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

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по курсу

«Data Science»

Слушатель

Герасимова
Валентина Вениаминовна

Москва, 2023

Содержание

Введение

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

Композиционный материал или композитный материал (КМ) – многокомпонентный материал, изготовленный (человеком или природой) из двух или более компонентов с существенно различными физическими и/или химическими свойствами, которые, в сочетании, приводят к появлению нового материала с характеристиками, отличными от характеристик отдельных компонентов и не являющимися простой их суперпозицией [26].

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.

В составе композита принято выделять матрицу и наполнитель, последние выполняют функцию армирования. В качестве наполнителей композитов как правило выступают углеродные или стеклянные волокна, а роль матрицы играет полимер. Сочетание разных компонентов позволяет улучшить характеристики материала и делает его одновременно лёгким и прочным. При этом отдельные компоненты остаются таковыми в структуре композитов, что отличает их от смесей и затвердевших растворов. Изменяя состав и соотношение матрицы и ориентацию наполнителя, можно получить большой спектр материалов определенным набором свойств. Многие композиты превосходят традиционные материалы и сплавы по своим механическим свойствам и в то же время они легче. Использование композитов обычно позволяет уменьшить массу конструкции при сохранении или улучшении её механических характеристик.

Современные композиты широко используются в различных областях: например, в судостроении, авиационной промышленности, в космической технике, в металлургии и т.д. Существенным недостатком производства композитов

является их стоимость. Зная характеристики компонентов, невозможно рассчитать свойства композита. Для получения заданных свойств требуется большое количество испытаний различных комбинаций. Сократить время и затраты на создание определенного материала могла бы помочь система поддержки производственных решений, построенная на принципах машинного обучения.

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

Задачи:

1) Провести разведочный анализ предложенных данных.
2) Провести предобработку данных.
3) Обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении.
4) Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель.
5) Разработать приложение с графическим интерфейсом или интерфейсом командной строки.

1. Аналитическая часть

1.1. Постановка задачи

Для работы были даны 2 файла: X_bp.xlsx, состоящий из 1023 строк и 10 столбцов и X_nup.xlsx, состоящий из 1040 строки и 3 столбцов.

По условию задачи файлы необходимо объединить по индексу, а тип объединения – INNER. Так как в исходных файлах разное количество строк, то часть строк из файла X_nup была отброшена. И дальнейшие исследования проводятся с объединенным датасетом, содержащим 13 признаков и 1023 строк или объектов.

Описание признаков объединенного датасета приведено в таблице 1. Все признаки имеют тип float64, то есть вещественный. Пропусков в данных нет. Все признаки, кроме «Угол нашивки», являются непрерывными, количественными. «Угол нашивки» принимает только два значения и будет рассматриваться как категориальный признак.

Таблица 1 – Описание признаков датасета

Название	Файл	Тип данных	Непустых значений	Уникальных значений
Соотношение матрица-наполнител	X_bp	float64	1023	1014
Плотность, кг/м3	X_bp	float64	1023	1013
модуль упругости, ГПа	X_bp	float64	1023	1020
Количество отвердителя, м.%	X_bp	float64	1023	1005
Содержание эпоксидных групп,%_2	X_bp	float64	1023	1004
Температура вспышки, С_2	X_bp	float64	1023	1003

Поверхностная плотность, г/м2	X_bp	float64	1023	1004
Модуль упругости при растяжении, ГПа	X_bp	float64	1023	1004
Прочность при растяжении, МПа	X_bp	float64	1023	1004
Потребление смолы, г/м2	X_bp	float64	1023	1003
Угол нашивки, град	X_nup	float64	1023	2
Шаг нашивки	X_nup	float64	1023	989
Плотность нашивки	X_nup	float64	1023	988

Гистограммы распределения переменных и диаграммы «ящик с усами» приведены на рисунках 1 - 4. По ним видно, что все признаки, кроме «Угол нашивки», имеют нормальное распределение и принимают неотрицательные значения. «Угол нашивки» принимает значения: 0, 90.

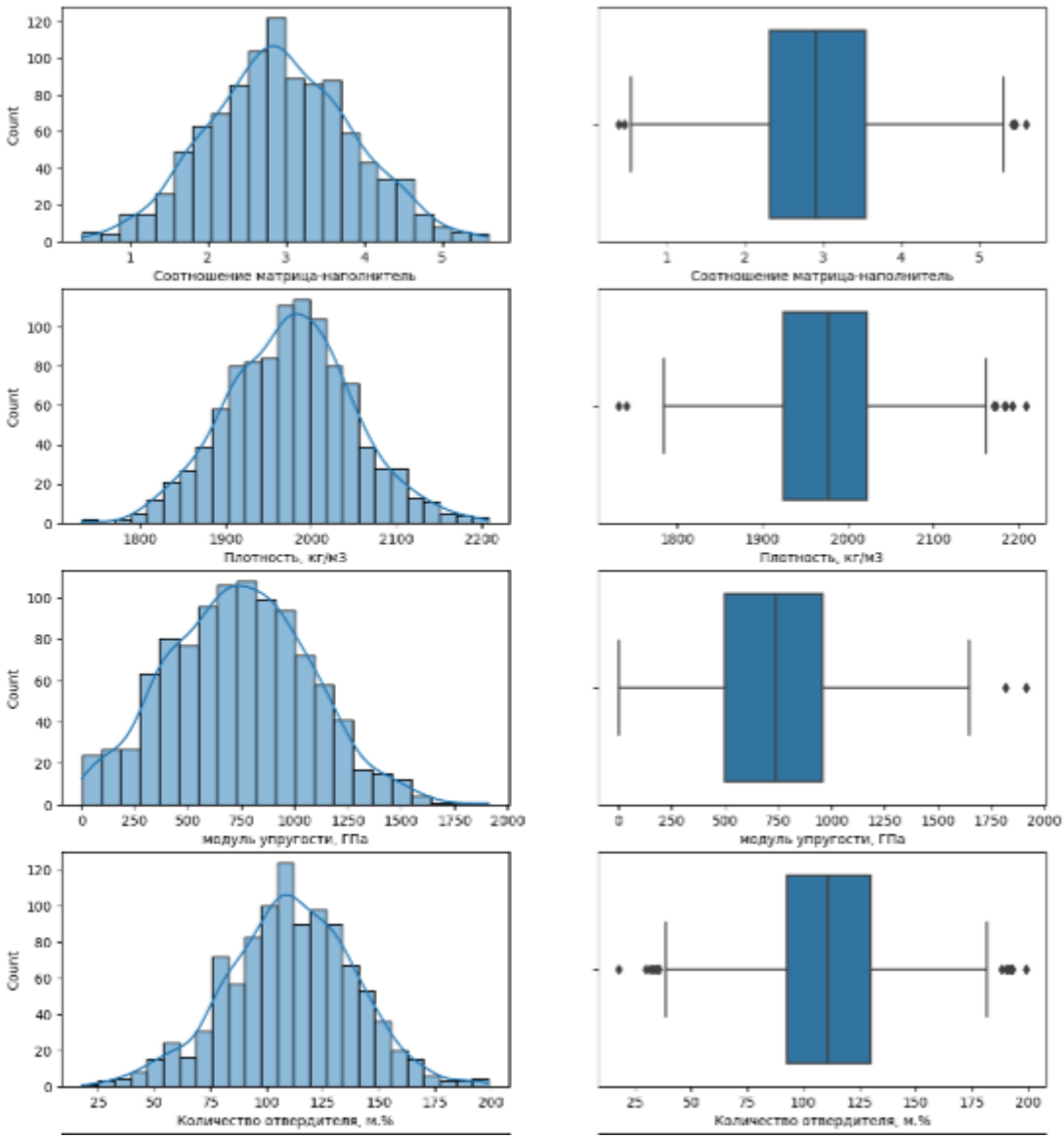

Рисунок 1 – Гистограммы распределения переменных
и диаграммы «ящик с усами»

