

**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

Зацепина Алла Ивановна

Москва, 2022

Содержание

<u>Содержание</u>	Error! Bookmark not defined.
<u>Введение</u>	Error! Bookmark not defined.
<u>1. Аналитическая часть</u>	Error! Bookmark not defined.
<u>1.1. Постановка задачи</u>	Error! Bookmark not defined.
<u>1.2. Описание используемых методов</u>	Error! Bookmark not defined.
<u>1.3. Разведочный анализ данных</u>	Error! Bookmark not defined.
<u>2. Практическая часть</u>	Error! Bookmark not defined.
<u>2.1. Предобработка данных</u>	Error! Bookmark not defined.
<u>2.2. Разработка и обучение модели</u>	Error! Bookmark not defined.
<u>2.3. Тестирование модели</u>	Error! Bookmark not defined.
<u>2.4. Написать нейронную сеть, которая будет рекомендовать соотношение «матрица-наполнитель»</u>	Error! Bookmark not defined.
<u>2.5. Разработка приложения</u>	Error! Bookmark not defined.
<u>2.6. Создание удалённого репозитория и загрузка</u>	Error! Bookmark not defined.
<u>Заключение</u>	Error! Bookmark not defined.
<u>Список используемой литературы и веб ресурсы</u>	Error! Bookmark not defined.

Введение

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

В процессе работы над выпускной квалификационной работой было определено, что необходимо решить классическую задачу регрессии.

Задачи регрессии на практике встречаются довольно часто. Например, предсказание цены объекта недвижимости - классическая регрессионная задача. В таких проблемах атрибутами выступают разные характеристики квартир или домов - площадь, этажность, расположение, расстояние до центра города,

количество комнат, год постройки. В разных наборах данных собрана разная информация и, соответственно, модели тоже должны быть разные.

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

Также в рамках данной работы было разработана нейронная сеть, которая предлагает соотношение матрица – наполнитель.

1. Аналитическая часть

1.1 Постановка задачи

На входе было дано 2 файла: первый файл с параметрами базальтопластика, второй файл с данными нашивок углепластика.

```
#Загрузка первого файла  
df_bp = pd.read_excel(r"/content/drive/MyDrive/X_bp.xlsx")  
#Вывод размерностей  
df_bp.shape
```

(1023, 11)

Рисунок 1 – размерность первого файла

```
#Загрузка второго файла  
df_nup = pd.read_excel(r"/content/drive/MyDrive/X_nup.xlsx")  
#Вывод размерности  
df_nup.shape
```

(1040, 4)

Рисунок 2 – размерность второго файла

Задача состоит в том, чтобы спрогнозировать ряд конечных свойств получаемых композиционных материалов.

Для решения предложенной задачи был составлен план работы, где были определены дальнейшие шаги над данной задачей.

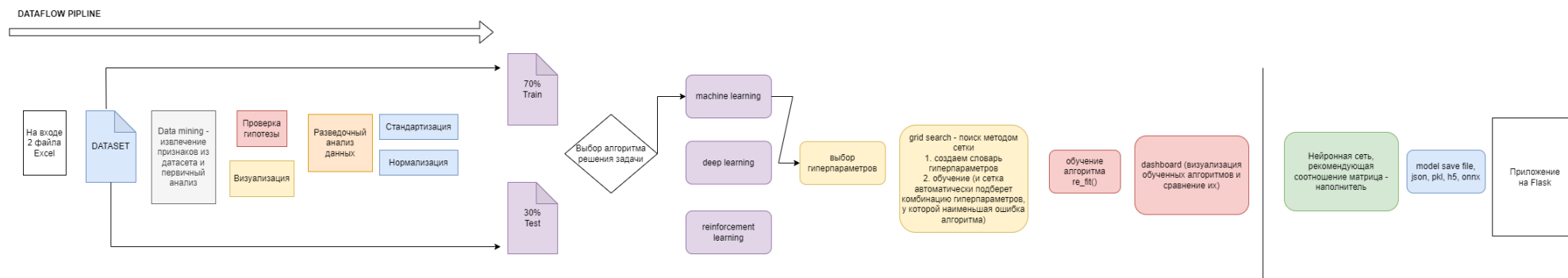


Рисунок 3 - Подробный план работы над решаемой задачей в рамках выпускной квалификационной работы

Во-первых, были объединены оба файла в один. Объединение было выполнено по индексу тип объединения INNER.

	0	1	2	3	4
Соотношение матрица-наполнитель	1.857143	1.857143	1.857143	1.857143	2.771331
Плотность, кг/м3	2030.000000	2030.000000	2030.000000	2030.000000	2030.000000
модуль упругости, ГПа	738.736842	738.736842	738.736842	738.736842	753.000000
Количество отвердителя, м.%	30.000000	50.000000	49.900000	129.000000	111.860000
Содержание эпоксидных групп,%_2	22.267857	23.750000	33.000000	21.250000	22.267857
Температура вспышки, С_2	100.000000	284.615385	284.615385	300.000000	284.615385
Поверхностная плотность, г/м2	210.000000	210.000000	210.000000	210.000000	210.000000
Модуль упругости при растяжении, ГПа	70.000000	70.000000	70.000000	70.000000	70.000000
Прочность при растяжении, МПа	3000.000000	3000.000000	3000.000000	3000.000000	3000.000000
Потребление смолы, г/м2	220.000000	220.000000	220.000000	220.000000	220.000000
Угол нашивки, град	0.000000	0.000000	0.000000	0.000000	0.000000
Шаг нашивки	4.000000	4.000000	4.000000	5.000000	5.000000
Плотность нашивки	57.000000	60.000000	70.000000	47.000000	57.000000

Рисунок 4 – Объединение файлов в один файл

Во-вторых, был проведен разведочный анализ данных. В рамках него были удалены выбросы, проведена проверка на наличие пропущенных значений, проведена проверка на наличие дубликатов.

Далее была построена гистограмма распределения каждой из переменных, диаграмма ящик с усами, попарные графики рассеяния точек.

