МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по курсу

«Data Science»

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

Слушатель Семина Юлия Васильевна

Москва, 2023

Содержание

Введение

Композиционные материалы – это материалы, которые состоят из двух или более отдельных компонентов, объединенных в единое целое. Они объединяют в себе свойства и преимущества различных компонентов, что позволяет создавать материалы с уникальными свойствами и характеристиками. Такие материалы используются во многих отраслях промышленности, включая авиацию, судостроение, машиностроение, энергетику и медицину. Их применение позволяет сократить вес и объем конструкций, повысить их прочность, жесткость и устойчивость к воздействию различных нагрузок и факторов.

Современные композиты изготавливаются из материалов: полимеры, керамика, стеклянные и углеродные волокна, но данный принцип сохраняется. У такого подхода есть и недостаток: даже если мы знаем характеристики исходных компонентов, определить характеристики композита, состоящего из этих компонентов, достаточно проблематично. Для решения этой проблемы есть два пути: физические испытания образцов материалов, или прогнозирование характеристик. Суть прогнозирования заключается в симуляции представительного элемента объема композита, на основе данных о характеристиках входящих компонентов (связующего и армирующего компонента).

На входе имеются данные о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.).

Цель: на выходе необходимо спрогнозировать ряд конечных свойств получаемых композиционных материалов. 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.

1 Аналитическая часть

1.1 Постановка задачи

Для исследовательской работы были даны 2 файла: X_bp.xlsx (с данными о параметрах базальтопластика, состоящий из 1023 строки и 10 столбцов) и X_nup.xlsx (данными нашивок углепластика, состоящий из 1040 строк и 3 столбцов).

In [ ]:
```
# загрузка данных из X_bp.
# удалим неинформативный столбец, содержащий индекс

X_bp = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/dataset/X_bp.xlsx', index_col=0)
X_bp.shape      #__определим размерность файла
```

Out[ ]: (1023, 10)

In [ ]:
```
X_bp.head()     #__отобразим первые пять строк датасета (по умолч.)
```

Out[ ]:

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2
0	1.857143	2030.0	738.736842	30.00	22.267857	100.000000	210.0	70.0	3000.0	220.0
1	1.857143	2030.0	738.736842	50.00	23.750000	284.615385	210.0	70.0	3000.0	220.0
2	1.857143	2030.0	738.736842	49.90	33.000000	284.615385	210.0	70.0	3000.0	220.0
3	1.857143	2030.0	738.736842	129.00	21.250000	300.000000	210.0	70.0	3000.0	220.0
4	2.771331	2030.0	753.000000	111.86	22.267857	284.615385	210.0	70.0	3000.0	220.0

Рисунок 1 – Данные датасета о параметрах базальтопластика

```
X_nup = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/dataset/X_nup.xlsx', index_col=0)
X_nup.shape      #__определим размерность файла
```

Out[ ]: (1040, 3)

In [ ]:
```
X_nup.head(1000)     #__отобразим первые пять строк датасета (по умолч.)
```

Out[ ]:

	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0	0	4.000000	57.000000
1	0	4.000000	60.000000
2	0	4.000000	70.000000
3	0	5.000000	47.000000
4	0	5.000000	57.000000

Рисунок 2 – Данные датасета о параметрах углепластика

Задача собрать исходные данные файлы в единый набор данных, чтобы разработать модели для прогноза модуля упругости при растяжении, прочности при растяжении и соотношения матрица-наполнитель. Понимаем, что эти исходные датасеты имеют разную размерность. Объединение датасетов выполнено по условию задачи по типу INNER.

```
In [ ]:
# объединение датасетов
df = X_bp.join(X_nup, how='inner')
df.shape

Out[ ]: (1023, 13)
```

Рисунок 3 – Объединение датасета по индексу

Объединенный датасет содержит 13 признаков и 1023 строки. Часть данных удалена на начальном этапе исследования из X_nup.

Столбец Угол нашивки, град показал всего два уникальных значения. Это категориальный признак. Применим бинарное кодирование: приведём столбец к значениям 0 и 1.

Если не приводить категориальный признак к числовому со значениями 0 и 1, это может повлиять на корректность вычисления описательной статистики, выбора и обучения модели, написания нейросети, так как большинство алгоритмов машинного обучения не могут работать с категориальными признаками напрямую.

Проведем разведочный анализ данных. Получим информацию объединенного датасета:

```
df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1023 entries, 0 to 1022
Data columns (total 13 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Соотношение матрица-наполнитель    1023 non-null   float64
 1   Плотность, кг/м3                   1023 non-null   float64
 2   модуль упругости, ГПа              1023 non-null   float64
 3   Количество отвердителя, м.%        1023 non-null   float64
 4   Содержание эпоксидных групп,%_2    1023 non-null   float64
 5   Температура вспышки, С_2           1023 non-null   float64
 6   Поверхностная плотность, г/м2      1023 non-null   float64
 7   Модуль упругости при растяжении, ГПа  1023 non-null   float64
 8   Прочность при растяжении, МПа      1023 non-null   float64
 9   Потребление смолы, г/м2            1023 non-null   float64
 10  Угол нашивки, град                 1023 non-null   int64
 11  Шаг нашивки                        1023 non-null   float64
 12  Плотность нашивки                  1023 non-null   float64
dtypes: float64(12), int64(1)
memory usage: 111.9 KB
```

Рисунок 4 – Информация о типах данных датасета

Разведочный анализ данных на первом этапе показывает, что пропусков и дубликатов нет.

Затем требуется получить описательную статистику данных, нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами (боксплоты), попарные графики рассеяния точек.

Изучить датасет на предмет выбросов в данных и применить выбранный способ удаления выбросов. Получить информацию очищенного от шумов и выбросов датасета.

Провести предобработку (препроцессинг) данных. Выбрать метод препроцессинга и получить графики распределения данных до и после предобработки.

Обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении. При построении модели необходимо 30% данных оставить на тестирование модели, на остальных происходит обучение моделей. При построении моделей провести поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10.

Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель.

Разработать приложение с графическим интерфейсом или интерфейсом командной строки, которое будет выдавать прогноз, полученный в задании 4 или 5 (один или два прогноза, на выбор учащегося).

Оценить точность модели на тренировочном и тестовом датасете.

Создать репозиторий в GitHub / GitLab и разместить там код исследования. Оформить файл README.

1.2 Описание используемых методов

В данном разделе приводится краткое описание методов, которые предполагается использовать для решения поставленной задачи.

Основным инструментом для анализа и построения моделей является язык программирования «Python» и его библиотеки.

Рассмотрим особенности, а также достоинства и недостатки каждого из используемых в данной работе методов.

Данная задача в рамках классификации категорий машинного обучения относится к машинному обучению с учителем и традиционно это задача регрессии. Цель любого алгоритма обучения с учителем — определить функцию потерь и минимизировать её, поэтому для наилучшего решения в процессе исследования были применены следующие методы:

– линейная регрессия;
– Лассо;
– метод опорных векторов;
– случайный лес;
– дерево решений;
– k-ближайших соседей;
– градиентный бустинг.

1.2.1 Метод линейной регрессии

Метод линейной регрессии (Linear regression) – это один из наиболее простых и популярных методов машинного обучения. Он используется для решения задач регрессии, когда требуется предсказать (определить) значение непрерывной переменной на основе значений одной или нескольких независимых переменных.

Особенности метода линейной регрессии:

– позволяет указывать наилучший способ предсказания изменений в зависимых переменных, выраженных через связанные независимые переменные;
– может работать с как с одной, так и с несколькими независимыми переменными;
– может использоваться для описания сложных систем в виде уравнений.

Достоинства метода линейной регрессии:

– метод хорошо интерпретируем и позволяет получать интерпретируемые результаты;

– может использоваться для прогнозирования будущих значений;

– обучение модели линейной регрессии не требует много вычислительных ресурсов;

– может использоваться для определения влияющих на переменную параметров.

Недостатки метода линейной регрессии:

– метод линейной регрессии является линейным, и поэтому не подходит для задач, в которых связи между переменными не являются линейными;

– чувствителен к наличию выбросов в данных, что может привести к неверным результатам;

– не способен учитывать естественные факторы, влияющие на переменные в реальной жизни, например, погоду;

– не подходит для задач, где данные не являются нормально распределенными.

1.2.2 Метод Лассо

LASSO (Least Absolute Shrinkage and Selection Operator) – это метод регуляризации, который применяется в машинном обучении для уменьшения размерности факторного пространства и выбора наиболее важных признаков.

Особенности метода Lasso:

– данный метод используется для избегания переобучения. Переобучение - это явление, при котором модель слишком хорошо подстраивается под обучающие данные, в результате чего ее точность на новых данных резко снижается;

– позволяет решать задачи, в которых число признаков велико или когда имеются признаки, несущественные для решения задачи;

– Lasso является способом регуляризации, который устраняет проблему мультиколлинеарности, когда признаки в данных сильно коррелируют между собой.

Достоинства метода Лассо:

– это мощный инструмент для выбора наиболее значимых признаков в данных, что может улучшить точность и скорость работы модели;

– уменьшает смещение (bias) модели, что приводит к более точным результатам прогнозирования;

– позволяет эффективно управлять сложностью модели путем настройки гиперпараметров.

– можно использовать для решения задач регрессии и классификации, в какой бы области не применялся.

Недостатки метода Лассо:

– может удалить релевантные признаки из набора данных, что приведет к ухудшению качества модели. Однако, этот недостаток можно уменьшить, используя более сложные методы, такие как Elastic Net;

– может работать хуже, если зависимость между признаками не линейная;

– требует настройки параметров, что может быть трудно для новичков в области машинного обучения;

– если используется большое количество переменных, то модель может стать слишком сложной и трудной для интерпретации.

1.2.3 Метод опорных векторов

Метод опорных векторов (Support Vector Regression, SVM) – это один из самых популярных методов машинного обучения, который используется для решения проблем классификации и регрессии. SVM работает с линейными и нелинейными данными и ищет гиперплоскость, которая разделяет обучающие примеры на различные классы с максимальным зазором. Позволяет решать задачи

классификации и регрессии как для линейных, так и для нелинейных данных и может обучаться на наборах данных с высокой размерностью.