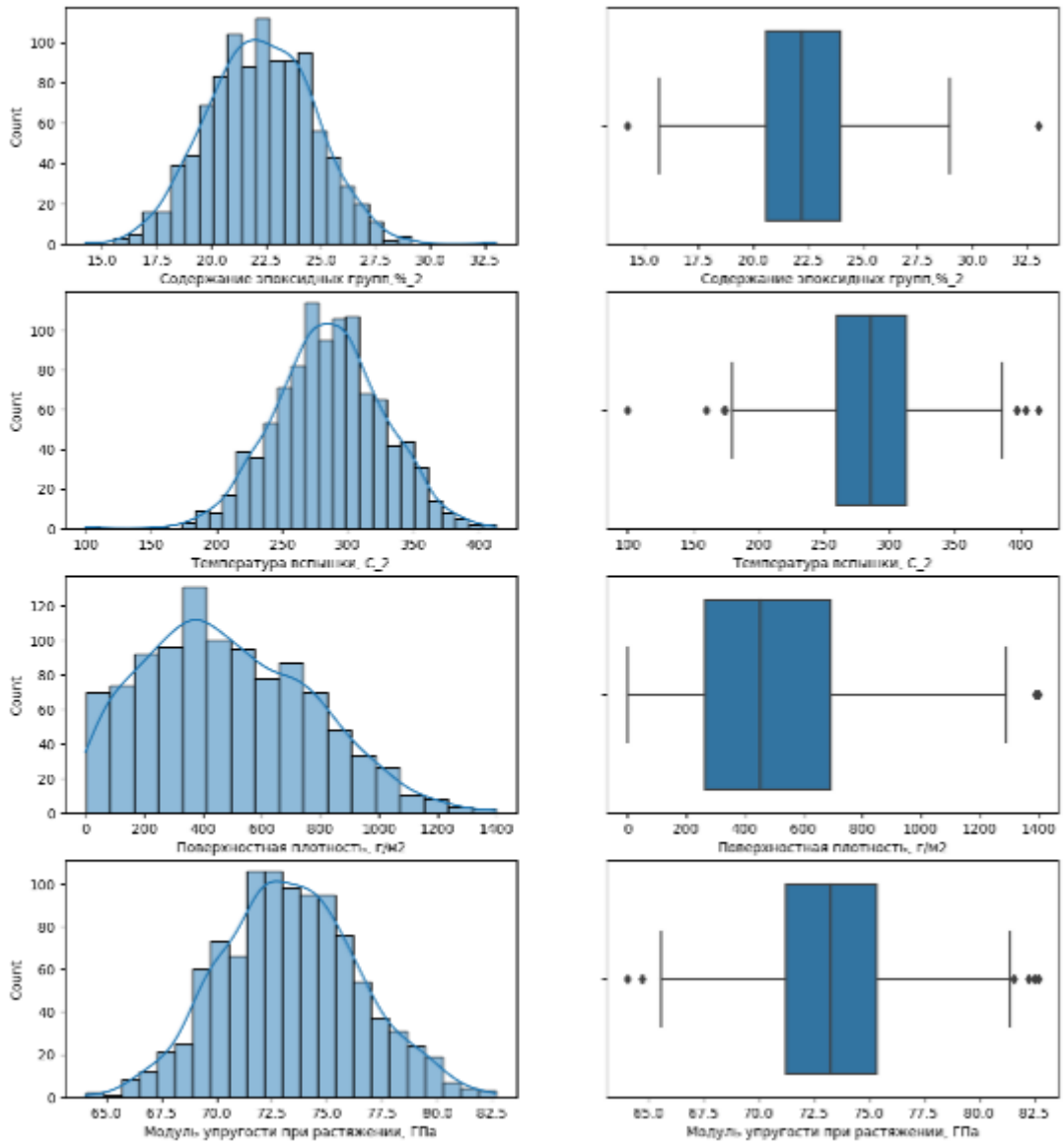

Рисунок 2 – Гистограммы распределения переменных
и диаграммы «ящик с усами»

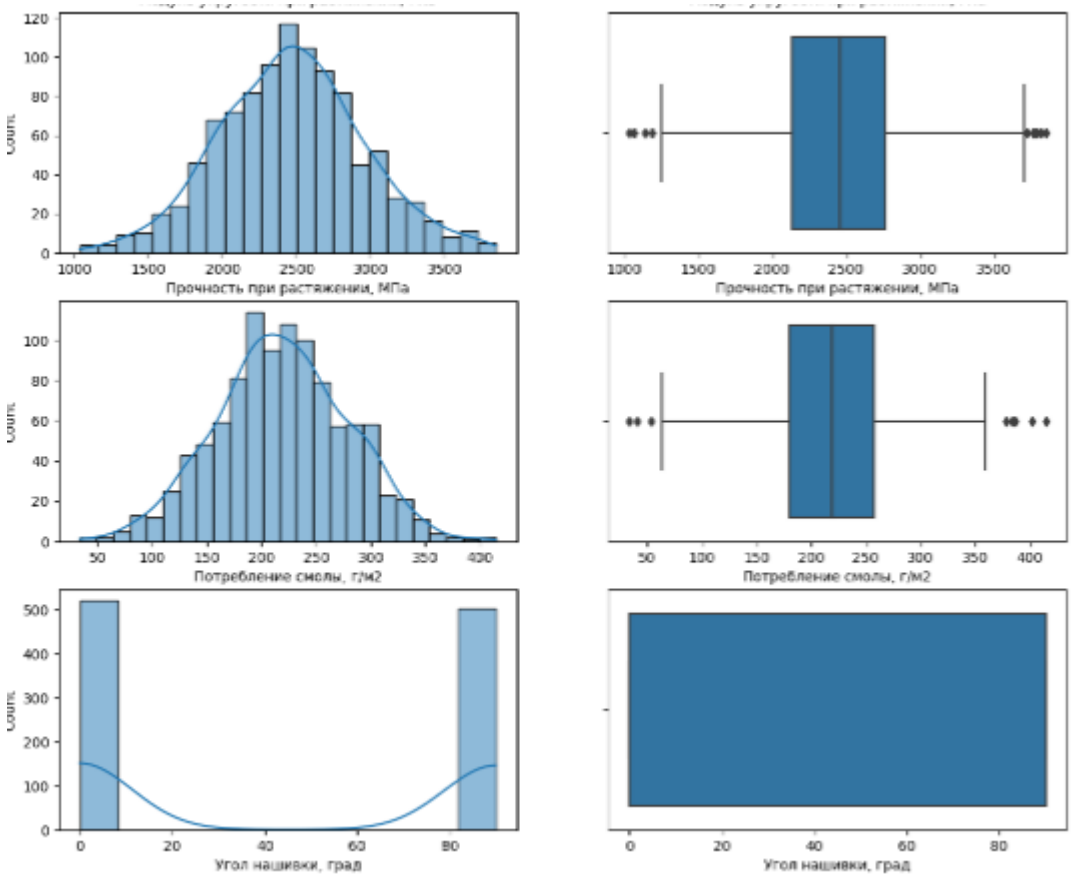

Рисунок 3 – Гистограммы распределения переменных
и диаграммы «ящик с усами»