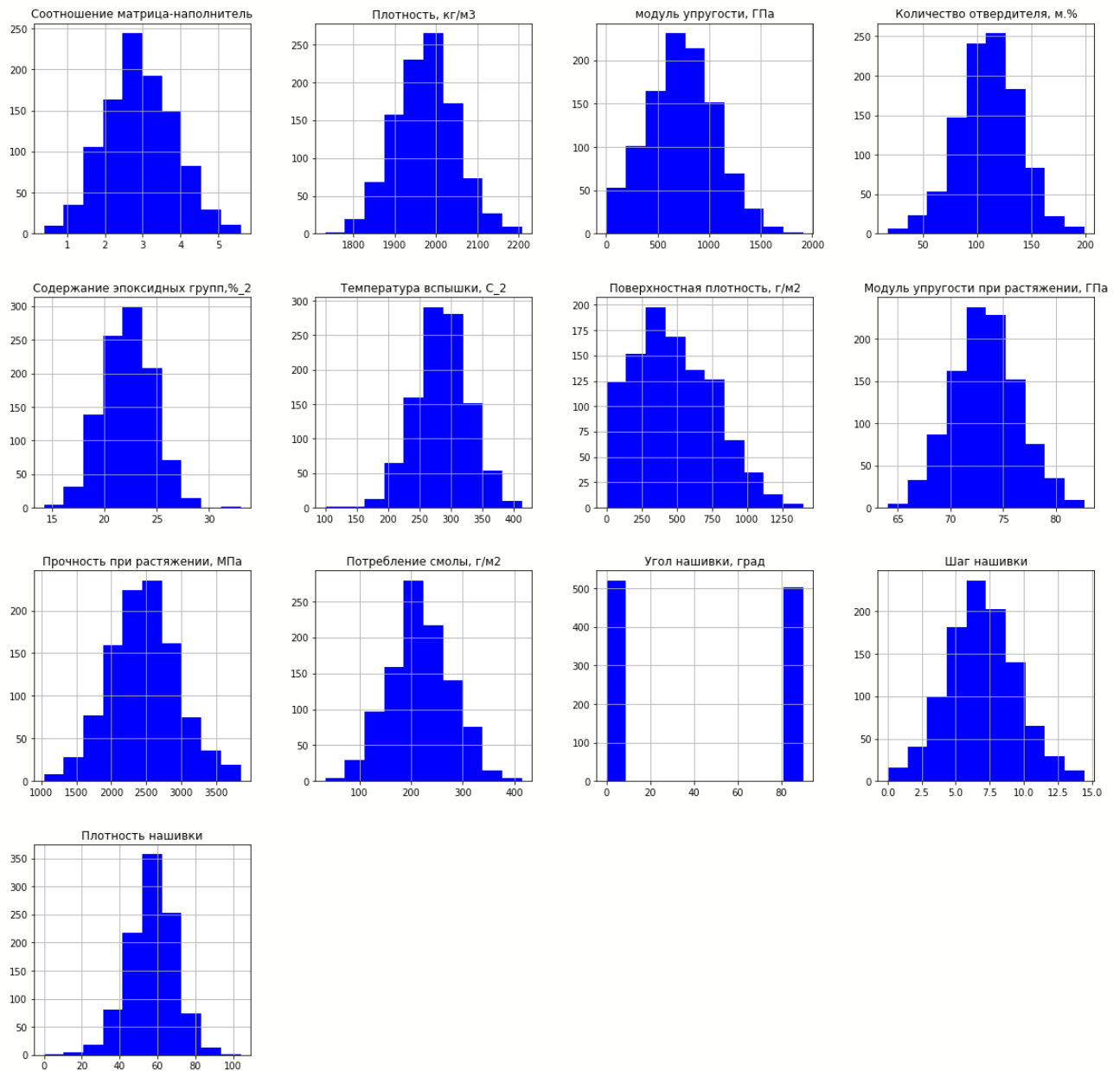


Рисунок 5 - Гистограмма распределения каждой из переменных

1.2 Описание используемых методов

В процессе работы над выпускной квалификационной работой было определено, что необходимо решить классическую задачу регрессии.

В результате были применены все методы, о которых было рассказано на данном курсе (применяемые для решения задач регрессии):

- 1) Метод случайного леса
- 2) Метод опорных векторов
- 3) Метод К ближайших соседей
- 4) Дерево решений
- 5) Метод линейной регрессии
- 6) Метод градиентного бустинга
- 7) Многослойный перцептрон

1) Метод случайного леса

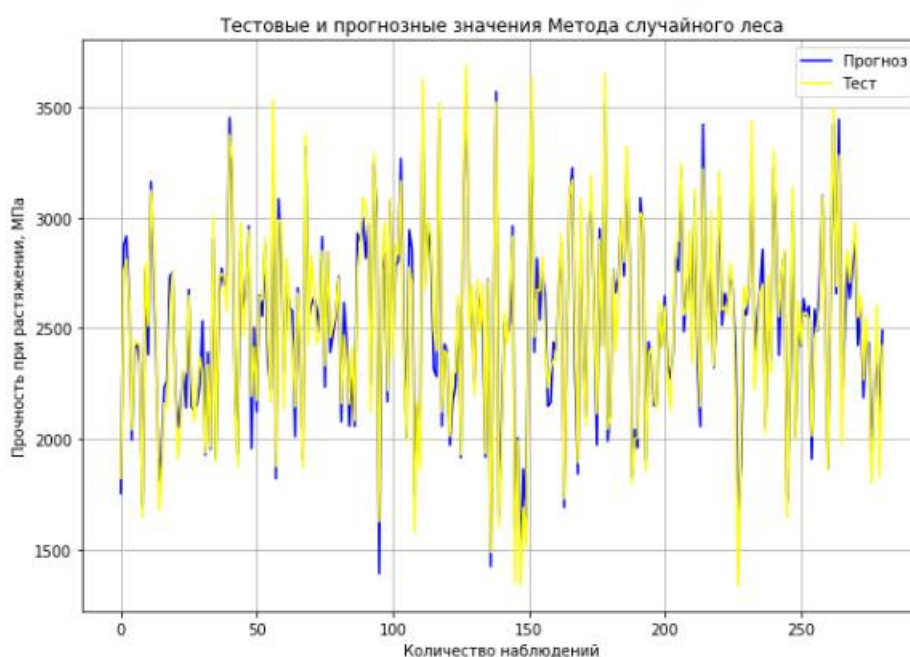


Рисунок 6 – График метода случайного леса для прочности при растяжении, МПа

Случайный лес — это контролируемый алгоритм машинного обучения, который в большинстве случаев дает отличный результат даже без гиперпараметров. Его можно использовать как для задач классификации, так и для регрессии. Он основан на концепции обучения по ансамблю, которая представляет собой процесс группировки нескольких классификаторов для решения сложной проблемы и повышения производительности модели.

Достоинства метода:

- Одним из самых больших преимуществ случайного леса является его универсальность. Его можно использовать как для задач регрессии, так и для задач классификации.
- Он способен обрабатывать большие наборы данных с высокой размерностью.
- Это повышает точность модели и предотвращает проблему переобучения.

