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 Орлов Д.О.

Москва, 2023

Оглавление

1. Введение

1.1. Определение понятия композиционных материалов

В источниках существует множество определений понятия композиционные материалы (композиты). Например:

1.1.1. «..Композиционные материалы (композиты) – многокомпонентные материалы, состоящие из,…, основы (матрицы), армированной наполнителем…. (1)

1.1.2. Композиционный матери́ал или композитный матери́ал (КМ), сокращённо компози́т — 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. В составе композита принято выделять матрицу/матрицы и наполнитель/наполнители…. (2)

1.1.3. Композиты – это материалы, состоящие из двух или более компонентов (армирующих элементов и скрепляющей их матрицы) и обладающие свойствами, отличными от суммарных свойств компонентов. При этом предполагается, что компоненты, входящие в состав композита, должны быть хорошо совместимыми и не растворяться или иным способом поглощать друг друга. (3)

1.1.4. Из анализа приведенных и множества иных подобных определений следует, что наиболее общими классифицирующими признаками композитов являются: 1). Наличие двух или более несмешиваемых фаз; 2). Наличие у композита свойств, отличных от свойств компонентов

Из изложенного следует, что в широком смысле композиционный материал – это любой материал с гетерогенной структурой, т. е. со структурой, состоящей минимум из двух фаз.

1.2 Немного истории

Композиты использовались человеком с незапамятных времен. Даже в самые первые, кирпичи и гончарные изделия, появившиеся за 5000 лет до н. э., древние часто добавляли измельченные камни или материалы органического происхождения, чтобы уменьшить усадку и растрескивание при сушке/обжиге. Из известнейших исторических примеров использования композитов можно привести композитные луки, появившиеся в Китае ок. 1000 лет до н.э. и широко известную булатную сталь. Так же стоит упомянуть первые бетоны, рецепты которых хитрые древние римляне подглядели в процессах взаимодействия вулканических выбросов с морской водой.

Таким образом, начало технологии композиционных материалов, уходит глубоко в историю, а комбинирование различных материалов остается наиболее важным путем создания новых материалов в настоящее время.

1.3. Вопросы классификации

1.3.1. Представляется, что систему классификации композитов следует начать с разделения на природные и искусственные.

1.3.2. По характеристикам матрицы:

1.3.2.1. Однокомпонентные и многокомпонентные матрицы

1.3.2.2. По состоянию фазы жидкие, гелеобразные и твердые матрицы

1.3.2.3. По химсоставу неорганические, органические, металлические и смешанные матрицы

1.3.2.4. По химическим, физическим и физико-химическим свойствам (реакционноспособность, термоциклические нагрузки, температура разложения и др.)

1.3.3. По характеристикам наполнителя:

1.3.3.1. Однокомпонентные и многокомпонентные матрицы

1.3.3.2. По состоянию фазы жидкие, гелеобразные и твердые наполнители

1.3.3.3. По химсоставу неорганические, органические, металлические и смешанные наполнители

1.3.3.4. По химическим, физическим и физико-химическим свойствам (реакционноспособность, термоциклические нагрузки, температура разложения и др.)

1.3.3.5. Геометрическая форма наполнителя (порошки (гранулы), пластины, плоскостные образования (графен), волокна разной структуры (углеволокно,нитевидные кристаллы), сферойды (фуллерены) , трубки, ткани и др.)

1.3.4. По вектору проявления свойств:

1.3.4.1. Анизотропные. Свойства анизотропных материалов зависят от направления в исследуемом объекте.

1.3.4.2. Изотропные. Изотропными называют материалы, которые имеют одинаковые свойства во всех направлениях.

1.3.4.3. Квазиизотропные. Термин «квазиизотропный» означает, что композит является анизотропным в микрообъеме, но изотропным в объеме всего изделия.

1.3.5. По структуре и расположению компонентов (внутренней архитектуре) композита В соответствии с этой классификацией композиты можно разделить на:

1.3.5.1. Композиты со стохастическим распределением наполнителя в матрице

1.3.5.2. Композиты с пространственно-организованной внутренней структурой (волокнистые, слоистые, каркасные, смешанные и др.)

1.3.5.3. Композиты со смешанной архитектурой внутренней структуры. (часть внутренней структуры организована стохастически, а часть упорядочена)

1.3.6 По химическим, физическим и физико-химическим свойствам композита

Приведенная здесь классификация основана (по мнению автора) на наиболее общих особенностях композитов и не носит закрывающий характер. Безусловно она может быть дополнена, например, классификацией по методам изготовления, по способам использования и еще многими другими. Более того, с развитием науки применимой к композитам, она обязательно будет расширятся и дополнятся. Однако, представляется, что данные расширения и дополнения, так или иначе, в основном будут имманентны приведенным пунктам классификации.

1.4. Применение и перспективы композитов

Благодаря различным уникальным свойствам композиты находят широчайшее применение в промышленности и обычной жизни. Композиты применяются в областях от строительства до производства летательных аппаратов, от электроники до медицины, от производства одежды и обуви до продуктов питания, от топлива до автомобилестроения. Пожалуй, трудно найти сторону жизни, где композиты не используются тем или иным образом.

Важным аспектом здесь является возможность изготовления композитов с заранее заданными свойствами. Собственно предмет настоящей работы является маленьким штришком к данному вопросу. Следует отметить, что применяемые сегодня композиты, как правило, монофункциональны, т.е. используется только одно проявляемое ими полезное свойство.

Представляется, что в перспективе получат интенсивное развитие разработки композитов, проявляющих несколько поленых свойств (мультифункциональность), и композитов с развитой внутренней архитектурой, как основой мультифункциональности. В данной области перед нами предстает просто непаханное поле! Тем более, что зачатки технологии таких композитов усматриваются уже сегодня.

2. Постановка задачи.

2.1. Смысловое описание решаемой задачи анализа данных.

Как указывалось в введении разработка новых композиционных материалов чрезвычайно актуально. Причем, поскольку свойства итогового композиционного материала могут существенно отличатся от свойств исходных, для получения новых материалов с заданными свойствами, необходимо каким-то образом оценить свойства продукта.

Для решения этой проблемы есть два пути: эксперименты в натуре, или моделирование (прогнозирование) характеристик. Суть моделирования (прогнозирования), в рассматриваемом аспекте заключается в симуляции требуемого свойства продукта, на основе данных о характеристиках входящих.

Совершенно очевидно, что компьютерное моделирование и прогнозирование осуществляется быстрее чем серия экспериментов в натуре, и, кроме того, существенно дешевле.

На входе в задании имеются данные о свойствах неизвестных компонентов неизвестных композиционных материалов, неизвестно как связанных с ними. На выходе необходимо спрогнозировать ряд конечных свойств неизвестных композиционных материалов.

Кроме того, в задании указывается, что кейс основан на реальных производственных задачах Центра НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана).

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

Задание по вкр дано в следующей редакции:

1. Датасет со свойствами композитов. Объединение делать по индексу тип объединения INNER
https://drive.google.com/file/d/1B1s5gBlvgU81H9GGolLQVw_SOi-vyNf2/view?usp=sharing
2. Обучить алгоритм машинного обучения, который будет определять значения:
- Модуль упругости при растяжении, ГПа
- Прочность при растяжении, МПа
3. Написать нейронную сеть, которая будет рекомендовать:
- Соотношение матрица-наполнитель
4. Написать приложение, которое будет выдавать прогноз полученный в задании 2 или 3 (один или два прогноза, на выбор учащегося)
5. Создать профиль на github.com
6. Сделать commit приложения на github.com
7. Сделать commit на веб-хостинг (По желанию учащегося)

8. Написать пояснительную записку к проекту, которая включает блок-схему и описание процесса подготовки, обучения моделей и инструкцию по установке и запуску приложения.

2.2 Характеристика датасета

2.2.1 Описание переменных в датасете

2.2.1.1 Соотношение матрица – наполнитель – по смыслу показывает на содержание матрицы по отношению к наполнителю. При этом:

- не указаны названия матрицы и наполнителя, (это обстоятельство не позволяет точно определить: это соотношение относится к конечному продукту, к части начального продукта (в виде порошка и эпоксидной смолы), к какому-то промежуточному продукту, к конечному продукту.
- не указана формула вычисления соотношения (хотя можно представить себе минимум 2 способа, которые дадут разный результат).
- не указаны размерности (массовая, объемная, молярная) в которых исчислялось соотношение

Таким образом, по мнению автора, данная переменная не является достаточно репрезентативной и согласованной.

По смыслу задания, данная переменная является исходной

2.2.1.2 Плотность, кг/м3 – по смыслу показывает на содержание массы чего-либо в одном кубическом метре.

- в нашем датасете не указано к какому объекту (исходному, промежуточному или конечному) относится данная характеристика

Таким образом, по мнению автора, данная переменная не является достаточно репрезентативной и согласованной.

По заданию, данная переменная является исходной

2.2.1.3. Модуль упругости, Гпа, по смыслу характеризует прочностные характеристики твердого тела.

- в нашем датасете не указано к какому объекту (исходному, промежуточному или конечному) относится данная характеристика

Таким образом, по мнению автора, данная переменная не является достаточно репрезентативной и согласованной.

По заданию, данная переменная является исходной

2.2.1.4. Количество отвердителя, м.% К.о должно указывать на количество отвердителя, однако не указано, в каких единицах дано данное количество. Кроме того, не указано, в каком объекте присутствует данное количество, в исходной смоле, в смеси с наполнителем, в пропитанной ткани и т.д.

Таким образом, по мнению автора, данная переменная не является достаточно репрезентативной и согласованной.

По заданию, данная переменная является исходной

2.2.1.5. Содержание эпоксидных групп, %_2. СЭГ должно указывать на количество эпоксидных групп, однако не указано, в каких единицах дано данное количество. Кроме того, не указано, в каком объекте присутствует данное количество, в исходной смоле, в смеси с наполнителем, в пропитанной ткани и т.д.

Таким образом, по мнению автора, данная переменная не является достаточно репрезентативной и согласованной.

По заданию, данная переменная является исходной

2.2.1.6. Температура вспышки, С_2 – по определению наименьшая температура летучего конденсированного вещества, при которой пары над поверхностью вещества способны вспыхивать в воздухе под воздействием источника зажигания, однако устойчивое горение после удаления источника зажигания не возникает. Под вспышкой здесь понимается —

быстрое сгорание смеси паров летучего вещества с воздухом, сопровождающееся кратковременным видимым свечением.

Не указывается на каком этапе технологического процесса осуществляется съем данной переменной. В связи с чем, по мнению автора, данная переменная не является достаточно репрезентативной и согласованной.

По заданию, данная переменная является исходной

2.2.1.7. Поверхностная плотность, г/м2, - параметр, применяемый для характеристики тканевых материалов, однако не указано имеется в виду П.П. до или после пропитки ткани смолой, до или после отверждения, до или после измерения температуры вспышки и т.д. В связи с чем, по мнению автора, данная переменная не является достаточно репрезентативной и согласованной.