Достоинства метода опорных векторов:

– имеет высокую точность классификации;

– способен обработать данные с небольшим объемом обучающих примеров и сохранить высокую точность классификации;

– не зависит от размерности данных, что делает его достаточно универсальным методом машинного обучения;

– использование функций в SVM позволяет работать с нелинейными данными и достигать хорошей точности на многих задачах.

Недостатки метода опорных векторов:

– SVM требует большого объема вычислительных ресурсов для решения задач с большими объемами данных;

– может давать неправильные результаты при наличии шума или выбросов в обучающих данных;

– параметры модели сложно интерпретировать.

Недостатки метода, такие как зависимость от объема данных, наличия шума или выбросов, являются хорошо изученными и могут быть учтены при разработке методов для работы с SVM.

1.2.4 Метод случайного леса

Метод случайный лес (Random Forest) – это алгоритм машинного обучения, который применяется для решения задач классификации, регрессии и кластеризации. Этот метод основан на использовании ансамблей деревьев решений.

Особенности метода случайный лес:

– состоит из множества деревьев решений, которые обучаются на подмножествах независимых переменных;

– каждое дерево в случайном лесу строится путем выбора случайного набора признаков из исходного набора данных;

– конечный результат предсказания получается путем усреднения предсказаний от всех деревьев.

Достоинства метода случайный лес:

– является одним из наиболее точных алгоритмов машинного обучения;

– может эффективно работать с большими наборами данных, имеющими высокую размерность;

– снижает вероятность переобучения и улучшает обобщающую способность модели;

– с помощью алгоритма можно проводить интерпретацию важности признаков, что позволяет лучше понять, какие переменные влияют на результат прогнозирования.

Недостатки метода случайный лес:

– обучение случайного леса занимает значительное время, особенно при использовании больших наборов данных и большого числа деревьев;

– интерпретация результатов может быть сложной, особенно если у деревьев различные глубины;

– несмотря на то, что метод может быть эффективен при работе с данными большого объема, он может показывать не лучшую производительность в задачах с небольшим набором данных.

1.2.5 Дерево решений

Метод Дерево решений (Decision Tree) – это алгоритм машинного обучения в задачах классификации и регрессии. Он основан на создании дерева, где каждый узел представляет условие, которое позволяет разделить данные на более чистые подгруппы. Особенностями метода является то, что алгоритм дерева решений строит дерево, которое позволяет определить важность каждого признака для создания модели. Дерево может содержать различные типы узлов, включая категориальные, числовые и бинарные.

Достоинства метода Дерево решений:

– является относительно простым алгоритмом машинного обучения, который легко интерпретировать и объяснить;

– алгоритм быстро работает при обучении и предсказании на небольших объемах данных;

– может использоваться для обработки данных с различными типами признаков, включая категориальные, числовые и бинарные.

Недостатки метода Дерево решений:

– деревья решений могут иметь тенденцию к переобучению, когда они слишком точно настраиваются на обучающую выборку и не могут обобщать знания для новых данных;

– возможно создание сложных деревьев, которые могут быть трудными для интерпретации и приводить к плохой обобщающей способности модели;

– могут быть чувствительны к шуму в данных, что может приводить к созданию неверных ветвей дерева.

В целом, метод Дерево решений является эффективным алгоритмом машинного обучения на небольших объемах данных. Однако, необходимо учитывать недостатки метода.

1.2.6 Метод k-ближайших соседей

Метод k-ближайших соседей (k Nearest Neighbours, KNN) – это один из самых простых алгоритмов машинного обучения для классификации и регрессии. Метод основан на сравнении расстояний между объектами и выборе тех объектов, которые находятся ближе всего к новому объекту.

Особенностями метода KNN является то, что алгоритм использует метрические методы для измерения расстояния между объектами. Количество соседей (k) может быть настраиваемым параметром, который влияет на точность и скорость работы алгоритма.

Достоинства метода k-ближайших соседей:

– является простым и легко интерпретируемым алгоритмом;

– обладает хорошей обобщающей способностью и может использоваться для классификации различных типов данных, таких как номинальные, бинарные, ординальные и числовые;

– не требует предварительной обработки данных и может использоваться для необработанных данных;

– допускает настройку нескольких параметров;

– находит лучшее решение из возможных;

– имеет низкую чувствительность к выбросам.

Недостатки метода k-ближайших соседей:

– полностью перебирает всю обучающую выборку при распознавании;

– не создаёт правил и не обобщает предыдущий опыт;

– невозможно сказать, на каком основании строятся ответы;

– сложно выбрать близость метрики;

– замедляется с ростом объёма данных, что может приводить к проблеме медленной скорости работы алгоритма;

– имеет высокую зависимость результатов классификации от выбранной метрики.

1.2.7 Метод градиентного бустинга

Метод градиентный бустинг (GradientBoostingRegressor) – это ансамбль деревьев решений, обученный с использованием градиентного бустинга. В основе данного алгоритма лежит итеративное обучение деревьев решений с целью минимизировать функцию потерь. Основная идея градиентного бустинга: строятся последовательно несколько базовых классификаторов, каждый из которых как можно лучше компенсирует недостатки предыдущих. Финальный классификатор является линейной композицией этих базовых классификаторов. Метод способен достигать высокой точности при обработке больших объемов данных, а также может обрабатывать как числовые, так и категориальные признаки.

Достоинства метода GradientBoosting:

– позволяет достичь высокой точности в задаче регрессии, которая превосходит точность других методов машинного обучения;

– хорошо работает с выбросами в данных, что уменьшает их влияние на качество модели;

– может обрабатывать большие объемы данных и выбирать релевантные признаки для их анализа;

– модель обучения GradientBoosting имеет возможность использовать как числовые, так и категориальные признаки.

Недостатки метода GradientBoosting:

– в случае большого объема данных построение модели может работать очень медленно;

– может переобучаться, если увеличить количество деревьев, поэтому необходимо тщательно выбирать критерии остановки;

– необходимо тщательно подготовить данные, чтобы гарантировать получение более точной модели;

– слабее и менее гибок чем нейронные сети.

1.2.8 Полносвязные нейронные сети

Полносвязные нейронные сети (Fully Connected Neural Networks) являются одним из наиболее широко используемых типов нейронных сетей для решения Pipeline задач. Они основаны на прямых связях (между всеми нейронами) между всеми слоями нейронов. Основные особенности: каждый нейрон в слое связан с каждым нейроном в следующем слое, что обеспечивает полную сеть со связями между всеми нейронами; способны эффективно обрабатывать сложную информацию, включая изображения, звук и текст; также они могут использоваться для задач классификации, регрессии и кластеризации. Достоинства:

– могут быть очень мощными инструментами машинного обучения и могут достигать высокой точности, если подобрать оптимальные параметры и архитектуру сети;

– устойчивы к шумам и перегрузкам в данных, что обеспечивает высокую точность в задачах классификации и регрессии;

– могут обучаться без привязки к конкретным характеристикам данных, что делает их более гибкими в использовании.

Недостатки:

– их построение может быть сложным процессом, требующим большого количества вычислительных ресурсов и времени на обучение модели;

– выбор оптимальных параметров и архитектуры сети может быть сложной задачей для новичков в области машинного обучения;

– метод может страдать от проблемы переобучения (overfitting), когда модель обучается на тренировочных данных слишком точно и не учитывает варианты, которых нет в обучающей выборке.

Для построения модели используется также библиотека «TensorFlow». Она используется для создания, обучения и применения различных моделей машинного обучения, таких как нейронные сети, рекуррентные нейронные сети, сверточные нейронные сети. Инструмент позволяет создавать сложные модели машинного обучения, включая глубокие нейронные сети, с помощью высокоуровневых и более низкоуровневых интерфейсов.

1.2.9 Перекрестная проверка. Поиск гиперпараметров по сетке. Метрики качества моделей

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 выборке, затем выполняется расчет метрик качества на валидационной. В результате мы получаем средние метрики качества для всех валидационных выборок. Перекрестную проверку реализует функция «cross_validate» из библиотеки «Scikit-learn».

«Sklearn» поддерживает все этапы классического процесса машинного обучения: от предобработки и создания модели, до оценки ее качества и использования для прогнозирования новых данных, содержит множество инструментов и алгоритмов для анализа данных, машинного обучения и статистического моделирования. Она включает в себя более двадцати модулей, в том числе для регрессии (regression), предобработки данных (preprocessing).

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. Перекрестная проверка уже встроена в этот класс.

В данной работе применены несколько различных метрик качества, применимых для регрессии.

«R2» или коэффициент детерминации измеряет долю дисперсии, объясненную моделью, в общей дисперсии целевой переменной. Отрицательные значение коэффициента детерминации означают плохую объясняющую способность модели.

«RMSE (Root Mean Squared Error)» или корень из средней квадратичной ошибки принимает значения в тех же единицах, что и целевая переменная. Метрика использует возведение в квадрат, поэтому хорошо обнаруживает грубые ошибки.

«MAE (Mean Absolute Error)» или средняя абсолютная ошибка так же принимает значения в тех же единицах, как и целевая переменная. При этом ошибка прогноза не смещена ни в большую, ни в меньшую сторону и она не зависит от масштаба данных, что делает её удобной метрикой для оценки качества модели.

«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 измерения целевой переменной.

Метрики принимают положительные значения. Но отображены в работе со знаком «-». Так корректно отработает выделение цветом лучших моделей и эти метрики надо минимизировать.

1.3 Разведочный анализ данных

Рассмотрим описательную статистику данных, результат в таблице 1.

Таблица 1 – Описательная статистика признаков датасета

Признак	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

Прежде чем передать данные в работу моделей машинного обучения, необходимо обработать и очистить их. Очевидно, что необработанные данные могут содержать искажения и пропущенные значения, что способно привести к крайне неверным результатам по итогам моделирования. Но безосновательно удалять что-либо тоже неправильно. Именно поэтому сначала набор данных надо изучить.

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, ранее мы его привели к количественному.

Инструменты разведочного анализа данных помогают исследовать и понимать данные, выявлять взаимосвязи между переменными и определять возможные выбросы или аномалии в данных, которые могут пригодиться в дальнейшей работе с данными.