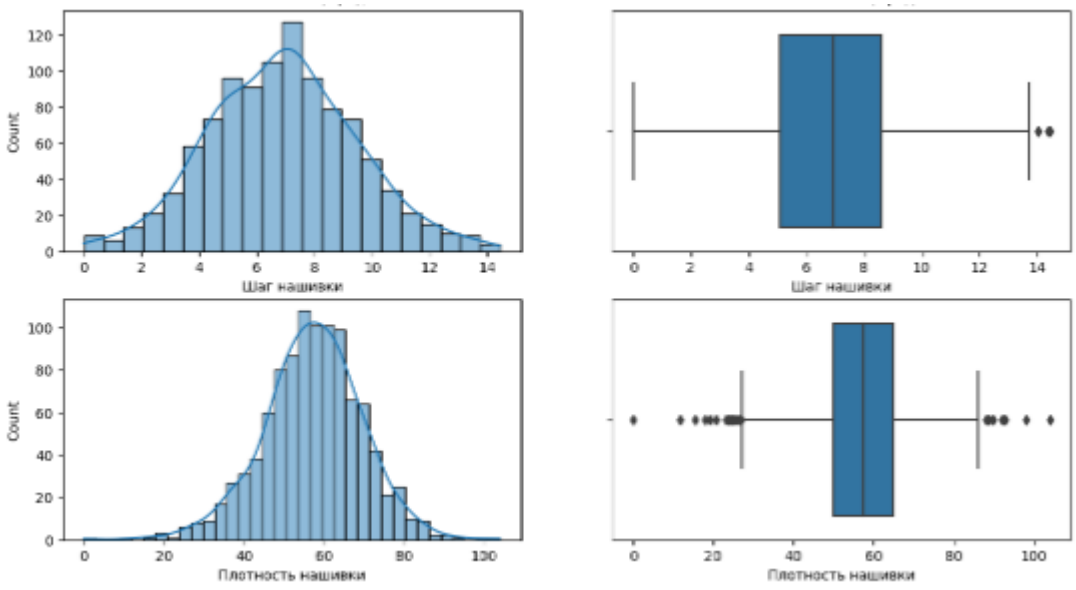

Рисунок 4 – Гистограммы распределения переменных
и диаграммы «ящик с усами»

После анализа данных выявлено, что в датасете отсутствуют пропуски, и дубликаты.

Приведем описательную статистику датасета – таблица 2.

Таблица 2 – Описательная статистика признаков датасета

	Среднее	Стандартное отклонение	Минимум	Максимум	Медиана
Соотношение матрица-наполнитель	2.9304	0.9132	0.3894	5.5917	2.9069
Плотность, кг/м3	1975.7349	73.7292	1731.7646	2207.7735	1977.6217
модуль упругости, ГПа	739.9232	330.2316	2.4369	1911.5365	739.6643
Количество отвердителя, м.%	110.5708	28.2959	17.7403	198.9532	110.5648
Содержание эпоксидных групп,%_2	22.2444	2.4063	14.2550	33.0000	22.2307
Температура вспышки, С_2	285.8822	40.9433	100.0000	413.2734	285.8968
Поверхностная плотность, г/м2	482.7318	281.3147	0.6037	1399.5424	451.8644
Модуль упругости при растяжении, ГПа	73.3286	3.1190	64.0541	82.6821	73.2688
Прочность при растяжении, МПа	2466.9228	485.6280	1036.8566	3848.4367	2459.5245
Потребление смолы, г/м2	218.4231	59.7359	33.8030	414.5906	219.1989
Угол нашивки, град	44.2522	45.0158	0.0000	90.0000	0.0000

Шаг нашивки	6.8992	2.5635	0.0000	14.4405	6.9161
Плотность нашивки	57.1539	12.3510	0.0000	103.9889	57.3419

Попарные графики рассеяния точек приведены на рисунке 5.

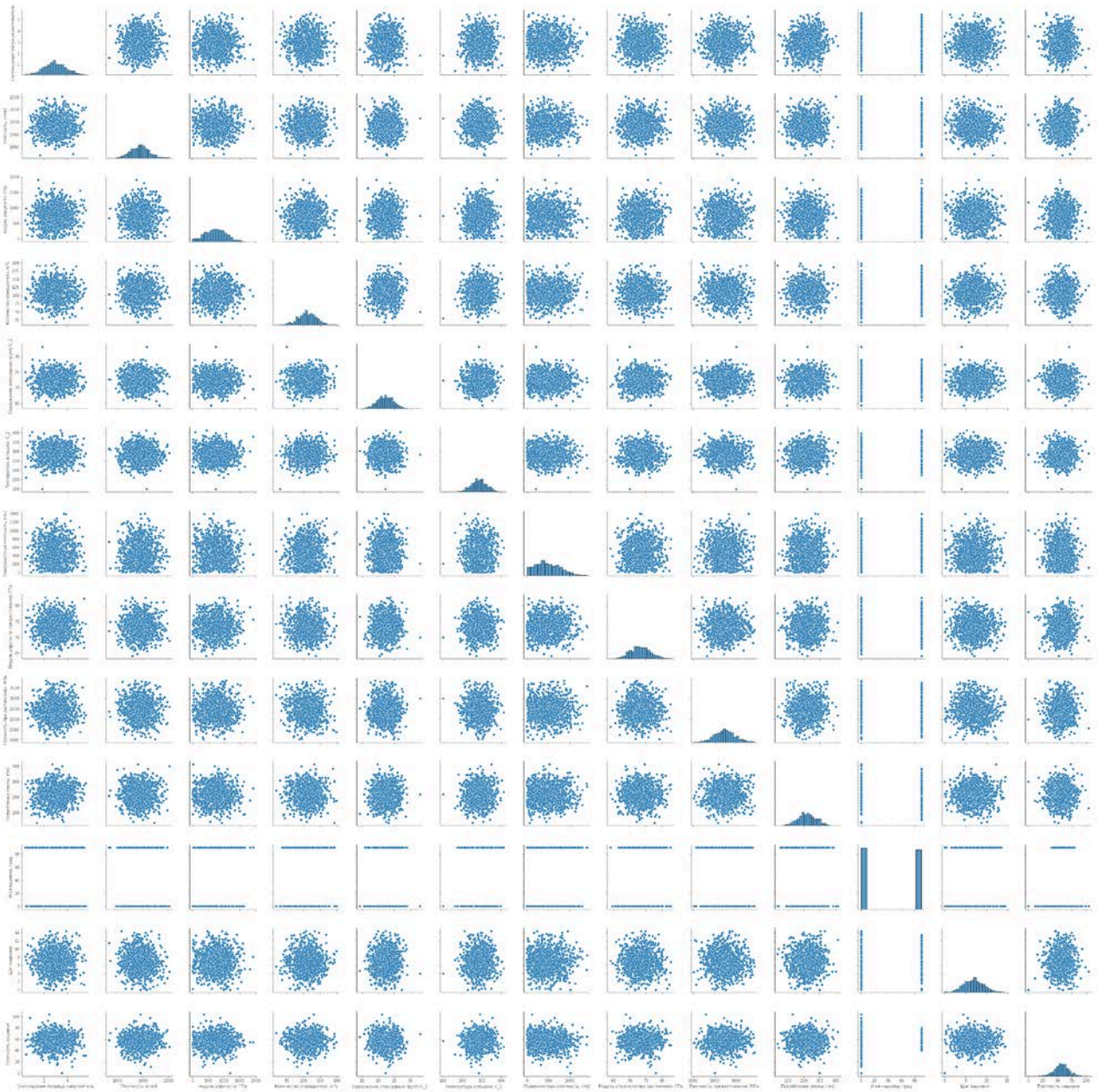

Рисунок 5 - Попарные графики рассеяния точек

По графикам рассеяния мы видим, что некоторые точки отстоят далеко от общего облака. Так визуально выглядят выбросы – аномальные, некорректные значения данных, выходящие за пределы допустимых значений признака.

Для выявления выбросов в датасете будем использовать метод 3-х сигм и IsolationForest (изолирующий лес) [1] .

После применения данных методов было найдено 109 выбросов.

После удаления выбросов в датасете осталось 914 строк и 13 признаков.

В задании целевыми переменными указаны:

– модуль упругости при растяжении, Гпа;

– прочность при растяжении, МПа;

– соотношение матрица-наполнитель.

1.2. Описание используемых методов

Предсказание значений вещественной, непрерывной переменной – это задача регрессии. Эта зависимая переменная должна иметь связь с одной или несколькими независимыми переменными, называемых также предикторами или регрессорами. Регрессионный анализ помогает понять, как значение зависимой переменной изменяется при изменении независимых переменных.