По заданию, данная переменная является исходной

2.2.1.8. Модуль упругости при растяжении, Гпа – параметр, описывающий прочностные характеристики какого-либо объекта. Однако не указано, имеется в виду М.У. до или после пропитки ткани смолой, до или после отверждения, до или после измерения температуры вспышки, по окончании технологического процесса или нет, и т.д. Не согласована размерность. В связи с чем, по мнению автора, данная переменная не является достаточно репрезентативной и согласованной.

По указанию постановщика задачи в задании, данная переменная является конечной.

2.2.1.9. Прочность при растяжении, Мпа параметр, описывающий прочностные характеристики какого-либо объекта. Однако не указано, имеется в виду П. Р. до или после пропитки ткани смолой, до или после

отверждения, до или после измерения температуры вспышки, по окончании технологического процесса или нет, и т.д. Не согласована размерность. В связи с чем, по мнению автора, данная переменная не является достаточно репрезентативной и согласованной.

По указанию постановщика задачи в задании, данная переменная является конечной.

2.2.1.10. Потребление смолы, г/м2 данный параметр характеризует расход некой смолы, отнесенный к площади. Однако не указано, имеется в виду П. с. до или после пропитки ткани смолой, по отношению к площади финального продукта или какого – либо промежуточного, по окончании технологического процесса или нет, и т.д. Не согласована размерность. В связи с чем, по мнению автора, данная переменная не является достаточно репрезентативной и согласованной.

По заданию, данная переменная является исходной

2.2.1.11. Угол нашивки, град является технологическими приемами для формирования готовых композитных изделий из пропитанных тканей.

По заданию, данная переменная является исходной

2.2.1.12. Шаг нашивки является технологическими приемами для формирования готовых композитных изделий из пропитанных тканей.

По заданию, данная переменная является исходной

2.2.1.13. Плотность нашивки является технологическими приемами для формирования готовых композитных изделий из пропитанных тканей.

По заданию, данная переменная является исходной

Таким образом, из анализа физической и химической природы данных предоставленных в датасете, «как есть», видно, что большинство переменных являются не вполне репрезентативными и согласованными. Что очевидно будет влиять на результат выполнения задания.

2.2.2 Характеристика датасета как объекта бигдата

Датасет состоит из двух файлов: X_bp и X_nup.

Файл X_bp содержит:

– признаков: 10 и индекс;

– строк: 1023.

Файл X_nup содержит:

– признаков: 3 и индекс;

– строк: 1040.

По заданию файлы требуют объединения по типу INNER по индексу.

При объединении часть строк из файла X_nup была отброшена.

Далее исследуем объединенный датасет, содержащий 13 признаков и 1023 строк или объектов.

Признаки объединенного датасета иллюстрируются табл. 1.

Табл. 1 Признаки датасета

```
 #   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
```

Двенадцать переменных содержат значения типа float64. и одна int64.

Качественные характеристики отсутствуют. Пропусков не имеется. Ни одна из записей не является null. Очистка не требуется.

Дополнительно проверим на пропуски. Датасет не имеет пропусков (см. табл. 2)

Табл. 2 Поиск пропусков

```
Соотношение матрица-наполнитель         0
Плотность, кг/м3                        0
модуль упругости, ГПа                   0
Количество отвердителя, м.%             0
Содержание эпоксидных групп,%_2         0
Температура вспышки, С_2                0
Поверхностная плотность, г/м2           0
Модуль упругости при растяжении, ГПа    0
Прочность при растяжении, МПа           0
Потребление смолы, г/м2                 0
Угол нашивки, град                      0
Шаг нашивки                             0
Плотность нашивки                       0
dtype: int64
```

Проверим датасет на дубликаты - дубликатов нет. (Рис. 1)

```
# Проверим датасет на дубликаты
rrr_dataset.duplicated().sum()
```

```
0
```

Рис 1. Поиск дубликатов

В принципе датасет готов к разведочному анализу и удалению аномалий и выбросов.

В задании целевыми переменными указаны: модуль упругости при растяжении, Гпа; прочность при растяжении, МПа; соотношение матрица-наполнитель.

2.3. Описание используемых методов

Регрессия в теории вероятностей и математической статистике — односторонняя стохастическая зависимость, устанавливающая соответствие между случайными переменными, то есть математическое выражение, отражающее связь между зависимой переменной y и независимыми переменными x при условии, что это выражение будет иметь статистическую значимость. В отличие от чисто функциональной зависимости y=f(x), когда каждому значению независимой переменной x соответствует одно определённое значение величины y, при регрессионной связи одному и тому же значению x могут соответствовать в зависимости от случая различные значения величины y. Соответственно задача регрессии в машинном обучении - это задача предсказания какой-то численной характеристики объекта предметной области по определенному набору его параметров (атрибутов). Такие независимые переменные называют так же предикторами или регрессорами. На текущий момент существует множество методов регрессионного анализа, для решения задач. Ниже опишем некоторые из них в соответствии с заданием.

2.3.1. Линейная регрессия

Линейная регрессия — используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной y от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) x с линейной функцией зависимости. В частном случае, когда фактор единственный (без учёта константы), говорят о парной или простейшей линейной регрессии y=ax+b. Когда количество факторов (без учёта константы) больше одного, то говорят о множественной регрессии Y=b0+b1*x1+b2*x2+⋯+bn*xn, где n - число входных переменных.

Линейная регрессия — первый тщательно изученный метод регрессионного анализа. Его главное достоинство — простота. Простота является и главным недостатком этого метода. Тем не менее, именно с линейной регрессии целесообразно начать подбор подходящей модели (от простого, к сложному).

На языке python линейная регрессия реализована в sklearn.linear_model.LinearRegression.

2.3.2. Лассо (LASSO) и гребневая (Ridge) регрессия

Гребневая(ридж) модель регрессии и модель регрессии Лассо – это регуляризованные линейные модели, хороший способ уменьшить переобучение и упорядочить модель: чем меньше у нее степеней свободы, тем сложнее будет переобучить данные. Простой способ регуляризации полиномиальной модели – уменьшить количество степеней полинома.

LASSO использует сжатие коэффициентов (shrinkage) и этим пытается уменьшить сложность данных, искривляя пространство, на котором они лежат. В этом процессе лассо автоматически помогает устранить или исказить сильно коррелированные и избыточные функции в методе с низкой дисперсией.

Регрессия лассо использует регуляризацию L1, то есть взвешивает ошибки по их абсолютному значению.

Гребневая регрессия или ридж-регрессия — так же вариация линейной регрессии, очень похожая на регрессию LASSO. Она так же применяет сжатие и хорошо работает для данных, которые демонстрируют сильную мультиколлинеарность.

Самое большое различие между ними в том, что гребневая регрессия использует регуляризацию L2, которая взвешивает ошибки по их квадрату, чтобы сильнее наказывать за более значительные ошибки.

Регуляризация позволяет интерпретировать модели. Если коэффициент стал 0 (для Lasso) или близким к 0 (для Ridge), значит данный входной признак не является значимым.

Эти методы реализованы в sklearn.linear_model.Lasso и sklearn.linear_model.Ridge.

2.3.3. Метод опорных векторов для регрессии

Метод опорных векторов — набор схожих алгоритмов обучения с учителем, использующихся для задач классификации и регрессионного анализа.

Чаще всего он применяется в постановке бинарной классификации.

Основная идея заключается в построении гиперплоскости, разделяющей объекты выборки оптимальным способом. Интуитивно, хорошее разделение достигается за счет гиперплоскости, которая имеет самое большое расстояние до ближайшей точки обучающей выборке любого класса. Максимально близкие объекты разных классов определяют опорные вектора.

Если в исходном пространстве объекты линейно неразделимы, то выполняется переход в пространство большей размерности.

Решается задача оптимизации.

Для вычислений используется ядерная функция, получающая на вход два вектора и возвращающая меру сходства между ними:

– линейная;

– полиномиальная;

– гауссовская (rbf).

Эффективность метода опорных векторов зависит от выбора ядра, параметров ядра и параметра C для регуляризации.

Преимущество метода — его хорошая изученность.

Недостатки:

– чувствительность к выбросам;

– отсутствие интерпретируемости.

Вариация метода для регрессии называется SVR (Support Vector Regression).

В python реализацию SVR можно найти в sklearn.svm.SVR.

2.3.4. Метод k-ближайших соседей

Еще один метод классификации, который адаптирован для регрессии - метод k-ближайших соседей (k Nearest Neighbors). На интуитивном уровне суть метода проста: посмотри на соседей вокруг, какие из них преобладают, таковым ты и являешься.

В случае использования метода для регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны.

Для реализации метода необходима метрика расстояния между объектами. Используется, например, эвклидово расстояние для количественных признаков или расстояние Хэмминга для категориальных.

Этот метод — пример непараметрической регрессии.

Он реализован в sklearn.neighbors.KneighborsRegressor.

2.3.5. Деревья решений

Деревья решений (Decision Trees) - еще один непараметрический метод, применяемый и для классификации, и для регрессии. Деревья решений используются в самых разных областях человеческой

деятельности и представляют собой иерархические древовидные структуры, состоящие из правил вида «Если ..., то ...».

Решающие правила автоматически генерируются в процессе обучения на обучающем множестве путем обобщения обучающих примеров. Поэтому их называют индуктивными правилами, а сам процесс обучения — индукцией деревьев решений.

Дерево состоит из элементов двух типов: узлов (node) и листьев (leaf).

В узлах находятся решающие правила и производится проверка соответствия примеров этому правилу. В результате проверки множество примеров, попавших в узел, разбивается на два подмножества: удовлетворяющие правилу и не удовлетворяющие ему. Затем к каждому подмножеству вновь применяется правило и процедура рекурсивно повторяется пока не будет достигнуто некоторое условие остановки алгоритма. В последнем узле проверка и разбиение не производится, и он объявляется листом.

В листе содержится не правило, а подмножество объектов, удовлетворяющих всем правилам ветви, которая заканчивается данным листом. Для классификации — это класс, ассоциируемый с узлом, а для регрессии — соответствующий листу интервал целевой переменной.

При формировании правила для разбиения в очередном узле дерева необходимо выбрать атрибут, по которому это будет сделано. Общее правило для классификации можно сформулировать так: выбранный атрибут должен разбить множество наблюдений в узле так, чтобы результирующие подмножества содержали примеры с одинаковыми метками класса, а количество объектов из других классов в каждом из этих множеств было как можно меньше. Для этого были выбраны различные критерии, например, теоретико-информационный и статистический.

Для регрессии критерием является дисперсия вокруг среднего. Минимизируя дисперсию вокруг среднего, мы ищем признаки, разбивающие выборку таким образом, что значения целевого признака в каждом листе примерно равны.

Огромное преимущество деревьев решений в том, что они легко интерпретируемы, понятны человеку. Они могут использоваться для извлечения правил на естественном языке. Еще преимущества — высокая точность работы, нетребовательность к подготовке 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.

Деревья решений реализованы в sklearn.tree.DecisionTreeRegressor.

2.3.6. Случайный лес

Случайный лес (RandomForest) — представитель ансамблевых методов.

