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 Астафьева Н.С.

Москва, 2023

**Содержание**

Введение3

1. Аналитическая часть4

1.1. Постановка задачи4

1.2. Описание используемых методов7

1.3. Разведочный анализ данных12

2. Практическая часть19

2.1. Предобработка данных19

2.2. Разработка и обучение модели23

2.3. Написать нейронную сеть, которая будет рекомендовать соотношение матрица. 27

2.4. Разработка приложения34

2.5. Создание удаленного репозитория и загрузка результатов работы на него.37

Заключение38

Список использованной литературы39

**Введение**

В последние годы сфера материаловедения стала одной из наиболее активно развивающихся отраслей, в которой большое внимание уделяется разработке новых материалов с улучшенными свойствами. Однако, создание новых материалов может быть дорогостоящим и время затратным процессом, поэтому актуальным является разработка эффективных методов прогнозирования конечных свойств новых материалов на основе их состава и структуры.

В рамках курса "Data Science" в МГТУ имени Н.Э. Баумана было проведено исследование применения методов машинного обучения и создания нейронной сети для прогнозирования свойств композиционных материалов на основе данных о их составе и структуре. В данной выпускной квалификационной работе рассмотрены различные методы обработки и анализа данных, используемые в процессе построения моделей прогнозирования, а также проанализированы результаты применения различных алгоритмов машинного обучения и нейронных сетей для решения данной задачи.

Целью данной работы является разработка эффективной модели прогнозирования конечных свойств новых композиционных материалов на основе данных об их составе и структуре. Для достижения этой цели в работе рассматриваются различные подходы к анализу данных и методы машинного обучения, а также проводится сравнительный анализ результатов работы различных моделей.

В ходе работы были использованы данные о композиционных материалах, содержащие информацию об их составе и структуре, а также о конечных свойствах, которые необходимо было прогнозировать. Были проведены эксперименты с различными алгоритмами машинного обучения, в том числе с использованием нейронных сетей, деревьев решений, метода опорных векторов и случайных лесов. Созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

Результаты работы могут быть использованы в дальнейших исследованиях в области материаловедения, а также могут быть применены в производственных условиях для оптимизации процесса разработки новых материалов с требуемыми свойствами.

**1.** **Аналитическая часть**

**1.1. Постановка задачи.**

Созданные прогнозные модели на основе машинного обучения и нейронных сетей помогут сократить количество проводимых испытаний для композиционных материалов, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

На входе имеются данные о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.). На выходе необходимо спрогнозировать ряд конечных свойств получаемых композиционных материалов.

Датасет представлен с помощью 13 переменных, которые говорят о тех или иных свойствах данного композиционного материала:

* Соотношение матрица-наполнитель;
* Плотность, кг/м3;
* Модуль упругости, Гпа;
* Количество отвердителя, м.%;
* Содержание эпоксидных групп, %\_2;
* Температура вспышки, С\_2;
* Поверхностная плотность, г/м2;
* Модуль упругости при растяжении, ГПа;
* Прочность при растяжении, Мпа;
* Потребление смолы, г/м2;
* Угол нашивки, град;
* Шаг нашивки;
* Плотность нашивки.

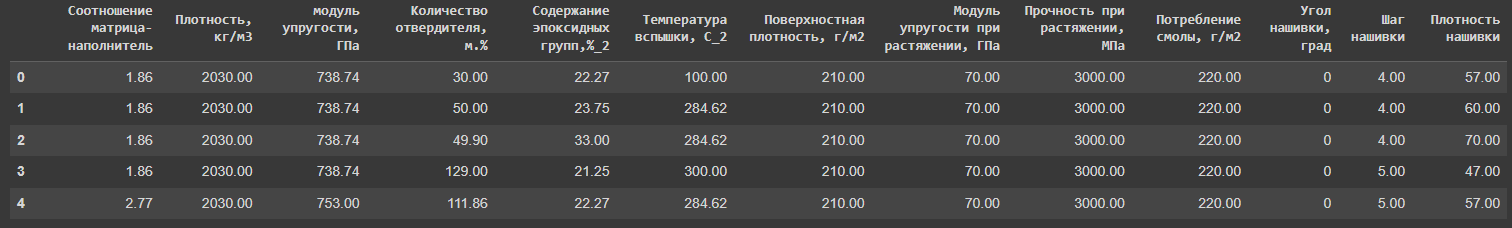


Рисунок 1. Датасет.

Датасет со свойствами композитов представлен двумя файлами (X\_bp.xlsx и X\_nup.xlsx) после объединения которых (по индексу тип объединения INNER) в соответствии с поставленной задачей его размерность представляет собой 1023 строк и 13 столбцов. В данном датасете отсутствуют пропущенные значения и дубликаты, но присутствуют выбросы. Графики приведённые в данной работе показывают на наличие выбросов во всех признаках кроме 'Угол нашивки, град'. В основном распределения данных близки к нормальному, за исключением столбца "Угол нашивки, град" который представлен двумя значениями 0 и 90.

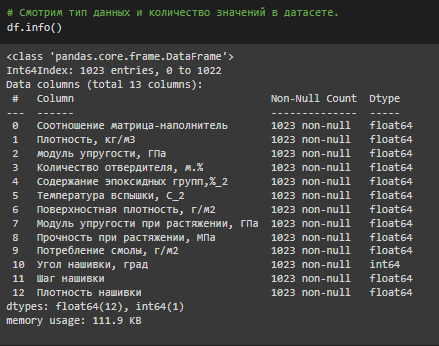


Рисунок 2. Тип данных и количество значений

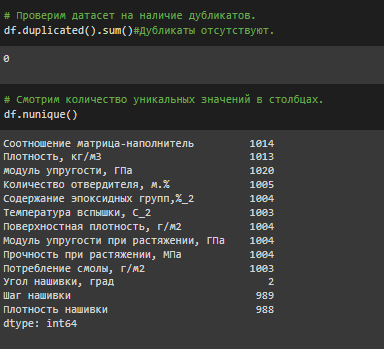


Рисунок 3. Дубликаты и количество уникальных значений

Для оценки корреляции между признаками была выбрана корреляция Пирсона. Корреляция Пирсона — это метод параметрической статистики, позволяющий определить наличие или отсутствие линейной связи между двумя количественными показателями, а также оценить ее тесноту и статистическую значимость. Также для наглядности зависимости между признаки была построена тепловая карта на которой заметно что корреляции между признаками не наблюдается.