В настоящее время разработано много методов регрессионного анализа. Например, простая и множественная линейная регрессия. Эти модели являются параметрическими в том смысле, что функция регрессии определяется конечным числом неизвестных параметров, которые оцениваются на основе данных. В работе будем использовать следующие методы: линейная регрессия, лассо, Байесовская регрессия, дерево решений, градиентный бустинг, случайный лес. Рассмотрим подробнее данные методы.

Линейная регрессия – используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной y от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) x линейной функцией зависимости [26].

Линейная регрессия основана на гипотезе, что искомая зависимость – линейная. Каждая независимая переменная вносит аддитивный вклад в результирующее значение с некоторым весом, называемом коэффициентом регрессии.

Модель линейной регрессии является часто используемой и наиболее изученной. А именно изучены свойства оценок параметров, получаемых различными методами при предположениях о вероятностных характеристиках факторов, и случайных ошибок модели. Предельные (асимптотические) свойства оценок нелинейных моделей также выводятся исходя из аппроксимации последних линейными моделями.

Достоинства линейной регрессии:

- Скорость и простота получения модели.
- Интерпретируемость модели. Линейная модель является прозрачной и понятной для аналитика. По полученным коэффициентам регрессии можно судить о том, как тот или иной фактор влияет на результат, сделать на этой основе дополнительные полезные выводы.
- Широкая применимость. Большое количество реальных процессов в экономике и бизнесе можно с достаточной точностью описать линейными моделями.
- Изученность данного подхода. Для линейной регрессии известны типичные проблемы (например, мультиколлинеарность) и их решения, разработаны и реализованы тесты оценки статической значимости получаемых моделей.

Главный недостаток линейной регрессии состоит в том, что она может моделировать только прямые линейные зависимости, в то время как часто возникает необходимость создания модели других типов отношений между данными.

Lasso (Least absolute shrinkage and selection operator) – метод оценивания коэффициентов линейной регрессионной модели [26].

Метод заключается во введении ограничения на норму вектора коэффициентов модели, что приводит к обращению в 0 некоторых коэффициентов модели. Метод приводит к повышению устойчивости модели в случае большого числа

обусловленности матрицы признаков X позволяет получить интерпретируемые модели – отбираются признаки, оказывающие наибольшее влияние на вектор ответов.

Байесовская линейная 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 [26].

Конечным результатом байесовского линейного моделирования является не единая оценка параметров модели, а распределение, которое мы можем использовать, чтобы сделать выводы о новых наблюдениях. Это распределение позволяет нам продемонстрировать нашу неопределенность в модели и является одним из преимуществ байесовских методов моделирования, по мере увеличения количества точек данных неопределенность должна уменьшаться, что свидетельствует о более высоком уровне достоверности в наших оценках.

Дерево принятия решений (также называют деревом классификации или регрессионным деревом) – средство поддержки принятия решений, использующееся в машинном обучении, анализе данных и статистике [26].

Структура дерева представляет собой «листья» и «ветки». На рёбрах («ветках») дерева решения записаны признаки, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах – признаки, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение.

Подобные деревья решений широко используются в интеллектуальном анализе данных. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе.

В отличие от остальных методов добычи данных, метод дерева принятия решений имеет несколько достоинств: прост в понимании и интерпретации; высокая точность работы; не требует специальной подготовки данных, как например: нормализации данных, добавления фиктивных переменных, а также удаления пропущенных данных.

Недостаток деревьев решений – склонность переобучаться. Переобучение в случае дерева решений – это точное распознавание примеров, участвующих в обучении и полная несостоятельность на новых данных. В худшем случае, дерево будет большой глубины и сложной структуры, а в каждом листе будет только один объект. Для решения этой проблемы используют разные критерии остановки алгоритма.

Метод случайного леса (англ. random forest) – алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев [26].

Алгоритм сочетает в себе две основные идеи: метод бэггинга Бреймана и метод случайных подпространств, предложенный Тин Кам Хо. Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.

Достоинства случайного леса: высокая точность предсказания; редко переобучается; практически не чувствителен к выбросам в данных; одинаково хорошо обрабатывает как непрерывные, так и дискретные признаки, данные с большим числом признаков; высокая параллелизуемость и масштабируемость.

Из недостатков можно отметить, что его построение занимает больше времени. Так же теряется интерпретируемость.

Градиентный бустинг (GradientBoosting) – это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений

[26]. В отличие от случайного леса, где каждый базовый алгоритм строится независимо от остальных, бустинг воплощает идею последовательного построения линейной комбинации алгоритмов. Каждый следующий алгоритм старается уменьшить ошибку предыдущего.

Чтобы построить алгоритм градиентного бустинга, нам необходимо выбрать базовый алгоритм и функцию потерь или ошибки (loss). Loss-функция – это мера, которая показывает насколько хорошо предсказание модели соответствуют данным [26].

Из недостатков алгоритма можно отметить только затраты времени на вычисления и необходимость грамотного подбора гиперпараметров.

Нейронная сеть – это последовательность нейронов, соединенных между собой связями [26]. Структура нейронной сети пришла в мир программирования из биологии. Вычислительная единица нейронной сети – нейрон или персептрон.

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

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

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

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

– входной слой – его размер соответствует входным параметрам;

– скрытые слои – их количество и размерность определяем специалист;

– выходной слой – его размер соответствует выходным параметрам.

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

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

Для обновления весов в модели используются различные оптимизаторы.

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

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

1.3. Разведочный анализ данных

Цель разведочного анализа данных – выявить закономерности в данных. Для корректной работы большинства моделей желательна сильная зависимость выходных переменных от входных и отсутствие зависимости между входными переменными.

На рисунке 5 изображен график попарного рассеяния точек. По форме «облаков точек» не заметно зависимостей, которые станут основой работы моделей. Помочь выявить связь между признаками может матрица корреляции, приведенная на рисунке 6.

