы взаимосвязи между независимой и зависимой переменными; не всегда лучше, чем пошаговая регрессия. Байесовская регрессия — это подход в линейной регрессии, в котором статистический анализ проводится в контексте байесовского вывода: когда регрессионная модель имеет ошибки, имеющие нормальное распределение, и, если принимается определённая форма априорного распределения, доступны явные результаты для апостериорных распределений вероятностей параметров модели.

Байесовская регрессия рассматривает параметры как случайные величины и встраивает их неопределенность в сам процесс моделирования.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Байесовская регрессия позволяет параметрам принимать разные значения с разной вероятностью, в зависимости от априорных предположений. Этот подход более естественно учитывает неопределенность, делая модель менее подверженной переобучению. Традиционная регуляризация находит компромисс между сложностью модели и точностью, но игнорирует вероятностную природу параметров, что в некоторых случаях ведет к субоптимальным решениям, тогда как байесовский подход учитывает эту неопределенность в параметрах на всех этапах моделирования.

Поскольку апостериорные распределения могут указывать на степень неопределенности в весах, байесовская регрессия более гибка и способна точнее оценивать влияние признаков, что особенно полезно в условиях ограниченных данных или мультиколлинеарности.

Достоинства байесовской регрессии: она адаптируется к имеющимся данным; её можно использовать для включения параметров регуляризации в процедуру оценки; более устойчива к некорректно поставленным задачам (задача, не обладающая каким-либо из свойств корректно поставленной задачи.

К недостаткам байесовской регрессии можно отнести: вывод модели может занять много времени.

1.3 Разведочный анализ данных

Разведочный анализ данных — анализ основных свойств данных, нахождение в них общих закономерностей, распределений и аномалий, построение начальных моделей, зачастую с использованием инструментов визуализации.

Понятие введено математиком Джоном Тьюки, который сформулировал цели такого анализа следующим образом:

- максимальное «проникновение» в данные;
- выявление основных структур;
- выбор наиболее важных переменных;

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

- обнаружение отклонений и аномалий;
- проверка основных гипотез;
- разработка начальных моделей.

При проведении разведочного анализа данных были использованы методы: проведен анализ переменных и определены их свойства; визуализации данных (построение гистограмм распределения переменных, диаграмм размаха, диаграмм рассеяния и тепловой карты); выявления сводных статистик и мер центральной тенденции (определение среднего арифметического (mean) каждой переменной, среднего квадратичного отклонения (std) каждой переменной, минимального (min) и максимального (max) значения каждой переменной, первого (25%), второго (50%) и третьего (75%) квартилей); проведен анализ выбросов, дубликатов и аномалий; проведен анализ корреляции переменных; проведен анализ распределения данных (включая проведение статистического теста Шапиро-Уилка).

Изучаем структуру и характеристики данных, чтобы полностью понять, с чем имеем дело. Это включает в себя обзор размера набора данных, типов переменных, наличия пропущенных значений, дубликатов и других важных аспектов. Разведка на этом уровне позволяет нам сформировать четкое представление о данных, что становится фундаментом для дальнейшего анализа.

С целью понимания структуры и характеристик данных была составлена таблица с наименованием и описанием переменных.

Названия столбцов	Описание полей
Соотношение матрица-наполнитель	матрица - это связующее вещество, в которое внедрен наполнитель (волокна, частицы, слои). Соотношение может быть выражено в процентах или в виде объемного или весового соотношения.
Плотность, кг/м3	физическая величина, характеризующая массу вещества, содержащегося в единице объема. Выражен в килограмме на кубический метр.
модуль упругости, ГПа	физическая величина, характеризующая способность твердого тела сопротивляться упругой деформации при приложении к нему силы. Выражен в Гигапаскаль.
Количество отвердителя, м.%	отвердитель – это химическое вещество, которое добавляется к лакокрасочным материалам, эпоксидным смолам и другим реакционноспособным олигомерам для ускорения или инициирования процесса отверждения (застывания, полимеризации).
Содержание эпоксидных групп,%_2	эпоксидные группы – это структурные единицы в молекулах эпоксидных смол, которые обеспечивают их способность к отвердеванию под действием отвердителей.
Температура вспышки, С_2	это самая низкая температура, при которой пары над поверхностью горючего вещества способны вспыхнуть при поднесении к ним источника зажигания, но устойчивое горение после удаления источника зажигания не наблюдается.
Поверхностная плотность, г/м2	величина, характеризующая массу вещества, приходящуюся на единицу площади. Выражен в грамм на метр кубический
Модуль упругости при растяжении, ГПа	ценивает упругость жестких или твердых материалов, которая является соотношением между деформацией материала и силой, необходимой для его деформации. Выражен в Гигапаскаль.
Прочность при растяжении, МПа	максимальное напряжение, которое материал может выдержать при растягивающей нагрузке до начала разрушения. Это важный показатель для оценки прочности и надежности материалов. Выжаражен в Мегапаскаль.
Потребление смолы, г/м2	потребление эпоксидной смолы. Выражен в граммах на квадратный метр
Угол нашивки, град	угол нашивки. Выражен в градусах
Шаг нашивки	шаг нашивки
Плотность нашивки	плотность нашивки

Рисунок 3 – наименование переменных и их описание

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Сводные статистики и меры центральной тенденции позволяют нам получить обобщенное представление о распределении данных и основных характеристиках. Это ключевые числовые метрики, которые помогают нам понять типичные и наиболее значимые значения в наборе данных.

Описательная статистика, а именно общее количество элементов (count) в таблице, среднее арифметическое (mean) переменной, среднее квадратичное отклонение (std) переменной, минимальное (min) и максимальное (max) значение переменной, первый (25%), второй (50%) и третий (75%) квартили переменной приведены на рисунке 4.

```
df.describe().T
```

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

Рисунок 4 – описательная статистика данных

Изучаем общую информацию о зафиксированных в таблице данных и проверяем таблицу на наличие пропусков, проверяем данные на наличие дубликатов.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 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
```

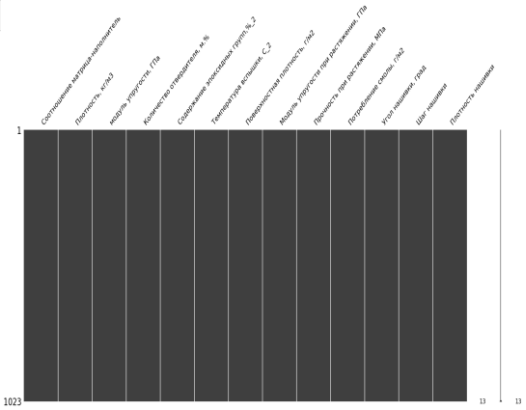

Рисунок 5 – общая информация о количество строк и столбцов, типах данных и диаграмма пропущенных значений

```
df.duplicated().sum()
```

```
np.int64(0)
```

Рисунок 6 – результаты проверки данных на наличие дубликатов

Визуализация данных позволяет нам увидеть и понять паттерны, тренды и взаимосвязи в данных через графику и диаграммы.

Гистограммы распределения данных используются для изучения распределения данных, выявления тенденций и закономерностей. Дополнительно на гистограммы была нанесена кривая оценки плотности ядра (KDE), которая помогает визуализировать распределение точек данных в наборе данных и выявлять закономерности, кластеры и аномалии в данных.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

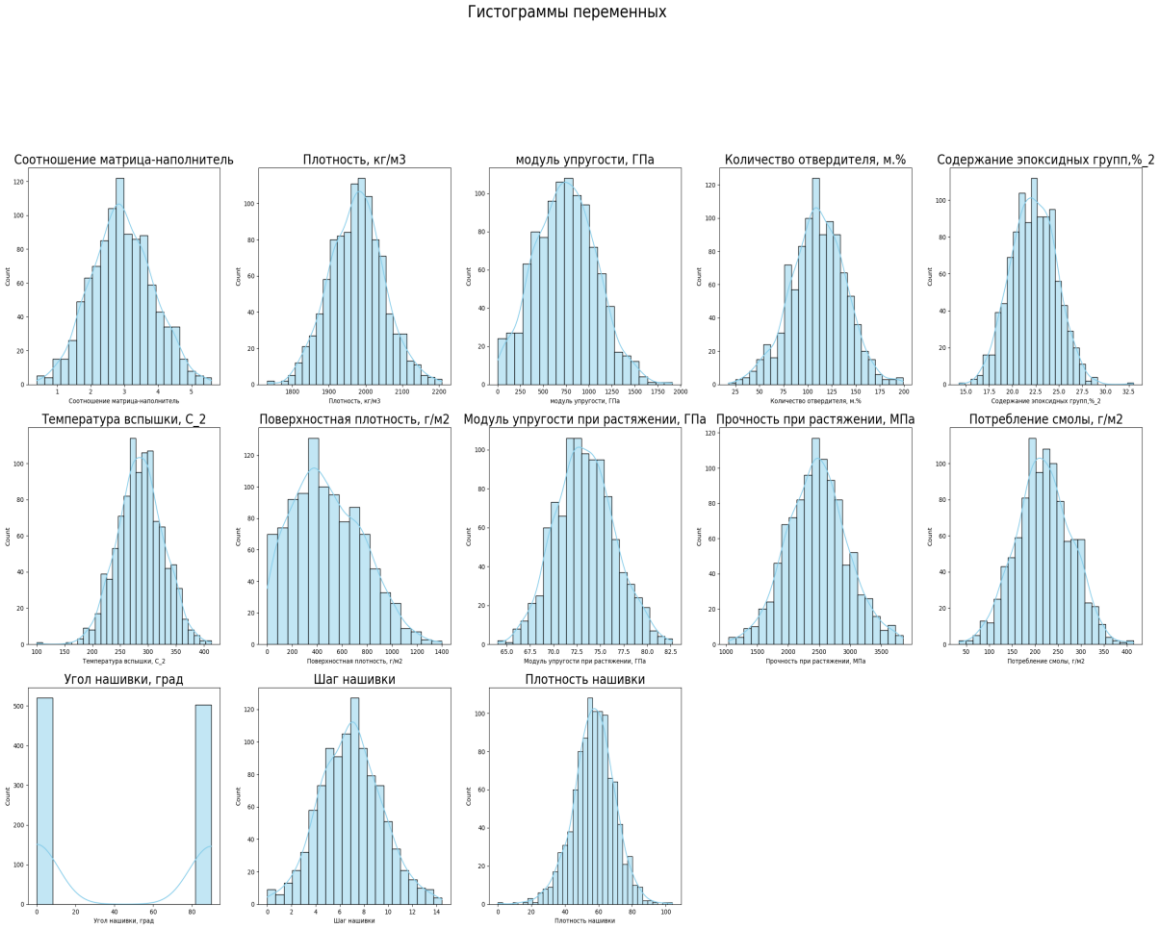

Рисунок 7 – гистограммы распределения каждой переменной в данных

Дополнительно провели статистический тест Шапиро-Уилка с целью проверки нормальности распределения переменных.