Гистограммы распределения переменных – это графическое представление, которое показывает, как часто появляются разные значения переменных в наборе данных. Это может помочь определить тип распределения переменной и понять, как она влияет на другие переменные.

```
c = 1 #__инициализация plot counter в цикле for
plt.figure(figsize = (29,20))
for col in df.columns:
  plt.subplot(5, 5, c)
  sns.histplot(data = df[col], kde=True, color = "#2F4F4F")
  sns.despine()
  plt.ylabel(None)
  c += 1
```

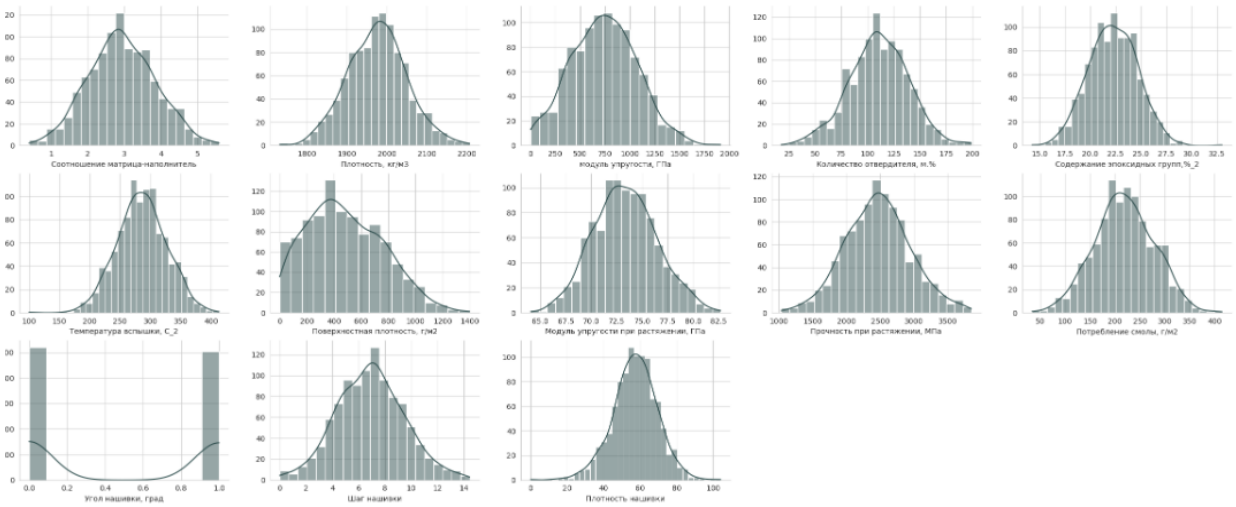

Рисунок 5 – Гистограммы распределения каждой переменной

Все признаки, кроме "Угол нашивки, град" стремятся к нормальному распределению. Гистограммы, исключая для угла нашивки, у которого два значения, показывают явные выбросы.

Диаграммы ящика с усами (Box Plot) – это график, который показывает сводную статистику данных, включая медиану, распределение данных по квартилям и выбросы. Это помогает визуально определить наличие аномалий и понять различия между разными группами в данных.

На рисунке 6 ящики с усами показали явные выбросы, которые заметны более, чем на гистограммах распределения.

```
c = 1 # инициализация plot counter в цикле for
plt.figure(figsize = (27,20))
sns.set_style('whitegrid')

for col in df.columns:
  plt.subplot(5, 5, c)
  sns.boxplot (data = df[col], orient="h", color = "#B0C4DE")
  plt.xlabel(col)
  sns.despine()
  c += 1
```

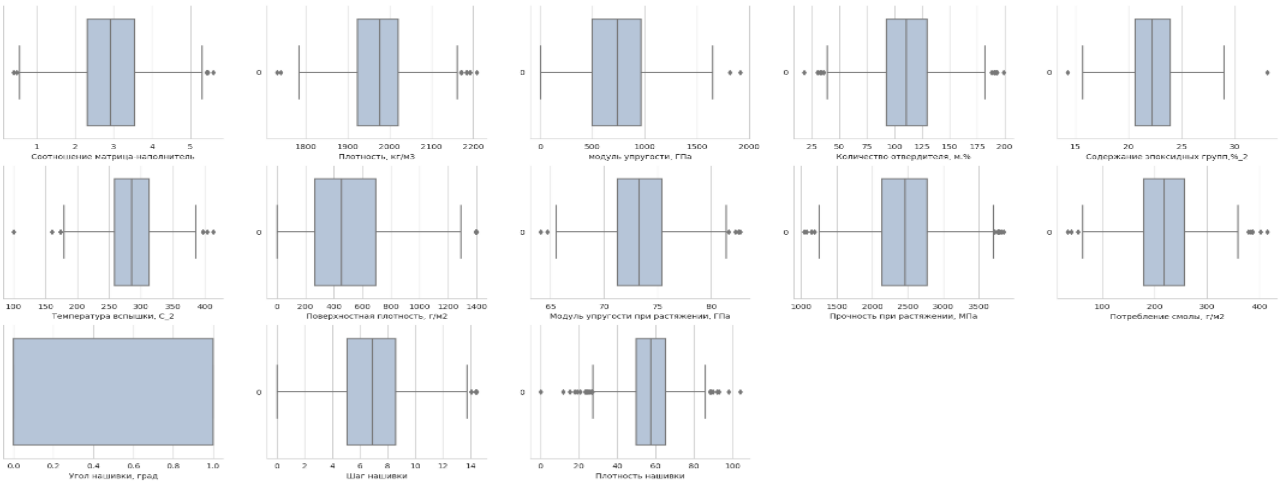

Рисунок 6 – Боксплоты («ящики с усами») распределения данных

Попарные графики рассеяния точек (Scatter Plot) представляет каждую точку как пару значений переменных. Это помогает определить существуют ли связи между двумя переменными и если да, то какая эта связь. Построим матрицу попарного рассеяния точек с выделением значений Угол нашивки.

На попарных графиках распределения на рисунке 7 не видно корреляции между признаками. Единственная зависимость, которую можно отметить – это меньшая дисперсия значений Плотности нашивки при 90 градусов Угла нашивки, по сравнению со значением 0 градусов.

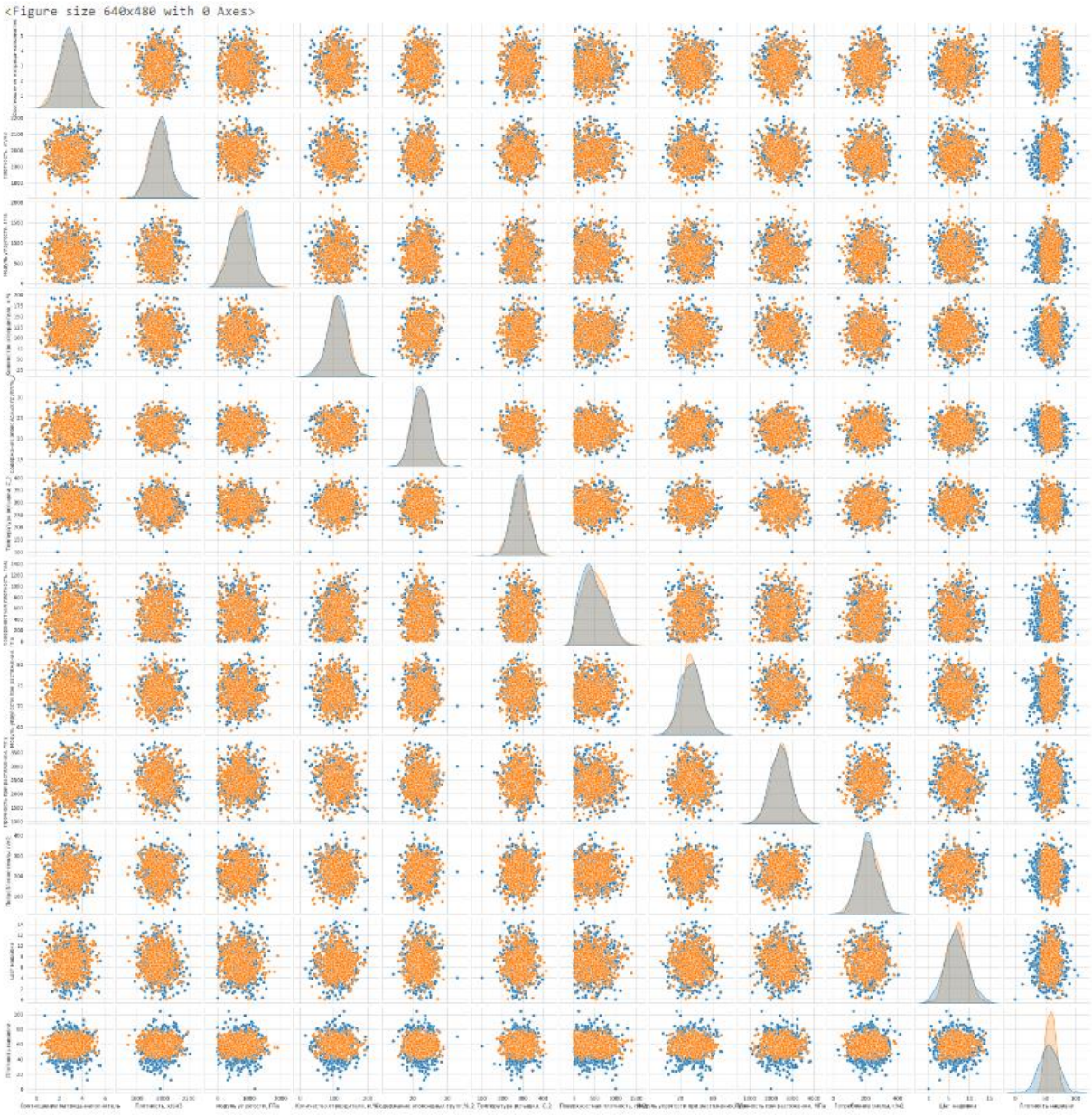

Рисунок 7 – Попарные графики рассеяния точек

Тепловая карта (Heatmap) – это график, который используется для визуализации матрицы корреляции между переменными. Она показывает, какие переменные сильно коррелируют (положительно или отрицательно) и какие нет.