Недостатки метода:

- Основное ограничение случайного леса заключается в том, что большое количество деревьев может сделать алгоритм слишком медленным и неэффективным для прогнозов в реальном времени. В целом, эти алгоритмы быстро обучаются, но довольно медленно создают прогнозы после обучения. Для более точного прогноза требуется больше деревьев, что приводит к более медленной модели. В большинстве реальных приложений алгоритм случайного леса достаточно быстр, но, безусловно, могут быть ситуации, когда важна производительность во время выполнения, и предпочтительнее использовать другие подходы.

- Случайный лес — это инструмент прогнозного моделирования, а не описательный инструмент, а это означает, что, если мы ищем описание взаимосвязей в наших данных, другие подходы были бы лучше.

2) Метод опорных векторов

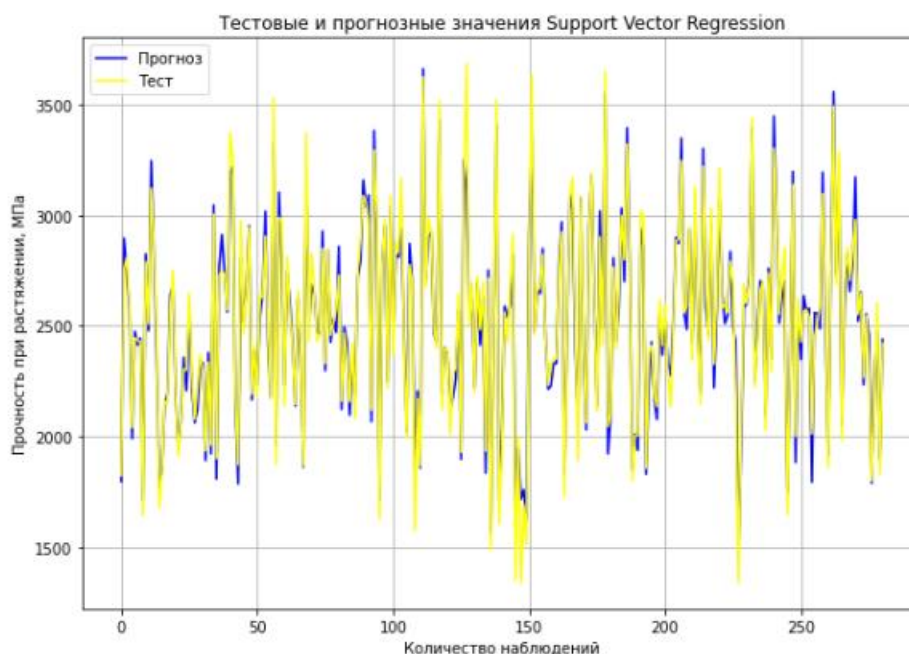


Рисунок 7 – График метода опорных векторов для прочности при растяжении, МПа

Метод опорных векторов — это модель машинного обучения, используемая для задач классификации и регрессии. Многие предпочитают метод опорных векторов, поскольку он обеспечивает заметную корректность при меньшей вычислительной мощности. В основном используется в задачах классификации.

Достоинства метода:

- Метод опорных векторов работает сравнительно хорошо, когда существует понятный предел диссоциации между классами.
- Он более производителен в пространствах с высокой размерностью.

- Он эффективен в тех случаях, когда количество измерений больше, чем количество образцов.

Недостатки метода:

- Алгоритм метода опорных векторов неприемлем для больших наборов данных.
- В случаях, когда количество свойств для каждой точки данных превышает количество обучающих выборок данных, метод опорных векторов будет работать неэффективно.
- Поскольку классификатор опорных векторов работает путем размещения точек данных выше и ниже классифицирующей гиперплоскости, для классификации нет вероятностного уточнения.

3) Метод К ближайших соседей

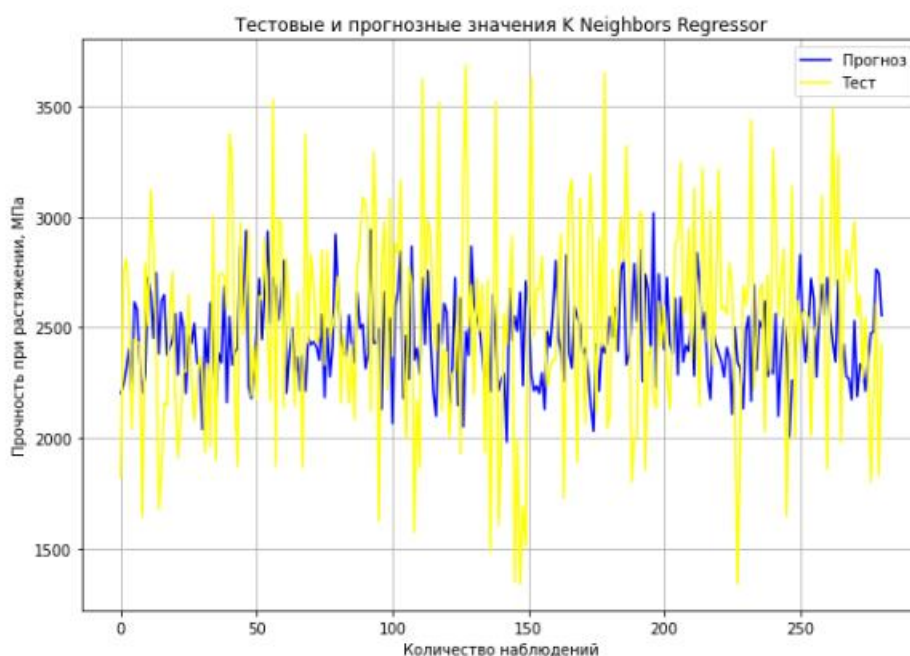


Рисунок 8 – График метода К ближайших соседей для прочности при растяжении, МПа

Метод К-ближайших соседей — это непараметрический метод, используемый для классификации и регрессии. Это один из самых простых

методов машинного обучения. Это ленивая модель обучения с локальным приближением.

Достоинства метода:

- Легкая и простая модель машинного обучения.
- Несколько гиперпараметров для настройки.

Недостатки метода:

- К следует выбирать мудро.
- Большие вычислительные затраты во время выполнения, если размер выборки велик.
- Надлежащее масштабирование должно быть обеспечено для справедливого обращения среди функций.

Метод К ближайших соседей против нейронных сетей:

- Нейронные сети нуждаются в больших данных обучения по сравнению с методом К ближайших соседей для достижения достаточной точности.
- Нейронные сети нуждаются в большом количестве настроек гиперпараметров по сравнению с методом К ближайших соседей.