```
for i in list(df.columns.values):
  stat, p = stats.shapiro(df[i]) # тест Шапиро-Уилк
  print(i)
  print('Statistics=%.3f, p-value=%.3f' % (stat, p))

  alpha = 0.05
  if p > alpha:
      print('Принять гипотезу о нормальности распределения\n')
  else:
      print('Отклонить гипотезу о нормальности распределения\n')
```

Рисунок 8 – тест Шапиро-Уилка на нормальность распределения

Диаграмма размаха (ящик с усами) — это статистический инструмент для визуализации распределения данных. Она выглядит как прямоугольник с линиями по краям, которые указывают на минимальные и максимальные значения.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Диаграммы размаха позволяют сделать наблюдения о таких о ключевых значениях, например, как:

- средний показатель, медиана 25го перцентиля и так далее;
- наличие выбросов и каковы их значения;
- о симметричности распределения данных, группировке данных;
- направления смещения данных.

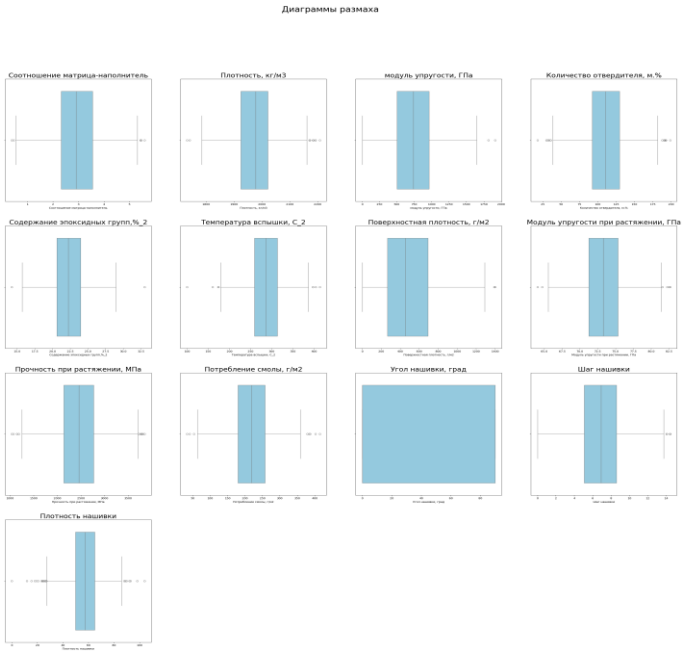

Рисунок 9 – диаграммы размаха по каждой переменной в наборе данных

Диаграмма рассеяния – это график, в котором каждая точка представляет собой отдельное наблюдение и показывает взаимосвязь между двумя переменными. Это может помочь нам определить, есть ли какая-либо зависимость или корреляция между ними.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

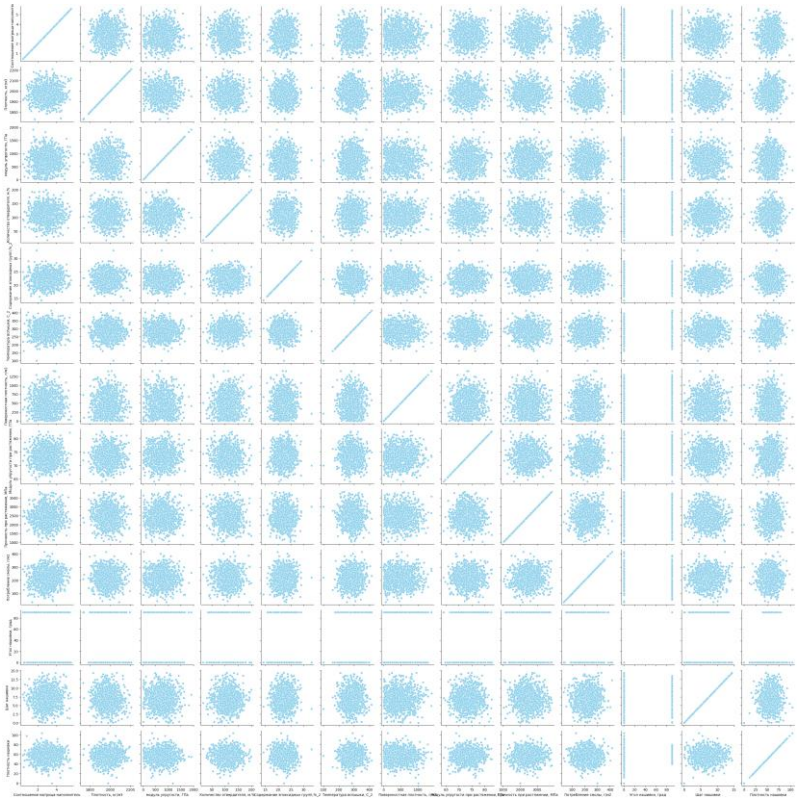

Рисунок 10 – Диаграмма рассеяния по связи каждой переменной в наборе данных

Осуществляем проверку переменных в наборе данных на наличие корреляции для чего строим тепловую карту.

Тепловая карта – это графическое представление матрицы данных, где цветовая шкала показывает степень взаимосвязи между переменными. Это помогает выявить паттерны и зависимости в больших наборах данных.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

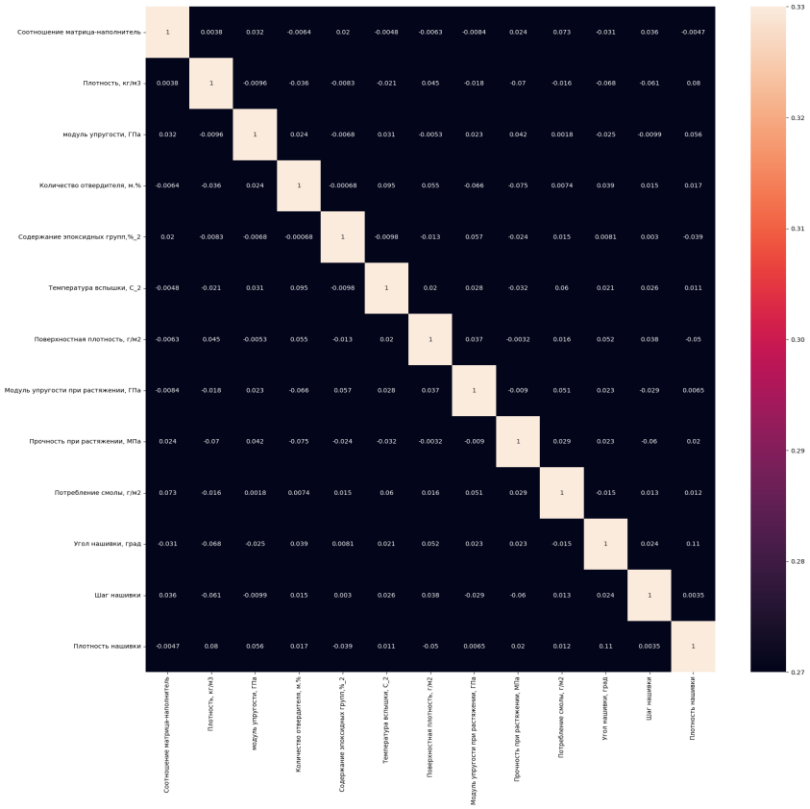

Рисунок 11 – Диаграмма рассеяния по связи каждой переменной в наборе данных

Корреляция — это статистическая взаимосвязь между двумя или более переменными. Она указывает на то, насколько изменения в одной переменной связаны с изменениями в другой. Если изменение одной переменной сопровождается систематическим изменением другой, то говорят о наличии корреляции. Корреляция не обязательно означает причинно-следственную связь, то есть изменение одной переменной не обязательно является причиной изменения другой, они могут быть связаны через третью переменную или просто случайно.

Для измерения силы и направления связи между переменными используется коэффициент корреляции, который обычно принимает значения от -1 до +1.

Значение, близкое к +1 или -1, указывает на сильную корреляцию, а значение, близкое к 0, указывает на слабую или отсутствующую корреляцию.

Отмечаем следующие характеристики и особенности предоставленных данных:

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

1) Размер таблицы с данными, предоставленными для исследования, составил 13 переменных (столбцов) и 1023 значения переменной (строки);

2) В значениях переменных отсутствуют пропуски;

3) В таблице с данными, предоставленными для исследования, отсутствуют дубликаты;

4) Обнаружены выбросы в значениях каждой переменной (за исключением переменной «Угол нашивки, град»);

5) Отмечаем большой разброс в значениях данных. Отмечаем необходимость масштабирования данных.

6) Отмечаем, что данные измерены в разных единицах измерениях. Многие из которых сложносочитаемы.

7) Диаграммы распределения показали, что распределение большинства переменных визуально похоже на нормальное. После проведения статистического теста Шапиро-Уилка сделали выводы о том, что значения переменных "Соотношение матрица-наполнитель", "Плотность, кг/м3","Количество отвердителя, м.%", "Содержание эпоксидных групп,%_2", "Температура вспышки, С_2", "Модуль упругости при растяжении, ГПа", "Прочность при растяжении, МПа", "Потребление смолы, г/м2", "Шаг нашивки" распределены нормально, а значение переменных "модуль упругости, ГПа", "Поверхностная плотность, г/м2", "Угол нашивки, град", "Плотность нашивки" распределены не нормально;

8) Отмечаем отсутствие линейной связи между переменными и практически полное отсутствие корреляции между переменными;

9) В таблице представлены в 2 типа данных – float64 (для данных, представленных в виде чисел с плавающей точкой) и int64 (для данных, предоставленных в виде целых чисел). Отмечаем, что переменная «Угол нашивки, град» имеет только два значения равные 0 или 90;

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

2 Практическая часть

2.1 Предварительная обработка данных

Обрабатываем значения переменной "Угол нашивки, град". Как установлено ранее, переменная имеет значения 0 или 90 градусов.

Угол нашивки в композитных материалах, или угол ориентации волокна, определяет направление волокон в слое композита. Он влияет на прочность и жесткость композитного материала, а также на его поведение при нагрузках. Угол нашивки (направления волокна) — это угол между направлением волокон в слое композитного материала и некоторым эталонным направлением (обычно осью X или L).

Угол нашивки обычно указывается в градусах. Например, 0° означает, что волокна параллельны эталонной оси, а 90° - перпендикулярны.

Угол нашивки влияет следующие на характеристики композита: прочность; жесткость; устойчивость к трещинам; вес.

Примеры: угол нашивки 0° обеспечивает максимальную прочность на растяжение в этом направлении; угол нашивки 90° обеспечивает высокую прочность на сжатие и сдвиг в этом направлении.

Обрабатываем значения переменной "Угол нашивки, град".

Необходимо отметить, что категориальные переменные выражены ограниченным диапазоном значений. В нашем случае "Угол нашивки, град" выражен только 2 значениями - 0 или 90. В данном случае "Угол нашивки, град" мы можем отнести к номинальной переменной, т.к. установить порядок и ранжирование значений невозможно.

При этом необходимо отметить, что угол нашивки будет влиять на свойство композитного материала в зависимости от ориентации волокон. При этом из таблицы не ясно в каком направлении применялась сила (перпендикулярно, параллельно или под иным углом) при замерах значений угла нашивки.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Чтобы оценить свойства композитного материала в зависимости от оси направления силы по отношению к углу нашивки необходимо проводить несколько испытаний с целью понимания насколько угол нашивки влияет на отдельные свойства композитного материала. Из данных мы видим, что в таблице № 2 было представлено по 520 значений угла нашивки, но исследование уникальных значений показало, что мы не можем определить, что происходило 2 замера одного и того же композитного материала в зависимости от направления угла нашивки. Так же мы не можем определить направление оси применения силы при замерах и направление применение силы которое необходимо учитывать при обучении алгоритма машинного обучения, который будет определять значения:

- Модуль упругости при растяжении, ГПа;
- Прочность при растяжении, МПа.

Ввиду вышеуказанного мы не можем ранжировать значения угла нашивки или говорить о том, что значение 90 лучше, чем значение 0, или наоборот.

В связи с чем мы кодируем переменную "Угол нашивки, град" как категориальную с использованием OneHotEncoder.

OneHotEncoder — это метод, используемый в машинном обучении для представления категориальных данных в виде числовых данных. Он преобразует каждую категорию в двоичный вектор, где только один элемент "включен" (1), а остальные "выключены" (0). Этот метод имеет решающее значение для алгоритмов, требующих числового ввода, и помогает предотвратить неправильное толкование порядковых отношений между категориями.