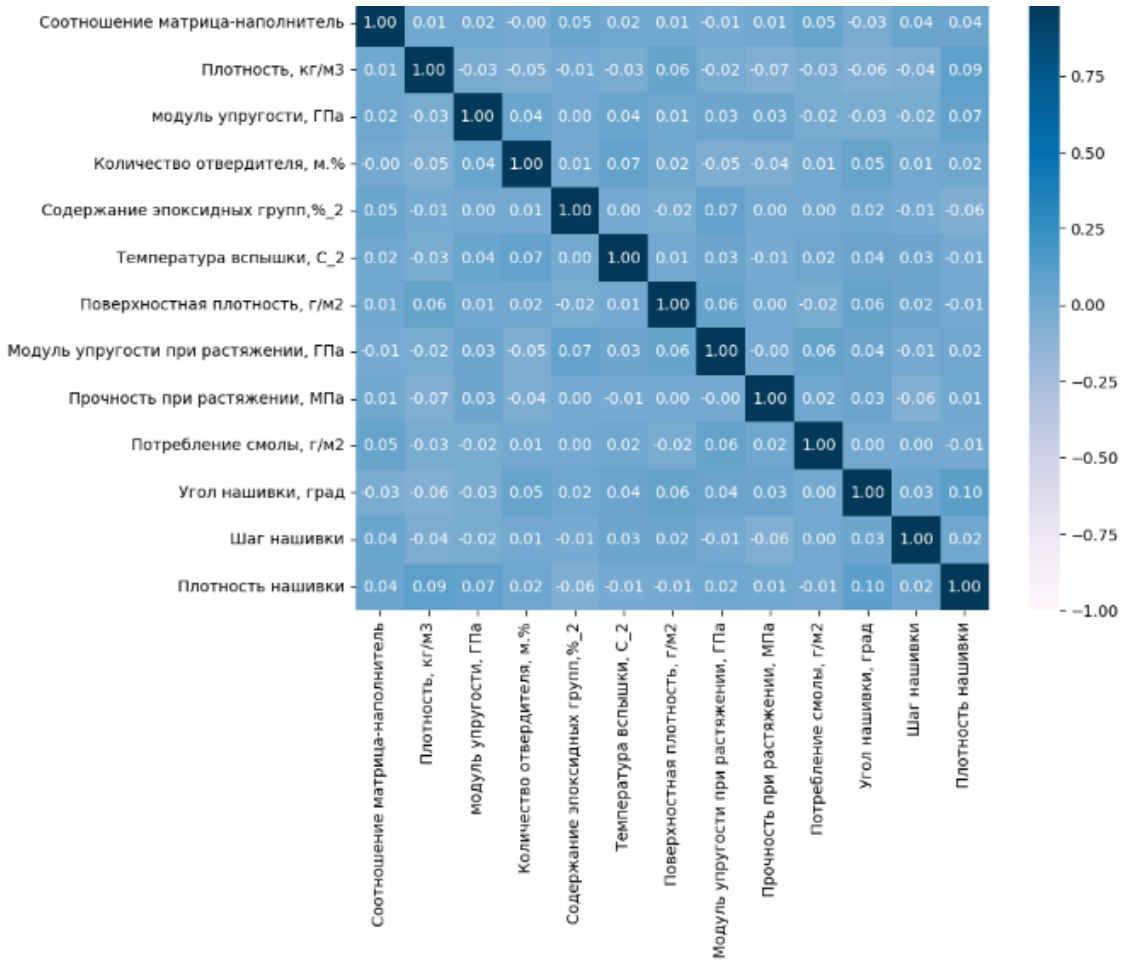

Рисунок 6 – Матрица корреляции

По матрице корреляции видно, что все коэффициенты корреляции близки к нулю, что означает отсутствие линейной зависимости между признаками. В задании целевыми переменными указаны модуль упругости при растяжении, Гпа; прочность при растяжении, МПа и соотношение матрица-наполнитель. Для каждой целевой переменной решим отдельные задачи. Для каждой задачи необходимо выполнить следующее:

- разделить данные на тренировочную и тестовую выборки. Для тестовой выборки необходимо оставить 30% данных;
- выполнить препроцессинг, то есть подготовку исходных данных;
- выбрать базовую модель для определения нижней границы качества предсказания.

- взять несколько моделей с гиперпараметрами по умолчанию, и используя перекрестную проверку, посмотреть их метрики на тренировочной выборке;
- подобрать для этих моделей гиперпараметры с помощью с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10;
- сравнить метрики моделей после подбора гиперпараметров и выбрать лучшую;
- получить предсказания лучшей и базовой моделей на тестовой выборке, сделать выводы;
- сравнить качество работы лучшей модели на тренировочной и тестовой выборке.

Произведем предварительную обработку данных (препроцессинг). Препроцессинг необходим для обеспечения корректной работы моделей.

Его необходимо выполнять после разделения на тренировочную и тестовую выборку.

Препроцессинг для категориальных и количественных признаков выполняется по-разному.

Категориальный признак один – «Угол нашивки, град». Он принимает значения 0 и 90. Модели отработают лучше, если мы превратим эти значения в 0 и 1 с помощью LabelEncoder.

Вещественных количественных признаков у нас большинство. Проблема вещественных признаков в том, что их значения лежат в разных диапазонах, в разных масштабах. Это видно в таблице 2. Необходимо провести одно из двух возможных преобразований:

- нормализацию – приведение в диапазон от 0 до 1 с помощью MinMaxScaler;
- стандартизацию – приведение к матожиданию 0, стандартному отклонению 1 с помощью StandartScaler.

Для данной работы буду использовать нормализацию и MinMaxScaler.

Для обеспечения статистической устойчивости метрик модели используем перекрестную проверку или кросс-валидацию. Чтобы ее реализовать, выборка разбивается необходимое количество раз на тестовую и валидационную. Модель обучается на тестовой выбоке, затем выполняется расчет метрик качества на валидационной. В качестве результата мы получаем средние метрики качества для всех валидационных выборок. Перекрестную проверку реализует функция cross_validate из sklearn.

Поиск гиперпараметров по сетке реализует класс GridSearchCV из sklearn. Он получает модель и набор гиперпараметров, поочередно передает их в модель, выполняет обучение и определяет лучшие комбинации гиперпараметры. Перекрестная проверка уже встроена в этот класс.

Существует множество различных метрик качества, применимых для регрессии. В этой работе я использую:

- R2 или коэффициент детерминации измеряет долю дисперсии, объясненную моделью, в общей дисперсии целевой переменной. Если он близок к единице, то модель хорошо объясняет данные, если же он близок к нулю, то прогнозы сопоставимы по качеству с константным предсказанием;
- RMSE (Root Mean Squared Error) или корень из средней квадратичной ошибки принимает значениях в тех же единицах, что и целевая переменная. Метрика использует возведение в квадрат, поэтому хорошо обнаруживает грубые ошибки, но сильно чувствительна к выбросам;
- MAE (Mean Absolute Error) - средняя абсолютная ошибка так же принимает значениях в тех же единицах, что и целевая переменная;

R2 в норме принимает положительные значения. Эту метрику надо максимизировать. Отрицательные значение коэффициента детерминации означают плохую объясняющую способность модели.

2. Практическая часть

2.1. Предобработка данных

В датасете столбец «Угол нашивки, град» принимает значения 0 и 90. Используем LabelEncoder для кодирования значений в 0 и 1.

Для нормализации будем использовать MinMaxScaler(). На рисунке 7 изображены данные до нормализации, а на рисунке 8 изображены данные после нормализации. Как видно из графиков, данные после нормализации находятся в одном диапазоне. Это позволит свести их вместе в одной модели для машинного обучения и обеспечит корректную работу вычислительных алгоритмов.

Рисунок 7 – графики распределения данных до нормализации

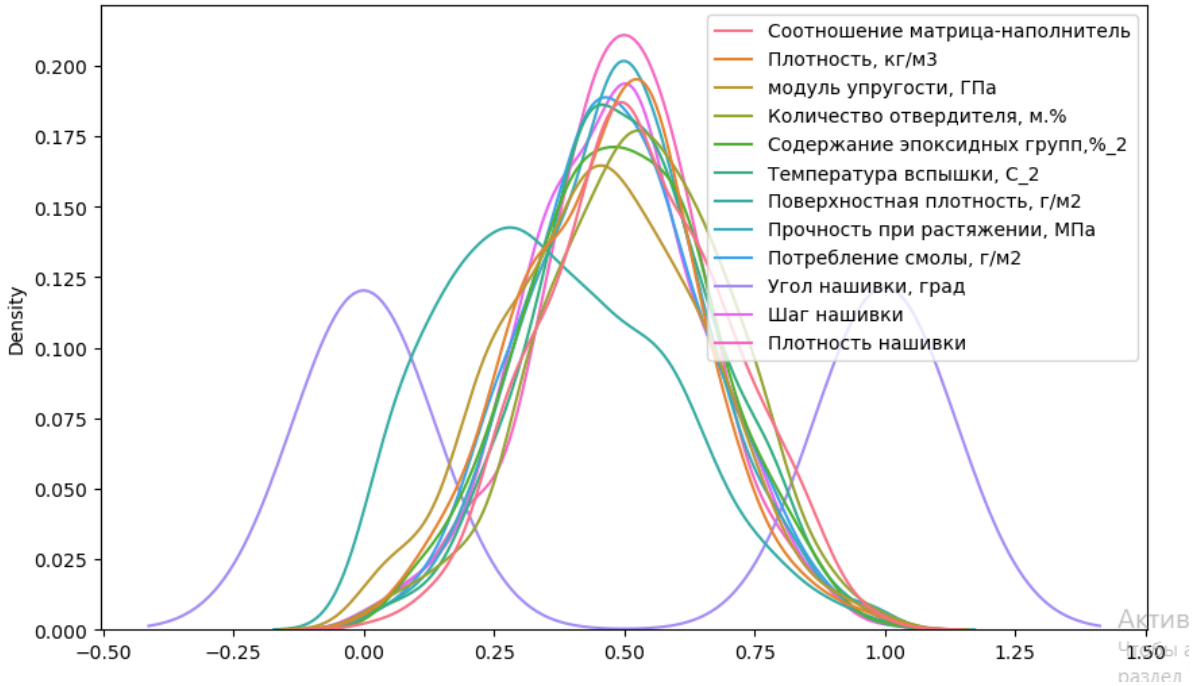

Рисунок 8 – графики распределения данных после нормализации

На рисунке 9 представлена описательная статистика данных до нормализации. На рисунке 10 – данные после нормализации.