4) Дерево решений

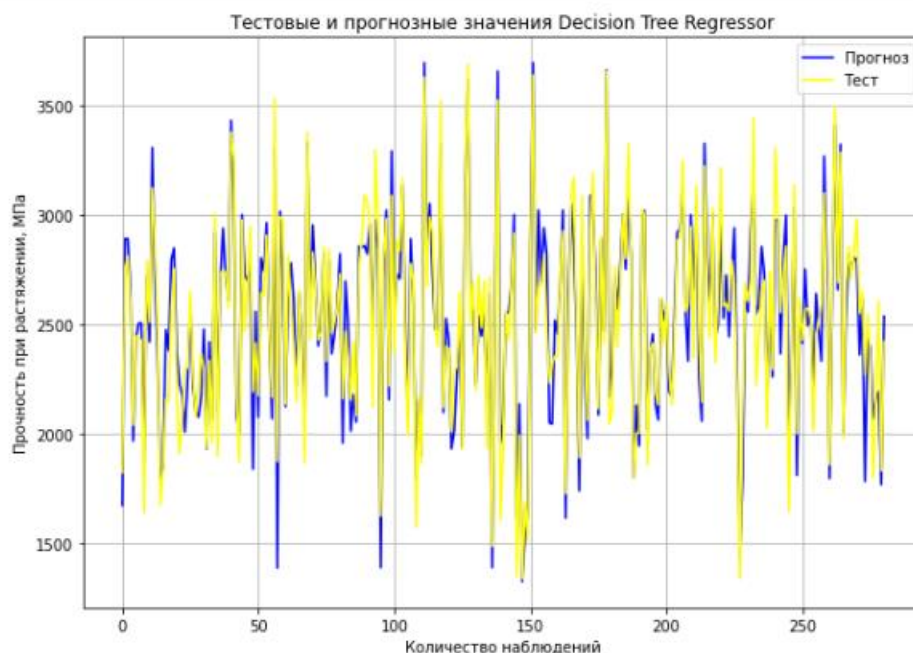


Рисунок 9 – График метода Дерево решений для прочности при растяжении, МПа

Дерево решений — это алгоритм на основе дерева, используемый для решения задач регрессии и классификации. Инвертированное дерево обрамлено, которое разветвляется от корневого узла с однородной вероятностью, до сильно неоднородных листовых узлов, для получения выходных данных. Деревья регрессии используются для зависимой переменной с непрерывными значениями, а деревья классификации используются для зависимой переменной с дискретными значениями.

Достоинства метода:

- Не требуется предварительной обработки данных.
- Нет предположений о распространении данных.
- Эффективно обрабатывает коллинеарность.
- Деревья решений могут дать понятное объяснение по поводу прогноза.

Недостатки метода:

- Возможности переоснащения модели, если мы продолжим строить дерево для достижения высокой чистоты. Для решения этой проблемы можно использовать обрезку дерева решений.
- Склонен к выбросам.
- Дерево может стать очень сложным во время обучения сложных наборов данных.
- Потеря ценной информации при обработке непрерывных переменных.

Дерево решений и случайный лес:

- Случайный лес — это набор деревьев решений, и в качестве прогнозируемого результата выбирается среднее / большинство голосов леса.
- Модель случайного леса будет менее подвержена переоснащению, чем дерево решений, и дает более обобщенное решение.
- Случайный лес более надежен и точен, чем деревья решений.

Дерево решений против метода К ближайших соседей:

- Оба являются непараметрическими методами.
- Дерево решений поддерживает автоматическое взаимодействие объектов, тогда как метод К ближайших соседей не может.
- Дерево решений быстрее из-за дорогостоящего выполнения метода К ближайших соседей в реальном времени.

Дерево решений против нейронной сети:

- Оба находят нелинейные решения и имеют взаимодействие между независимыми переменными.
- Деревья решений лучше, когда в обучающих данных имеется большой набор категориальных значений.

- Деревья решений лучше, чем нейронная сеть, когда сценарий требует объяснения по поводу решения.
- Нейронная сеть превосходит дерево решений, когда имеется достаточно данных для обучения.

Дерево решений против метода опорных векторов:

- Метод опорных векторов использует трюк ядра для решения нелинейных задач, тогда как деревья решений выводят гипер-прямоугольники во входном пространстве для решения проблемы.
- Деревья решений лучше для категориальных данных, и это обеспечивает коллинеарность лучше, чем метод опорных векторов.

5) Метод линейной регрессии

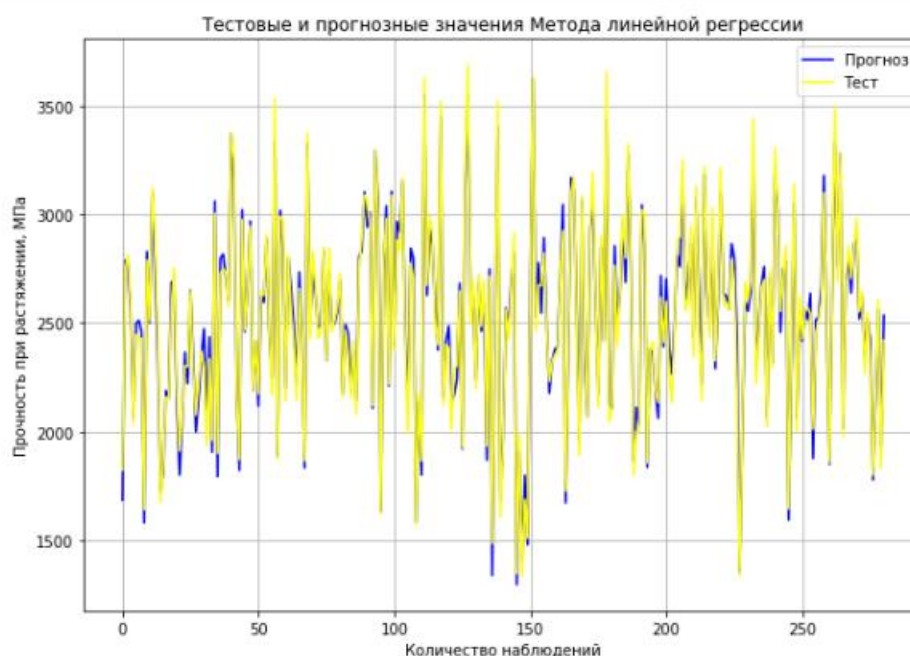


Рисунок 10 – График метода линейной регрессии для прочности при растяжении, МПа

Линейная регрессия — это модель регрессии, то есть она будет принимать функции и прогнозировать непрерывный выход, например: цена акций, зарплата и т. д. Линейная регрессия, как следует из названия, находит решение для линейной кривой для каждой проблемы.