Максимальная корреляция между плотностью нашивки и углом нашивки 0.11, значит нет зависимости между этими данными. Корреляция между всеми параметрами очень близка к 0, корреляционные связи между переменными не наблюдаются (рисунок 8).

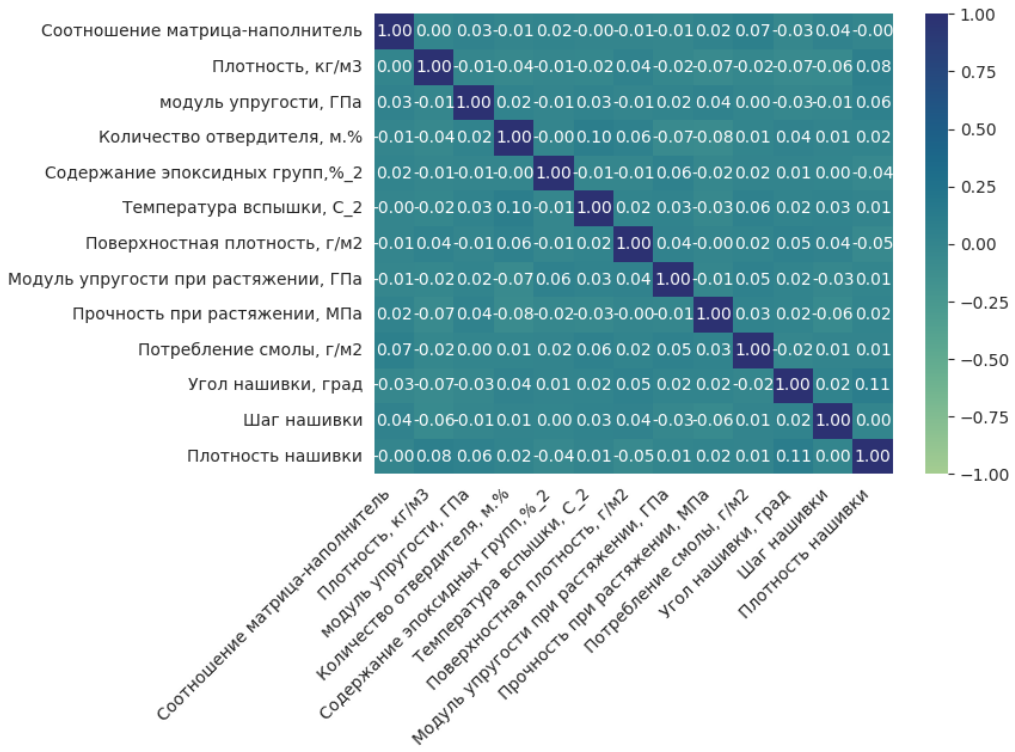

Рисунок 8 – Матрица корреляции между переменными

Тепловая карта показывает практически отсутствие корреляции между признаками и целевыми переменными. Максимальная корреляция между плотностью нашивки и углом нашивки 0.11, значит нет зависимости между этими данными. Корреляция между всеми параметрами очень близка к 0, корреляционные связи между переменными не наблюдаются.

1.3.1 Поиск и исключение выбросов

Для расчета обнаружения выбросов этих данных мы будем использовать методы трех сигм и межквартильного расстояния. Данные, значительно отличающиеся от выборки, будут полностью удалены.

Метод 3-х сигм найдено выбросов: «24». Метод межквартильных расстояний найдено выбросов: «93».

Метод трёх сигм не гарантирует, что все выбросы будут удалены, так как есть вероятность того, что выбросы могут находиться внутри границ трёх стандартных отклонений.

Большинство признаков распределены нормально. Всё же, во избежание потери достоверности и полноты, применим метод трёх сигм.

```
In [ ]:
# Удалить выбросы методом 3-х сигм
outliers = pd.DataFrame(index=df.index)
for column in df:
    zscore = (df[column] - df[column].mean()) / df[column].std()
    outliers[column] = (zscore.abs() > 3)
df = df[outliers.sum(axis=1)==0]
df.shape

Out[ ]: (1000, 13)
```

Рисунок 9 – Результат удаления выбросов методом трёх сигм

Теперь, после очистки данных от выбросов, датасет содержит тринадцать признаков и тысячу значений для каждого.

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	на
mean	2.936299	1975.402478	738.675486	110.821904	22.235549	285.957299	479.855825	73.318178	2464.864198	218.254011	0.496	6
50%	2.908811	1977.321002	741.148111	110.652620	22.221462	285.853960	450.869535	73.230375	2456.394188	218.697660	0.000	6

Рисунок 10 – Средние и медианные значения признаков

Средние и медианные значения признаков близки, что говорит о пригодности данных для построения моделей предсказания.

Повторный разведочный анализ очищенных данных так же показывает отсутствие пропусков и дубликатов, тип значения численный.

```
In [ ]:
# проверка на пропуски
# датасет чистый
df.isnull().sum()

Out[ ]: 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

In [ ]:
#поиск дубликатов в датасете
df.duplicated().sum()

Out[ ]: 0
```

Рисунок 11 – Повторный разведочный анализ

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1000.0000	2.9363	0.9088	0.3894	2.3193	2.9088	3.5539	5.5917
Плотность, кг/м3	1000.0000	1975.4025	72.9537	1784.4822	1923.6287	1977.3210	2021.1595	2192.7388
модуль упругости, ГПа	1000.0000	738.6755	327.5452	2.4369	500.7730	741.1481	961.6508	1649.4157
Количество отвердителя, м.%	1000.0000	110.8219	27.8696	29.9561	92.5238	110.6526	129.8531	192.8517
Содержание эпоксидных групп,%_2	1000.0000	22.2355	2.3842	15.6959	20.5832	22.2215	23.9749	28.9551
Температура вспышки, С_2	1000.0000	285.9573	40.2315	173.4849	259.1038	285.8540	313.0291	403.6529
Поверхностная плотность, г/м2	1000.0000	479.8558	277.7086	0.6037	266.9787	450.8695	691.5284	1291.3401
Модуль упругости при растяжении, ГПа	1000.0000	73.3182	3.1138	64.0541	71.2488	73.2304	75.3266	82.6821
Прочность при растяжении, МПа	1000.0000	2464.8642	485.0154	1036.8566	2134.5359	2456.3942	2760.1630	3848.4367
Потребление смолы, г/м2	1000.0000	218.2540	58.9450	41.0483	179.8122	218.6977	257.4748	386.9034
Угол нашивки, град	1000.0000	0.4960	0.5002	0.0000	0.0000	0.0000	1.0000	1.0000
Шаг нашивки	1000.0000	6.9106	2.5577	0.0376	5.1058	6.9222	8.5888	14.4405
Плотность нашивки	1000.0000	57.2763	11.8458	20.5716	49.8930	57.4720	64.9309	92.9635

Рисунок 12 – Описательная статистика очищенного набора данных

2 Практическая часть

2.1 Предобработка данных

В данном разделе приводится графики распределения для каждого признака до и после нормализации.

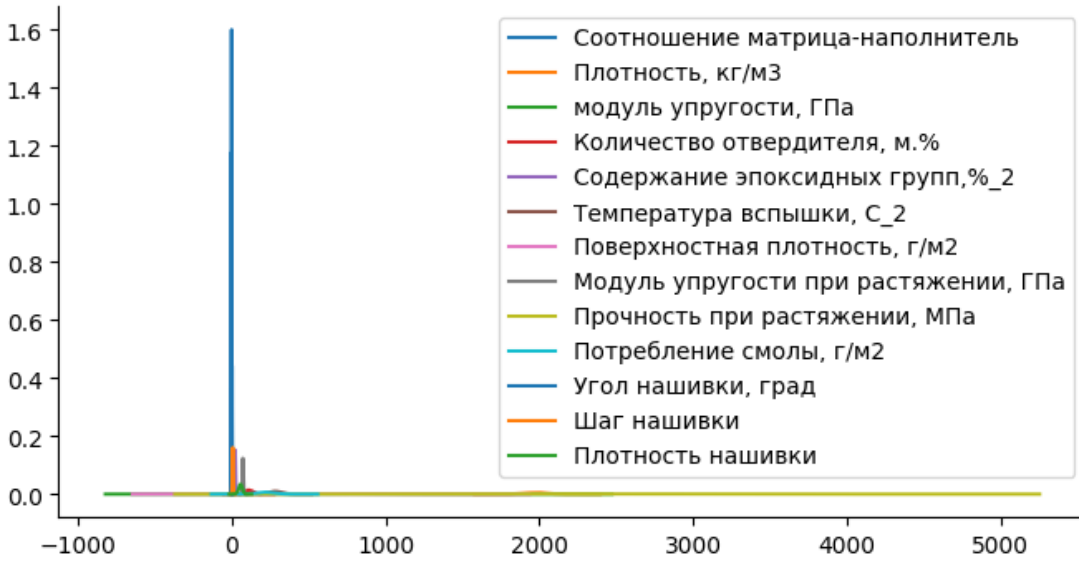

Рисунок 13 – Оценка плотности ядра

Отношения между переменными показывают, что данные лежат в разных диапазонах. Данные могут вести себя плохо, если отдельные функции не выглядят более или менее как стандартные нормально распределенные.

Нормализация масштабирует каждую входную переменную до диапазона от нуля до единицы – диапазон значений, где мы имеем наибольшую точность.

Стандартизация методом «StandardScaler» преобразует данные таким образом, чтобы среднее значение каждой переменной было равно «0», а стандартное отклонение равнялось «1», что соответствует стандартному нормальному распределению.

Препроцессинг данных выполнен в целях демонстрации полученных знаний в обучении. Имеем один подготовленный датасет и три задания, три разных прогноза. В таком случае к стандартизации будем приводить копии исходного датасета, предварительно разделенные на тренировочную и тестовую выборки.

На рисунке 14 показано сравнение распределения исходных и отшкалированных данных.