```
# Осуществляем кодирование переменной "Угол нашивки, град"
ohe = OneHotEncoder(sparse_output=False)
ohe.fit_transform(df[['Угол нашивки, град']])
ohe.get_feature_names_out()
df[['Угол нашивки, град_0', 'Угол нашивки, град_90']] = ohe.fit_transform(df[['Угол нашивки, град']])
df.drop('Угол нашивки, град', axis=1, inplace=True)
```

Рисунок 12 – использование OneHotEncoder

Данные так же содержат переменную "Температура вспышки, С_2".

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Температура вспышки – это минимальная температура, при которой вещество выделяет достаточно паров, чтобы при поднесении источника зажигания произошло воспламенение паров и их кратковременное горение, но без устойчивого горения после удаления источника зажигания. Это важный параметр для оценки пожарной опасности веществ, особенно при их транспортировке, хранении и использовании. Температура вспышки не оказывает влияния на прочность и упругость уже изготовленного композитного материала. В связи с чем принимаем решения исключить из набора данных переменную "Температура вспышки, С_2".

```
# исключаем переменную "Температура вспышки, С_2"
df.drop('Температура вспышки, С_2', axis=1, inplace=True)
```

Рисунок 13 – исключение переменной «Температура вспышки, С_2»

В данных также есть переменные "Потребление смолы г/м2" и "Содержание эпоксидных групп, %_2".

Зная, что эпоксидные группы – это структурные единицы в молекулах эпоксидных смол, которые обеспечивают их способность к отвердеванию под действием отвердителей. Они представляют собой тричленный цикл, содержащий один атом кислорода, и также называются оксиранами или эпоксидами. Эти группы реагируют с отвердителями, образуя сшитые (сетчатые) полимеры, которые отличаются высокой прочностью и устойчивостью к различным воздействиям. Из предоставленных данных мы обнаруживаем, что "Содержание эпоксидных групп, %_2" дано в процентном соотношении и знаем о том, что эпоксидные группы содержатся в смолах. Значения смол приведено граммах в столбце "Потребление смолы г/м2".

Принимаем решение вычислить значение эпоксидных смол в содержащихся в смолах. Проводим вычисления и изменяем данные и фиксируем итоговое значение эпоксидных групп и количество смол (за вычетом содержащихся в них эпоксидных групп).

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

```
df['Содержание эпоксидных групп, г'] = df['Потребление смолы, г/м2'] * (df['Содержание эпоксидных групп,%_2'] / 100)
df.drop('Содержание эпоксидных групп,%_2', axis=1, inplace=True)
df['Потребление смолы, г/м2 без ЭГ'] = df['Потребление смолы, г/м2'] - df['Содержание эпоксидных групп, г']
df.drop('Потребление смолы, г/м2', axis=1, inplace=True)
```

Рисунок 14 – операции над переменными

Таким образом, добавлены две новых переменных "Содержание эпоксидных групп, г" и "Потребление смолы, г/м2 без ЭГ".

Работа с выбросами.

Выброс — это экстремальные значения во входных данных, которые находятся далеко за пределами других наблюдений.

Многие алгоритмы машинного обучения чувствительны к разбросу и распределению значений признаков обрабатываемых объектов. Соответственно, выбросы во входных данных могут исказить и ввести в заблуждение процесс обучения алгоритмов машинного обучения, что приводит к увеличению времени обучения, снижению точности моделей и, в конечном итоге, к снижению результатов. Даже до подготовки предсказательных моделей на основе обучающих данных выбросы могут приводить к ошибочным представлениям и в дальнейшем к ошибочной интерпретации собранных данных.

Для удаления выбросов используем метод межквартильного размаха (IQR). Межквартильный размах — один из самых популярных методов обнаружения выбросов, особенно для ненормально распределённых данных. Он устойчив к асимметрии и экстремальным значениям.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