Достоинства метода:

- Простая и простая реализация.
- Космическое комплексное решение.
- Быстрое обучение.
- Значение коэффициентов θ дает предположение о значимости признака

Недостатки метода:

- Применимо, только если решение является линейным. Во многих реальных сценариях это может быть не так.
- Алгоритм предполагает, что входные остатки (ошибка) распределены нормально, но не всегда могут быть выполнены.
- Алгоритм предполагает, что входные функции взаимно независимы (без коллинеарности).

Поскольку линейная регрессия является алгоритмом регрессии, мы сравним его с другими алгоритмами регрессии. Одним из основных отличий линейной регрессии является то, что линейная регрессия может поддерживать только линейные решения. Не существует лучших моделей в машинном обучении, которые превосходят все остальные, а эффективность основана на типе распределения обучающих данных.

Линейная регрессия против дерева решений:

- Деревья решений поддерживают нелинейность, где линейная регрессия поддерживает только линейные решения.
- Когда имеется большое количество объектов с меньшим количеством наборов данных (с низким уровнем шума), линейные регрессии могут превзойти деревья решений / случайные леса. В общем случае деревья решений будут иметь лучшую среднюю точность.
- Для категориальных независимых переменных деревья решений лучше, чем линейная регрессия.
- Деревья решений обрабатывают коллинеарность лучше, чем линейная регрессия.

Линейная регрессия против метода опорных векторов:

- Метод опорных векторов поддерживает как линейные, так и нелинейные решения с использованием трюка ядра.
- Метод опорных векторов лучше справляется с выбросами, чем линейная регрессия.
- Оба хорошо работают, когда тренировочных данных меньше, и имеется большое количество функций.

Линейная регрессия против метода К ближайших соседей:

- Метод К ближайших соседей является непараметрической моделью, тогда как линейная регрессия является параметрической моделью.
- Метод К ближайших соседей работает медленно в реальном времени, так как он должен отслеживать все обучающие данные и находить соседние узлы, тогда как линейная регрессия может легко извлекать выходные данные из настроенных θ коэффициентов.

Линейная регрессия против нейронных сетей:

- Нейронные сети нуждаются в больших данных обучения по сравнению с моделью линейной регрессии, тогда как линейная регрессия может хорошо работать даже с меньшим количеством данных обучения.
- Нейронная сеть будет медленнее по сравнению с линейной регрессией.
- Средняя точность всегда будет лучше с нейронными сетями.

6) Метод градиентного бустинга

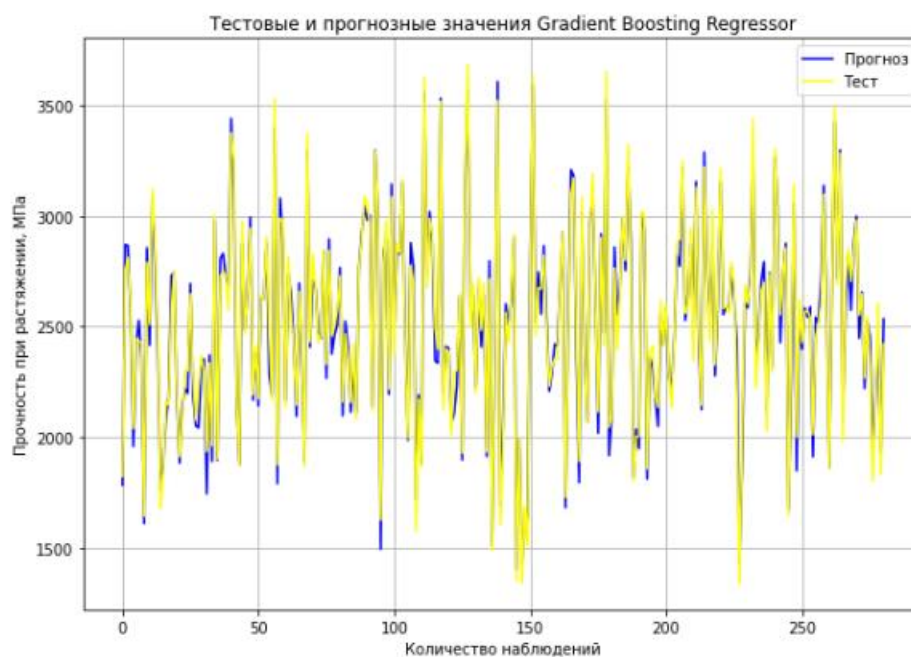


Рисунок 11 – График метода градиентного бустинга для прочности при растяжении, МПа

Градиентный бустинг деревьев решений (далее 'градиентный бустинг') – это ансамблевый алгоритм машинного обучения для проблем регрессии и классификации. Он может применяться в планировании потребности, когда требуется учитывать несколько внешних условий при расчете прогноза (среднюю температуру в определенные периоды времени, цену и т.д.).

Преимущества метода:

- В отличие от алгоритма мультилинейной регрессии, где предикторы на самом деле не зависят друг от друга, градиентный бустинг можно использовать также в случаях, когда между предикторами может наблюдаться несколько зависимостей. Для получения более точного прогноза можно сочетать мультилинейную регрессию с градиентным бустингом в одной модели прогноза.
- Этот алгоритм может обрабатывать множество потенциальных предикторных переменных.
- Отсутствует необходимость в преобразованиях ввода, таких как применение логарифмической функции к некоторым независимым переменным.
- Нерелевантные предикторные переменные определяются автоматически и не влияют на прогноз.
- Этот алгоритм менее чувствителен к резко выделяющимся значениям, чем другие алгоритмы.

Недостатки метода:

- Он может легко переобучить данные обучения, то есть прогноз может соответствовать истории продаж слишком сильно или точно и поэтому быть менее достоверным.
- Древовидная структура зависит от данных обучения, поэтому алгоритм не может дать хорошие прогнозы для случаев, когда будущие значения независимых переменных в значительной степени отличаются от соответствующих прошлых значений.

- Он часто требует множества вычислений, что может увеличить время выполнения.