Если точность дерева решений оказалось недостаточной, мы можем множество моделей собрать в коллектив. Формула итогового решателя (3) — это усреднение предсказаний отдельных деревьев.

$$a(x) = \frac{1}{N}\sum_{i=1}^{N} b_i(x),$$

где

N – количество деревьев;

i – счетчик для деревьев;

b – решающее дерево;

x – сгенерированная нами на основе данных выборка.

Для определения входных данных каждому дереву используется метод случайных подпространств. Базовые алгоритмы обучаются на различных подмножествах признаков, которые выделяются случайным образом.

Преимущества 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.

Метод реализован в sklearn.ensemble.RandomForestRegressor.

2.3.7. Градиентный бустинг

Градиентный бустинг (GradientBoosting) — еще один представитель ансамблевых методов.

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. Используя градиентный спуск и обновляя

предсказания, основанные на скорости обучения (learning rate), ищем значения, на которых loss минимальна.

Бустинг, использующий деревья решений в качестве базовых алгоритмов, называется градиентным бустингом над решающими деревьями. Он отлично работает на выборках с «табличными», неоднородными данными и способен эффективно находить нелинейные зависимости в данных различной природы. На настоящий момент это один из самых эффективных алгоритмов машинного обучения. Благодаря этому он широко применяется во многих конкурсах и промышленных задачах. Он проигрывает только нейросетям на однородных данных (изображения, звук и т. д.).

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

Метод градиентного бустинга реализован в библиотеке sklearn — sklearn.ensemble.GradientBoostingRegressor. Существуют и другие реализации, в т.ч. более мощные, например, XDGBoost.

2.3.8. Нейронная сеть

Нейронная сеть — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. Вычислительная единица нейронной сети — нейрон или персептрон.

НС представляет собой систему соединённых и взаимодействующих между собой простых виртуальных процессоров (искусственных нейронов). Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он

периодически посылает другим процессорам. Будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие по отдельности простые процессоры вместе способны решать довольно сложные задачи.

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 и др.

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 определяет 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.

НС сегодня это самый мощный, гибкий и широко применяемый инструмент в машинном обучении, используется для решения задач регрессии, классификации, распознавания образов и речи, компьютерного зрения и др.

2.4. Разведочный анализ данных

Разведочный анализ данных (РАД)— анализ основных свойств данных, нахождение в них общих закономерностей, распределений и аномалий, построение начальных моделей.

Понятие РАД впервые ввел математик Джон Тьюки, который сформулировал цели РАД следующим образом:

максимальное «проникновение» в данные,

выявление основных структур,

выбор наиболее важных переменных,

обнаружение отклонений и аномалий,

проверка основных гипотез,

разработка начальных моделей. (4)

В РАД важное значение занимают инструменты визуализации. Основатель науки о РАД Дж. Тьюки отмечал: «График имеет наибольшую ценность тогда, когда он вынуждает нас заметить то, что мы совсем не ожидали увидеть.» (5)

Основные средства разведочного анализа — изучение вероятностных распределений переменных, построение и анализ корреляционных матриц, факторный анализ, дискриминантный анализ, многомерное шкалирование. Таким образом РАД. Важный этап работы с данными, между скраббингом и моделированием.

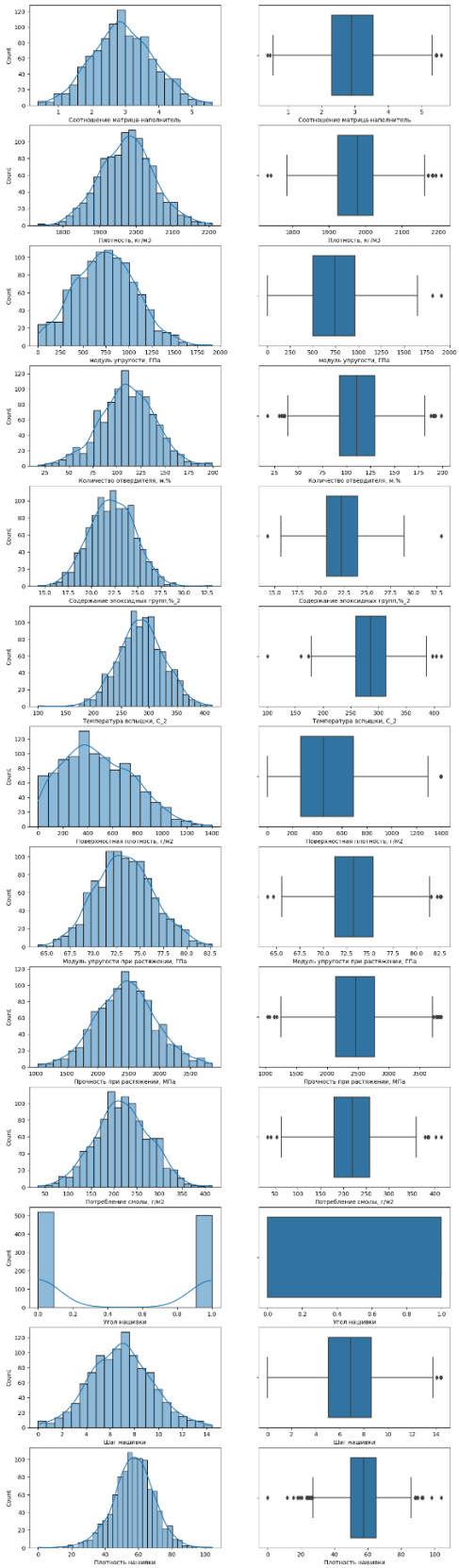
Выведем описательную статистику по каждому столбцу (табл. 3):

Табл. 3 Описательная статистика датасет

	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	1023.0	0.491691	0.500175	0.000000	0.000000	0.000000	1.000000	1.000000
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

На рис. 1-2 приведены гистограммы распределения переменных и диаграммы «ящик с усами». Из полученных данных видно, что все признаки, кроме «Угол нашивки», имеют нормальное распределение и принимают неотрицательные значения. «Угол нашивки» принимает значения: 0, 90.

Рис 1, 2


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

Попарные графики рассеяния точек приведены на рисунке 3.

По данным графикам мы видим, что некоторые точки отстоят далеко от облака, это выбросы — аномальные, значения данных, выходящие за допустимые пределы значений переменной.

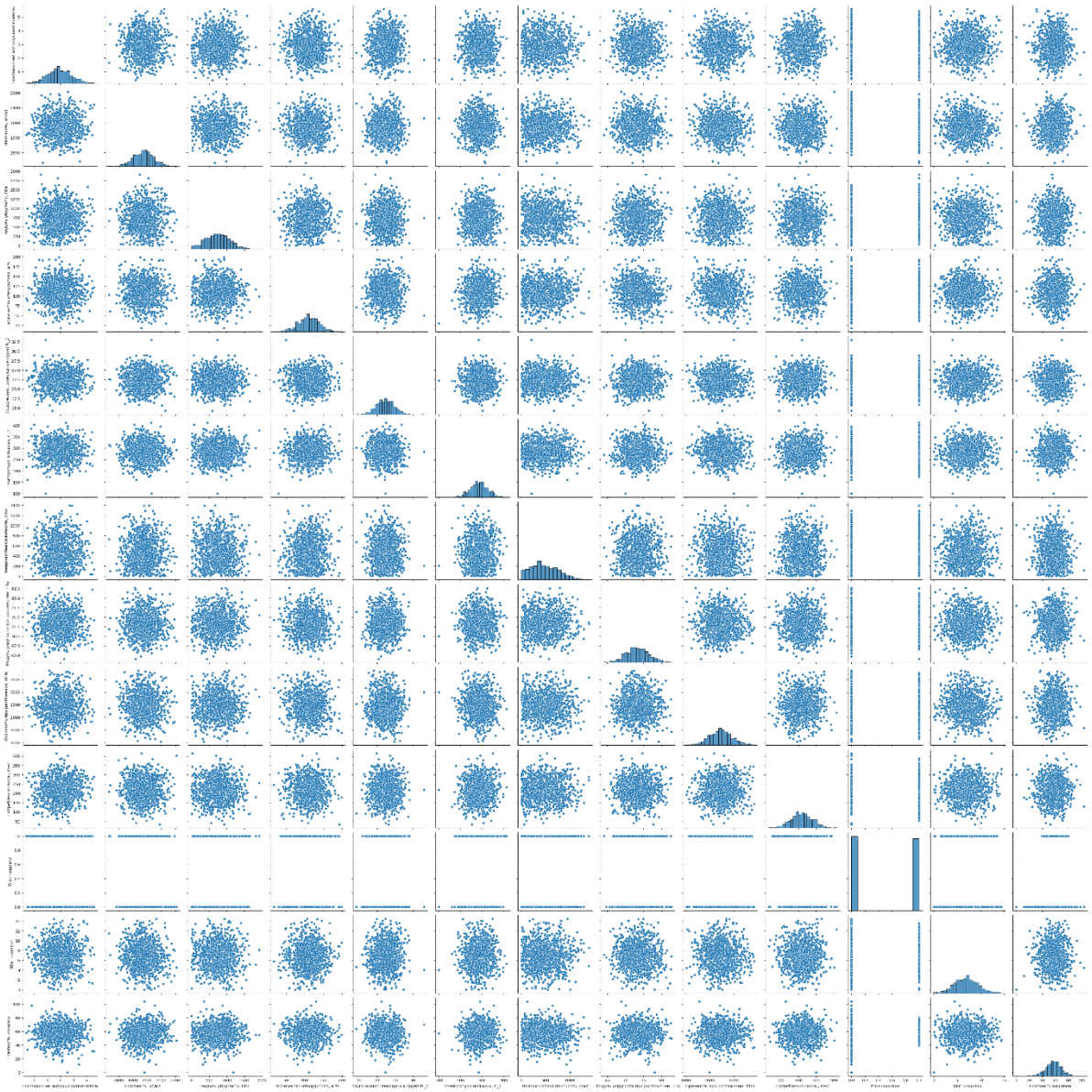


Рис.3 Попарные графики рассеяния точек

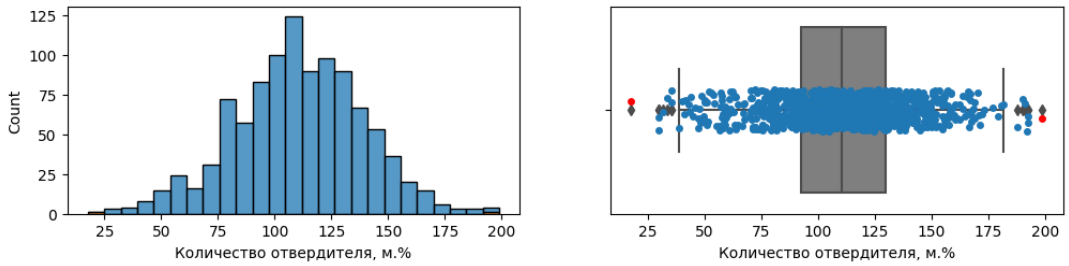
Итак, все признаки, кроме "Угол нашивки, град" имеют нормальное распределение. Они количественные, вещественные. Принимают неотрицательные значения.

"Угол нашивки, град" принимает 2 значения. Можно превратить в бинарный признак.