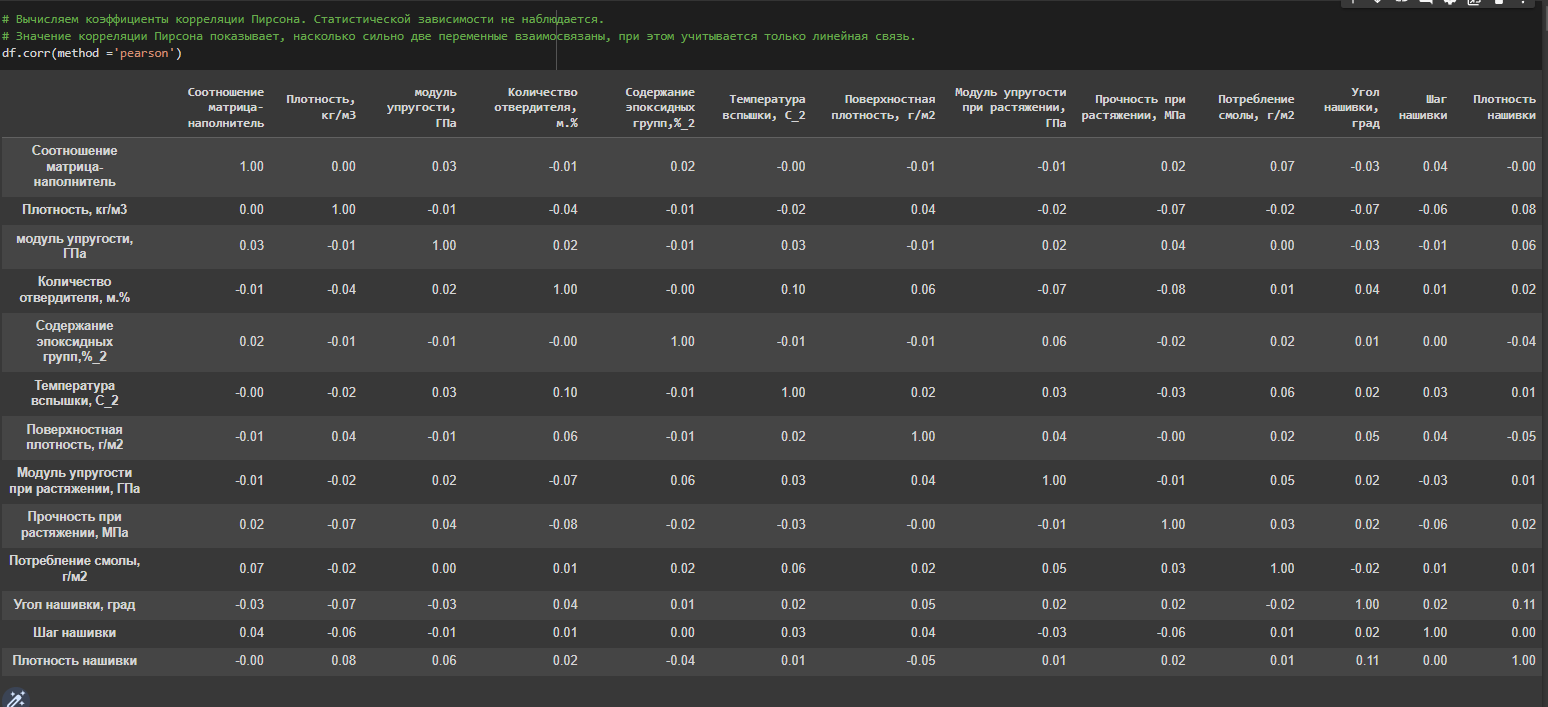


Рисунок 4. Корреляция Пирсона

В качества задачи для данной выпускной квалификационной работы ставится обучение алгоритма машинного обучения, который будет определять значения: “Модуль упругости при растяжении, ГПа" и “ Прочность при растяжении, МПа ”, а также создание нейронной сети, которая будет рекомендовать “Соотношение матрица-наполнитель”.

**1.2. Описание используемых методов**

Для решения данной задачи регрессии было использовано несколько методов машинного обучения:

* BayesianRidge;
* DecisionTreeRegressor;
* DummyRegressor;
* ElasticNet;
* ElasticNetCV;
* GradientBoostingRegressor;
* KernelRidge;
* Lasso;
* LassoLars;
* LinearRegression;
* RandomForestRegressor;
* Ridge;
* SVR.

Bayesian Ridge (Байесовская линейная регрессия) - это метод регрессии, который использует байесовский подход для оценки коэффициентов регрессии и шума в данных. Он минимизирует сумму квадратов ошибок и моделирует коэффициенты регрессии как случайные переменные с нормальным распределением.

Decision Tree Regression (Дерево решений регрессии) — это метод нелинейной модели, который использует дерево решений для моделирования зависимости между признаками и целевой переменной. Дерево решений представляет собой структуру данных, которая состоит из узлов и листьев. Узлы представляют собой разделение признаков на две или более ветви, а листья — это конечные значения, которые являются предсказанными значениями для данного наблюдения. Дерево решений строится по обучающей выборке путем рекурсивного разделения на подгруппы, каждый раз выбирая признак, который максимально снижает дисперсию (или уменьшает ошибку).

DummyRegressor – это регрессор, который делает прогнозы, используя простые правила. Этот регрессор полезен в качестве простой основы для сравнения с другими (реальными) регрессорами. DummyClassifier делает прогнозы, которые игнорируют входные функции. Этот классификатор служит простой базой для сравнения с другими более сложными классификаторами. Конкретное поведение базовой линии выбирается с помощью параметра стратегии.

ElasticNet (Эластичная\_сеть) – метод регрессии, использующий комбинацию L1 и L2 регуляризации для уменьшения количества признаков и предотвращения переобучения.

ElasticNetCV – метод регрессии, использующий комбинацию L1 и L2 регуляризации и кросс-валидацию для выбора оптимальных параметров модели. это класс перекрестной проверки, который может искать несколько альфа-значений и применять лучшее из них.

GradientBoostingRegressor (Градиентный бустинг регрессии) — это метод ансамблирования множества слабых моделей регрессии, чтобы получить более сильную модель. Он работает путем последовательного добавления слабых моделей, которые корректируют ошибки предыдущих моделей.

Kernel Ridge (Ядерная регрессия) — это метод регрессии, который использует ядерные функции для построения нелинейной модели регрессии. Он минимизирует сумму квадратов ошибок и добавляет штраф на сложность модели за счет добавления к функционалу потерь L2-нормы коэффициентов регрессии.

Lasso (Лассо-регрессия) — это метод регуляризации линейной регрессии, который использует L1-регуляризацию. Он работает путем добавления штрафа за сумму абсолютных значений коэффициентов регрессии, что приводит к сокращению некоторых коэффициентов до нуля, что позволяет выполнять отбор признаков.

LassoLars – это модель лассо, реализованная с использованием алгоритма LARS, и в отличие от реализации, основанной на спуске по координатам, она дает точное решение, кусочно-линейное как функцию нормы его коэффициентов. Модель Лассо соответствует регрессии с наименьшим углом, также известной как Ларс.

LinearRegression (Линейная регрессия) — это простой метод машинного обучения для построения линейной модели, которая моделирует связь между входными признаками и выходными целевыми значениями. Он работает путем минимизации суммы квадратов ошибок между прогнозируемыми и реальными значениями.

RandomForestRegressor (Случайный лес регрессии) — это метод ансамблирования множества деревьев решений, чтобы получить более сильную модель. Он работает путем построения множества деревьев решений, которые каждое используют подмножество признаков и данных, и затем усредняет результаты прогнозирования всех деревьев.

Ridge (Гребневая регрессия) — это метод регуляризации линейной регрессии, который использует L2-регуляризацию. Он работает путем добавления штрафа за квадраты значений коэффициентов регрессии, что ограничивает их значения и уменьшает переобучение.

SVR (Метод опорных векторов для регрессии) – это метод регрессии, который использует максимальный зазор между опорными векторами для построения линейной или нелинейной модели регрессии. Он минимизирует сумму квадратов ошибок и добавляет ограничения на значения коэффициентов регрессии в зависимости от расстояний до опорных векторов.