8) Многослойный перцептрон

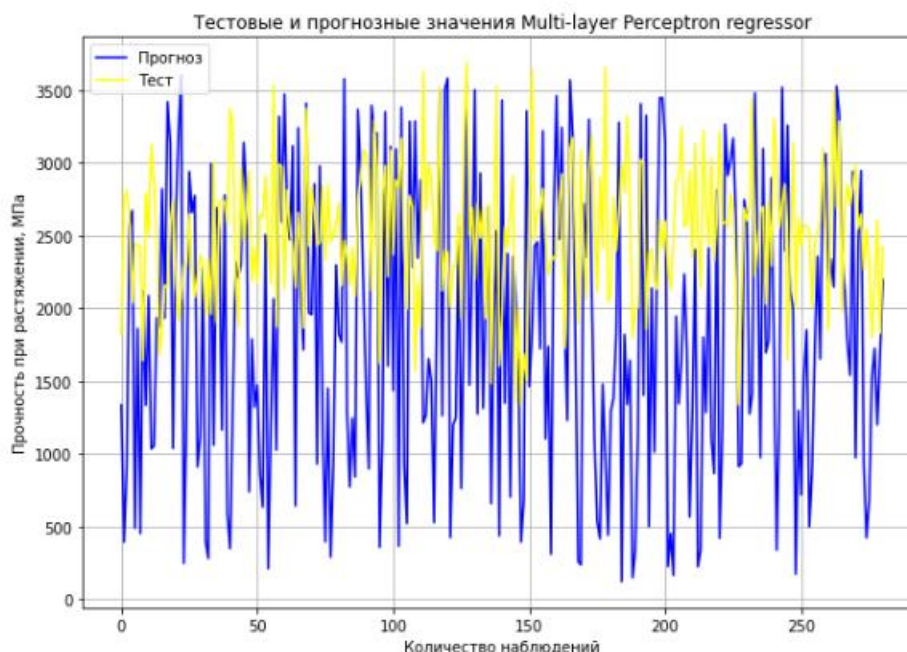


Рисунок 12 – График многослойного перцептрона для прочности при растяжении, МПа

Многослойный перцептрон — это алгоритм обучения с учителем, который изучает функцию $f(\cdot) : R^m \rightarrow R^o$ обучением на наборе данных, где m — количество измерений для ввода и o — количество размеров для вывода. Учитывая набор функций $X = x_1, x_2, \dots, x_m$ и цель y , он может изучить аппроксиматор нелинейной функции для классификации или регрессии. Он отличается от логистической регрессии тем, что между входным и выходным слоями может быть один или несколько нелинейных слоев, называемых скрытыми слоями.

Достоинства метода:

- Возможность изучать нелинейные модели.

- Возможность изучения моделей в режиме реального времени (онлайн-обучение) с использованием `partial_fit`.

Недостатки метода:

- Многослойный персептрон со скрытыми слоями имеют невыпуклую функцию потерь, когда существует более одного локального минимума. Поэтому разные инициализации случайных весов могут привести к разной точности проверки.
- Многослойный персептрон требует настройки ряда гиперпараметров, таких как количество скрытых нейронов, слоев и итераций.
- Многослойный персептрон чувствителен к масштабированию функций.

1.3 Разведочный анализ

Разведочный анализ данных – предварительное исследование датасета с целью определения его основных характеристик, взаимосвязей между признаками, а также сужения набора методов, используемых для создания модели машинного обучения.

Датасет был проверен на наличие дублирующих записей. Дублирующие записи не только искажают статистические показатели датасета, но и снижают качество обучения модели.

Датасет был проверен на наличие пропусков. Стоит помнить, что в случае, если пропусков у признака слишком много (более 70%), такой признак обычно удаляют.

Самый легкий способ обнаружить выбросы – визуальный. Поэтому была построена диаграмма "ящик с усами".

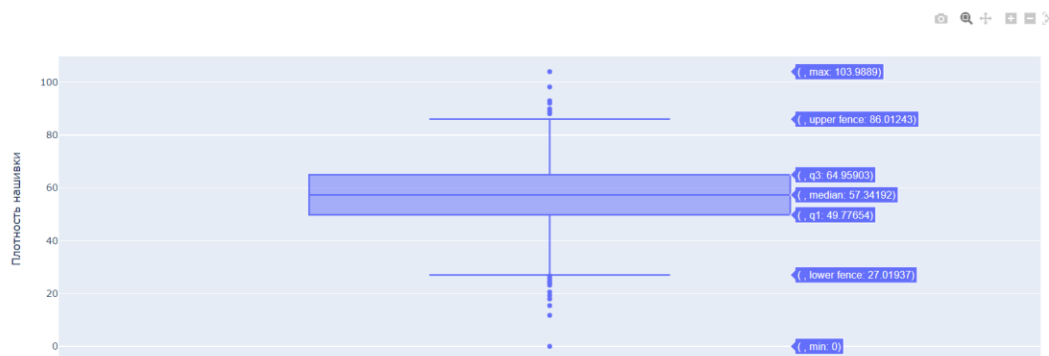


Рисунок 13 – Один из вариантов диаграммы ящик с усами

На данном рисунке видны скучковавшиеся окружности в верхней и нижней частях изображения – это есть аномалии, и от них, как правило, избавляются с помощью удаления выбросов.

Также при проведении разведочного анализа были построены гистограммы распределения каждой из переменной, попарные графики рассеяния точек, оценка плотности ядра.

2. Практическая часть

2.1 Предобработка данных

По условиям задачи необходимо было нормализовать данные. Для этого были применены 2 варианта нормализации данных: MinMaxScaler и Normalizer.