```
for col in df.columns:
    q75, q25 = np.percentile(df.loc[:, col], [75, 25])
    intr_qr = q75 - q25

    max = q75 + (1.5 * intr_qr)
    min = q25 - (1.5 * intr_qr)

    df.loc[df[col] < min, col] = np.nan
    df.loc[df[col] > max, col] = np.nan
```

```
df = df.dropna(axis=0)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 925 entries, 1 to 1022
Data columns (total 13 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   Соотношение матрица-наполнитель       925 non-null    float64
 1   Плотность, кг/м3                      925 non-null    float64
 2   модуль упругости, ГПа                 925 non-null    float64
 3   Количество отвердителя, м.%           925 non-null    float64
 4   Поверхностная плотность, г/м2         925 non-null    float64
 5   Модуль упругости при растяжении, ГПа  925 non-null    float64
 6   Прочность при растяжении, МПа         925 non-null    float64
 7   Шаг нашивки                           925 non-null    float64
 8   Плотность нашивки                     925 non-null    float64
 9   Угол нашивки, град_0                  925 non-null    float64
 10  Угол нашивки, град_90                 925 non-null    float64
 11  Содержание эпоксидных групп, г        925 non-null    float64
 12  Потребление смолы, г/м2 без ЭГ        925 non-null    float64
dtypes: float64(13)
memory usage: 101.2 KB
```

Рисунок 15 – обнаруживаем и исключаем выбросы

Проводим несколько итераций исключения выбросов. Удостоверяемся в том, что выбросы полностью исключены с помощью построения диаграмм размаха и вышеуказанной формулы.

Масштабирование данных.

В значениях переменных есть большой разброс в их масштабе. Отдельные значения могут быть равны 0 и/или близки к нему, а другие измеряются в тысячах.

Масштабирование данных — это процесс преобразования числовых признаков в наборе данных к одному масштабу, чтобы они имели одинаковый диа-

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

пазон значений. Это необходимо, потому что многие алгоритмы машинного обучения чувствительны к разным масштабам признаков, и масштабирование помогает избежать ситуации, когда один признак доминирует над другими.

Масштабирование применяется для:

- Улучшение работы алгоритмов.
- Предотвращение доминирования признаков. Если один признак имеет большой диапазон значений, а другой – маленький, то первый может оказывать большее влияние на модель, даже если его важность не выше. Масштабирование выравнивает вклад каждого признака.
- Оптимизация обучения. Масштабирование может ускорить процесс обучения модели и повысить ее точность.

Масштабирования данных будет произведено непосредственно перед созданием обучающей и тестовой выборок при построении моделей машинного обучения и модели нейросети.

Общая информация об обработанных данных.

Далее отображены рисунки с наименованием и описанием переменных, описательная статистика итоговой таблицы, диаграммы распределения, диаграммы размаха, диаграммы рассеяния, тепловую карту.

Наименование столбца	Описание поля
Соотношение матрица-наполнитель	Матрица - это связующее вещество, в которое внедрен наполнитель (волокна, частицы, слои). Соотношение может быть выражено в процентах или в виде объемного или весового соотношения.
Плотность, кг/м3	Физическая величина, характеризующая массу вещества, содержащегося в единице объема. Выражен в килограмме на кубический метр.
модуль упругости, ГПа	Физическая величина, характеризующая способность твердого тела сопротивляться упругой деформации при приложении к нему силы. Выражен в Гигапаскаль.
Количество отвердителя, м.%	Отвердитель – это химическое вещество, которое добавляется к лакокрасочным материалам, эпоксидным смолам и другим реакционноспособным олигомерам для ускорения или инициирования процесса отверждения (застывания, полимеризации).
Поверхностная плотность, г/м2	Величина, характеризующая массу вещества, приходящуюся на единицу площади. Выражен в грамм на метр кубический
Модуль упругости при растяжении, ГПа	Оценивает упругость жестких или твердых материалов, которая является соотношением между деформацией материала и силой, необходимой для его деформации. Выражен в Гигапаскаль.
Прочность при растяжении, МПа	Максимальное напряжение, которое материал может выдержать при растягивающей нагрузке до начала разрушения. Это важный показатель для оценки прочности и надежности материалов. Выжаражен в Мегапаскаль.
Шаг нашивки	Шаг нашивки
Плотность нашивки	Плотность нашивки
Угол нашивки, град_0	Угол нашивки равный 0 градусам
Угол нашивки, град_90	Угол нашивки равный 90 градусам
Содержание эпоксидных групп, г	Количество эпоксидных групп в смоле (в граммах). Эпоксидные группы – это структурные единицы в молекулах эпоксидных смол, которые обеспечивают их способность к отвердеванию под действием отвердителей.
Потребление смолы, г/м2 без ЭГ	Потребление смолы за вычетом эпоксидных групп в смоле (в граммах/м2)

Рисунок 16 – наименование столбцов и их описание обработанных данных

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	925.0	2.919068	0.897082	0.547391	2.314541	2.903233	3.546415	5.314144
Плотность, кг/м3	925.0	1974.319439	71.143154	1784.482245	1923.255135	1977.302956	2020.828331	2161.565216
модуль упругости, ГПа	925.0	735.328695	327.705942	2.436909	498.519344	738.736842	956.246864	1628.000000
Количество отвердителя, м.%	925.0	111.086088	26.830806	38.668500	92.834720	111.157879	130.163998	181.828448
Поверхностная плотность, г/м2	925.0	483.435186	279.103156	0.603740	267.140817	460.721126	694.589685	1291.340115
Модуль упругости при растяжении, ГПа	925.0	73.304924	3.011295	65.793845	71.279418	73.253725	75.309657	81.203147
Прочность при растяжении, МПа	925.0	2460.116851	450.916069	1250.392802	2149.974687	2456.395009	2751.235671	3636.892992
Шаг нашивки	925.0	6.918383	2.514591	0.037639	5.117477	6.947322	8.606411	13.732404
Плотность нашивки	925.0	57.568830	11.065402	28.661632	50.280645	57.615327	64.854936	86.012427
Угол нашивки, град_0	925.0	0.486486	0.500088	0.000000	0.000000	0.000000	1.000000	1.000000
Угол нашивки, град_90	925.0	0.513514	0.500088	0.000000	0.000000	1.000000	1.000000	1.000000
Содержание эпоксидных групп, г	925.0	48.129428	13.667150	13.158372	38.137716	48.089010	57.358358	84.967714
Потребление смолы, г/м2 без ЭГ	925.0	168.856094	44.576887	50.427004	139.606267	168.753112	199.907147	284.828567

Рисунок 17 – Описательная статистика обработанных данных

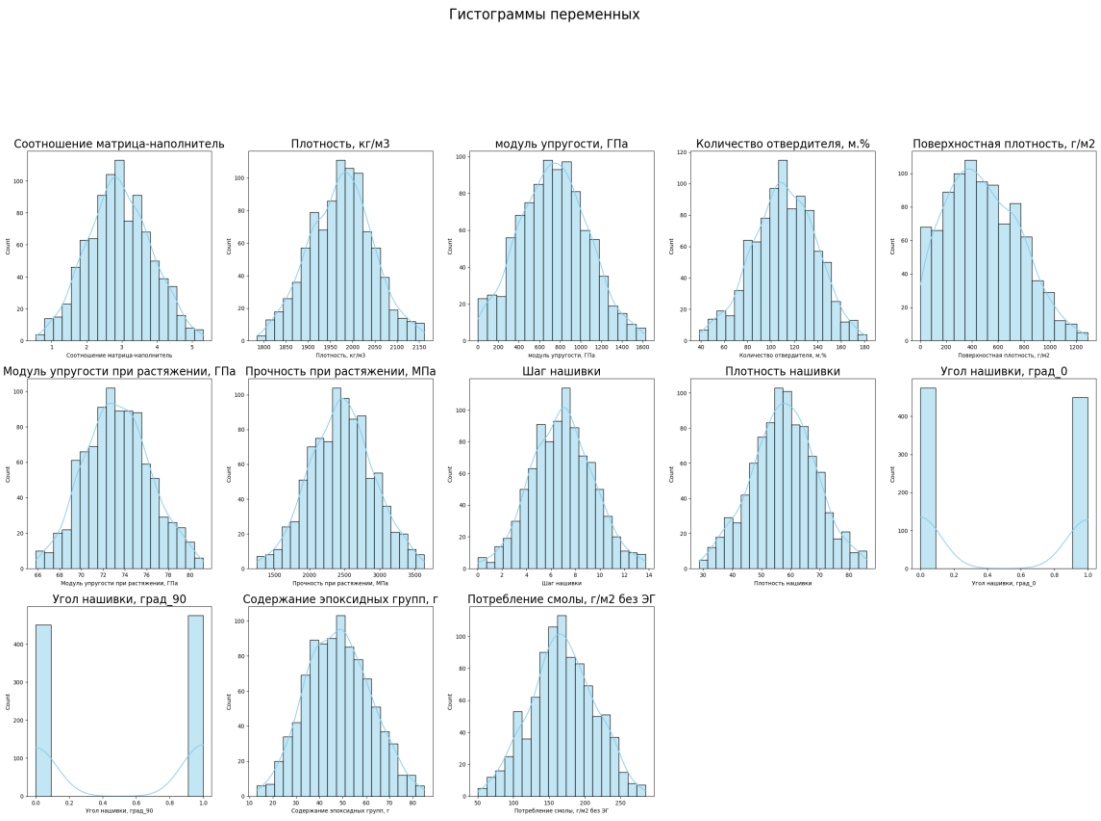

Рисунок 18 – гистограммы распределения переменных

На основании гистограмм распределения и статистического теста Шапиро-Уилка делаем выводы о том, что значения переменных "Соотношение матрица-наполнитель", "Плотность, кг/м3", "Прочность при растяжении, МПа", "Шаг

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

нашивки", "Плотность нашивки" распределены нормально. Значения переменных "модуль упругости, ГПа", "Количество отвердителя, м.%", "Модуль упругости при растяжении, ГПа", "Поверхностная плотность, г/м2", "Угол нашивки, град_0", "Угол нашивки, град_0","Содержание эпоксидных групп, г", "Потребление смолы, г/м2 без ЭГ" распределены не нормально.

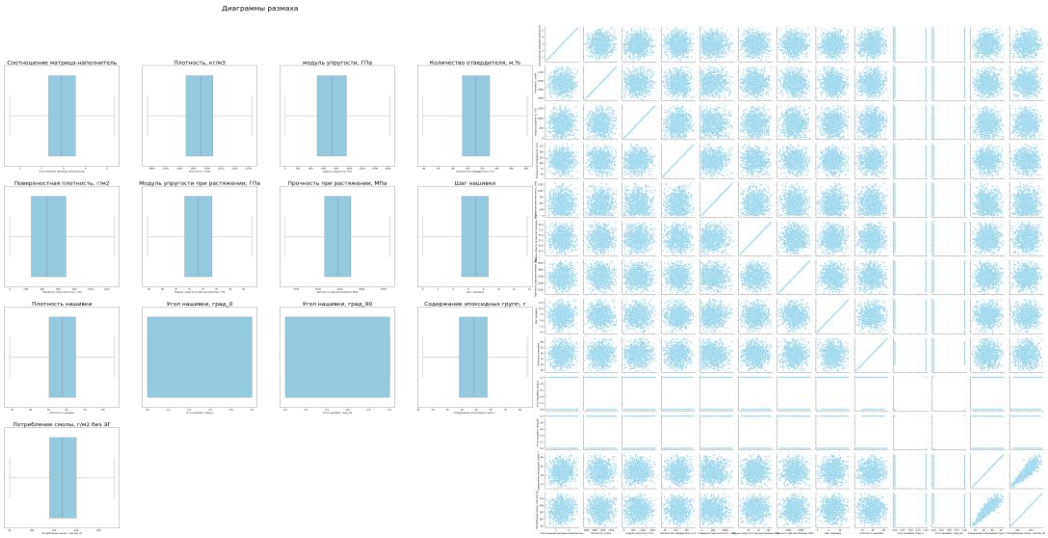

Рисунок 19 – диаграммы размаха и попарного рассеяния переменных

Обращаем внимание на то, что существуем сильная корреляция между созданными переменными Содержание эпоксидных групп, г" и "Потребление смолы, г/м2 без ЭГ"

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

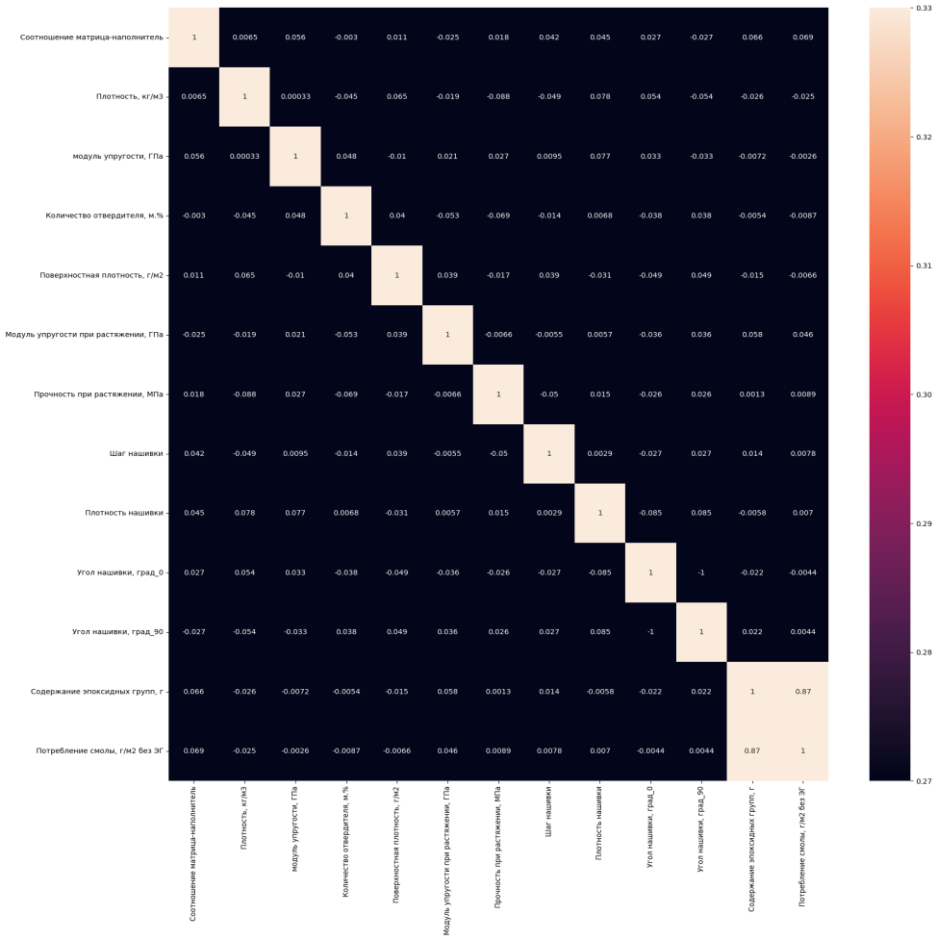

Рисунок 20 – тепловая карта обработанных данных

Необходимо отметить, что многие переменные не распределены нормально, практически полное отсутствие корреляции между переменными и линейной связи, что может ухудшить предсказательную способность прогнозирующих моделей машинного обучения.

2.2 Разработка и обучение модели

Подготавливаем таблицы для создания обучающей и тестовой выборки, а именно исключаем и выносим в отдельные таблицы переменные, в отношении которых необходимо подготовить прогнозирующую модель.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