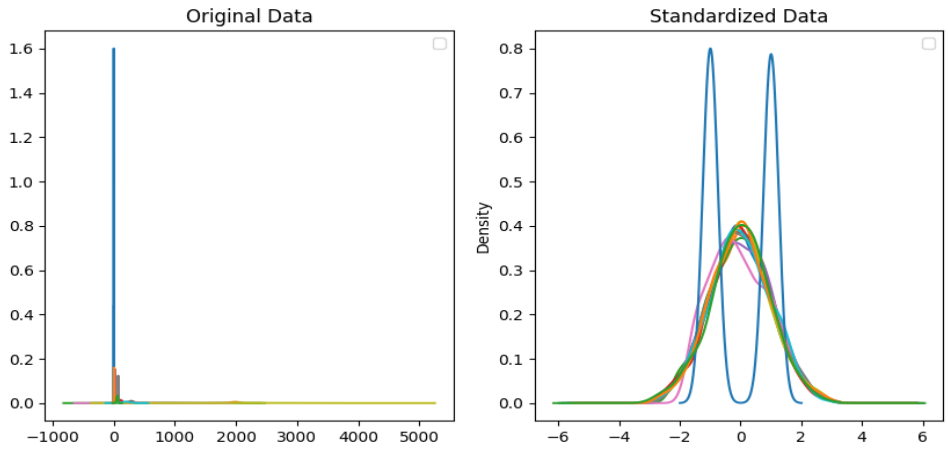

Рисунок 14 – Сравнение распределения данных до и после стандартизации

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	на
mean	0.00	-0.00	0.00	-0.00	-0.00	0.0	0.0	0.00	-0.00	-0.00	0.00	
50%	-0.03	0.03	0.01	-0.01	-0.01	-0.0	-0.1	-0.03	-0.02	0.01	-0.99	

Рисунок 15 – Средние и медианные значения стандартизованных данных

Рисунок 15 показывает почти отсутствие различий между средними и медианными значениями для всех признаков.

2.2 Разработка и обучение модели

В данной части приводится список моделей, которые будут использоваться для прогноза модуля упругости при растяжении и прочности при растяжении.

Используются несколько метрик для сравнения моделей. Для статистической устойчивости результатов подбираются коэффициенты кросс-валидации.

При построении моделей проводится поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно

«10». Для определения нижней границы качества модели наиболее удобно использовать метрику «MAE», она показывает среднюю абсолютную ошибку прогноза модели по всем наблюдениям.

В работе мы использовали «DummyRegressor» с параметром «mean», параметр предсказывает среднее значение целевой переменной, которое было вычислено на тренировочных данных.

При построении модели необходимо 30% данных оставить на тестирование модели, на остальных происходит обучение моделей.

Целевая переменная для первой модели: «Модуль упругости при растяжении, ГПа». Целевая переменная препроцессингу не подвергается. Препроцессинг выполнен с помощью метода «StandardScaler». Далее рассматривается для каждой модели описательная статистика до и после нормализации.

	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	-2.653	-2.628	-2.308	-2.900	-2.686	-2.791	-1.716	-3.039	-0.989	-2.679	-3.121
max	2.754	2.968	2.795	2.969	2.783	2.851	2.967	2.865	1.011	2.971	3.022
mean	0.000	0.000	0.000	-0.000	0.000	0.000	-0.000	-0.000	0.000	0.000	-0.000
std	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001

Рисунок 16 – Пример описательной статистики тренировочных данных

Таблица 2 – Сравнение качества моделей перекрестной проверки

Модель	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.029222	-3.026484	-2.435078	-0.033249	-7.706106
LinearRegression	-0.043235	-3.042794	-2.446686	-0.033413	-7.713844
Lasso	-0.029222	-3.026484	-2.435078	-0.033249	-7.706106
DecisionTree	-1.375941	-4.526988	-3.629757	-0.049523	-11.931894
RandomForest	-0.109272	-3.138957	-2.507197	-0.034235	-8.035295
SVR	-0.099814	-3.126372	-2.501624	-0.034180	-8.107903
KNeighbors	-0.323109	-3.424050	-2.731698	-0.037274	-8.982734

Метрики работы выбранных моделей с гиперпараметрами по умолчанию, полученные с помощью перекрестной проверки на тестовом множестве, приведены в таблице 2.

Метрики показали, что ни одна из моделей не точна для решения поставленной задачи предсказания. Лучше других проявили себя линейные модели, хуже отработало дерево решений.

Подберём параметры кросс-валидации для некоторых моделей.

Результат выведем в сравнительную таблицу.

Таблица 3 – Сравнение моделей по параметрам кросс-валидации для модуля упругости при растяжении

Модели с лучшими параметрами	R2	RMSE	MAE	MAPE	max_error
Lasso(alpha=1)	-0.02922	-3.02648	-2.43507	-0.03324	-7.7061
SVR (C=0.025, kernel='sigmoid')	-0.02969	-3.02724	-2.42950	-0.03316	-7.6676
SVR (C=0.1, kernel='sigmoid')	-0.03331	-3.03247	-2.42928	-0.03317	-7.6146
RandomForestRegressor (bootstrap=False, criterion='absolute_error', max_depth=2, max_features=1, n_estimators=10, random_state=42)	-0.04273	-3.04394	-2.44754	-0.03339	-7.68630
DecisionTreeRegressor (criterion='absolute_error', max_depth=1, max_features=4, random_state=38, splitter='random')	-0.0245	-3.0199	-2.4248	-0.0331	-7.7355

Модель с Деревьями решений при подборе гиперпараметров научилась и показала метрики лучше, чем при параметрах по умолчанию. В целом, метрики близки к базовой модели. По таблице 3 мы видим, что при кропотливом подборе гиперпараметров можно улучшить модель предсказания.

Теперь сделаем визуализацию для сравнения работы с данными по лучшей и базовой модели предсказания.

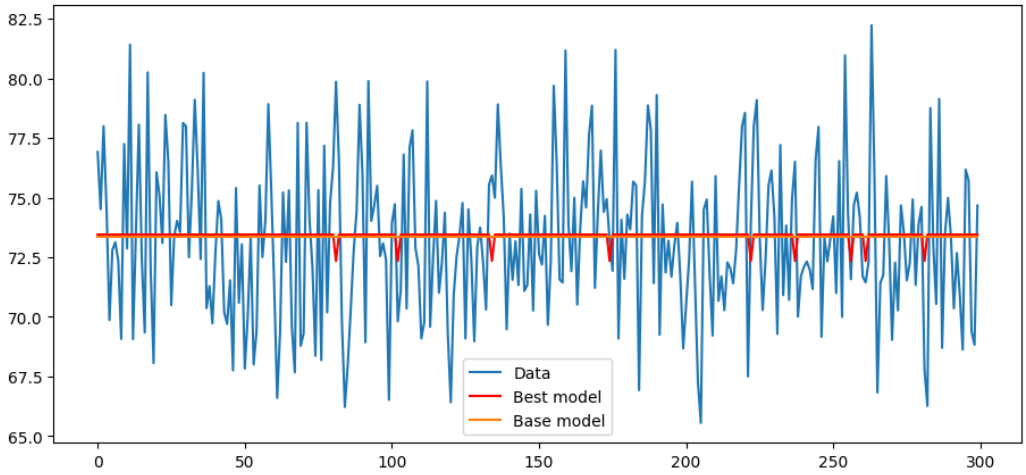

Рисунок 17 – Визуализация качества базовой и лучшей модели

Целевая переменная для второй модели: «Прочность при растяжении, МПа». Целевая переменная препроцессингу не подвергается. Препроцессинг выполнен с помощью метода «StandardScaler».

	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	-2.750	-2.577	-2.265	-2.970	-2.760	-2.777	-1.694	-3.049	-1.014	-2.676	-3.116
max	2.890	3.004	2.792	3.007	2.848	2.810	2.758	2.958	0.986	2.890	2.786
mean	0.000	-0.000	0.000	-0.000	-0.000	0.000	-0.000	0.000	0.000	0.000	0.000
std	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001

Рисунок 18 – Описательная статистика тренировочной выборки после этапа препроцессинга

Сравним качество моделей предсказания для прочности при растяжении при перекрестной проверке.

Таблица 4 – Сравнение качества моделей перекрестной проверки

Модель	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.022646	-485.766658	-380.550456	-0.168633	-1267.370957
Lasso	-0.038292	-488.696718	-385.230924	-0.170347	-1272.027386
DecisionTree Regressor	-1.154126	-699.859479	-553.085798	-0.240743	-1905.650070

Продолжение таблицы 4

Модель	R2	RMSE	MAE	MAPE	max_error
SVR	-0.022372	-485.698613	-380.578720	-0.168202	-1268.151424
GradientBoosting Regressor	-0.114121	-506.607014	-403.137241	-0.177888	-1328.962027

Качество модели, полученной методом опорных векторов, показывает метрики лучше базовой модели. Хуже проявили себя метрики модели градиентный бустинг.

Подберём параметры кросс-валидации для моделей и сравним с результатами по перекрестной проверке. Результат выведем в сравнительную таблицу.

Таблица 5 – Сравнение по кросс-валидации для прочности при растяжении

Модель	R2	RMSE	MAE	MAPE	max_error
DecisionTreeRegressor (max_depth=5, max_features=7, random_state=44, splitter='random')	-0.028920	-486.55490	-379.19597	-0.168115	-1289.26078
DecisionTreeRegressor (max_depth=3, max_features=3, random_state=49, splitter='random')	-0.026133	-486.36524	-379.84327	-0.168349	-1253.77565
GradientBoostingRegressor (max_depth=6, max_features=1, n_estimators=1, random_state=3128)	-0.018646	-484.82637	-379.50171	-0.168228	-1269.36172
GradientBoostingRegressor (loss='absolute_error', max_depth=5, max_features=1, n_estimators=1, random_state=3128)	-0.021328	-485.48386	-380.46830	-0.168037	-1268.91142

Модель Градиентный бустинг при подборе гиперпараметров показала метрики лучше, чем при параметрах по умолчанию, что видно в таблице 5. По сравнению с другими моделями, градиентный бустинг будет более качественной и точнее предсказывающей моделью.

Теперь сделаем визуализацию для сравнения работы с данными по лучшей и базовой модели предсказания.