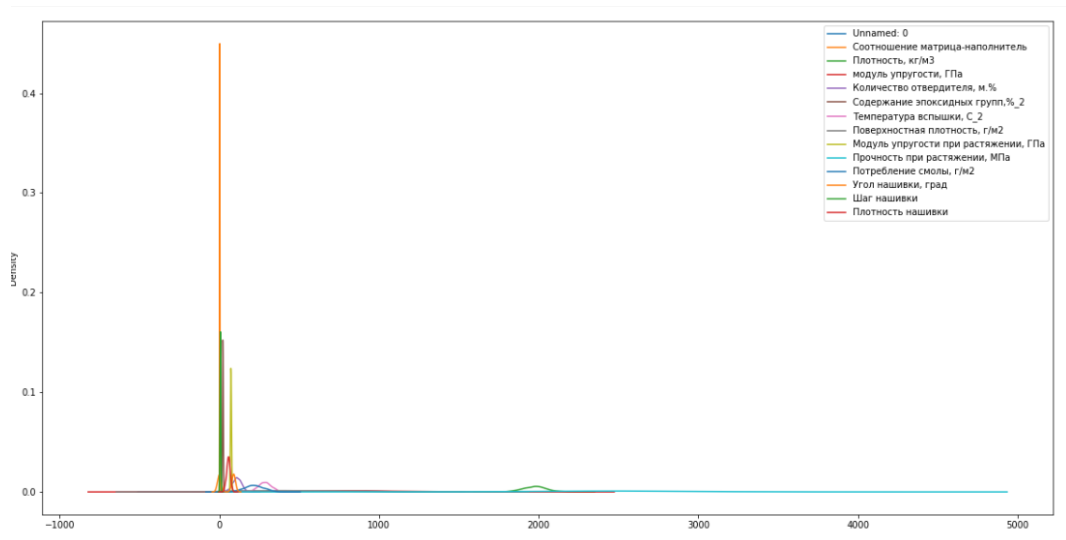


Рисунок 14 – График до нормализации

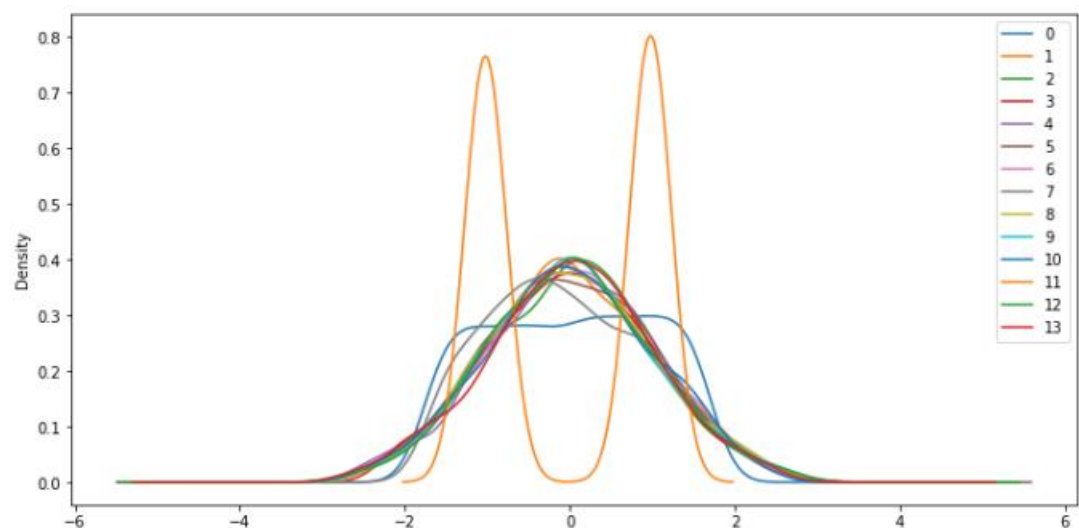


Рисунок 15 – График после нормализации методом MinMaxScaler

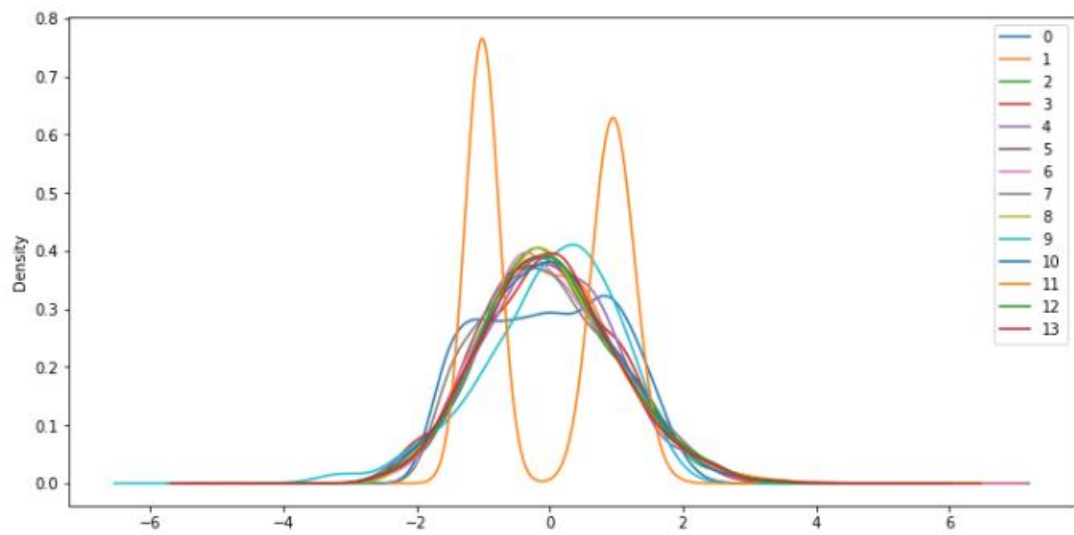


Рисунок 16 – График после нормализации методом Normalizer

2.2 Разработка и обучение модели

Разработка и обучение моделей машинного обучения осуществлялась для двух выходных параметров Прочность при растяжении и Модуль упругости при растяжении. Для решения задачи применимы следующие методы:

- 1) Метод случайного леса
- 2) Метод опорных векторов
- 3) Метод К ближайших соседей
- 4) Дерево решений
- 5) Метод линейной регрессии
- 6) Метод градиентного бустинга
- 7) Многослойный перцептрон

2.3 Тестирование модели

После обучения каждой из моделей была проведена оценка точности этих моделей на тренировочной и тестирующей части выборки.

В качестве параметра оценки моделей использовалась средняя квадратическая ошибка (MSE) и среднеквадратическая ошибка (RMSE). MSE - метрика, которая обозначает среднеквадратичную разницу между прогнозируемыми значениями и фактическими значениями в наборе данных. Чем ниже MSE, тем лучше модель соответствует набору данных. RMSE - метрика, которая обозначает квадратный корень из средней квадратичной разницы между прогнозируемыми значениями и фактическими значениями в наборе данных. Чем ниже RMSE, тем лучше модель соответствует набору данных.

	Regressor	MAE
0	RandomForest	83.629415
1	Support Vector	68.846370
2	KNeighbors	402.584361
3	DecisionTree	121.081762
4	Linear Regression	59.132081
5	GradientBoosting	59.891014
6	MLP	1011.609230

Рисунок 17 – Сравнительная таблица с результатами MAE для прочности при растяжении

	Regressor	RMSE
0	RandomForest	105.445633
1	Support Vector	94.234929
2	KNeighbors	497.791178
3	DecisionTree	153.203851
4	Linear Regression	72.785352
5	GradientBoosting	74.525747
6	MLP	1230.396757

Рисунок 18 – Сравнительная таблица с результатами RMSE для прочности при растяжении

	Regressor	MAE
0	RandomForest	2.482423
1	Support Vector	3.162011
2	KNeighbors	2.573453
3	DecisionTree	3.722941
4	Linear Regression	2.412238
5	GradientBoosting	2.425166
6	MLP	3.309768

Рисунок 19 – Сравнительная таблица с отображением MAE модуля упругости при растяжении

	Regressor	RMSE
0	RandomForest	3.045093
1	Support Vector	3.911841
2	KNeighbors	3.231927
3	DecisionTree	4.445104
4	Linear Regression	2.963983
5	GradientBoosting	2.987870
6	MLP	3.994959

Рисунок 20 – Сравнительная таблица с результатами RMSE для прочности при растяжении

Результаты ошибок MSE и RMSE при использовании многослойного перцептрона для прочности при растяжении, а также для метода К ближайших соседей являются неудовлетворительными. Остальные методы также показали довольно большие значения MSE и RMSE.