Таблица 1 – Плюсы и минусы каждого метода

|  |  |  |
| --- | --- | --- |
| **Метод машинного обучения** | **Плюсы** | **Минусы** |
| BayesianRidge | Хорошо работает на данных с большим количеством признаков, устойчив к шуму в данных | Может быть медленным на больших данных, требует подбора оптимальных параметров. |
| DecisionTreeRegressor | Прост в использовании, работает быстро на небольших данных, может обрабатывать как числовые, так и категориальные данные. | Склонен к переобучению, не очень хорошо работает на больших данных. |
| DummyRegressor | Прост в использовании, быстро работает, может использоваться как базовая модель для сравнения с другими моделями | Не очень точен, не учитывает сложности данных. |
| ElasticNet | Хорошо работает на данных с большим количеством признаков, устойчив к шуму в данных | Может быть медленным на больших данных. |
| ElasticNetCV | Хорошо работает на данных с большим количеством признаков, устойчив к шуму в данных, автоматически выбирает оптимальные параметры модели. | Может быть медленным на больших данных. |
| GradientBoostingRegressor | Хорошо работает на данных с большим количеством признаков, устойчив к шуму в данных, может обрабатывать как числовые, так и категориальные данные. | Может быть медленным на больших данных. |
| KernelRidge | Хорошо работает на данных с нелинейной зависимостью, устойчив к шуму в данных. | Может быть медленным на больших данных, требует подбора оптимальных параметров. |
| Lasso | Хорошо работает на данных с большим количеством признаков, устойчив к шуму в данных. | Может быть медленным на больших данных. |
| LassoLars | Хорошо работает на данных с большим количеством признаков, устойчив к шуму в данных | Может быть медленным на больших данных. |
| LinearRegression | Прост в использовании, быстро работает, хорошо работает на данных с линейной зависимостью. | Не учитывает сложности данных, склонен к переобучению. |
| RandomForestRegressor | Хорошо работает на данных с большим количеством признаков, устойчив к шуму в данных, может обрабатывать как числовые, так и категориальные данные. | Может быть медленным на больших данных. |
| Ridge | Хорошо работает на данных с большим количеством признаков, устойчив к шуму в данных | Может быть медленным на больших данных. |
| SVR | Хорошо работает на данных с нелинейной зависимостью, устойчив к шуму в данных | Может быть медленным на больших данных, требует подбора оптимальных параметров. |

**1.3. Разведочный анализ данных**

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

В данном проекте были использованы следующие методы разведочного анализа данных:

* Описательная статистика данного датасета.
* Визуальный анализ гистограмм
* Визуальный анализ диаграмм размаха («ящик с усами»)
* Проверка нормальности распределения по критерию Пирсона
* Анализ попарных графиков рассеяния переменных
* Корреляционный анализ c целью поиска коэффициентов

Для просмотра статистических данных для в каждого столбца используется метод describe( ). Этот метод показывает количество строк в столбце - count, среднее значение столбца - mean, столбец стандартное отклонение - std, минимальные (min) и максимальные (max) значения, а также границу каждого квартиля - 25%, 50% и 75%. Любые значения NaN автоматически пропускаются.

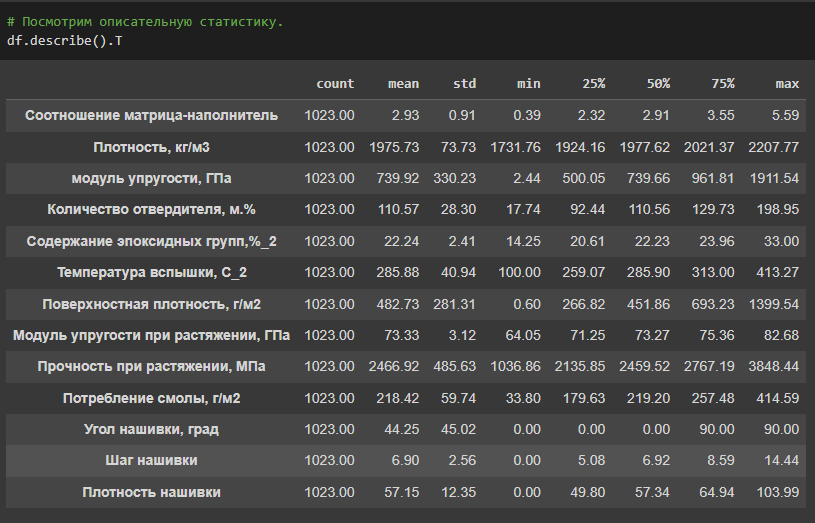


Рисунок 5. Описательная статистика

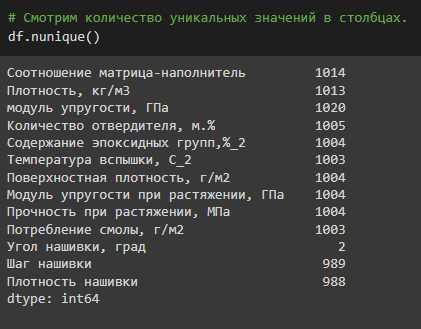


Рисунок 6. Количество уникальных значений в столбцах.

Для просмотра количества дубликатов в датасете используем метод duplicated().

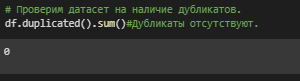


Рисунок 7. Проверка на наличие дубликатов

Для оценки величины и характера разброса данных построены гистограммы распределения для каждого из признаков.

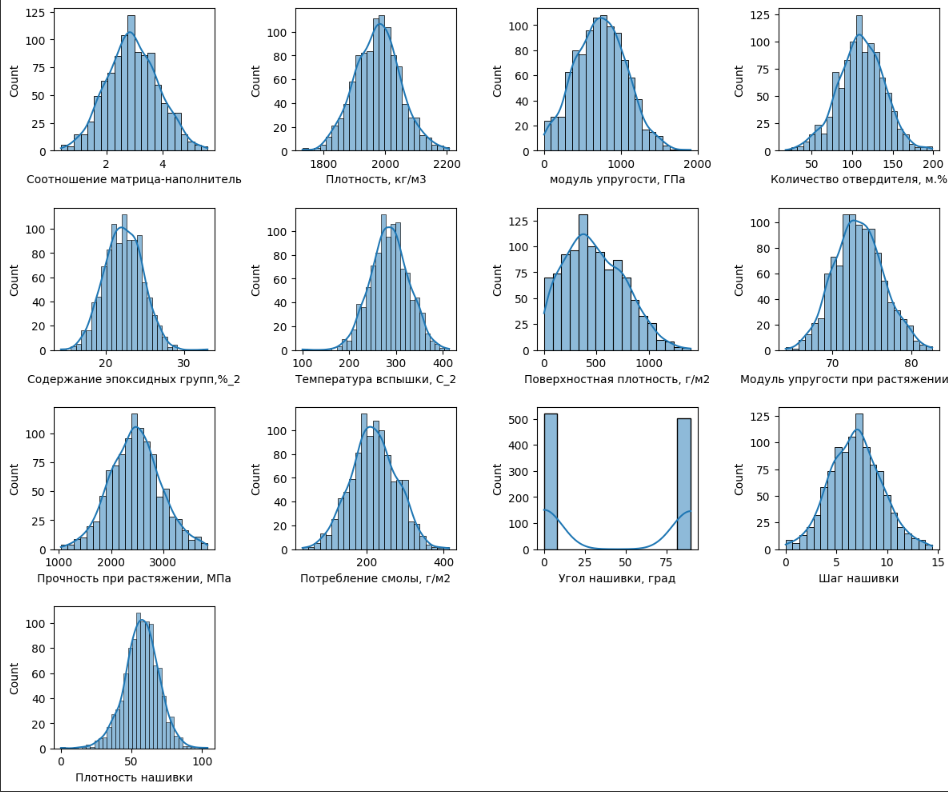


Рисунок 8. Гистограммы распределения

Гистограммы показывают что в основном распределения данных близки к нормальному, за исключением столбца с данными “Угол нашивки, град”, который представлен двумя значениями.