```
data_for_models = df.copy()
X_proch = df.drop("Прочность при растяжении, МПа", axis=1)
y_proch = df['Прочность при растяжении, МПа']

X_mod_uprug = df.drop("Модуль упругости при растяжении, ГПа", axis=1)
y_mod_uprug = df["Модуль упругости при растяжении, ГПа"]
```

Рисунок 21 – подготовка таблиц, исключение зависимых переменных

Нормализация (Max-Min Normalization, Min-Max Scaling) – техника преобразования значений признака, масштабирующая значения таким образом, что они располагаются в диапазоне от 0 до 1. Цель такого преобразования – изменить значения числовых столбцов в наборе данных так, чтобы сохранить различия их диапазонов. В Машинном обучении данные требует нормализации, когда признаки имеют разные диапазоны и тем самым способствуют искажению восприятия взаимоотношений между переменными-предикторами (Predictor Variable) и целевой переменной (Target Variable).

Принимаем решения о нормализации данных. Используем MinMaxScaler из модуля sklearn.

```
mms_proch = MinMaxScaler()
col = X_proch.columns
result = mms_proch.fit_transform(X_proch)

X_proch = pd.DataFrame(result, columns = col)
X_proch.describe()
```

Рисунок 22 – нормализуем данные с помощью MinMaxScaler

Создаем обучающие и тестирующие выборки для прогноза по каждой зависимой переменной. Разделение по принципу 70% данных относится к обучающей выборки и 30% относится к тестирующей выборки. Используем инструмент train_test_split из модуля sklearn.

Далее разрабатываем модели машинного обучения.

```
# При построении модели пользуемся библиотекой sklearn.linear_model
regressor = LinearRegression()
regressor.fit(X_train_proch, y_train_proch)
y_pred_proch = regressor.predict(X_test_proch)
```

Рисунок 23 – код построения модели линейной регрессии

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

```
# при построении модели используем библиотеку sklearn
knn = KNeighborsRegressor()

knn.fit(X_train_proch, y_train_proch)
y_pred = knn.predict(X_test_proch)
```

Рисунок 24 – Построение и тестирование модели Регрессии на основе k-ближайших соседей (KNeighborsRegressor) для предсказания переменной "Прочность при растяжении, МПа"

К-ближайших соседей (K-Nearest Neighbors или просто KNN) — алгоритм классификации и регрессии, основанный на гипотезе компактности, которая предполагает, что расположенные близко друг к другу объекты в пространстве признаков имеют схожие значения целевой переменной или принадлежат к одному классу.

Проведем поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10. Используем объект GridSearchCV из библиотеки sklearn, который поможет выбрать наилучшие параметры.