2.4 Написать нейронную сеть, которая будет рекомендовать соотношение матрица – наполнитель

До создания нейронной сети были нормализованы данные, а также выборка разбита на обучающую и тестовую. Затем была создана с заранее определенным случайным образом количеством слоев.

```
#Создаем модель
def create_model(lyrs=[32], act='softmax', opt='SGD', dr=0.1):
    seed = 7
    np.random.seed(seed)
    tf.random.set_seed(seed)

    model = Sequential()
    model.add(Dense(lyrs[0], input_dim=x_train.shape[1], activation=act))
    for i in range(1, len(lyrs)):
        model.add(Dense(lyrs[i], activation=act))

    model.add(Dropout(dr))
#создаем выходной слой
    model.add(Dense(3, activation='tanh'))
    model.compile(loss='binary_crossentropy', optimizer=opt, metrics=['mae', 'accuracy'])
    return model
```

Рисунок 21 – Создание нейронной сети

Затем было проведено исследование для определения оптимального количества слоев.

```
#Попытка определить лучшие значения для модели
model = KerasClassifier(build_fn=create_model, epochs=50, batch_size=4, verbose=0)

layers = [[8],[16, 4],[32, 8, 2],[14, 6, 3], [64, 64, 3], [128, 64, 16, 3]]
param_grid = dict(lyrs=layers)

grid = GridSearchCV(estimator=model, param_grid=param_grid, cv=10, verbose=2)
grid_result = grid.fit(x_train, y_train)

[CV] END .....lyrs=[8]; total time= 11.3s
[CV] END .....lyrs=[8]; total time= 10.9s
[CV] END .....lyrs=[8]; total time= 9.6s
[CV] END .....lyrs=[8]; total time= 9.8s
[CV] END .....lyrs=[8]; total time= 11.0s
[CV] END .....lyrs=[8]; total time= 11.0s
[CV] END .....lyrs=[8]; total time= 11.0s
[CV] END .....lyrs=[8]; total time= 9.7s
```

Рисунок 22 – Попытка определит лучшие значения для модели

Была создана модель с рекомендуемым количеством слоев.

Далее данная модель была обучена и проведена оценка полученной нейронной сети.

```
#Оценка нейронной сети
```

```
scores = model.evaluate(x_test, y_test)
print("\ns: %.2f%%" % (model.metrics_names[1], scores[1]*100))
```

```
10/10 [=====] - 0s 2ms/step - loss: -4.5820 - mae: 2.3606 - accuracy: 0.0000e+00
```

```
mae: 236.06%
```

Рисунок 23 – Оценка нейронной сети

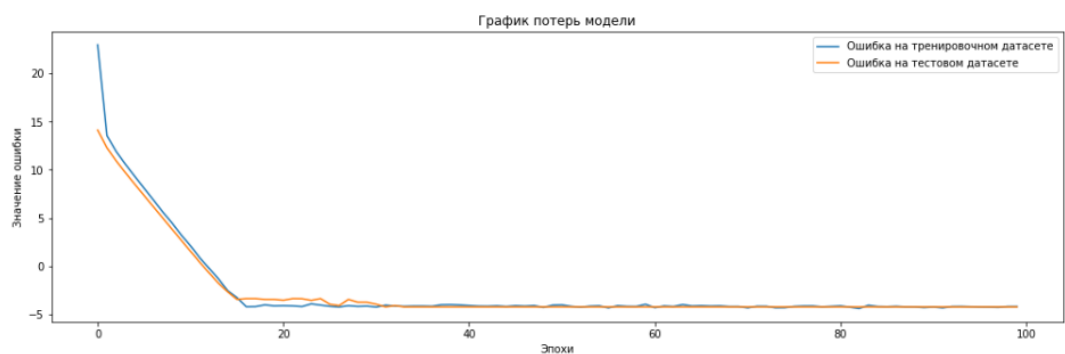


Рисунок 24 – График потерь модели

2.5 Разработка приложения

Было разработано приложение, позволяющее предсказывать соотношение матрица – наполнитель.

**Прогнозное значение параметра
Соотношение матрица-наполнитель**

Введите значение плотности, кг/м³

Введите значение модуля упругости, ГПа

Введите значение количества отвердителя, м.%

Введите значение содержания эпоксидных групп, %_2

Введите значение температуры вспышки, С_2

Введите значение поверхностной плотности, г/м²

Введите значение модуля упругости при растяжении, ГПа

Введите значение прочности при растяжении, МПа

Введите значение потребления смолы, г/м²

Введите значение угла нашивки, град

Введите значение шага нашивки

Введите значение плотности нашивки

Отправить

Рисунок 25 – Скриншот пользовательского приложения

Для получения прогнозного значения параметра Соотношение матрица – наполнитель выполните следующие действия:

1. Перейти по следующей ссылке [Прогнозное значение параметра Соотношение матрица-наполнитель](#).
2. В каждом поле ввода ввести нужные значения.

3. Нажать кнопку Отправить.
4. В результате отобразится прогнозное значение параметра Соотношение матрица – наполнитель.

2.6 Создание удаленного репозитория и загрузка результатов на него

На github.com был создан следующий репозиторий:

<https://github.com/Alla-tech/VKR>

Заключение

В процессе работы над выпускной квалификационной работы был разработан подробный план работ и каждый шаг был декомпозирован до более детальных подпунктов.

В итоге не удалось разработать модель с достоверными результатами. Созданные модели содержат довольно большие значения среднеквадратичных ошибок. При это наиболее лучшие значения MAE для модуля упругости и прочности при растяжении у метода линейной регрессии.

Для получения достоверных сведений необходимо наличие более компетентного специалиста в том, числе со знаниями о композиционных материалах, а также наличие большего количества датасетов с данными.

Список используемой литературы, веб-ресурсы

1. Регрессия как задача машинного обучения <https://koroteev.site/text/ml1/>
2. Сравнительное изучение алгоритмов классического машинного обучения
<https://machinelearningmastery.ru/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222/>
3. Алгоритм случайного леса <https://itchef.ru/articles/58561/>
4. Метод опорных векторов в машинном обучении
https://translated.turbopages.org/proxy_u/en-ru.ru.b7146247-6398e653-d7a96c13-74722d776562/https/www.geeksforgeeks.org/support-vector-machine-in-machine-learning/
5. Градиентный бустинг деревьев принятия решений
https://help.sap.com/docs/SAP_INTEGRATED_BUSINESS_PLANNING/feature3cea3cc549aaa9d9de7d363a83e6/a2a3eada1d954aeaaaaa8259ac2720767.html?version=2202