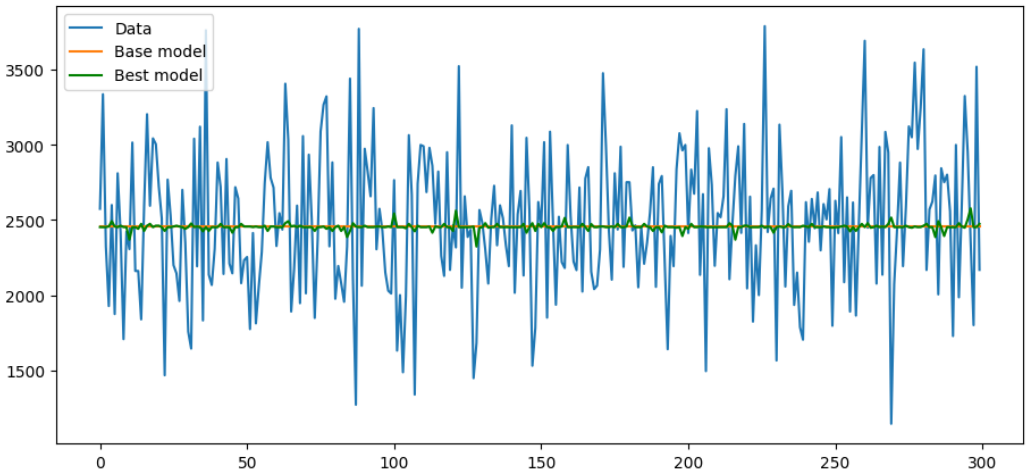

Рисунок 19 – Визуализация качества базовой и лучшей модели для прогноза прочности при растяжении

2.3 Тестирование модели

Сравним качество лучшей модели Дерево решений на тренировочных и тестовых данных для модуля упругости при растяжении.

Таблица 6 – Оценка качества модели Дерево решений

Данные	R2	RMSE	MAE	MAPE	max_error
Модуль упругости, train	0.006730	-3.023163	-2.410568	-0.032922	-9.400634
Модуль упругости, test	-0.008675	-3.300236	-2.665111	-0.036518	-8.782905

Как видно в таблице 6, на данных тестовых лучшая модель для модуля упругости при растяжении показывает метрики хуже. Качество модели требует улучшений и не гарантирует точность предсказания.

Сравним качество лучшей модели Градиентный бустинг на тренировочных и тестовых данных для прочности при растяжении, результат также сведем в итоговой таблице по метрикам качества.

Таблица 7 – Оценка качества модели Градиентный бустинг

Данные	R2	RMSE	MAE	MAPE	max_error
Прочность при растяжении, train	0.033631	-477.8012	-374.10805	-0.165670	-1415.5576
Прочность при растяжении, test	0.00914	-483.0871	-389.41809	-0.16625	-1372.9174

Для модели Градиентный бустинг хоть и близкий к 0, но положительным получился коэффициент детерминации. «MAE» на тестовом множестве незначительно больше, чем на тренировочном, максимальная ошибка чуть меньше на тестовом множестве. Значит, какую-то зависимость модель обнаружила и обучилась, а не просто подстроилась к данным.

В итоге не удалось получить пригодной точной модели для предсказания модуля упругости при растяжении и прочности при растяжении. Возможно, отражается присутствие выбросов данных. Подбор гиперпараметров занимает длительное время, но способен значительно улучшить качество модели.

2.4 Нейронная сеть для рекомендаций соотношения матрица-наполнитель

Описывается выбранная архитектура нейронной сети и ее результаты.

В качестве метрики для оценки моделей возьмём среднюю абсолютную ошибку («mae»), также на более удачных моделях сравним метрики на тренировочной и тестовой выборках, коэффициент детерминации (R2).

Строим, компилируем полносвязную нейронную сеть. Выбор сделан в пользу библиотеки «TensorFlow». Всего было написано шесть моделей с различными архитектурами: разными количествами нейронов в слоях, количеством слоев, параметрами активации, количеством итераций, разделений на тренировочную и валидационную выборку. Представим наиболее удачную модель. Применены функции активации такие, как «ReLU», «Leaky ReLU», сигмоид-функция.

Также написаны нейросети с ранней остановкой для борьбы с переобучением, помимо дропаут-слоев.

Архитектура и параметры обучения выбранной модели, как наиболее точной, представлены на рисунке 20.

```
# модель 3 полносвязной нейронной сети с более сложной архитектурой, тренируем
modelMF_3 = tf.keras.Sequential([
                keras.layers.Dense(32,  input_dim=12, activation = 'sigmoid'),
                keras.layers.Dropout(0.5),
                keras.layers.Dense(64, activation = 'sigmoid'),
                keras.layers.Dropout(0.5),
                keras.layers.Dense(32, activation = 'sigmoid'),
                keras.layers.Dense(16, activation = 'sigmoid'),
                keras.layers.Dense(1, activation = 'linear')])
modelMF_3.compile(optimizer='adam', loss='mse', metrics=['mae'])
modelMF_3.summary()
history = modelMF_3.fit(X_train, y_train,
                    epochs=150,
                    validation_split=0.1,
                    verbose=2)
plt.plot(history.history['mae'], label = 'Точность train')
plt.plot(history.history['val_mae'], label = 'Точность test')
plt.xlabel = ('Epochs')
plt.ylabel = ('MAE')
plt.legend()
plt.show()
```

```
Model: "sequential_120"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_453 (Dense)           (None, 32)                416

 dropout_119 (Dropout)       (None, 32)                0

 dense_454 (Dense)           (None, 64)                2112

 dropout_120 (Dropout)       (None, 64)                0

 dense_455 (Dense)           (None, 32)                2080

 dense_456 (Dense)           (None, 16)                528

 dense_457 (Dense)           (None, 1)                 17

=================================================================
Total params: 5,153
Trainable params: 5,153
Non-trainable params: 0
```

Рисунок 20 – Создание и архитектура третьей модели в TensorFlow

Добавление слоев «Dropout» помогает избежать переобучения модели.

Оптимизатор «Adam», который является адаптивным методом стохастической оптимизации, позволяет эффективно и быстро обучать модель; «loss=mse»

здесь функция потерь, которая используется для обучения модели. В данном случае используется среднеквадратическая ошибка (MSE), которая является самой распространенной функцией потерь для регрессионных задач; «metrics=['mae']» метрика, используемая для измерения качества модели в процессе обучения. В данном случае используется средняя абсолютная ошибка (MAE), которая является простой метрикой, измеряющей среднее абсолютное отклонение прогнозов от фактических значений.

«Sigmoid» (сигмоид) – это функция активации, широко используемая в нейронных сетях. Она преобразует любое значение в диапазоне от 0 до 1.

Результат поведения модели на тренировочных и тестовых выборках отображен на графике рисунка 21. Целевая переменная не подвергалась преобразованию (нормализации).

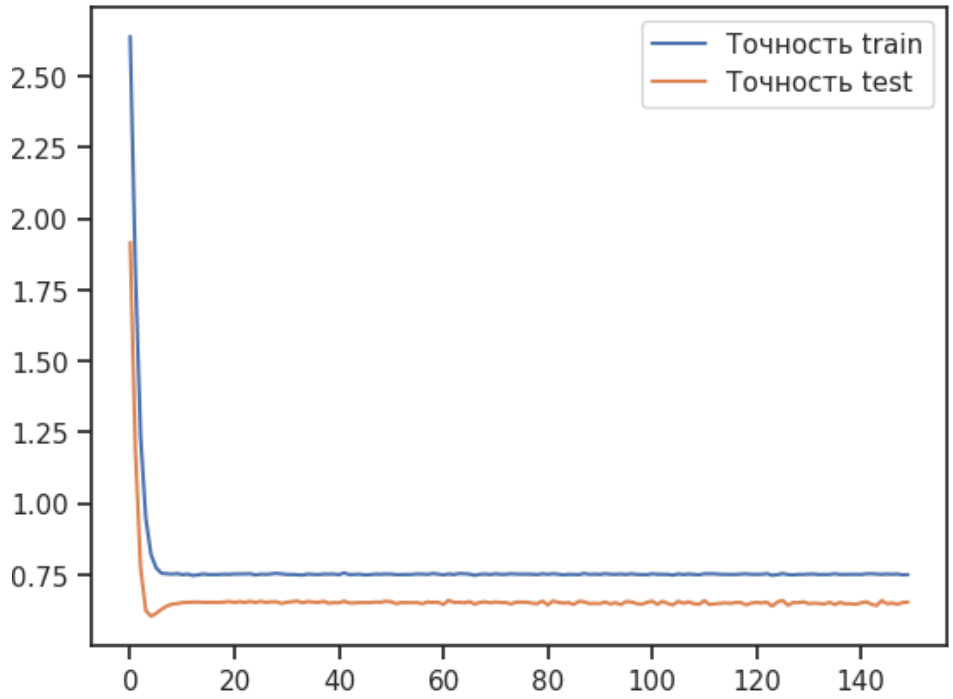

Рисунок 21 – Результат обучения. Визуализация потерь модели

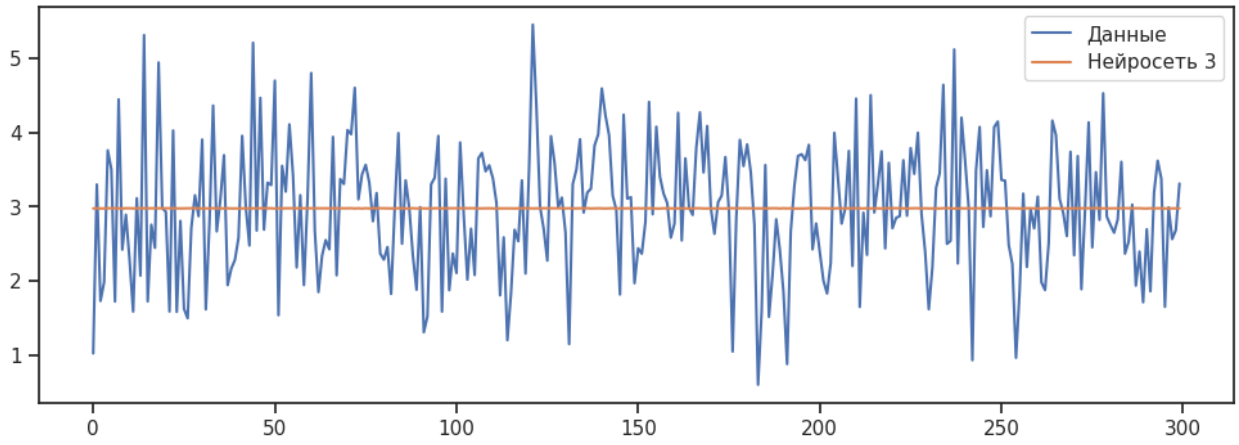

Рисунок 22 – Визуализация работы нейросети-3 на тестовых данных

Как видно по рисунку 22 модель нашла нелинейную зависимость данных.