	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 9 – описательная статистика данных до нормализации

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	905.0	0.513209	0.178451	0.0	0.392589	0.510464	0.636563	1.0
Плотность, кг/м3	905.0	0.477730	0.173157	0.0	0.352835	0.482835	0.587714	1.0
модуль упругости, ГПа	905.0	0.459297	0.194139	0.0	0.320086	0.457179	0.590467	1.0
Количество отвердителя, м.%	905.0	0.522108	0.171822	0.0	0.402974	0.519494	0.643543	1.0
Содержание эпоксидных групп,%_2	905.0	0.489338	0.175596	0.0	0.370514	0.486678	0.619612	1.0
Температура вспышки, С_2	905.0	0.502994	0.175168	0.0	0.383337	0.502815	0.620667	1.0
Поверхностная плотность, г/м2	905.0	0.367543	0.209374	0.0	0.206505	0.348479	0.526737	1.0
Модуль упругости при растяжении, ГПа	905.0	0.512807	0.176103	0.0	0.395174	0.507795	0.630032	1.0
Прочность при растяжении, МПа	905.0	0.496558	0.171111	0.0	0.378564	0.492681	0.601861	1.0
Потребление смолы, г/м2	905.0	0.495995	0.171316	0.0	0.384295	0.495770	0.611168	1.0
Угол нашивки, град	905.0	0.501657	0.500274	0.0	0.000000	1.000000	1.000000	1.0
Шаг нашивки	905.0	0.487693	0.176315	0.0	0.362216	0.490455	0.606312	1.0
Плотность нашивки	905.0	0.489191	0.166831	0.0	0.385108	0.492545	0.593333	1.0

Рисунок 9 – описательная статистика данных после нормализации

2.2 Разработка и обучение модели

Для обучения модели разделим датасет в соотношении 30 на 70. Т.е. при построении модели 30% данных оставим на тестирование модели, а на остальных 70% происходит обучение моделей. По условиям задания необходимо обучить несколько моделей для прогноза модуля упругости при растяжении и прочности при растяжении. Так же при построении моделей необходимо провести поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10.

Для работы были выбраны следующие модели:

1) Лассо-регрессия (Lasso);
2) Линейная регрессия (LinearRegression);
3) Байесовская регрессия (BayesianRidge);
4) Регрессор дерева решений (DecisionTreeRegressor);
5) Градиентный бустинг регрессии (GradientBoostingRegressor);
6) Случайный лес регрессии (RandomForestRegressor).

В ходе работы для каждой модели подсчитаны коэффициент детерминации (R^2), среднеквадратичная ошибка (RMSE) и средняя абсолютная ошибка (MAE). Так же была посчитана максимальная остаточная ошибка. Она фиксирует худшую ошибку случае между предсказанным значением и истинным значением. В идеально подобранной модели регрессии с одним выходом max_error будет находиться 0 в обучающем наборе, этот показатель показывает степень ошибки, которую имела модель при подборе.

В ходе обучения был использован GridSearchCV из библиотеки sklearn. GridSearchCV – это очень мощный инструмент для автоматического подбирания параметров для моделей машинного обучения. GridSearchCV находит наилучшие параметры, путем обычного перебора: он создает модель для каждой возможной комбинации параметров. Важно помнить, что данный метод является времязатратным.

2.3 Тестирование модели

Рассмотрим работу моделей для параметра – модуль упругости при растяжении. В процессе работы было сделано сравнение моделей с параметрами по умолчанию – рисунок 10.

	R2	RMSE	MAE	max_error
Lasso	-0.016429	-0.174638	-0.141512	-0.418983
LinearRegression	-0.033449	-0.176069	-0.142452	-0.420930
BayesianRidge	-0.018933	-0.174847	-0.141812	-0.419632
DecisionTreeRegressor	-1.246245	-0.258515	-0.206703	-0.690937
GradientBoostingRegressor	-0.154025	-0.186047	-0.150441	-0.476617
RandomForestRegressor	-0.071823	-0.179288	-0.145024	-0.440930

Рисунок 10 – сравнение моделей с параметрами по умолчанию

После использования GridSearchCV – поиск гиперпараметров по сетке, получены следующие результаты – рисунок 11.

	R2	RMSE	MAE	max_error
Lasso	-0.016429	-0.174638	-0.141512	-0.418983
LinearRegression	-0.033449	-0.176069	-0.142452	-0.420930
BayesianRidge	-0.021877	-0.175091	-0.141887	-0.418650
DecisionTreeRegressor	-0.065786	-0.178876	-0.144632	-0.423173
GradientBoostingRegressor	-0.028914	-0.175747	-0.142425	-0.421910
RandomForestRegressor	-0.022156	-0.175171	-0.141818	-0.421669

Рисунок 11 – сравнение моделей с параметрами после поиска гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой

Ни одна из выбранных моделей не оказалась подходящей для данного датасета.

Коэффициент детерминации R2 близок к 0. Значит, они не лучше базовой модели. Поиск гиперпараметров по сетке не дал значительных результатов. Но подбирая гиперпараметры, можно значительно улучшить предсказание выбранной модели.

Все модели крайне плохо описывают исходные данные – не удалось добиться положительного значения R2. Самая лучшая модель лассо, дает коэффициент детерминации близкий к нулю, что соответствует базовой модели.

Другие модели совпадают с базовой моделью. Их характеристики улучшились, но не значительно.

Для прогнозирования модели прочности при растяжении будем рассматривать те же модели, что и для модуля упругости при растяжении. Рассмотрим работу моделей для параметра – модуль прочности при растяжении. На рисунке 12 представлены результаты работы моделей на параметрах по умолчанию, рисунок 13 – результаты работы моделей после поиска гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой.

	R2	RMSE	MAE	max_error
Lasso	-0.014169	-0.171250	-0.136350	-0.455913
LinearRegression	-0.025649	-0.172298	-0.137538	-0.458594
BayesianRidge	-0.019121	-0.171683	-0.136813	-0.457807
DecisionTreeRegressor	-1.033244	-0.240739	-0.194158	-0.634070
GradientBoostingRegressor	-0.074812	-0.176176	-0.140056	-0.477073
RandomForestRegressor	-0.070036	-0.175810	-0.141375	-0.471743

Рисунок 12 – сравнение моделей с параметрами по умолчанию

	R2	RMSE	MAE	max_error
Lasso	-0.016886	-0.171565	-0.136861	-0.459283
LinearRegression	-0.025649	-0.172298	-0.137538	-0.458594
BayesianRidge	-0.017493	-0.171592	-0.136922	-0.457358
DecisionTreeRegressor	-0.037498	-0.173159	-0.138561	-0.459825
GradientBoostingRegressor	-0.028381	-0.172446	-0.137954	-0.447979
RandomForestRegressor	-0.011780	-0.171130	-0.136244	-0.449757

Рисунок 13 – сравнение моделей с параметрами после поиска гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой

Все модели крайне плохо описывают исходные данные – не удалось добиться положительного значения R2. Самая лучшая – случайный лест, дает коэффициент детерминации близкий к нулю, что соответствует базовой модели.

Другие модели совпадают с базовой моделью. Их характеристики улучшились, но не значительно.