Для поиска выбросов существуют различные методы. Например: 3-х сигм, межквартильных расстояний, изолирующего леса и др. Метод трех сигм известен своей эффективностью в уменьшении количества дефектов. Метод трех сигм не требует сложной статистической обработки данных и может быть легко реализован.

Поскольку из приведенных выше графиков видно, что распределение большинства переменных близко к нормальному, и так как датасет был предварительно очищен, выберем метод 3 сигм.

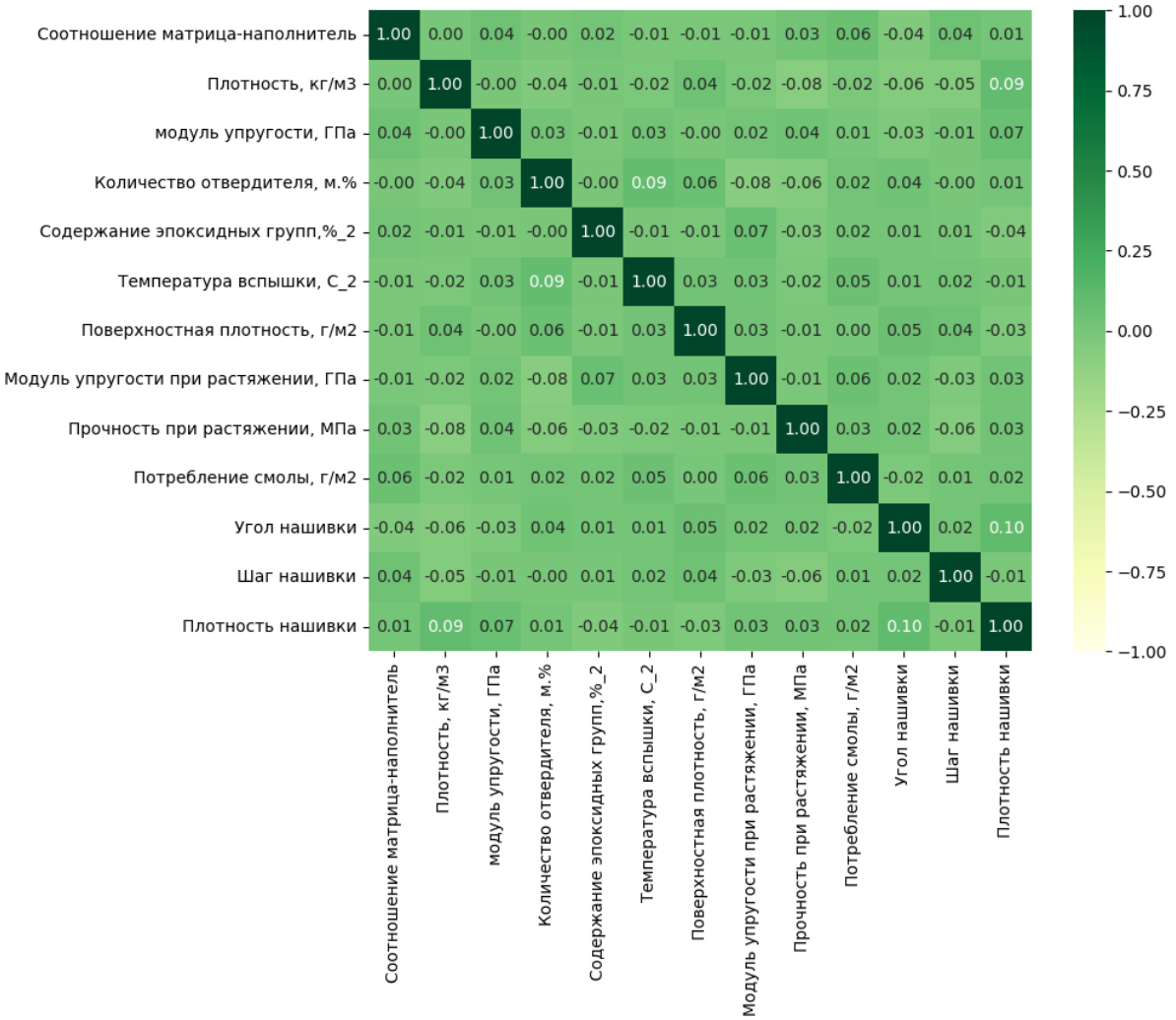
Осуществляем поиск выбросов. Пример выбросов на гистограмме распределения и диаграмме «ящик с усами» приведен на (рис. 4)



(рис. 4)

Значения, определенные как выбросы, удаляем. После этого в датасете осталось 1000 строк и 13 признаков-переменных.

Как ранее указывалось на рисунке 3 мы вывели графики попарного рассеяния точек. По форме «облаков точек» видимых зависимостей, которые могут стать основой работы моделей, не обнаружено. Попробуем увидеть связь между признаками с помощью матрицы корреляции (рис. 5).



(рис. 5)

Коэффициенты корреляции, близки к 0, показывают отсутствие линейной зависимости между признаками. Тогда как для нормальной работы модели по предсказанию выходных переменных выходные переменные должны показывать зависимость от входных.

Очевидно применение линейных моделей регрессии не даст приемлемого результата.

2.4.1 Выбор признаков

В данном параграфе следует учитывать, что никакой дополнительной информации по ходу технологических процессов, по действительно

значимым параметрам процесса (например Т отверждения и др.) постановщиком задания не предоставлено. По указанной причине будем исходить исключительно из поставленной задачи, предоставленных данных «как есть» и из того обстоятельства, что домыслы при обработке данных не допустимы (см., например, ст. 759 ГК РФ.)

Исходя из сказанного и задания разделяем данные на входные и выходные признаки. На выходе у нас 3 целевых признака:

- Модуль упругости при растяжении ГПа, (МУР)

- Прочность при растяжении МПа, (ППР)

- Соотношение матрица – наполнитель (СМН)

Отметим, что по условиям задачи СМН является целевым, по отношению к МУР и ППР, а МУР и ППР, в свою очередь, целевыми по отношению к СМН.

Для каждого из целевых признаков составим отдельную модель, получим 3 отдельных набора переменных для построения моделей и будем решать 3 отдельные задачи.

2.4.2. Алгоритм решения задачи.

Наметим следующий алгоритм решения задач:

- в соответствии с заданием делим данные на тренировочную (70% данных) и тестовую выборки (30% данных);

- выполнить предварительную обработку данных (приведение к виду, необходимому для подачи на вход алгоритму) (препроцесинг);

- выбрать базовую модель для определения нижней границы качества предсказания;

- базовой моделью, возвращаем среднее значение целевого признака, лучшая модель по своим характеристикам должна быть лучше базовой;

- взять несколько моделей с гиперпараметрами по умолчанию, и ис-пользуя 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.

2.4.3. Предварительная обработка данных.

Цель предварительной обработки данных (ПОД) в обеспечении корректной работы моделей.

ПОД выполняется после разделения данных на тренировочную и тестовую выборку, как будто мы не знаем параметров тестовой выборки (минимум, максимум, мат. ожидание, стандартное отклонение).

ПОД для категориальных и количественных признаков выполняется разными методами.

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 или OrdinalEncoder.

Вещественных количественных признаков в датасете большинство. Их значения лежат в разных диапазонах и в разных масштабах. Что подтверждается таблицей 3. Необходимо 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.

В данной работе используем стандартизацию и StandardScaler.

ПОД необходимо повторить в приложении для введенных данных. Реализуем предварительную обработку с помощью ColumnTransformer, а потом сохраняем и перезагружаем данный объект аналогично объекту модели. Выходные переменные изменению не подвергаются.

2.4.4. Перекрестная проверка

Для обеспечения статистической устойчивости метрик модели используем перекрестную проверку или кроссвалидацию. Чтобы ее реализовать, выборка разбивается необходимое количество раз на тестовую и валидационную. Модель обучается на тестовой выборке, 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.

2.4.5. Поиск гиперпараметров по сетке

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 гиперпа-раметры. Перекрестная проверка уже встроена в этот класс.

2.4.6. Метрики качества моделей

Существует множество различных метрик качества, применимых для регрессии. В данной работе будут использованы следующие:

- R2 или коэффициент детерминации измеряет долю дисперсии, объяс-ненную 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;

- MAPE (Mean Absolute Percentage Error) или средняя абсолютная процентная ошибка — безразмерный показатель, представляющий собой взвешенную версию MAE;

- max error или максимальная ошибка данной модели в единицах изме-рения целевой переменной.

RMSE, MAE, MAPE и max error принимают положительные значения, тем не менее для лучшего представления информации отображаться в работе они будут со знаком «-».

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

3. Практическая часть

3.1. Разбиение и предобработка данных (РПД)

3.1.1 РПД для прогнозирования модуля упругости при растяжении

Признаки датасета делим на входные и выходные, а строки - на тренировочное и тестовое множество. Размерности полученных наборов данных показаны на рисунке 8. Описательная статистика входных признаков до и после предобработки показана на рисунке 9. Описательная статистика выходного признака показана на рисунке 10.

```
x1_train: (700, 11) y1_train: (700, 1)
x1_test: (300, 11) y1_test: (300, 1)
```

Рисунок 8 - Размерности тренировочного и тестового множеств после разбиения для 1-й задачи

Описательная статистика входных данных до предобработки

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
min	0.547391	1784.482245	2.436909	33.624187	15.695894	173.973907	1.668002	41.048278	0.000000	0.037639	20.571633
max	5.591742	2192.738783	1649.415706	192.851702	28.907470	403.652861	1288.691844	386.903431	1.000000	14.033215	92.963492
mean	2.943860	1972.286516	738.627618	112.119243	22.179055	286.449560	481.805877	216.838475	0.495714	6.880379	57.403269
std	0.902194	73.148332	326.130594	28.056458	2.335087	40.645101	278.253589	58.108052	0.500339	2.590968	12.036623

Описательная статистика входных данных после предобработки

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Потребление смолы, г/м2	Шаг нашивки	Плотность нашивки	Угол нашивки, град
min	-2.658166	-2.569280	-2.258964	-2.799754	-2.778397	-2.769241	-1.726774	-3.027393	-2.642886	-3.062152	0.000000
max	2.937033	3.015925	2.794707	2.879558	2.883502	2.885639	2.901896	2.928795	2.762655	2.956448	1.000000
mean	0.000000	0.000000	-0.000000	-0.000000	-0.000000	0.000000	0.000000	-0.000000	0.000000	-0.000000	0.495714
std	1.000715	1.000715	1.000715	1.000715	1.000715	1.000715	1.000715	1.000715	1.000715	1.000715	0.500339

Рисунок 9 - Описательная статистика входных признаков до и после предобработки для 1-й задачи

	0
count	700.00000
min	64.05406
max	82.23760
mean	73.39876
std	3.12858

Рисунок 10 - Описательная статистика выходного признака для 1-й задачи

3.1.2. РПД для прогнозирования прочности при растяжении.