Для проверки датасета на наличие выбросов был проведён визуальный анализ диаграмм размаха. Диаграммы размаха показывают медиану, верхний квартиль, нижний квартиль, межквартильный размах, выбросы. На «ящиках с усами» хорошо заметны наличия выбросов в данном датасете.

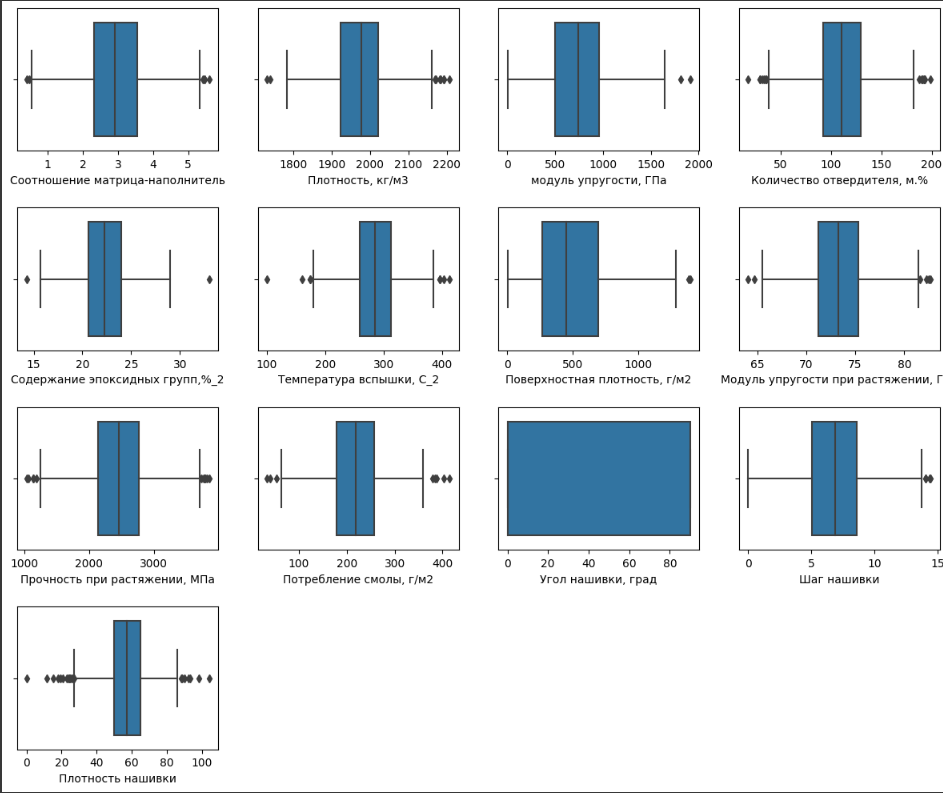


Рисунок 9. Диаграмма размаха

Так же был проведён анализ попарных графиков рассеяния переменных.

Попарный график помогает нам визуализировать распределение отдельных переменных, а также взаимосвязи между двумя переменными. Это отличный метод определения тенденций между переменными для последующего анализа. Если точки на графике расположены вдоль прямой линии, это может указывать на линейную связь между переменными. Если точки расположены в случайном порядке на графике, это может указывать на отсутствие связи между переменными или на то, что связь между переменными сложна и не может быть описана простой линейной связью. Также попарный график рассеяния может помочь идентифицировать выбросы, что являются отдельными наблюдениями, которые выделяются на графике и могут оказывать большое влияние на результаты анализа.

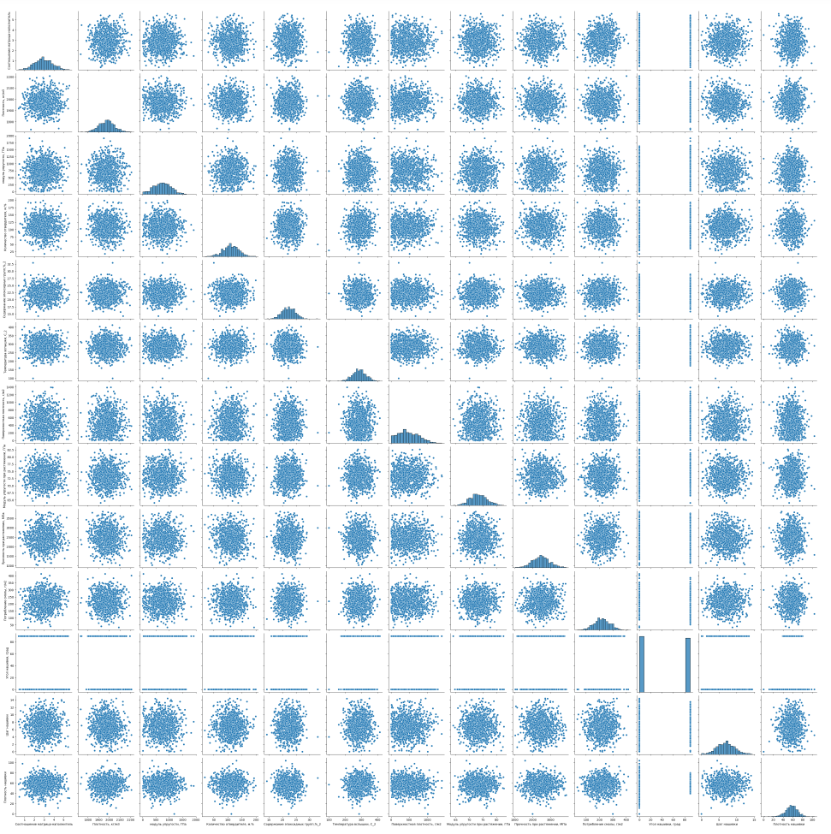


Рисунок 10. Попарный график рассеяния

График оценки плотности показывает, как вероятность распределена по значению непрерывной случайной величины. Он может использоваться для визуализации формы распределения, а также для получения информации о вероятности различных значений этой случайной величины.

График оценки плотности может использоваться для выявления выбросов, определения типа распределения, сравнения распределений между собой, оценки вероятности событий и принятия решений на основе статистических данных.

Оценка плотности ядра показывает, что значения переменных находятся в разных диапазонах поэтому требуется нормализация данных.