Выведем в таблицу сравнение метрик качества нейросети на тренировочных и тестовых выборках.

Таблица 8 – Качество нейросети на тренировочной и тестовой выборках

	R2	RMSE	MAE	MAPE	max_error
Соотношение матрица-наполнитель,тренировочный	-0.002767	-0.923670	-0.741881	-0.330315	-2.613384
Соотношение матрица-наполнитель, тестовый	-0.000423	-0.874308	-0.695222	-0.296718	-2.479797

На тестовой выборке модель показала метрики чуть лучше

Значение метрики R2 близко к нулю, что говорит о том, что модель не может объяснить разброс данных признака «Соотношение матрица-наполнитель».

Максимальная ошибка составляет 2.479797, что также указывает на значительную неточность в предсказаниях модели. Также стоит учитывать, что данные для признака имеют довольно большой разброс, что может затруднять обучение модели и снижать ее точность.

В целом, текущая модель не показывает хорошей точности предсказаний.

Также приведем итоговую таблицу сравнения нейросетей по средней абсолютной ошибке.

Таблица 9 – Сравнение моделей по абсолютной средней ошибке

Версия нейросети	MAE
Нейросеть 1	1.969044
Нейросеть 2	0.695281
Нейросеть 3	0.695293
Нейросеть 4	1.965212
Нейросеть 5	0.702981
Нейросеть 6	0.704050

Визуализируем распределение ошибки гистограммой (рисунок 23).

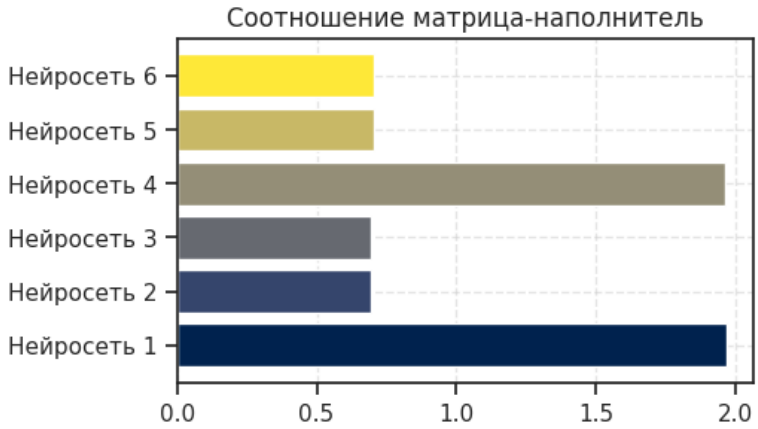

Рисунок 23 – Гистограмма распределения ошибки

Для улучшения результатов необходимо провести дополнительные исследования, возможно, изменить выбранный алгоритм моделирования. Также возможно повлияло наличие выбросов в данных.

Для интеграции в приложение была сохранена выбранная модель нейросети для соотношения матрица-наполнитель.

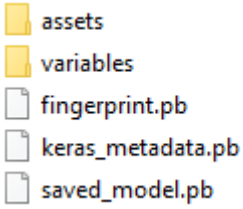

Рисунок 24 – Структура и файлы сохранения из TensorFlow

2.5 Разработка приложения

Разработать веб-приложение для рекомендации матрица-наполнитель, интегрировать нейросеть из предыдущей задачи. Инструменты: «Python», среда разработки «PyCharm Community», фреймворк «Flask» и шаблонизатор «Jinja».

В приложении А описан функционал приложения и инструкция для пользователя.

2.6 Создание удаленного репозитория

Для размещения ноутбуков с кодом, в которых проводилось исследование, сохраненной модели, Flask-приложения создан удаленный репозиторий на «GitHub». Ссылки кликабельны.

Адрес страницы:

https://github.com/maaliskuussa

Адрес репозитория:

https://github.com/maaliskuussa/DS_course_BMSTU2023

Commits:

https://github.com/maaliskuussa/DS_course_BMSTU2023/commits?author=maaliskuussa

Информация о содержании репозитория находится в файле README.md.

Заключение

В процессе исследования изучены теоретические основы и методы решения поставленной задачи. Спрогнозированы по входным параметрам ряд конечных свойств получаемых композиционных материалов при используемых признаках. Получен набор реальных экспериментальных данных.

Обучено нескольких моделей прогнозов «Модуль упругости при растяжении, ГПа» и «Прочность при растяжении, МПа». «Scikit-learn» поддерживает все этапы классического машинного обучения: от предобработки данных и создания модели, до оценки ее качества и использования для прогнозирования новых данных.

Написана нейронная сеть с помощью библиотеки «TensorFlow» для «соотношения матрица-наполнитель». Библиотека позволяет создавать сложные модели машинного обучения, включая глубокие нейронные сети, с помощью высокоуровневых и более низкоуровневых интерфейсов. В библиотеке также есть множество инструментов анализа данных.

В результате работы ожидаемая точность всех моделей не была получена, модели и нейросети нашли слабые закономерности данных.

Полученный результат работы является модельным шаблоном для создания реальной модели прогнозирования. Сказывается присутствие выбросов.

Построение моделей и нейросетей должно быть в контексте имеющихся данных. На этом основании будет лучше понята стратегия подбора коэффициентов кросс-валидации, построение архитектуры нейросети.

Возможный выход из ситуации и более высокая точность прогнозирования будет получена при добавлении в датасет синтетических данных. Это искусственные данные, имитирующие реальные наблюдения и используемые для подготовки моделей машинного обучения, когда получение реальных данных в ходе эксперимента представляет собой сложность.

Также требуется более углубленная практика в машинном обучении и погружение в предметную область.

Список использованных источников

1. Бизли, Д. Python. Подробный справочник: учебное пособие. – Пер. с англ. – СПб.: Символ-Плюс, 2010. – 864 с., ил.
2. Гафаров, Ф.М., Галимянов А.Ф. Искусственные нейронные сети и приложения: учеб. пособие /Ф.М. Гафаров, А.Ф. Галимянов. – Казань: Издательство Казанского университета, 2018. – 121 с.
3. 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
4. Документация по библиотеке numpy: – Режим доступа: https://numpy.org/doc/1.22/user/index.html#user. (дата обращения: 3.03.2023).
5. Документация по библиотеке pandas: – Режим доступа: https://pandas.pydata.org/docs/user_guide/index.html#user-guide. (дата обращения: 4.04.2023).
6. Документация по библиотеке scikit-learn: – Режим доступа: https://scikit-learn.org/stable/user_guide.html. (дата обращения: 5.04.2023).
7. Документация по библиотеке seaborn: – Режим доступа: https://seaborn.pydata.org/tutorial.html. (дата обращения: 6.04.2023).
8. Документация по библиотеке Tensorflow: – Режим доступа: https://www.tensorflow.org/overview (дата обращения: 10.04.2023).
9. Документация по языку программирования python: – Режим доступа: https://docs.python.org/3.8/index.html. (дата обращения: 2.02.2023).
10. Иванов Д.А., Ситников А.И., Шляпин С.Д – Композиционные материалы: учебное пособие для вузов, 2019. 13 с.
11. Шитиков В.К., Мастицкий С.Э. (2017) Классификация, регрессия и другие алгоритмы Data Mining с использованием R. 351 с. − Электронная книга, адрес доступа: https://github.com/ranalytics/data-mining
12. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.. : Пер. с англ. - М. : Издательский дом ‘‘Вильямс’’, 2007. - 1408 с.

13. Box G.E.P., Jenkins G.M., Reinsel G.C., Ljung G.M. Time Series Analysis: Forecasting and Control - 5th Edition. — Wiley, 2015. — 712 p. — ISBN: 978-1-118-67502-1.

14. Nisbet R., Elder J., Miner G. Handbook of Statistical Analysis and Data Mining Applications. - Academic Press, 2009. — 864 p. — ISBN: 0123747651

15. В. Ш. Берикашвили, С. П.Оськин Статистическая обработка данных, планирование эксперимента и случайные процессы : учебное пособие для вузов - 2-е изд., испр. и доп. — М. : Юрайт, 2021. — 163 с.

16. Документация по программной системе Deductor. Режим доступа: https://basegroup.ru/deductor/manual (дата обращения: 10.04.2023).

17. Документация по программной системе Loginom. Режим доступа: https://help.loginom.ru/userguide/ (дата обращения: 10.04.2023).

18. Дауни Аллен. Рекомендательные системы на практике / пер. с англ. Д. М. Павлова. – М.: ДМК Пресс, 2020. – 448 с.: ил.

19. 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.

Приложение А

Инструкция и описание пользовательского приложения на веб микрофреймворке Flask. Приложение рекомендует соотношение матрица-наполнитель для новых композиционных материалов на основе полученных данных.

В примере запуск осуществляется из среды разработки «PyCharm».