Признаки датасета делим на входные и выходные, а строки - на тренировочное и тестовое множество. Размерности полученных наборов данных показаны на рисунке 11. Описательная статистика входных признаков до и после предобработки показана на рисунке 12. Описательная статистика выходного признака показана на рисунке 13.

```
x1_train: (700, 11) y1_train: (700, 1)
x1_test: (300, 11) y1_test: (300, 1)
```

Рисунок 11 - Размерности тренировочного и тестового множеств после разбиения для 2-й задачи

Описательная статистика входных данных до предобработки

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
min	0.547391	1784.482245	2.436909	33.624187	15.695894	173.973907	1.668002	41.048278	0.000000	0.037639	20.571633
max	5.591742	2192.738783	1649.415706	192.851702	28.907470	403.652861	1288.691844	386.903431	1.000000	14.033215	92.963492
mean	2.943860	1972.286516	738.627618	112.119243	22.179055	286.449560	481.805877	216.838475	0.495714	6.880379	57.403269
std	0.902194	73.148332	326.130594	28.056458	2.335087	40.645101	278.253589	58.108052	0.500339	2.590968	12.036623

Описательная статистика входных данных после предобработки

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Потребление смолы, г/м2	Шаг нашивки	Плотность нашивки	Угол нашивки, град
min	-2.567145	-2.753587	-2.225848	-2.727895	-2.679251	-2.825478	-1.713713	-2.972700	-2.802971	-3.198418	0.000000
max	2.893896	2.918573	2.752658	3.128279	2.626492	3.029659	2.942989	2.723003	2.846822	3.161091	1.000000
mean	0.027286	-0.144307	-0.000482	0.159048	-0.075626	0.041830	0.023519	-0.077706	-0.040665	0.037182	0.495714
std	0.976720	1.016295	0.985831	1.031879	0.937766	1.036153	1.006775	0.956950	1.045933	1.057398	0.500339

Рисунок 12 - Описательная статистика входных признаков до и после предобработки для 2-й задачи

Описательная статистика выходной переменной

count	700.00000
min	64.05406
max	82.23760
mean	73.39876
std	3.12858

Рисунок 13 - Описательная статистика выходного признака для 2-й задачи

3.1.3. РПД для прогнозирования соотношения матрица-наполнитель

Признаки датасета делим на входные и выходные, а строки - на тренировочное и тестовое множество. Описательная статистика входных признаков до и после предобработки показана на рисунке 14. Описательная статистика входных признаков до и после предобработки показана на рисунке 15. Описательная статистика выходного признака показана на рисунке 16.

x3_train: (700, 12) y3_train: (700, 1)

x3_test: (300, 12) y3_test: (300, 1)

Рисунок 14 - Размерности тренировочного и тестового множеств после разбиения для 3-й задачи

Описательная статистика входных данных до предобработки

	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
min	1784.482245	2.436909	33.624187	15.695894	173.973907	1.668002	64.054061	1071.123751	41.048278	0.000000	0.037639	20.571633
max	2192.738783	1649.415706	192.851702	28.907470	403.652861	1288.691844	82.237600	3848.436732	386.903431	1.000000	14.033215	92.963492
mean	1972.286516	738.627618	112.119243	22.179055	286.449560	481.805877	73.398761	2469.109198	216.838475	0.495714	6.880379	57.403269
std	73.148332	326.130594	28.056458	2.335087	40.645101	278.253589	3.128575	493.531741	58.108052	0.500339	2.590968	12.036623

Описательная статистика входных данных после предобработки

	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	Шаг нашивки	Плотность нашивки	Угол нашивки, град
min	-2.753587	-2.225848	-2.727895	-2.679251	-2.825478	-1.713713	-2.955472	-2.979732	-2.972700	-2.802971	-3.198418	0.000000
max	2.918573	2.752658	3.128279	2.626492	3.029659	2.942989	2.965684	3.000494	2.723003	2.846822	3.161091	1.000000
mean	-0.144307	-0.000482	0.159048	-0.075626	0.041830	0.023519	0.087468	0.030468	-0.077706	-0.040665	0.037182	0.495714
std	1.016295	0.985831	1.031879	0.937766	1.036153	1.006775	1.018767	1.062693	0.956950	1.045933	1.057398	0.500339

Рисунок 15 - Описательная статистика входных признаков до и после предобработки для 3-й задачи

Описательная статистика выходных данных

	0
count	700.00000
min	0.54739
max	5.59174
mean	2.94386
std	0.90219

Рисунок 16 - Описательная статистика выходного признака для 3-й задачи

3.2. Разработка и обучение моделей для прогнозирования модуля упругости при растяжении

Для подбора лучшей модели для этой задачи берем следующие модели:

- LinearRegression — линейная регрессия (раздел 2.2.1);

- Ridge — гребневая регрессия (раздел 2.2.2);

- Lasso — лассо-регрессия (раздел 2.2.3);

- SVR — метод опорных векторов (раздел 2.2.4);

- KneighborsRegressor — метод ближайших соседей (раздел 2.2.5);

- DecisionTreeRegressor — деревья решений (раздел 2.2.6);

- RandomForestRegressor — случайный лес (раздел 2.2.7).

В качестве базовой модели взят DummyRegressor, возвращающий среднее значение целевого признака.

Метрики работы выбранных моделей с гиперпараметрами по умолчанию, полученные с помощью перекрестной проверки на тестовом множестве, приведены на рисунке 17.

Ни одна из выбранных мной моделей не оказалась подходящей для наших данных.

Коэффициент детерминации R2 близок к 0 для линейных моделей и метода опорных векторов. Значит, они не лучше базовой модели. И остальные метрики у них примерно совпадают с базовой моделью.

Гораздо хуже линейных моделей с гиперпараметрами по умолчанию отработали метод ближайших соседей и деревья решений.

Случайный лес отработал лучше, чем одно дерево решений, но хуже, чем линейные модели.

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.019376	-3.126837	-2.510495	-0.034288	-7.798105
LinearRegression	-0.018532	-3.123936	-2.502366	-0.034179	-8.098392
Ridge	-0.018463	-3.123834	-2.502325	-0.034178	-8.097452
Lasso	-0.019376	-3.126837	-2.510495	-0.034288	-7.798105
SVR	-0.041456	-3.157875	-2.499637	-0.034118	-8.357012
KNeighborsRegressor	-0.238674	-3.443022	-2.725185	-0.037216	-8.823389
DecisionTreeRegressor	-1.034156	-4.403633	-3.589790	-0.048987	-11.822403
RandomForestRegressor	-0.075323	-3.208305	-2.555245	-0.034924	-8.380008

Рисунок 17 — Результаты моделей с гиперпараметрами по умолчанию

	R2	RMSE	MAE	MAPE	max_error
Ridge(alpha=80, positive=True, solver='lbfgs')	-0.016604	-3.121555	-2.494491	-0.034068	-7.851330
Lasso(alpha=0.05)	-0.012094	-3.114368	-2.500839	-0.034157	-7.965382
SVR(C=0.01, kernel='linear')	-0.017814	-3.123659	-2.500515	-0.034147	-8.061850
KNeighborsRegressor(n_neighbors=29)	-0.036593	-3.147992	-2.512539	-0.034342	-8.157406
DecisionTreeRegressor(max_depth=2, max_features=2, random_state=42)	-0.018267	-3.125442	-2.490189	-0.034017	-8.154902
RandomForestRegressor(bootstrap=False, criterion='absolute_error', max_depth=5, max_features=1, n_estimators=50, random_state=42)	-0.017821	-3.124964	-2.497013	-0.034103	-8.146335

Рисунок 18 — Результаты моделей после подбора гиперпараметров

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

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

Метод опорных векторов в процессе подбора гиперпараметры лучшим ядром выбрал линейное и отработал аналогично линейным моделям.

Метод ближайших соседей увеличением количества соседей радикально улучшил качество работы. Но его лучшие результаты все равно немного, но отстают от линейных моделей.

Деревья решений при кропотливом подборе параметров превзошли результат линейной модели. Но они не являются объясняющей зависимость моделью.

Собирая деревья в ансамбли, можно улучшать характеристики. Но подбор параметров для леса затруднен тем, что это затратный по времени процесс. По этой причине не удалось получить комбинацию параметров для леса, которая была был лучше дерева решений. Поэтому в качестве лучшей модели выбираю дерево решений.

На рисунке 19 приведена визуализация работы лучшей модели на тестовом множестве.

Сложно визуализировать регрессию в многомерном пространстве. Но даже на таком графике мы видим, насколько не соответствует лучшая модель исходным данным и насколько она неудачна.

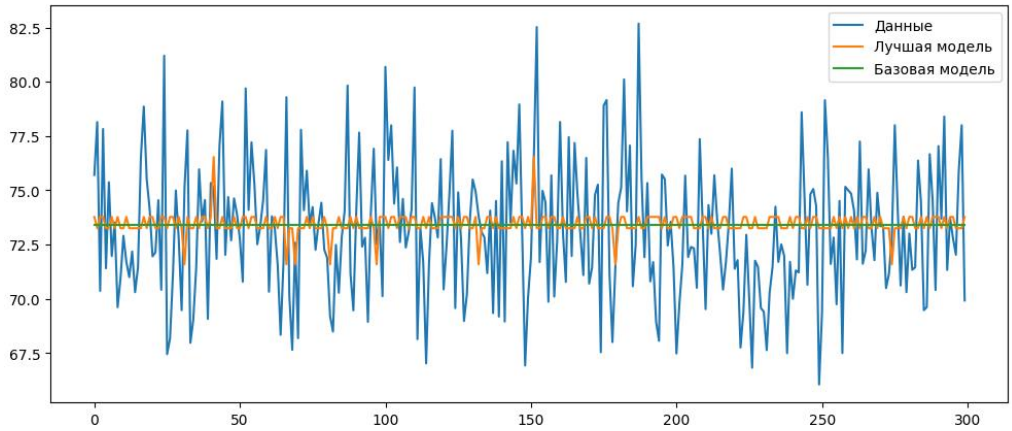


Рисунок 19 — Визуализация работы модели

	R2	RMSE	MAE	MAPE	max_error
Базовая модель	-0.007651	-3.082670	-2.479138	-0.033976	-9.283290
Лучшая модель (дерево решений)	-0.009086	-3.084865	-2.480036	-0.034013	-9.272048

Рисунок 20 - Метрики работы лучшей модели на тестовом множестве

Метрики работы лучшей модели на тестовом множестве и сравнение с базовой отражены на рисунке 20. Они подтверждают: полученная модель хуже базовой. Результат исследования отрицательный. Не удалось получить модель, которая могла бы оказать помощь в принятии решений специалисту предметной области.

3.3. Разработка и обучение моделей для прогнозирования прочности при растяжении.

Для подбора лучшей модели применим следующие модели:

- LinearRegression — линейная регрессия (раздел 2.2.1);

- Ridge — гребневая регрессия (раздел 2.2.2);

- Lasso — лассо-регрессия (раздел 2.2.2);

- SVR — метод опорных векторов (раздел 2.2.3);

- DecisionTreeRegressor — деревья решений (раздел 2.2.5);

- GradientBoostingRegressor — случайный лес (раздел 2.2.7).

В качестве базовой модели взят DummyRegressor, возвращающий среднее значение целевого признака.