2.4 Разработка нейронной сети для прогнозирования соотношения матрица-наполнитель

Для разработки нейронной сети для прогнозирования соотношения матрица-наполнитель была использована последовательная модель – Sequential из библиотеки Keras. Модель Sequential представляет собой линейный стек слоев.

Опишем параметры нейронной сети для заданной задачи:

1) Последовательная модель (Sequential) нейронной сети.
2) Модель состоит из трёх скрытых слоев (Dense), с количеством нейронов, в которых равно 8, и выходного слоя с одним нейроном.
3) Функция активации слоев выбран гиперболический тангенс (tanh), а для выходного слоя – сигмодида (sigmoid).
4) Количество эпох обучения равно 100.
5) Оптимизатор – Adam.
6) Loss-функция – среднеквадратическая ошибка.
7) Метрика – Mean Absolute Error.
8) Для валидации будет использовано 30% обучающих данных.

```
model = Sequential()
# Создаем слои
model.add(Dense(8, activation='tanh'))
model.add(Dense(8, activation='tanh'))
model.add(Dense(1, activation = 'sigmoid')) # выходной слой
model.compile(loss = 'MSE', optimizer = 'adam', metrics = ['MAE'])

history = model.fit(X_smn_train,
                    y_smn_train,
                    batch_size=250,
                    epochs=100,
                    validation_data=(X_smn_test,  y_smn_test),
                    verbose = 1)
model.summary()
```

Рисунок 14 – нейронная сеть для прогнозирования соотношения матрица-наполнитель

Отобразим структуру нейронной сети с помощью метода summery(). Результаты представлены на рисунке 15.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 8)                 104