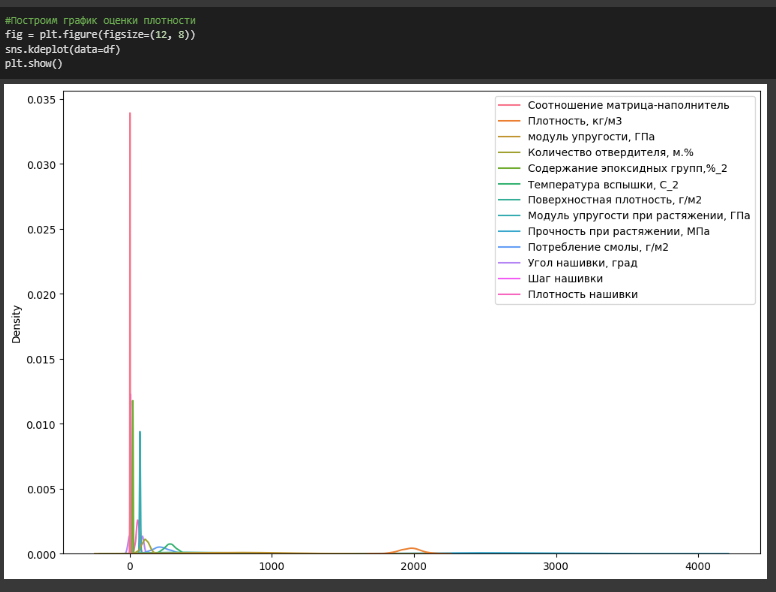


Рисунок 11. График оценки плотности распределения

Для визуализации коэффициентов корреляции и определения наличия между переменными зависимости, была построена тепловая карта коэффициентов корреляции методом Пирсона.

Корреляция Пирсона — это статистический показатель, который используется для измерения степени линейной связи между двумя переменными. Она измеряет силу и направление линейной связи между двумя непрерывными переменными.

Коэффициент корреляции Пирсона принимает значения от -1 до 1. Значение -1 указывает на полную обратную линейную связь, значение 1 указывает на полную прямую линейную связь, а значение 0 указывает на отсутствие линейной связи между переменными.

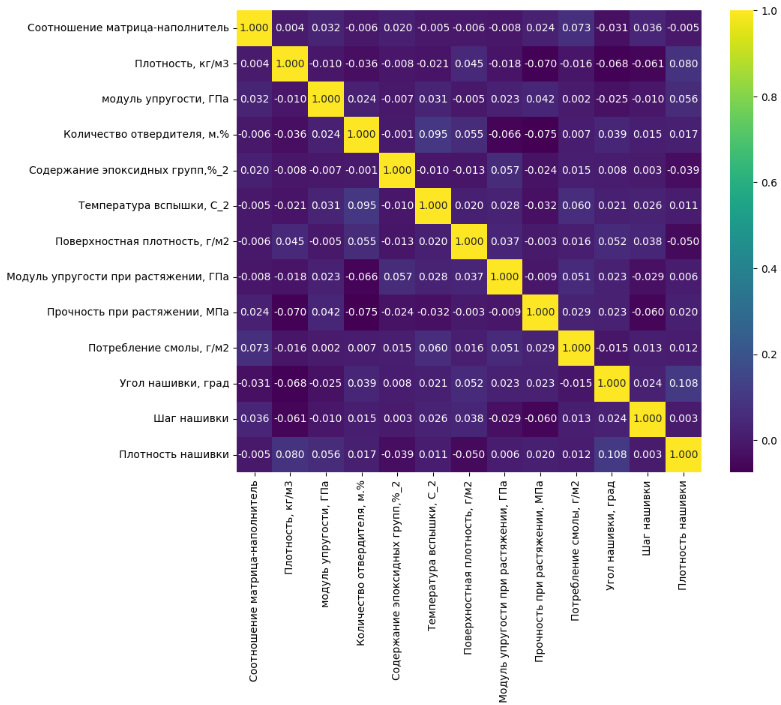


Рисунок 12. Тепловая карта

Корреляции между переменными практически не наблюдается.

**2. Практическая часть**

**2.1 Предобработка данных**

Для удаления выбросов в датасете был выбран метод исключения выбросов с помощью правила трёх сигм.

Правило трех сигм — это статистический метод для определения выбросов в наборе данных. Этот метод основан на стандартном отклонении данных от среднего значения. Согласно правила трех сигм, большинство данных должны находиться в пределах трех стандартных отклонений от среднего значения. Используя эту концепцию, можно определить, какие данные являются выбросами.

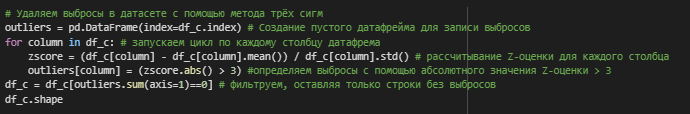


Рисунок 13. Удаления выбросов

Очищенный от выбросов датасет содержит 1000 строк. С помощью диаграмм размаха проверяем на сколько хорошо были удалены выбросы в датасете.

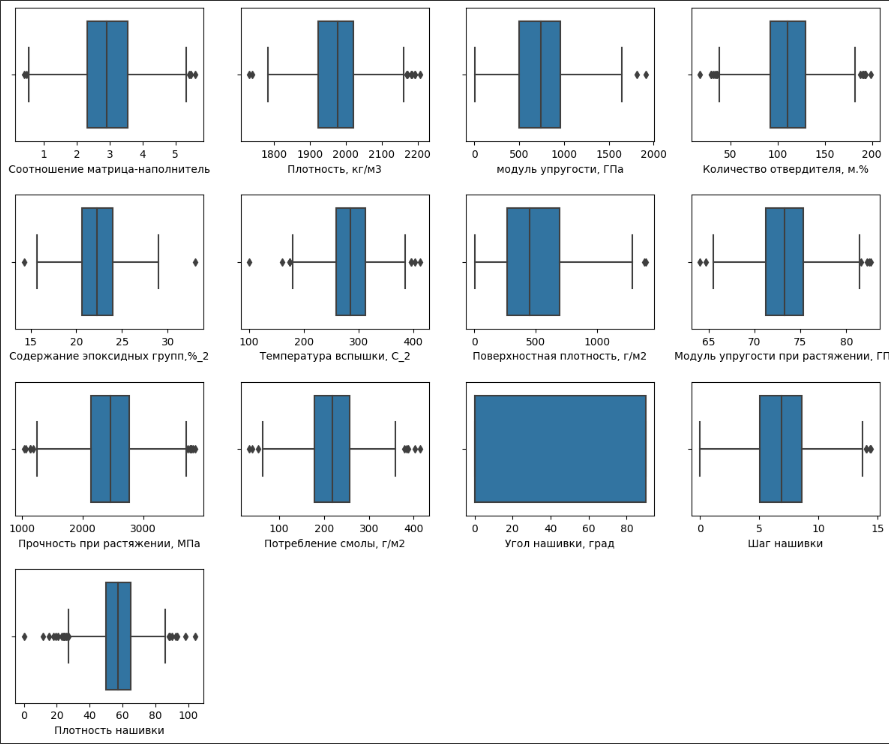


Рисунок 14. Диаграмма рассеивания

Для приведения всех признаков в датасете к одинаковой шкале была сделана нормализация с помощью метода MinMaxScaler из библиотеки sklearn.

Нормализация — это процесс приведения данных к определенному масштабу или диапазону значений. Нормализация может использоваться для приведения различных признаков в датасете к единому масштабу, чтобы они имели одинаковый вклад в результаты анализа или моделирования.

Нормализация данных является важным шагом в подготовке данных для машинного обучения и может улучшить результаты алгоритмов, повысить их скорость и устойчивость к шуму.