Метрики работы выбранных моделей с гиперпараметрами по умолчанию, полученные с помощью перекрестной проверки на тестовом множестве, приведены на рисунке 21.

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.022944	-493.539876	-391.010975	-0.171456	-1281.791709
LinearRegression	-0.014804	-491.329446	-391.262712	-0.171170	-1305.947015
Ridge	-0.014749	-491.316685	-391.248131	-0.171164	-1305.883895
Lasso	-0.013580	-491.039900	-390.926249	-0.171039	-1304.848543
SVR	-0.021077	-493.116843	-390.543237	-0.170362	-1279.655107
DecisionTreeRegressor	-1.097452	-700.282012	-561.980002	-0.244517	-1784.349498
GradientBoostingRegressor	-0.033926	-496.051787	-398.082126	-0.172868	-1274.138037

Рисунок 21 — Результаты моделей с гиперпараметрами по умолчанию

Ни одна из моделей не соответствует данным.

R2 близок к 0 для линейных моделей и метода опорных векторов. Значит, они не лучше базовой модели. Остальные метрики у них примерно идентичны базовой модели.

Гораздо хуже линейных моделей с гиперпараметрами по умолчанию отработали деревья решений.

Градиентный бустинг с параметрами по умолчанию отработал лучше деревьев решений. Он тоже близок к базовой модели.

После подбора гиперпарметров по сетке с перекрестной проверкой, получили метрики, приведенные на рисунке 22.

	R2	RMSE	MAE	MAPE	max_error
Ridge(alpha=710, solver='sparse_cg')	-0.010171	-490.370725	-389.300830	-0.170584	-1291.112095
Lasso(alpha=20)	-0.010149	-490.357498	-389.214674	-0.170515	-1293.747766
SVR(C=0.02, kernel='linear')	-0.021227	-493.150121	-390.526504	-0.170386	-1279.607766
DecisionTreeRegressor(max_depth=1, max_features=6, random_state=42, splitter='random')	-0.019887	-492.777398	-390.140019	-0.171163	-1281.728096
GradientBoostingRegressor(max_depth=2, max_features=1, random_state=42)	0.009853	-485.655328	-386.455905	-0.168528	-1276.277562

Рисунок 22 — Результаты моделей после подбора гиперпараметров

Подбор гиперпараметров не помог получить модель, превосходящую базовую. Все модели крайне плохо описывают исходные данные. Не удалось добиться коэффициента детерминации, больше нуля.

Линейные модели после подбора гиперпараметров немного улучшили характеристики.

Метод опорных векторов отработал аналогично линейным моделям.

Деревья решений после подбора параметров улучшили неудачный результат с параметрами по умолчанию.

Лучший результат дает градиентный бустинг. Значения ошибок примерно такие же, как у дерева решений. Но коэффициент детерминации немного больше, что показывает чуть лучшую объясняющую способность модели.

По данным основаниям в качестве лучшей модели принимаем градиентный бустинг. На рисунке 23 приведена визуализация работы лучшей модели на тестовом множестве.

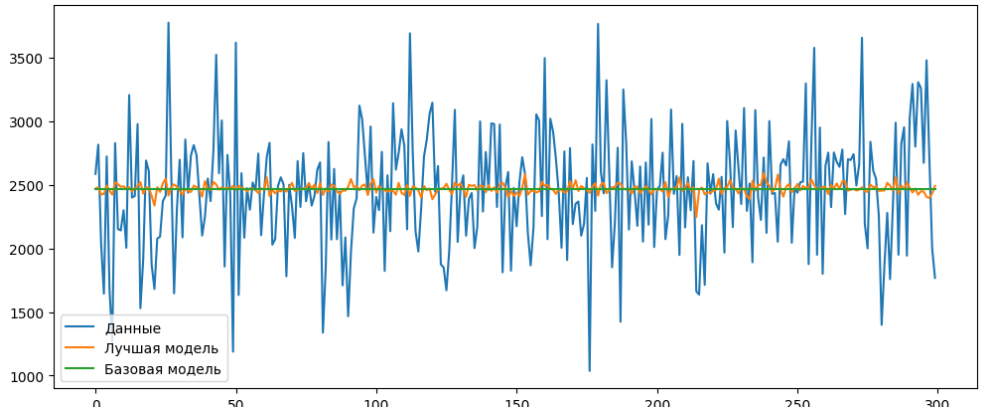


Рисунок 23 — Визуализация работы модели

На графиках результатов применения градиентного бустинга с выбранными параметрами, мы видим насколько они далеки от исходных данных.

Сравнение прогнозов лучшей модели с базовой на тестовом множестве отражено на рисунке 24. Градиентный бустинг показывает результат лишь чуть-чуть лучше базового. Но очевидно, что результат исследования отрицательный. Не удалось получить модели, которая могла бы оказать помощь в принятии решений специалисту предметной области.

	R2	RMSE	MAE	MAPE	max_error
Базовая модель	-0.000928	-464.631542	-363.886617	-0.162616	-1432.252593
Лучшая модель (градиентный бустинг)	-0.006114	-465.833555	-364.992369	-0.162767	-1399.342924

Рисунок 24. Сравнение прогнозов базовой и лучшей моделей на тестовом наборе.

3.4. Разработка нейронной сети для прогнозирования соотношения матрица-наполнитель

По заданию для соотношения матрица-наполнитель необходимо построить нейросеть. Но для сравнения нам также понадобится базовая модель DummyRegressor, возвращающая среднее целевого признака.

3.4.1. MLPRegressor из библиотеки sklearn

Строю нейронную сеть с помощью класса MLPRegressor следующей архитектуры:

- слоев: 6;

- нейронов на каждом слое: 24; 48, 48, 48, 24, 24

- активационная функция: relu;

- оптимизатор: adam;

- пропорция разбиения данных на тестовые и валидационные: 30%;

- ранняя остановка, если метрики на валидационной выборке не улучшаются;

- количество итераций: 5000.

Нейросеть обучилась за 1,28 сек и 64 итерации. График обучения приведен на рисунке 25.

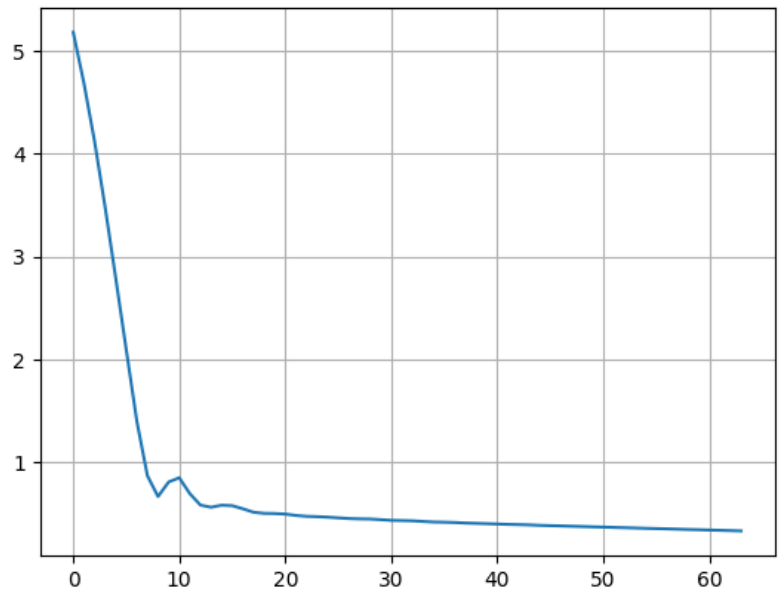


Рисунок 25 — График обучения MLPRegressor

Визуализация результатов, полученных нейросетью, приведены на рисунке 26. Из графиков усматривается, что нейросеть пыталась подстроится под исходные данные, но не получилось.

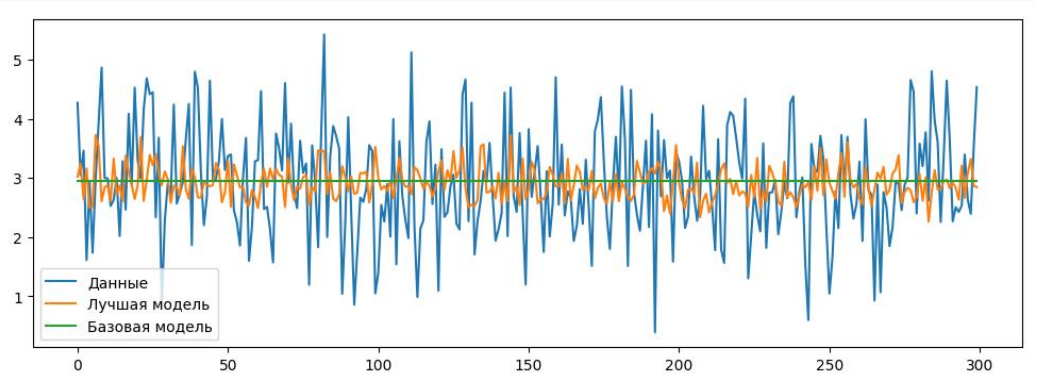


Рисунок 26 — Визуализация работы модели

Метрики работы нейросети MLPRegressor на тестовом множестве и сравнение с базовой моделью отражены на рисунке 27. Несмотря на красивый график с рисунка 26, метрики говорят об отсутствии результата, который можно внедрить. Значения ошибок нейросети **составляет 30,7%, а ее значения ошибок** хуже, чем у базовой модели.

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.000744	-0.924041	-0.739327	-0.340221	-2.554458
MLPRegressor	-0.092706	-0.965565	-0.774559	-0.349162	-2.683511

Рисунок 27 — Метрики работы нейросети MLPRegressor на тестовом множестве

3.4.2. Нейросеть из библиотеки tensorflow

Для более обоснованных выводов построим нейронную сеть с помощью класса keras.Sequential со следующими параметрами:

- входной слой для 12 признаков;

- выходной слой для 1 признака;

- скрытых слоев: 8;

- нейронов на каждом скрытом слое: 24;

- активационная функция скрытых слоев: relu;

- оптимизатор: Adam;

- loss-функция: MeanAbsolutePercentageError.