```
# Подберём параметры используя GridSearchCV
n_neighbors = list(range(1, 50))
p = [1, 2]
weights = ["uniform", "distance"]
metric = ["mahalanobis", "minkowski", "cosine", "chebyshev", "correlation", "euclidean"]
algorithm = ["ball_tree", "kd_tree", "brute"]

hyperparameters = dict(n_neighbors=n_neighbors, weights=weights, p=p, metric=metric, algorithm=algorithm)
```

```
knn = KNeighborsRegressor()
search = GridSearchCV(knn, hyperparameters, cv=10, verbose=1)
best_model = search.fit(X_train_proch, y_train_proch)
```

```
Fitting 10 folds for each of 3528 candidates, totalling 35280 fits
```

```
best_model.best_estimator_
```

KNeighborsRegressor

▼ Parameters

n_neighbors	48
weights	'uniform'
algorithm	'ball_tree'
leaf_size	30
p	1
metric	'minkowski'
metric_params	None
n_jobs	None

```
knn = best_model.best_estimator_
knn.fit(X_train_proch, y_train_proch)
y_pred = knn.predict(X_test_proch)
```

Рисунок 25 – поиск гиперпараметров с помощью объекта GridSearchCV из библиотеки sklearn

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

2.3 Тестирование модели

Метрики в машинном обучении – это количественные показатели, используемые для оценки производительности моделей. Они позволяют оценить, насколько хорошо модель выполняет поставленную задачу, будь то классификация, регрессия или другие типы задач. Выбор правильной метрики критически важен для оценки и сравнения различных моделей, а также для оптимизации их работы.

При оценке моделей использованы следующие метрики:

1) Корень средней квадратичной ошибки (RMSE) является одним из двух основных показателей эффективности для моделей прогнозирования регрессии. Это исследование устанавливает среднюю разницу между значениями, спрогнозированными моделями прогнозирования и фактическими значениями. Он дает оценку того, что хорошая модель может предсказать целевое значение (оценку точности);

2) MAE, или Mean Absolute Error (Средняя абсолютная ошибка), это метрика, которая используется для оценки точности моделей машинного обучения, в частности, регрессионных моделей. Она показывает среднее значение абсолютных разностей между прогнозируемыми и фактическими значениями. Чем меньше значение MAE, тем лучше модель предсказывает данные;

3) R2 - Статистический показатель, отражающий объясняющую способность регрессии f: X→Y и определяемый как доля дисперсии зависимой переменной, объясненная регрессионной моделью с данным набором независимых переменных. Коэффициент детерминации изменяется в диапазоне от $-\infty$ до 1. Если он равен 1, это соответствует идеальной модели, когда все точки наблюдений лежат точно на линии регрессии, т.е. сумма квадратов их отклонений равна 0. Если коэффициент детерминации равен 0, это означает, что связь между переменными регрессионной модели отсутствует, и вместо нее для оценки значения

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

выходной переменной можно использовать простое среднее ее наблюдаемых значений;

4) MAPE, или средняя абсолютная процентная ошибка (Mean Absolute Percentage Error), это метрика, используемая для оценки точности моделей прогнозирования. Она показывает среднее значение абсолютной процентной разницы между фактическими и прогнозируемыми значениями. Другими словами, MAPE измеряет, насколько в среднем прогнозы отличаются от реальных значений в процентах.

```
# При построении модели пользуемся библиотекой sklearn.linear_model
regressor = LinearRegression()
regressor.fit(X_train_mod_uprug, y_train_mod_uprug)
y_pred_mod_uprug = regressor.predict(X_test_mod_uprug)

# Визуализируем на диаграмме рассеяния соотношение предсказанных и настоящих значений
plt.figure(figsize=(5, 5))
plt.scatter(y_test_mod_uprug, y_pred_mod_uprug, color="skyblue")
plt.xlabel("Настоящие значения")
plt.ylabel("Предсказанные значения")
plt.show()

# Выводим значения метрик
print("Корень из среднеквадратичной ошибки:", np.sqrt(metrics.mean_squared_error(y_test_mod_uprug, y_pred_mod_uprug)))
print("MAE:", metrics.mean_absolute_error(y_test_mod_uprug, y_pred_mod_uprug))
print("R2", metrics.r2_score(y_test_mod_uprug, y_pred_mod_uprug))
print("MAPE:", metrics.mean_absolute_percentage_error(y_test_mod_uprug, y_pred_mod_uprug))
print("Коэффициенты регрессии:", regressor.coef_)
print("Константа:", regressor.intercept_)
```

Рисунок 26 – пример, кода с обучением и тестированием линейной регрессии

Ниже придем метрики моделей и сравнительные гистограммы итоговых метрик моделей

	RMSE	MAE	R2	MAPE
LinearRegression_proch	451.342041	365.029937	-0.003965	0.159865
LinearRegression_uprug	2.986901	2.421378	-0.012082	0.032981
KNR_GSCV_proch	451.342041	365.029937	-0.003965	0.159865
KNR_GSCV_upr	2.982421	2.434199	-0.009048	0.033143
RFR_GSCV_proch	449.623872	365.954796	0.003665	0.160283
RFR_GSCV_upr	2.988808	2.426734	-0.013374	0.033051
BayesR_proch	450.450401	361.279222	-0.000002	0.158812
BayesR_upr	2.971975	2.416087	-0.001993	0.032909
Lasso_proch	451.339347	365.027883	-0.003953	0.159864
Lasso_upr	2.984351	2.421021	-0.010355	0.032976

Рисунок 27 – итоговые метрики моделей

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Рисунок 28 – гистограммы сравнения итоговых метрик моделей

На основании сравнения можем сделать вывод о том, что в целом все использованные модели машинного обучения дали приблизительно равный результат. Поэтому предлагается использовать линейную регрессию либо среднее арифметическое.

Кроме того, было проведено дополнительное исследование с целью проверки шумов/синтезированных данных отдельных значений. Гипотеза не подтвердилась либо необходимо провести повторной исследование с минимальной предварительной обработкой данных (исключением переменных, формированием новых переменных).

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

2.4 Создание модели нейронной сети, которая будет рекомендовать соотношение матрицы

Нейронная сеть (нейросеть) — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации нервных сетей (биологических нейронных сетей) — сетей нервных клеток (нейронов) живого организма.

Подготавливаем таблицы для создания обучающей и тестовой выборки, а именно исключаем и выносим в отдельный таблицы переменные, в отношении которых необходимо подготовить прогнозирующую модель.

```
df_x_smn = df.drop(["Соотношение матрица-наполнитель"], axis=1)
df_y_smn = df[["Соотношение матрица-наполнитель"]]
```

Рисунок 29 – подготовка таблиц, исключение зависимой переменной «соотношение матрица-наполнитель»

Создание обучающей и тестирующей выборки для прогноза по переменной «соотношение матрица-наполнитель». Разделение по принципу 70% данных относится к обучающей выборки и 30% относится к тестирующей выборки.

```
X_train_smn, X_test_smn, y_train_smn, y_test_smn = train_test_split(df_x_smn, df_y_smn, test_size=0.3, random_state=1
)
```

Рисунок 30 – пример кода создания обучающей и тестирующей выборки

Создаем нормализационный слой. Это предварительной обработки, который нормализует непрерывные признаки. Этот слой будет сдвигать и масштабировать входные данные в распределение, центрированное вокруг 0 со стандартным отклонением 1.

```
normalizer = layers.Normalization(axis=-1)
normalizer.adapt(np.array(X_train_smn))
```

Рисунок 31 – создание нормализационного слоя

Строим последовательную модель нейронной сети. Модель нейронной сети строится на основе класса Sequential из библиотеки tensorflow.keras.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

```
model_smn = tf.keras.Sequential(
    [
        normalizer,
        layers.Dense(16, activation="tanh"),
        layers.Dense(8, activation="relu"),
        layers.Dense(8, activation="relu"),
        layers.Dense(1),
    ]
)