Нормализация (MinMaxScaler) является одним из методов нормализации данных. Этот метод используется для масштабирования признаков в диапазоне от 0 до 1. Он работает путем пересчета значений признаков на основе их минимального и максимального значений в наборе данных.

Формула для MinMax-нормализации следующая:

где:

x\_scaled - отмасштабированное значение признака;

x - оригинальное значение признака;

x\_min - минимальное значение признака в наборе данных;

x\_max - максимальное значение признака в наборе данных.

Процесс нормализации MinMaxScaler включает в себя следующие шаги:

* Определение минимального и максимального значений признака в наборе данных.
* Применение формулы MinMax-нормализации для каждого значения признака.
* Получение отмасштабированных значений признаков в диапазоне от 0 до 1.

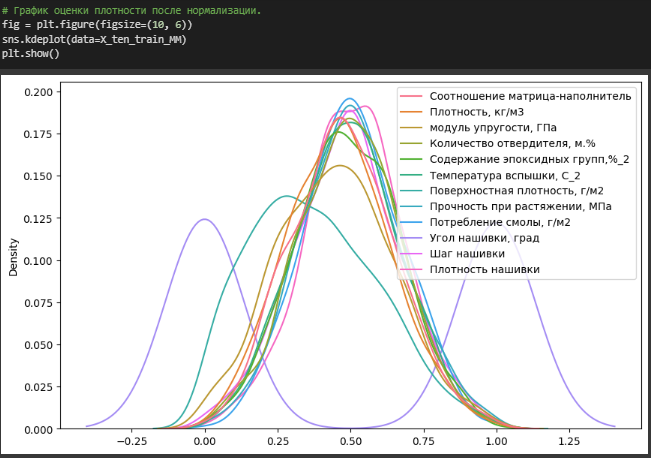


Рисунок 15. График оценки плотности распределения после нормализации

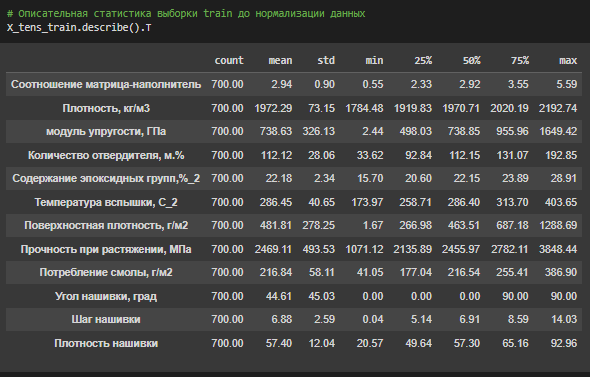


Рисунок 16. Описательная статистика до нормализации

Описательная статистика тренировочной выборки после нормализации.

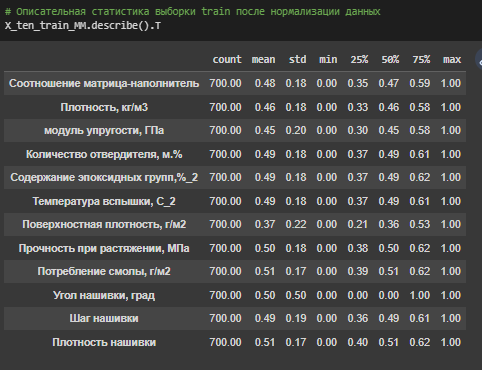


Рисунок 17. Описательная статистика после нормализации

* 1. **Разработка обучение и тестирование модели.**

Для решения поставленной задачи данные были разделены на обучающую и тестовую выборки в соотношении 70/30. Для каждой модели был создан словарь с гиперпараметрами и с помощью поиска по сетке с перекрестной проверкой были найдены лучшие гиперпараметры для каждой модели. Оценка качества работы каждой из моделей выполнялась с помощью вычисления коэффициента детерминации (R²), среднеквадратичной ошибки (RMSE) и средней абсолютной ошибка (MAE).

Для прогнозирования модуля упругости при растяжении были использованы следующие методы машинного обучения:

* DecisionTreeRegressor;
* DummyRegressor;
* ElasticNet;
* GradientBoostingRegressor;
* LassoLars;
* Lasso;
* LinearRegression;
* RandomForestRegressor;
* Ridge;
* KernelRidge.

Коэффициент детерминации, имеющий отрицательные значения близкие к нулю, говорит о том, что результат использования моделей не точнее использования для прогноза среднего значения прогнозируемого параметра.

Наилучший результат для “Модуль упругости при растяжении” с подобранными гиперпараметрами на тренировочной выборке показала модель KernelRidge.

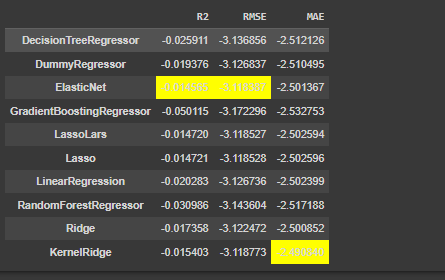


Рисунок 18. Показатели моделей на тренировочной выборке, предсказывающих модуль упругости при растяжении.

Все модели для расчёта “модуля упругости при растяжении” на тестовой выборке показали неудовлетворительные результаты, коэффициент детерминации у всех моделей показал отрицательное значение. Лучшие показатели на тестовой выборке показывает модель ElasticNet.

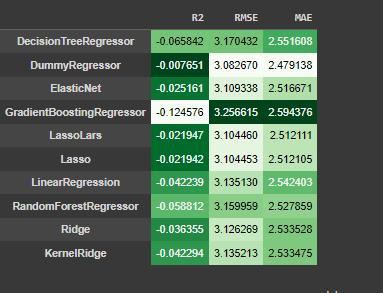


Рисунок 19. Показатели моделей на тестовой выборке, предсказывающих модуль упругости при растяжении.

Все модели показали неудовлетворительные результаты, коэффициент детерминации у лучших моделей находится около нуля. Лучшие показатели на тестовой выборке показывает модель DummyRegressor.

Для прогнозирования прочности при растяжении были использованы следующие методы машинного обучения:

* BayesianRidge;
* DecisionTreeRegressor;
* DummyRegressor;
* ElasticNetCV;
* GradientBoostingRegressor;
* Lasso;
* LinearRegression;
* KernelRidge;
* RandomForestRegressor;
* SVR.

Единственная модель для расчёта “Прочности при растяжении” на тренировочной выборке, у которой коэффициент детерминации стал положительным - стала модель SVR (Метод опорных векторов).

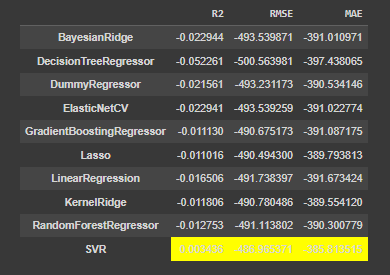


Рисунок 20. Показатели моделей на тренировочной выборке, предсказывающих “Прочности при растяжении”.

Все модели для расчёта “Прочности при растяжении” на тестовой выборке показали неудовлетворительные результаты, коэффициент детерминации у всех моделей показал отрицательное значение.

Все модели показали неудовлетворительные результаты, коэффициент детерминации у лучших моделей находится около нуля. Лучшие показатели на тестовой выборке показывает модель DummyRegressor.