Архитектура нейросети приведена на рисунке 28.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_1 (Dense)             (None, 24)                312

 dense_2 (Dense)             (None, 24)                600

 dense_3 (Dense)             (None, 24)                600

 dense_4 (Dense)             (None, 24)                600

 dense_5 (Dense)             (None, 24)                600

 dense_6 (Dense)             (None, 24)                600

 dense_7 (Dense)             (None, 24)                600

 dense_8 (Dense)             (None, 24)                600

 out (Dense)                 (None, 1)                 25

=================================================================
Total params: 4,537
Trainable params: 4,537
Non-trainable params: 0
_________________________________________________________________
```

Рисунок 28 — Архитектура нейросети в виде summary

Запускаю обучение нейросети со следующими параметрами:

- пропорция разбиения данных на тестовые и валидационные: 30%;

- количество эпох: 50.

- раннюю остановку не использую.

График обучения приведен на рисунке 30, ошибка — в таблице 2.

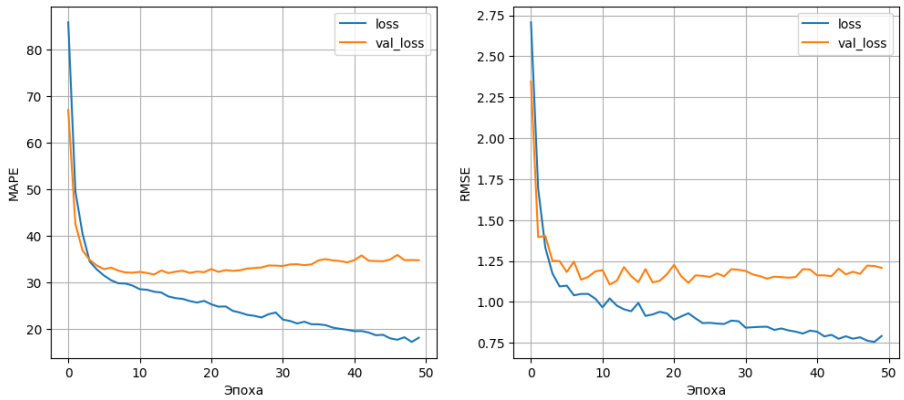


Рисунок 30 — График обучения нейросети

Видно, что примерно до 8 эпохи обучение шло хорошо, а потом сеть начала переобучаться. Значение loss на тестовых выборках продолжило уменьшаться, а на валидационной начало расти.

Одним из способов борьбы с переобучением может быть ранняя остановка обучения, если val_loss начинает расти. Для этого в tensorflow используются callbacks. Логично будет взять нейросеть с той же архитектурой и запустить обучение с ранней остановкой. График обучения приведен на рисунке 24, а ошибка — в таблице 2. Очевидно, что решение проблемы переобучения повышает точность модели на новых данных.

Еще одним методом борьбы с переобучением является добавление Dropout - слоев. Построим модель аналогичной архитектуры, только после

каждого скрытого слоя добавим слой Dropout с параметром 0.05. Такой слой выключит 5% случайных нейронов на каждом слое.

График обучения приведен на рисунке 25, а ошибка — в таблице 2. Видно, что Dropout-слои справились с переобучением.

Использование ранней остановки сокращает время на обучение модели, а использование Dropout увеличивает. Но уменьшается риск, что мы остано-вились слишком рано.

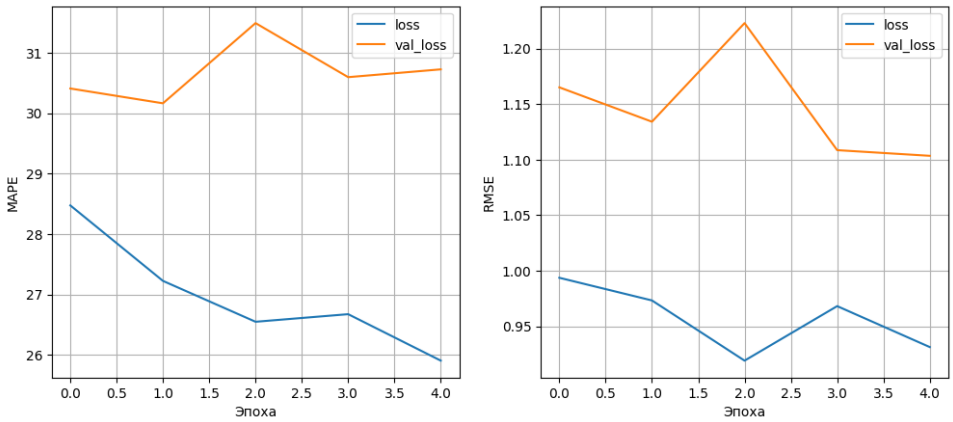


Рисунок 31 — График обучения нейросети с ранней остановкой

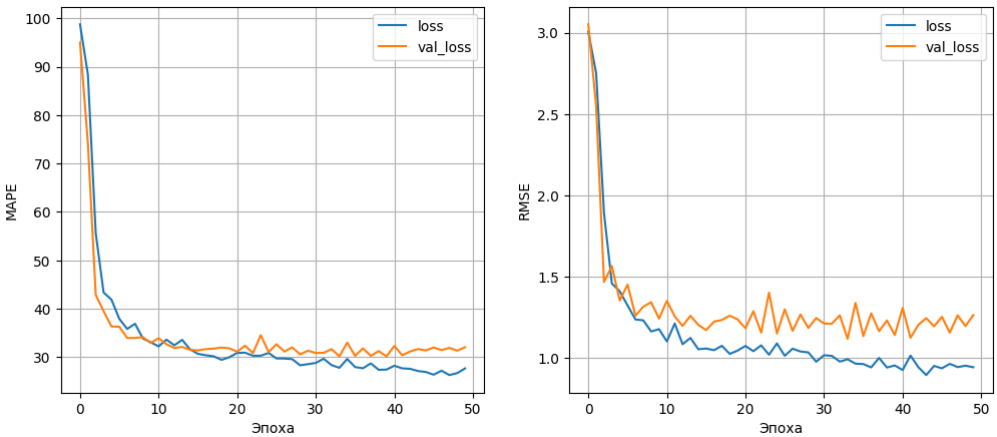


Рисунок 32 — График обучения нейросети с Dropout-слоем

Визуализация результатов работы нейросетей отображена на рисунке 33, а их метрики — на рисунке 34.

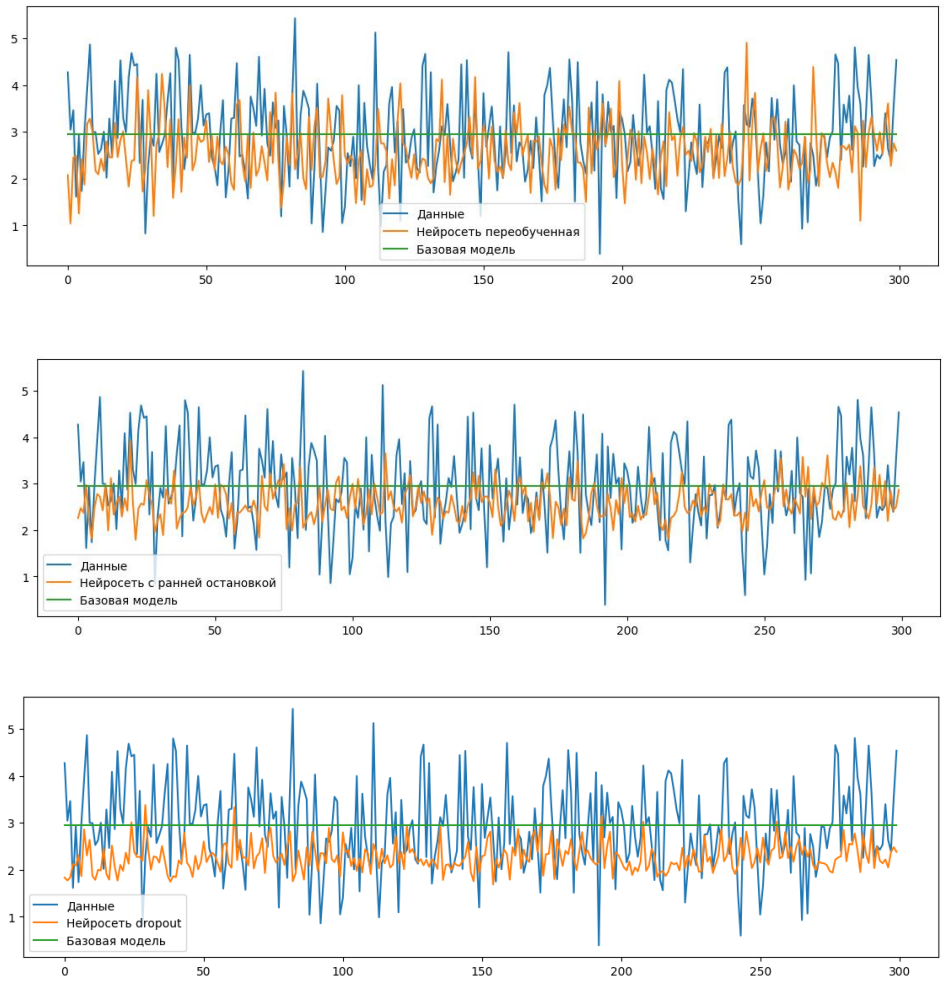


Рисунок 33 - Визуализация результатов работы нейросетей

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.000744	-0.924041	-0.739327	-0.340221	-2.554458
Нейросеть переобученная	-0.461445	-1.116660	-0.895166	-0.354564	-3.230687
Нейросеть с ранней остановкой	-0.271748	-1.041671	-0.813417	-0.320615	-3.378014
Нейросеть dropout	-0.593824	-1.166138	-0.923011	-0.332316	-3.672032

Рисунок 34 -Метрики работы нейросетей на тестовом множестве

Визуализация результатов показывает, что нейросеть из библиотеки tensorflow старалась подстроиться к данным. Выглядят результаты

«похоже», но метрики объективно не лучшие, предсказывает гораздо хуже базовой модели.

3.5 Тестирование модели

Согласно заданию, необходимо сравнить ошибку каждой модели на тренировочной и тестирующей части выборки.

Модель для предсказания модуля упругости при растяжении - {DecisionTreeRegressor(criterion='absolute_error', max_depth=2, max_features=10, random_state=3128, splitter='random')} Сравнение ее ошибок показано на рисунке 35.

	R2	RMSE	MAE	MAPE	max_error
Модуль упругости, тренировочный	0.015191	-3.102502	-2.469764	-0.033736	-9.199664
Модуль упругости, тестовый	-0.009086	-3.084865	-2.480036	-0.034013	-9.272048

Рисунок 35 - Сравнение ошибок модели для модуля упругости при растяжении на тренировочном и тестовом датасете.

Дерево решений имеет ошибку на тренировочном датасете меньше, чем на тестовом, потому что какое то обучение все же имело место. Но даже на тренировочном датасете оно не нашло закономерности во входных данных. Задачу решить не удалось.

Если модуль упругости при растяжении лежит в диапазоне [64.05-82.68], то наша модель делает предсказание с точностью ±8.15. Она работает не точнее среднего, и бесполезна для применения в реальных условиях.

Модель для предсказания прочности при растяжении - GradientBoostingRegressor(max_depth=1, max_features=1, n_estimators=50, random_state=42) Сравнение ее ошибок показано на рисунке 36.

	R2	RMSE	MAE	MAPE	max_error
Прочность при растяжении / train	0.056663	-479.002875	-379.538641	-0.166241	-1369.735087
Прочность при растяжении / test	-0.006114	-465.833555	-364.992369	-0.162767	-1399.342924

Рисунок 36 - Сравнение ошибок модели для прочности при растяжении на тренировочном и тестовом датасете.

Градиентный бустинг показал положительный, но близкий к 0 коэффициент детерминации. Ошибка на тестовом множестве незначимо больше, чем на тренировочном. Из чего следует, что модель нашла следы зависимости, а не выучила данные. Но задача не решена.

Если прочность при растяжении лежит в диапазоне [1071.12-3848.44], то наша модель дает предсказание с точностью ±1384.85. Она работает не точнее среднего, и бесполезна для применения в реальных условиях.

Модель для предсказания соотношения матрица-наполнитель — нейросеть из tensorflow, обученная с ранней остановкой. Сравнение ее ошибок показано на рисунке 37.

	R2	RMSE	MAE	MAPE	max_error
Соотношение матрица-наполнитель, тренировочный	-0.124062	-0.955839	-0.746561	-0.268580	-2.944175
Соотношение матрица-наполнитель, тестовый	-0.271748	-1.041671	-0.813417	-0.320615	-3.378014

Рисунок 37 - Сравнение ошибок модели для соотношения матрица-наполнитель на тренировочном и тестовом датасете.

У нейросети показатели для тестовой выборки сильнее отличаются в худшую сторону от показателей тренировочной. Это указывает на то, что НС не нашла закономерностей, а стала учить данные из тестовой выборки. Возможно, требуется иное построение архитектуры нейронной сети, чтобы получить лучший результат. Но задача не решена.

Если соотношение матрица-наполнитель лежит в диапазоне [0.39-5.46], то наша модель может предсказать с точностью ±3.08. Она работает не точнее среднего, и бесполезна для применения в реальных условиях.

3.6 Разработка приложения

Пригодных к внедрению моделей в ходе работы получить не удалось, однако можно разработать функционал приложения. В дальнейшем, в случае построения адекватной реальности модели, будет возможно внедрить ее в готовое приложение.

В приложении в соответствии с п.п.3,4 Задания реализовано:

- ввод входных параметров;

- получение и отображение прогноза прогноза значения матрица-наполнитель.

Веб-приложение разработано с помощью языка Python, фреймворка Flask, языка HTML. Инструкция: Открываем приложение по указанному ниже адресу, вводим данные, получаем прогноз.

Приложение выложено по адресу https://prediction-matrix.onrender.com/ и в репозитории GitHub по адресу: https://github.com/DdddOrlov/VKR-app-matrix001

3.7. Создание удаленного репозитория

Для данного исследования был создан удаленный репозиторий на GitHub, который находится по адресу https://github.com/DdddOrlov/DddddVKR и https://github.com/DdddOrlov/VKR-app-matrix001. На него были загружены

результаты работы: исследовательский notebook, код приложения, презентация, приложение.

Заключение

В ходе выполнения данной работы мы прошли практически весь Dataflow pipeline, рассмотрели большую часть операций и задач, которые приходится выполнять специалисту по работе с данными.

Этот поток операций и задач включает:

- изучение теоретических методов анализа данных и машинного обучения;

- изучение основ предметной области, в которой решается задача;