model_smn.compile(
    optimizer=tf.keras.optimizers.Adam(0.001),
    loss="mean_squared_error",
    metrics=[tf.keras.metrics.RootMeanSquaredError()],
)
```

Рисунок 32 – код создания нейронной сети

Model: "sequential"

Layer (type)	Output Shape	Param #
normalization (Normalization)	(None, 12)	25
dense (Dense)	(None, 16)	208
dense_1 (Dense)	(None, 8)	136
dense_2 (Dense)	(None, 8)	72
dense_3 (Dense)	(None, 1)	9

Total params: 1,302 (5.09 KB)

Trainable params: 425 (1.66 KB)

Non-trainable params: 25 (104.00 B)

Optimizer params: 852 (3.33 KB)

Рисунок 33 – архитектура нейронной сети

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Осуществляем обучение нейронной сети и визуализируем результаты, выводим метрики.

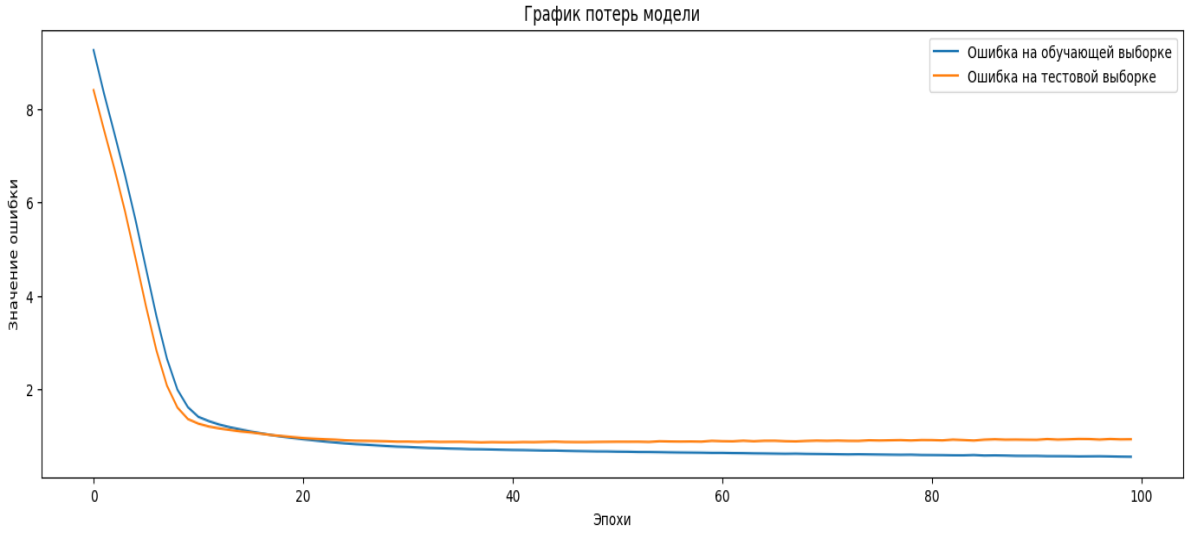

Рисунок 34 – график потерь модели

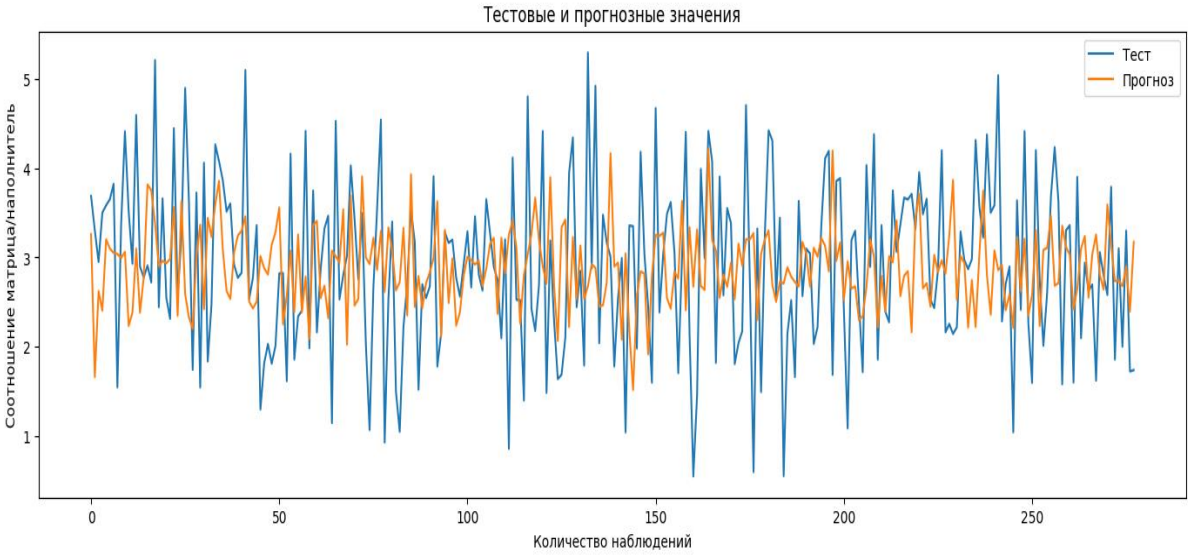

Рисунок 35 – тестовые и прогнозные значения модели

```
Значения метрик:
Корень из среднеквадратичной ошибки: 1.0132991274489693
MSE: 1.0267751216888428
MAE: 0.8330175876617432
R2 -0.1282951831817627
MAPE: 0.3753160536289215
```

Рисунок 36 – итоговые метрики модели

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Сохраняем итоговую модель для последующего ее использования в приложении.

```
with open('model_smn.pkl', 'wb') as file:
    pickle.dump(model_smn, file)
```

Рисунок 37 – сохраняем модель с использованием библиотеки pickle

2.5 Разработка приложения

При разработке приложения нам понадобилась библиотека Flask, при помощи которой мы смогли создать и реализовать функционал нашего приложения. Помимо библиотеки Flask мы ещё использовали библиотеку tensorflow и pickle для внедрения наших моделей в приложение.

```
import numpy as np
import flask
from flask import Flask, render_template
import pickle


app = Flask(__name__, template_folder='templates')

@app.route('/', methods=['POST','GET'])


@app.route('/index', methods=['POST','GET'])
def main():
    if flask.request.method == 'GET':
        return render_template('main.html')

    if flask.request.method == 'POST':
        with open('model_smn.pkl', "rb") as file:
            model_smn = pickle.load(file)

        x_1 = float(flask.request.form['plot'])
        x_2 = float(flask.request.form['m_uprg'])
        x_3 = float(flask.request.form['otv'])
        x_4 = float(flask.request.form['popv'])
        x_5 = float(flask.request.form['mupr'])
        x_6 = float(flask.request.form['pr_ras'])
        x_7 = float(flask.request.form['s_nash'])
        x_8 = float(flask.request.form['pl_nash'])
        x_9 = float(flask.request.form['ug_0'])
        x_10 = float(flask.request.form['ug_90'])
        x_11 = float(flask.request.form['seg'])
        x_12 = float(flask.request.form['psmol'])
        spisok = np.array([[x_1, x_2, x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_10,x_11,x_12,]])
        y_predict = model_smn.predict(spisok)

        return render_template('main.html', result = y_predict)