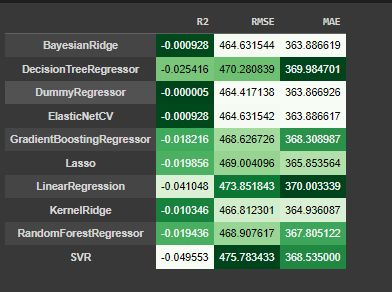


Рисунок 21. Показатели моделей на тестовой выборке, предсказывающих “Прочности при растяжении”.

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

Для решения поставленной задачи по рекомендации значения параметра “Соотношение матрица-наполнитель” были написаны нейросети с помощью MLPRegressor из библиотеки sklearn и с помощью Sequential из библиотеки Keras.

Для работы с нейросетью датасет разделен на новые тестовую и обучающую выборку в соотношении 70/30, с целевым параметром «Соотношение матрица/наполнитель».

Для подбора лучших параметров для MLPRegressor был использован метод GridSearchCV из библиотеки sklearn.

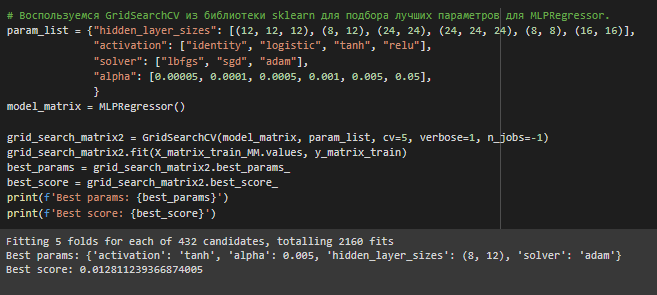


Рисунок 22. Подбор гиперпараметров

Преимущества многослойного перцептрона:

* Возможность изучать нелинейные модели.
* Возможность изучения моделей в режиме реального времени.

К недостаткам многослойного персептрона (MLP) можно отнести:

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

Архитектура и параметры нейронной сети многослойного персептрона MLPRegressor:

* Последовательная модель нейронной сети.
* Модель состоит из трёх скрытых слоев, в каждом слое по 12 нейронов и выходного слоя с одним нейроном.
* Функция активации слоев выбран гиперболический тангенс (tanh). В качестве оптимизатора нейронной сети используется SGD.
* (стохастический градиентный спуск) с импульсом ускорения = 0.5.
* В нейронной сети используется функция ранней остановки обучения.
* Параметр регуляризации L2 (регуляризация весов), который уменьшает переобучение модели - 0.005.
* Максимальное количество итераций для обучения модели – 500.
* Для валидации будет использовано 30% обучающих данных.

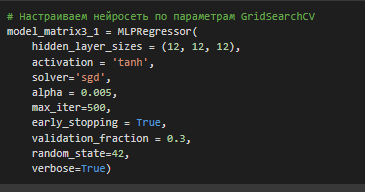


Рисунок 23. Настройка нейронной сети

В процессе обучения нейронной сети сработала ранняя остановка обучения на 17-ой эпохе обучения, т.к. в течение 10-ти эпох не наблюдалось улучшения результата потерь на валидационной выборке.

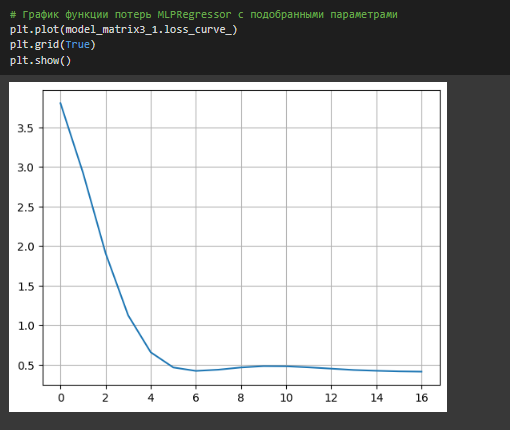


Рисунок 24. График ошибки.

Коэффициент детерминации для MLPRegressor составил: -0.019444.

Среднеквадратичная ошибка для MLPRegressor: 0.932634.

График прогнозных данных, полученных с помощью нейронной сети MLPRegressor.

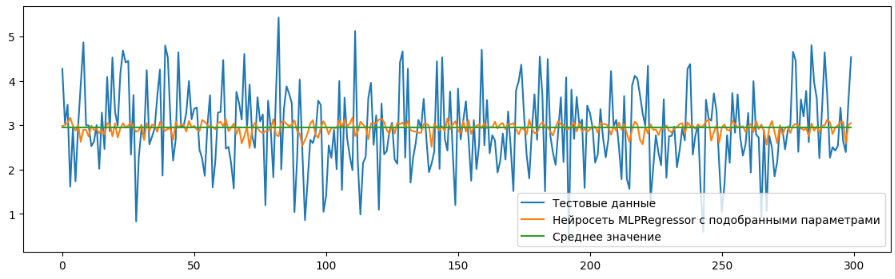


Рисунок 25. График прогнозных данных

Модель нейронной сети, созданной с помощью MLPRegressor показала неудовлетворительный результат. Коэффициент детерминации, имеющий значение близкое к нулю, говорит о том, что результат использования нейронной сети не точнее использования для прогноза среднего значения прогнозируемого параметра.

Архитектура и параметры нейронной сети из библиотеки Keras:

* Последовательная модель (Sequential) нейронной сети.
* Модель состоит из трёх скрытых слоев (Dense), с количеством нейронов, в которых равно 8, 16, 24 и выходного слоя с одним нейроном, так же в нейронной сети есть три слоя регуляризации (Dropout) со значениями (0.2, 0.3, 0.5).
* Функция активации слоев выбран гиперболический тангенс (tanh).
* В качестве оптимизатора нейронной сети используется SGD (стохастический градиентный спуск) со скоростью обучения 0,005, функцией потерь "среднеквадратическая ошибка" и метрикой оценки качества "среднеквадратичная ошибка".
* В нейронной сети используется функция ранней остановки обучения, если в течение пяти эпох не наблюдается улучшения потерь на валидационной выборке.
* Количество эпох обучения равно 100
* Для валидации будет использовано 30% обучающих данных.

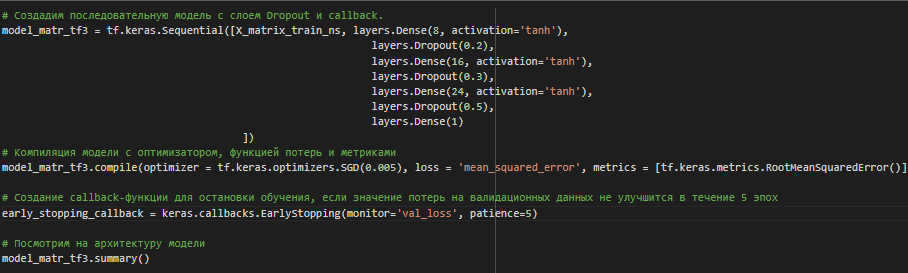


Рисунок 26. Нейронная сеть

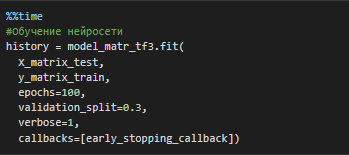


Рисунок 27. Обучение нейронной сети

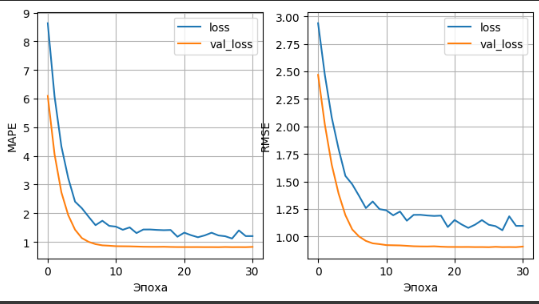


Рисунок 28. График ошибки

Коэффициент детерминации для последовательной нейронной сети из библиотеки Keras: -0.038714

Среднеквадратичная ошибка для последовательной нейронной сети из библиотеки Keras: 0.941408

График прогнозных значений, полученных с помощью последовательной нейронной сети из библиотеки Keras.