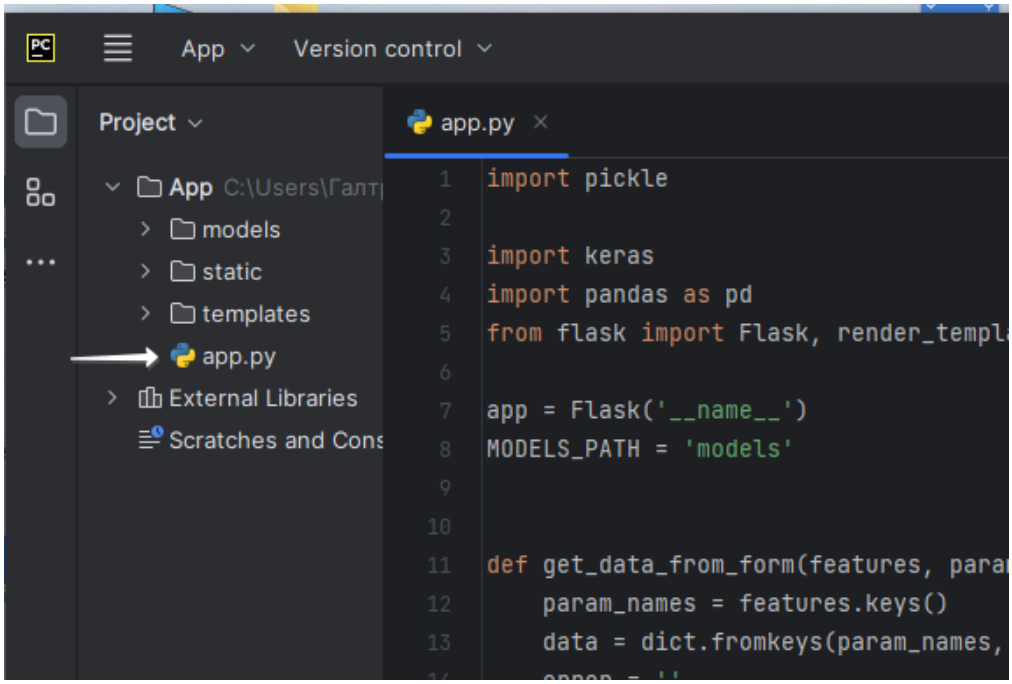

Рисунок 25 – Запуск файла приложения из IDE

Запускается файл «app.py». Далее пользователь открывает командную строку и переходит по адресу. Приложение загружается в браузер.

Рисунок 26 – Запуск приложения на сервер

При запуске приложения пользователь видит приветственное окно с заголовком и ссылку «Получить рекомендации». Проходя по ссылке, пользователь переходит в окно для расчетов.

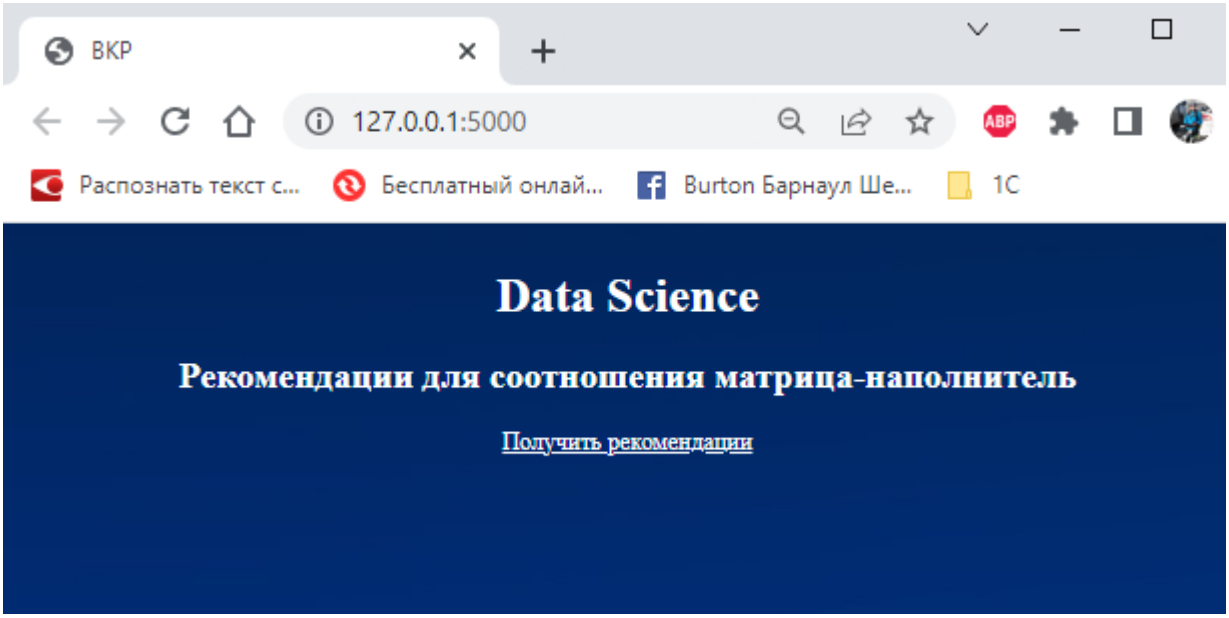

Рисунок 27 – Приветственное окно

Далее пользователь видит окно для ввода числовых данных признаков. Данные для примера взяты из исследуемого набора данных. Заполняются все поля.

Для получения рекомендаций пользователь нажимает «Отправить».

Идёт процесс вычисления на основе интегрированной модели нейросети.

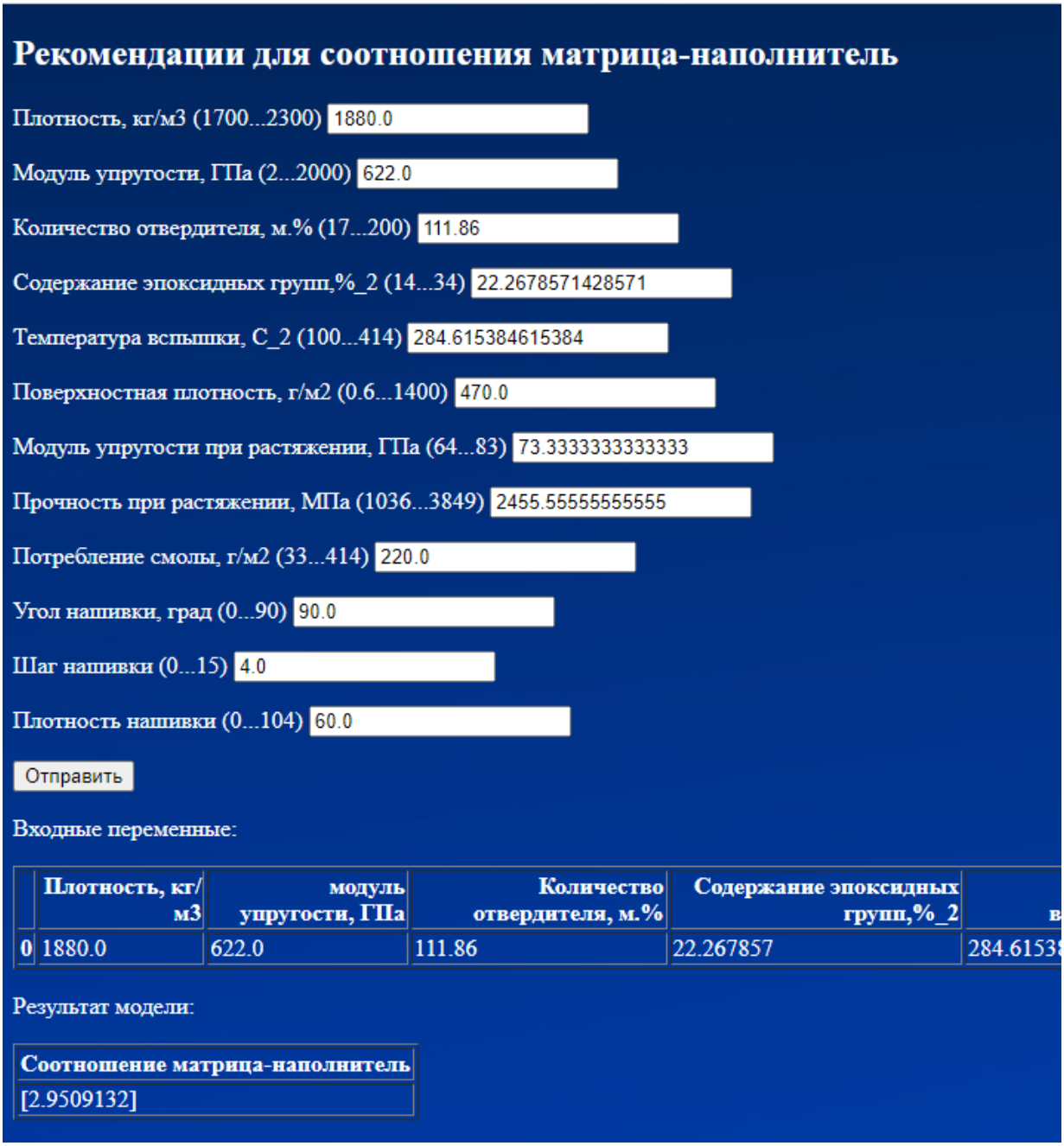

Рисунок 28 – Вывод результата на экран

Если пользователь ввёл не все значения, то приложение вернет предупреждение.

Если пользователь ввел данные вне диапазона значений, то приложение вернет предупреждение с указанием, в каком признаке допущена ошибка.

Рисунок 29 – Предупреждение о пропуске данных

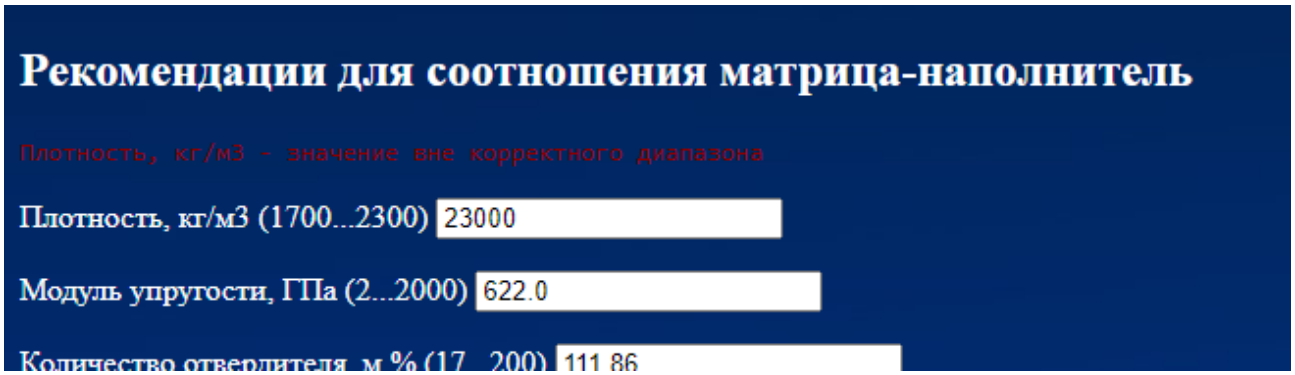

Рисунок 30 – Предупреждение о некорректном вводе значений