if __name__ == '__main__':
    app.run(debug=True)
```

Рисунок 38 – код приложения

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

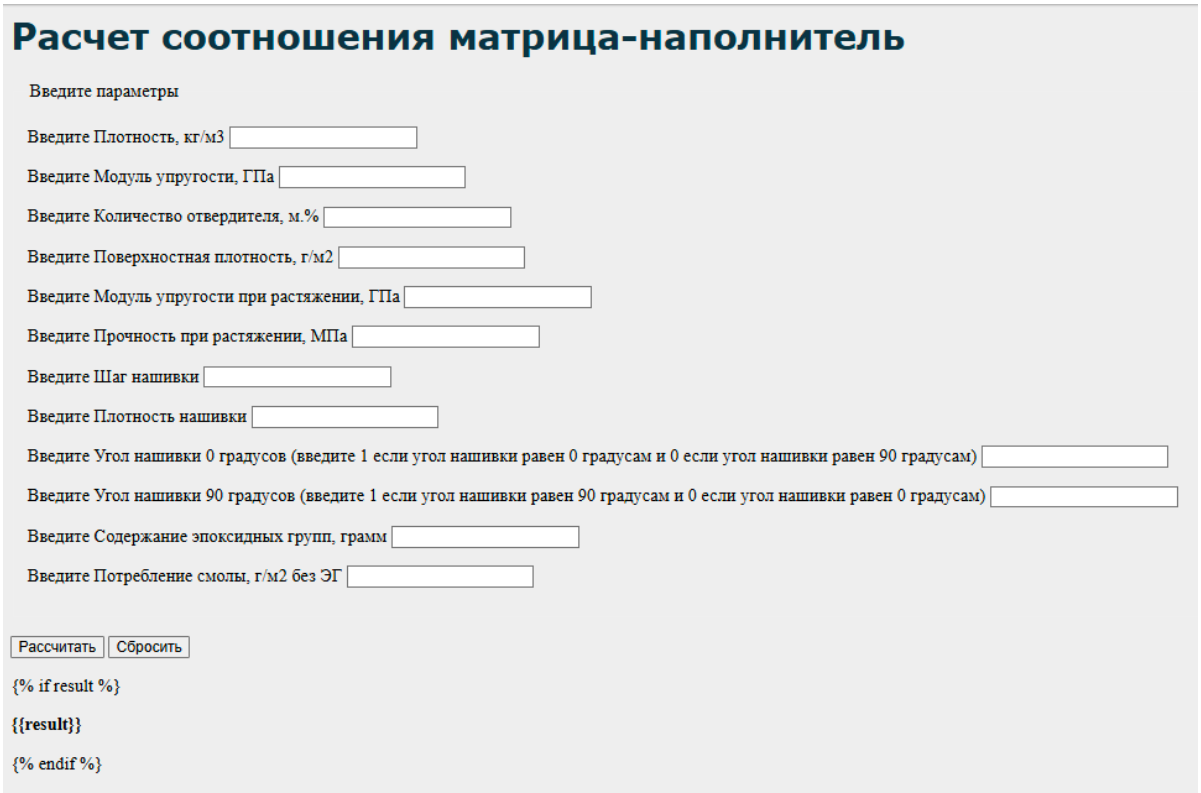

Рисунок 39 – интерфейс приложения

Для использования приложения необходимо:

1) Запустить файл my_app.py

2) Выполнить код из файла my_app.py

3) Перейти в интернет браузере по адресу: http://127.0.0.1:5000. По данному адресу открывает веб-интерфейс приложения.

4) В веб-интерфейса необходимо ввести все независимые переменные. Все поля являются обязательными для заполнения. Значения вводятся в виде целых чисел или чисел с плавающей точкой (запятые или текстовые символы не используются).

5) Для получения прогноза необходимо нажать кнопку «рассчитать».

6) Для удаления заполненных значений можно также использовать кнопку «сброс».

7) После нажатия кнопки «рассчитать». Выводится прогнозное значение.

8) Для остановки работы программы в окне терминала (IDLE) нажимается команда ctrl + c

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

2.6 Создание удаленного репозитория

Репозиторий был создан на платформе github.com по адресу: https://github.com/CyberNychu/VKR_Data_science.

В репозиторий размещены все необходимые файлы и документы.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Заключение

Процессе выполнения выпускной квалификационной работы были исполнены следующие задачи:

1) Изучены теоретические основы и методы решения поставленной задачи;
2) Проведен разведочный анализ предложенных данных;
3) Проведена предварительная обработка предложенных данных;
4) Построены, обучены и протестированы модели машинного обучения для прогноза значений переменных «Модуль упругости при растяжении, ГПа» и «Прочность при растяжении, МПа»;
5) Построена, обучена и протестирована модель нейронной сети, которая прогнозирует значение переменной «Соотношение матрица-наполнитель»;
6) Разработано приложение, которое использует модель нейронной сети и выдает прогнозное значение переменной «Соотношение матрица-наполнитель». Подготовлена инструкция по использованию программы;
7) Создан репозиторий в GitHub. В нем размещены итоговые документы и код исследования, оформлен файл README.

Необходимо отметить, что все построенные модели дали приблизительно равные результаты. Ни одна из моделей не была дала достаточно точна в прогнозе.

Коэффициент детерминации R2 у всех моделей близок к 0 и указывает что связь между переменными регрессионной модели отсутствует, и вместо нее для оценки значения выходной переменной можно использовать простое среднее ее наблюдаемых значений.

Можно отметить, что на результаты прогнозной способности моделей могли повлиять:

- отсутствие линейной связи / корреляции между переменными (что было выявлено на этапе исследования данных);
- отсутствие нормального распределения у отдельных переменных;

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

- измерение переменных в разных единицах (проценты, г/м3, м.%, С_2, %_2, г/м2, ГПа, МПа);

- отсутствие четкого понимания как происходил сбор данных и какие переменные могут быть производными или соотноситься между собой;

- не все переменные достаточно объяснены и замерены (например, угол нашивки);

- кроме того, необходимо учитывать, что композитные материалы являются сложными материалами и в результате объединения компонентов их свойства могут изменяться существенным образом.

Таким образом, для улучшения прогнозной способности модели необходимо получить информацию о том, как собирались данные, какие признаки могли быть синтезированы, проработать возможность группировки переменных и приведения их в схожие единицы измерения.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

Список использованной литературы

1 Сара Бослаф, Статистика для всех. / Пер. с англ. П.А. Волкова, И.М. Флямер, М.В. Либерман, А.А. Голицына. – М.: ДМК Пресс, 2017. – 586 с.: ил.

2 Уэс Маккинни, Python и анализ данных: Первичная обработка данных с применением pandas, NumPy и Jupiter. / Пер. с англ. А.А. Слинкина. 3-е изд. – М.: ДМК Пресс, 2023. – 536 с.: ил.

3 Орельен Жерон, Прикладное машинное обучение с помощью Skikit-Learn, Keras и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем, 2-е изд. / Пер. с англ. – СПб.: ООО «Диалектика», 2020. – 1040 с.: ил. – Парал. тит. англ.

4 Джоэл Грас, Data Science. Наука о данных с нуля. / Пер. с англ. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2024. – 416 с.: ил.

5 Томас Нилд, Математика для Data Science. Управляем данными с помощью линейной алгебры, теории вероятностей и статистики. – Астана: «Спринт Бук», 2025. – 352с.: ил.

6 Питер Брюс, Практическая статистика для специалистов Data Science. / Пер. с англ. / П. Брюс, Э.Брюс, П. Гедек. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2024. – 352 с.: ил.

7 Франсуа Шолле, Глубокое обучение на Python. 2-е межд. издание. – СПб.: Питер, 2025. – 576 с.: ил. – (Серия «Библиотека программиста»).

8 Владимир Соловьев, Статистика и котики / Савельев Владимир. – Москва: Издательство АСТ, 2024. – 192 с. – (Звезда нонфикшн).

9 Бхарат Рамсундар, TensorFlow для глубокого обучения. / Пер. с англ. / Б. Рамсундар, Р.Б. Заде. – СПб.: БХВ-Петербург, 2020. – 256 с.: ил.

10 Луис Серрано, Грокаем машинное обучение. – СПб.: Питер, 2025. – 512 с.: ил. – (Серия «Библиотека программиста»).

11 Эндрю Траск, Грокаем глубокое обучение. – СПб.: Питер, 2025. – 352 с.: – ил. – (Серия «Библиотека программиста»).

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

12 Адитья Бхаргава, Грокаем алгоритмы. 2-е изд. – СПб.: Питер, 2025. – 352 с.: ил. – (Серия «Библиотека программиста»).

13 Юлий Васильев, Python для data science. – СПб.: Питер, 2023. – 272 с.: ил. – (Серия «Библиотека программиста»).

14 Леонард Апельцин, Data science в действии. – СПб.: Питер, 2023. – 736 с.: ил. – (Серия «Для профессионалов»).

15 Александр Васильевич Маркин, Программирование на SQL: учебник и практикум для вузов / А.В. Маркин – 3-е изд. перераб. и доп. – Москва: Издательство Юрайт, 2025. – 805 с. – (Высшее образование). – Текст: непосредственный.

16 Эрик Мэтиз, Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. – СПб.: Питер, 2023. – 512 с.: ил. – (Серия «Библиотека программиста»).

17 Дэн Бейдер, Дэвид Эймос, Джоанна Яблонски, Флетчер Хейслер, Знакомство с Python. – СПб.: Питер, 2023. – 512 с.: ил. – (Серия «Библиотека программиста»).

18 Себастьян Рашка, Машинное обучение с PyTorch и Scikit-Learn. / Пер. с англ. / С. Рашка, Ю.Лю, В. Мирджалили. – Астана: Фолиант, 2024. – 688 с.: ил.

19 Пол Бэрри, Изучаем программирование на Python / Пол Бэрри; [пер. с англ. М.А. Райтман]. – Москва: Эксмо, 2022. – 624 с.: ил. – (Мировой компьютерный бестселлер).

20 Марк Лутц, Изучаем Python, том 1, 5-е изд.: Пер. с англ. – СПб.: ООО «Диалектика», 2020. – 832 с.: ил. – Парал. тит. англ.

21 Марк Лутц, Изучаем Python, том 2, 5-е изд.: Пер. с англ. – СПб.: ООО «Диалектика», 2020. – 720 с.: ил. – Парал. тит. англ.

22 Композиционные материалы: учебное пособие для вузов / Д. А. Иванов, А. И. Ситников, С. Д. Шляпин ; под редакцией А. А. Ильина. — Москва: Издательство Юрайт, 2019 — 253 с. — (Высшее образование). — Текст: непосредственный.

Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

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

24 Ф.М. Гафаров, А.Ф. Галимянов, Искусственные нейронные сети и приложения: учеб. пособие /Ф.М. Гафаров, А.Ф. Галимянов. – Казань: Издательство Казанского университета, 2018. – 121 с.

25 Валерий Костиков, Физико-химические основы технологии композиционных материалов: директивная технология композиционных материалов: учеб. пособие / В.И. Костиков. – Изд. Дом МИСиС, 2011. – 163 с.

26 Документация по библиотеке pandas: – Режим доступа: https://pandas.pydata.org/docs/user_guide/index.html#user-guide/. (дата обращения: 4.07.2025)

27 Документация по библиотеке numpy: – Режим доступа: https://numpy.org/doc/1.22/user/index.html#user/. (дата обращения: 4.07.2025)

27 Документация по библиотеке matplotlib: – Режим доступа: https://matplotlib.org/stable/users/index/. (дата обращения: 1.07.2025)

29 Документация по библиотеке skikit-learn: – Режим доступа: https://scikit-learn.org/stable/user_guide.html/. (дата обращения: 1.07.2025)

30 Документация по библиотеке tensorflow: – Режим доступа: https://www.tensorflow.org/guide?hl=ru/. (дата обращения: 3.07.2025)

31 Документация по библиотеке flask: – Режим доступа: https://flask.palletsprojects.com/en/stable/. (дата обращения: 3.07.2025)