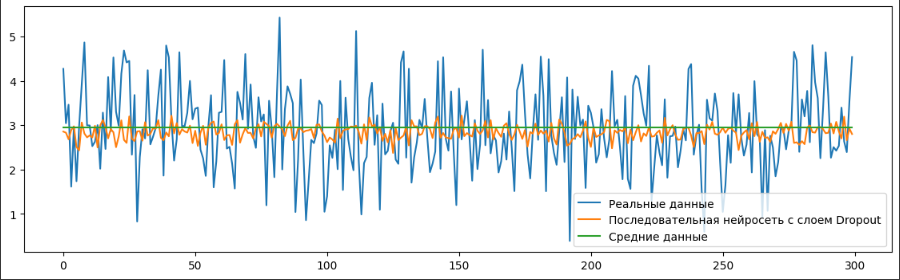


Рисунок 29. График прогнозных значений

Модель последовательной нейронной сети, созданной с помощью библиотеки Keras показала неудовлетворительный результат. Коэффициент детерминации, имеющий значение близкое к нулю, говорит о том, что результат использования нейронной сети не точнее использования для прогноза среднего значения прогнозируемого параметра.

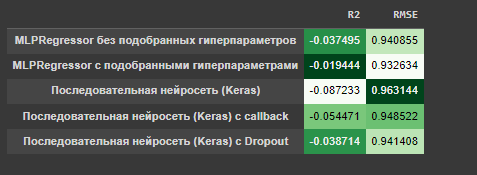


Рисунок 30. Результаты работы нейронных сетей для предсказания “Соотношение матрица-наполнитель”

**2.4 Web-приложение на Flask**

Веб-приложение для прогноза “соотношения матрица-наполнитель”. написано на языке программирования Python с использованием библиотеки Flask, для написания шаблонов страниц был использован язык разметки HTML.

1. Для запуска приложения необходимо:
2. Запустить среду разработки Visual Studio Code.
3. Запустить приложение командой " python app\_fix.py ".
4. Перейти по сгенерированной ссылке.
5. Ввести значение для каждого параметра.
6. Нажать кнопку "Отправить"

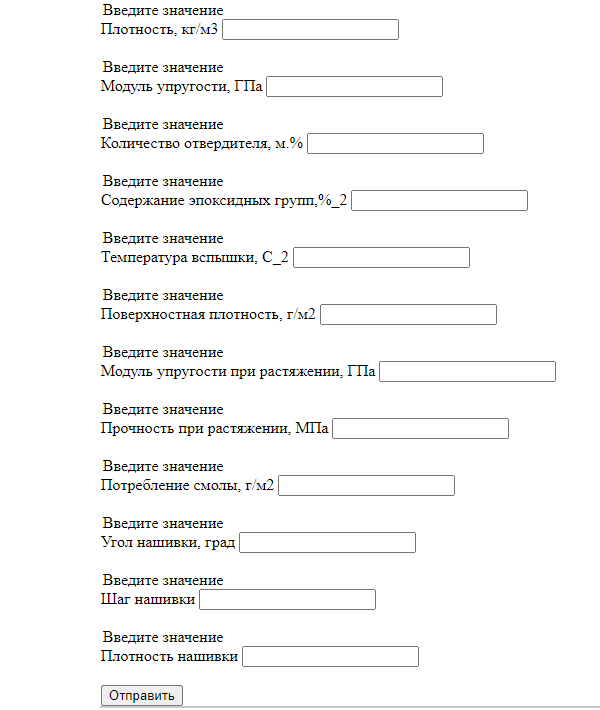


Рисунок 31. Страница ввода данных.

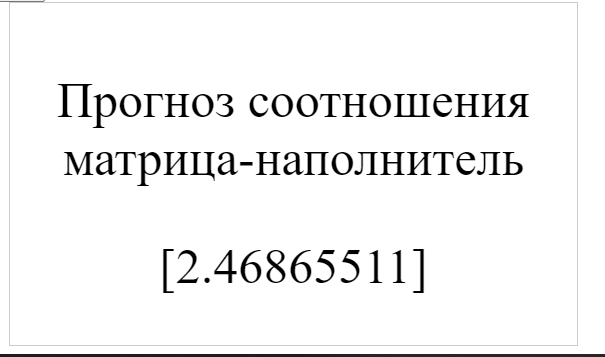


Рисунок 32. Спрогнозированное значение

**2.5 Создание удаленного репозитория и загрузка результатов работы на него.**

Ссылка на репозиторий в GitHub: <https://github.com/Natalia-1810/-.-.git>

**Заключение**

В данной работе не удалось разработать эффективную модель прогнозирования конечных свойств новых композиционных материалов на основе данных об их составе и структуре, точность предсказанных значений практически не превосходило среднее значения. Возможно для улучшения работы предсказательных систем следует уделить больше внимания качеству и количеству признаков в датасете.

Тем не менее получен существенный практический опыт по анализу, визуализации и предобработке данных, созданию нейронных сетей и моделей машинного обучения, а также опыт по созданию веб-приложения на основе этих прогнозных моделей.

**Список использованной литературы**

1. Композиционные материалы: Справочник /Под. ред. В.В. Васильева, Ю.М.Тарнопольского. –М.: Машиностроение, 1990. –512 с.
2. Библиотека Keras - инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow / пер. с англ. Слинкин А. А. - М.: ДМК Пресс, 2018. - 294 с.
3. Силен Дэви, Мейсман Арно, Али Мохамед. Основы Data Science и Big Data. Python и наука о данных. – СПб.: Питер, 2017. – 336 с.: ил.
4. Платформа scikit-learn [Электронный ресурс]: – Режим доступа: https://scikit-learn.org/stable/ (дата обращения: 27.03.2023).
5. Библиотека Seaborn- Режим доступа: https://seaborn.pydata.org/. (дата обращения 22.03.2023)
6. Язык программирования Python- Режим доступа: https://www.python.org/. (дата обращения 27.03.2023)
7. Библиотека Pandas – Режим доступа: https://pandas.pydata.org/ (дата обращения 27.03.2023)
8. Библиотека Sklearn – Режим доступа: https://scikit-learn.org/stable/ (дата обращения 27.03.2023)
9. Библиотека Pandas- Режим доступа: https://pandas.pydata.org/. (дата обращения 27.03.2023)
10. Библиотека Matplotlib- Режим доступа: https://matplotlib.org/. (дата обращения 27.03.2023)
11. Библиотека Tensorflow: Режим доступа: https://www.tensorflow.org/. (дата обращения 31.03.2023)