- извлечение и трансформацию данных. Здесь нам был предоставлен готовый набор данных, поэтому через трудности работы с разными источниками и парсингом данных мы еще не соприкоснулись;

- проведение разведочного анализа данных статистическими методами;

- DataMining — извлечение признаков из датасета и их анализ;

- разделение имеющихся, в нашем случае размеченных, данных на обучающую, валидационную, тестовую выборки;

- выполнение предобработки (препроцессинга) данных для обеспечения корректной работы моделей;

- построение аналитического решения. Это включает выбор алгоритма решения и модели, сравнение различных моделей, подбор гиперпараметров модели;

- визуализация модели и оценка качества аналитического решения;

- сохранение моделей;

- разработка и тестирование приложения для поддержки принятия решений специалистом предметной области, которое использовало бы найденную модель;

- внедрение решения и приложения в эксплуатацию. Этот блок задач мы тоже пока не затронули.

В данной работе произведены действия над предоставленным датасетом в соответствии с выданным заданием. При этом положительного решения задания получить не удалось, не разработаны модели, которые бы описывали закономерности предметной области. Указанный результат был прогнозируем уже на этапе анализа переменных.

Так, например, переменная Содержание эпоксидных групп, %, колеблется в границах 14 - 33. В соответствии с ГОСТ Р 56211-2014 (Таблица 1.) Такое содержание эпоксидных групп охватывает смолы, как минимум 4 марок.

Наименование показателя	Норма для марки										Метод испытания
	ЭД-22		ЭД-20		ЭД-16		ЭД-14	ЭД-10	ЭД-8		
	высший сорт	первый сорт	высший сорт	первый сорт	высший сорт	первый сорт			высший сорт	первый сорт	
3 Массовая доля эпоксидных групп, %	р 22,1-23,6		20,0-22,5		16,0-18,0		13,9-15,9	10,0-13,0	8,5-10,0	8,0-10,0	По ГОСТ 12497 и 7.5 настоящего стандарта

Из указанного следует, что в датасете были смешаны данные для совершенно разных исходных веществ (и соответственно испытуемых систем). Возможно, часть признаков была сгенерирована синтетически.

Эпоксидные смолы, их композиции и их свойства изучаются очень давно и детально. Накоплен огромный фактический материал. Даже поверхностный анализ литературы показывает, что точки экстремума прочностных характеристик для разных смол, при отверждении имеют разные координаты. Очевидно, что при сложении функций зависимости прочностных характеристик от разных смол, будут получены новые точки экстремума (для суммы). Но они не будут совпадать с соответствующими точками для отдельных систем. Фактически они не будут отражать реальность.

Кроме того, графики функций могут пересекаться, а это означает, что при смешении данные из одной системы могут по ошибке включатся в другую систему и так же удалять результат от реальности.

Кроме того, на прочностные характеристики эпоксидных композитов в разрезе предоставленного датасета, влияет множество скрытых переменных. Например: внутренняя поверхность наполнителя, степень затекания смолы во внутренние полости, температура отверждения, влажность и др. Из данного обстоятельства следует, что любая корреляция для данного датасета будет априори ложной.

Следует учесть, что удалять данные из датасета было неоднократно запрещено постановщиком задачи.

Таким образом основные причины отрицательного результата при решении задачи:

- несогласованность данных датасета

- очевидное наличие скрытых переменных

- отсутствие значимой дополнительной информации

Дальнейшие возможные пути решения этой задачи, следующие:

- получение дополнительной информации

- согласование данных

- изучение скрытых переменных

- преобразование датасета к виду, который мог бы учесть изложенные выше недостатки

- углубленное изучение аппарата датасайнс

- математический анализ

Литература:

1. с. 443, т.2, Химическая энциклопедия., Москва 1990

2. Википедия, статья: Композитный материал. Режим доступа:https://ru.wikipedia.org/wiki/Композитный_материал (дата обращения: 14.04.2023)

3. Л.И. Бондалетова, В.Г. Бондалетов Полимерные композиционные материалы Изд. Томского политехнического университета, 2013. с. 10

4. П. Брюс, Э. Брюс. 1. Разведочный анализ данных // Практическая статистика для специалистов Data Science. — СПб.: БХВ-Петербург, 2018. — С. 19—58.

5. Дж. Тьюки. Анализ результатов наблюдений, разведочный анализ. Изд. Мир, 1981 с.с. 5-38

6. ГОСТ Р 56211-2014 Смолы эпоксидно-диановые неотвержденные. Технические условия. Дата введения 2016-01-01

7. Ли. Х., Невилл К., Справочное руководство по эпоксидным смолам, М. Энергия, 1973 г., С 415

8. Композиционные материалы, Т. 2, Л. Браутман, Р. Крок "Мир", Москва, 1978, С. 558

9. Справочник по сопротивлению материалов С.П. Фесик, Москва, 1982 г. с. 280, с. 6 и др.

10. Адгезивы и адгезионные соединения, Л. Ли, Москва, Мир, 1988, 226 с.

11. Теоретические основы переработки полимеров Р.В. Торнер, М, Химия, 1977 г. , Гл. 1 с. 15-44

12.Эпоксидные полимеры и композиции, Чернов И.З., Химия, 1982, с. 214

13. Документация по языку программирования python: – Режим доступа: https://docs.python.org/3.8/index.html. (дата обращения: 14.04.2023)

14. Документация по библиотеке numpy: – Режим доступа: https://numpy.org/doc/1.22/user/index.html#user. (дата обращения: 15.04.2023)

15. Документация по библиотеке pandas: – Режим доступа: https://pandas.pydata.org/docs/user_guide/index.html#user-guide. (дата обращения: 15.04.2023)

16. Документация по библиотеке matplotlib: – Режим доступа: https://matplotlib.org/stable/users/index.html. (дата обращения: 15.04.2023)

17. Документация по библиотеке seaborn: – Режим доступа: https://seaborn.pydata.org/tutorial.html. (дата обращения: 16.04.2023)

18. Документация по библиотеке sklearn: – Режим доступа: https://scikit-learn.org/stable/user_guide.html. (дата обращения: 16.04.2023)

19. Документация по библиотеке keras: – Режим доступа: https://keras.io/api/.(дата обращения: 16.04.2023)

20. Билл Любанович. Простой Python. Современный стиль программирования. — СПб.: Питер, 2016. — 480 с.

21. Руководство по быстрому старту в flask: – Режим доступа: https://flask-russian-docs.readthedocs.io/ru/latest/quickstart.html. (дата обращения: 24.04.2023)

22. Loginom Вики. Алгоритмы: – Режим доступа: https://wiki.loginom.ru/algorithms.html. (дата обращения: 24.04.2023)

23. Andre Ye. 5 алгоритмов регрессии в машинном обучении, о которых вам следует знать: – Режим доступа:https://habr.com/ru/company/vk/blog/513842/.(дата обращения:25.04.2023)

24. Alex Maszański. Метод k-ближайших соседей (k-nearest neighbour): – Режим доступа: https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19. (дата обращения: 25.04.2023)

25. Yury Kashnitsky. Открытый курс машинного обучения. Тема 3. Классификация, деревья решений и метод ближайших соседей: – Режим доступа: https://habr.com/ru/company/ods/blog/322534/.(дата обращения: 25.04.2023)

26. Yury Kashnitsky. Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес: – Режим доступа: https://habr.com/ru/company/ods/blog/324402/.(дата обращения: 25.04.2023)

27. Alex Maszański. Машинное обучение для начинающих: алгоритм случайного леса (Random Forest): – Режим доступа: https://proglib.io/p/mashinnoe-obuchenie-dlya-nachinayushchih-algoritm-sluchaynogo-lesa-random-forest-2021-08-12. (дата обращения: 25.04.2023)

28. Alex Maszański. Решаем задачи машинного обучения с помощью алгоритма градиентного бустинга: – Режим доступа: https://proglib.io/p/reshaem-zadachi-mashinnogo-obucheniya-s-pomoshchyu-algoritma-gradientnogo-bustinga-2021-11-25. (дата обращения: 25.04.2023)

29. Жерон, Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. Пер. с англ. - СпБ.: ООО "Альфа-книга': 2018. - 688 с.: ил

30. Д. Рутковская, М. Пилиньский, Л. Рутковский Нейронные сети, генетические алгоритмы и нечеткие системы. - М.: Горячая Линия - Телеком. - 2013. - 384 с.

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

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

33. Адитья Бхаргава Грокаем алгоритмы , Москва 2017 , 288 с.

Приложение А. Скриншот веб-приложения