 dense_1 (Dense)             (None, 8)                 72

 dense_2 (Dense)             (None, 1)                 9

=================================================================
Total params: 185
Trainable params: 185
Non-trainable params: 0
_________________________________________________________________
```

Рисунок 15 – структура нейронной сети методом summery()

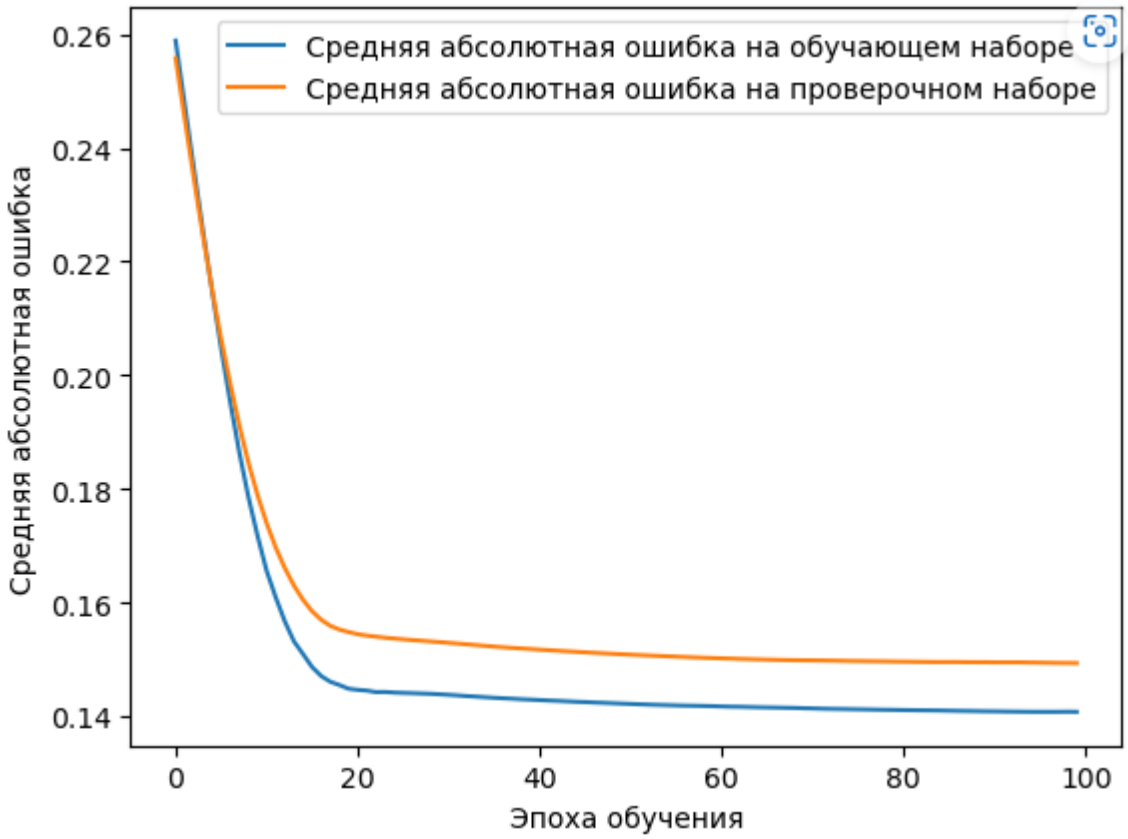

Рисунок 16 – график обучения нейросети

Из графика видно, что обучение нейросети прошло успешно. Количество слоев и нейронов подобрано в оптимальном варианте.

2.5. Разработка приложения

Приложение для прогноза соотношения матрица-наполнитель написано на языке программирования Python с использованием библиотеки Flask.

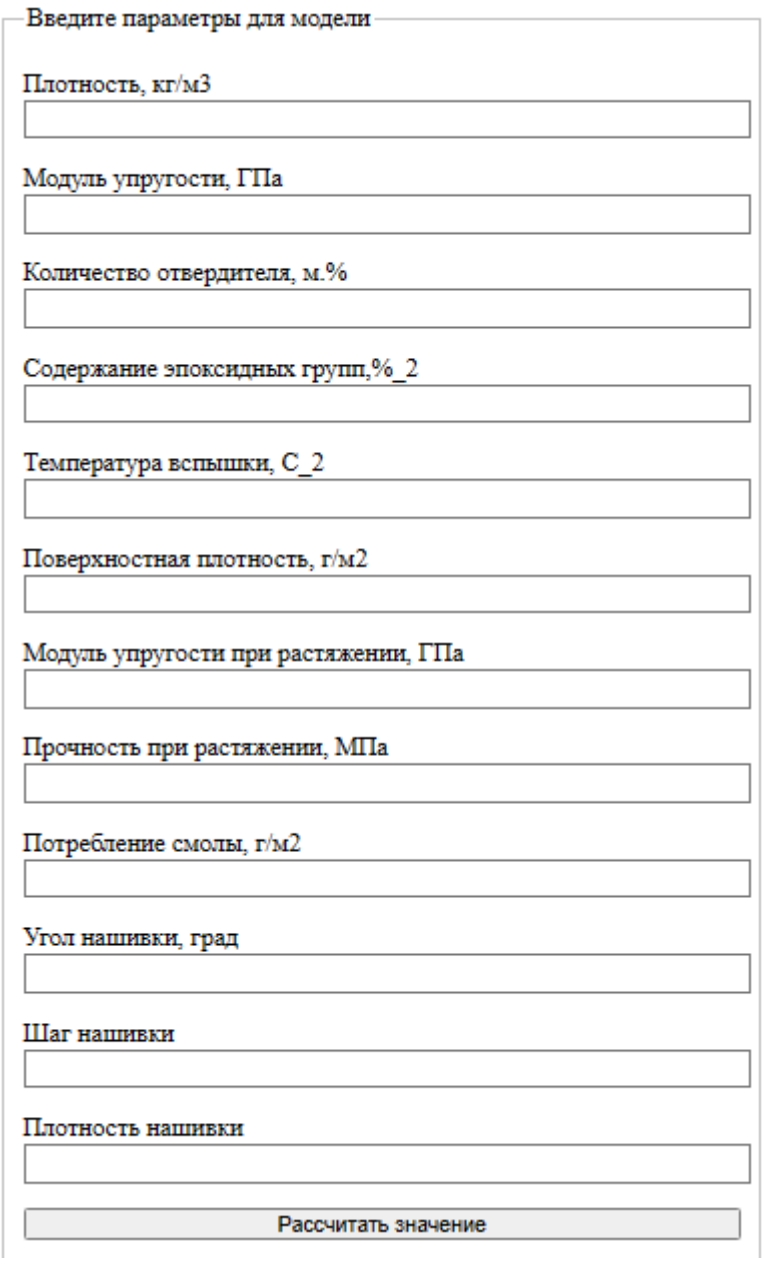

Рисунок 17 – форма ввода данных для прогноза соотношения матрица-наполнитель

На рисунке 17 представлена форма ввода данных для прогноза соотношения матрица наполнитель.

Алгоритм запуска приложения

1) Запустить приложение app.py.
2) Перейти по ссылке http://127.0.0.1:5000/.
3) Ввести параметры.
4) Нажать на кнопку «Рассчитать значение».

2.6. Создание удаленного репозитория

Для данного исследования был создан удаленный репозиторий на GitHub, который находится по адресу https://github.com/ValGeras/vkr. В данный репозиторий выложены все материалы по выпускной квалификационной работе.

Заключение

Композиционные материалы широко применяются во всех областях человеческой деятельности. Благодаря своим уникальным свойствам, композиты имеют большое преимущество перед традиционными металлами и сплавами. Для получения композиционных материалов проводится множество исследований и испытаний. Для минимизации затрат, целесообразно некоторые виды анализа проводить с помощью машинного обучения.

В ходе выполнения работы был проведен разведочный анализ предложенных данных. Нарисованы гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек. Для каждой колонки получено среднее, медианное значение, проведен анализ и исключение выбросов. Датасет проверен на наличие пропусков.

Была проведена предобработка данных – частичное удаление выбросов с помощью метода изолирующий лес и метода 3 сигм. Для нормализации был применен метод MinMaxScaler, подсчитаны корреляции между параметрами.

Так же обучено несколько моделей для прогноза модуля упругости при растяжении и прочности при растяжении. При построении модели 30% данных было оставлено на тестирование модели, на остальных происходило обучение моделей. При построении моделей проведен поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10. Использованные при разработке моделей подходы не позволили получить сколько-нибудь достоверных прогнозов. Применённые модели регрессии не показали высокой эффективности в прогнозировании свойств композитов. Лучшие метрики для модуля упругости при растяжении, ГПа – метод лассо-регрессия, для прочности при растяжении, МПа – случайный лес.

Написана нейронная сеть, которая рекомендует соотношение матрица-наполнитель. В процессе работы было выявлено, что предоставленных данных не хватает для построения оптимальной модели.

Разработано приложение с графическим интерфейсом, которое выдает прогноз соотношение матрица-наполнитель.

Таким образом, на сегодняшний день нейронные сети являются самым современным подходам к решению такого рода задач. Они способны находить скрытые и нелинейные зависимости в данных. Но выбор оптимальной архитектуры нейронной сети является довольно непростой задачей.

Библиографический список

1. А.В. Груздев Предварительная подготовка данных в Python: Том 1. Инструменты и валидация. – М: ДМК Пресс, 2023 – 816 с.
2. А.В. Груздев Предварительная подготовка данных в Python: Том 2. План, примеры и метрики качества. – М: ДМК Пресс, 2023 – 814 с.
3. Аллен Б. Дауни – Основы Python. Научитесь думать как программист / Аллен Б. Дауни.; пер. с англ. С. Черникова ; [науч. ред. А. Родионов]. – Москва : Манн, Иванов и Фербер, 2021. – 304 с.
4. Андерсон, Карл Аналитическая культура. От сбора данных до бизнес-результатов / Карл Андерсон ; пер. с англ. Юлии Константиновой ; [науч. ред. Руслан Салахиев]. – М. : Манн, Иванов и Фербер, 2017. – 336 с
5. Антонио Джулли, Суджит Пал 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.
6. В.К. Шитиков, С.Э. Мастицкий. Классификация, регрессия и другие алгоритмы Data Mining с использованием R. 351 с. − Электронная книга, адрес доступа: https://github.com/ranalytics/data-mining
7. В.Ш. Берикашвили Статистическая обработка данных, планирование эксперимента и случайные процессы: учебное пособие для вузов / В.Ш. Берикашвили, С.П. Оськин. – 2-е изд. испр. и доп. – Москва: Издательство Юрайт, 2022 – 164 с.
8. Грас Д. Data Science. Наука о данных с нуля: Пер. с англ. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2021. – 416 с.
9. Грас, Джоэл. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2021. - 416 с.: ил
10. Д. Рутковская, М. Пилиньский, Л. Рутковский Нейронные сети, генетические алгоритмы и нечеткие системы. – М.: Горячая Линия – Телеком. – 2013. – 384 с.

11. Д. Фостер Генеративное глубокое обучение. Творческий потенциал нейронных сетей. – СПб.: Питер. – 2020. – 336 с.

12. Дэвид Шпигельхалтер Искусство статистики. Как находить ответы в данных.: перевод на рус. яз. – ООО «Манн, Иванов и Фербер», 2021. – 645 с.

13. Жерон, Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем, 2-е изд.: Пер. с англ. – СПб.: ООО «Диалектика»: 2020. – 1040 с.

14. Любанович Билл Простой Python. Современный стиль программирования. 2-е изд. – СПб.: Питер, 2021 – 592с.

15. Митчелл, Райан. Современный скрапинг веб-сайтов с помощью Python. 2-е межд. издание . – СПб.: Питер, 2021.

16. П. Брюс. Практическая статистика для специалистов Data Science: пер с англ. / П. Брюс, Э. Брюс. – СПб.: БХВ-Петербург, 2018. – 304 с.

17. Рашид, Тарик. Создаем нейронную сеть.: пер. с англ. – СПб.: ООО «Альфа-книга», 2017 – 272с.

18. С. Николенко, А. Кадурин, Е. Архангельская, Глубокое обучение. Погружение в мир нейронных сетей. – СПб.: Питер. – 2020. – 480 с.

19. С. Рассел, П. Норвиг. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2007. – 1408 с.

20. С.В. Буканов Разработка приложений с графическим пользовательским интерфейсом на языке Python: учебное пособие / С.В. Буканов, О.В. Буканова. – Санкт-Петербург: Лань, 2023 – 88 с. – Текст: непосредственный

21. Сара Бослаф. Статистика для всех. / Пер. с англ. П.А. Волкова, И.М. Либерман, А.А. Галицына. – М.: ДМК Пресс, 2015 – 586 с.

22. Седжвик, Роберт, Уэйн, Кевин, Дондеро, Роберт. Программирование на языке Python: учебный курс.: пер. с англ. – СПб.: ООО «Альфа-книга», 2017. – 736 с.

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

24. Ян Эрим Солек. Программирование компьютерного зрения на языке Python / пер. с англ. Слинкн А. А. – М.: ДМК Пресс, 2016 – 312 с.

25. Matplotlib: Научная графика в Python: – Режим доступа: https://pythonworld.ru/novosti-mira-python/scientific-graphics-in-python.html (дата обращения: 20.04.2023).

26. Википедия: – Режим доступа: https://ru.wikipedia.org/wiki (дата обращения: 20.04.2023).

27. Документация по библиотеке numpy: – Режим доступа: https://numpy.org/doc/1.22/user/index.html#user. (дата обращения: 03.03.2023).

28. Документация по библиотеке pandas: – Режим доступа: https://pandas.pydata.org/docs/user_guide/index.html#user-guide. (дата обращения: 04.04.2023).

29. Документация по библиотеке scikit-learn: – Режим доступа: https://scikit-learn.ru/. (дата обращения: 05.04.2023).

30. Документация по библиотеке seaborn: – Режим доступа: https://seaborn.pydata.org/tutorial.html. (дата обращения: 08.04.2023).

31. Документация по библиотеке Tensorflow: – Режим доступа: https://www.tensorflow.org/overview?hl=ru. (дата обращения: 16.04.2023).