



Москва, ул. 2-я Бауманская, д.5, стр.1
+7 (495) 187-85-85, do@bmstu.ru
do.bmstu.ru

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

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
по курсу
«Data Science Pro»

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

Слушатель

Островерх Владимир Степанович

Москва, 2025

Содержание

Введение	3
1. Аналитическая часть.	5
1.1 Постановка задачи.	5
1.2 Описание используемых методов.	7
1.3 Разведочный анализ данных.	12
2. Практическая часть	18
2.1 Предобработка данных.	18
2.2 Разработка и обучение модели.	18
2.3 Тестирование модели.	19
2.4. Создание нейронной сети.	21
2.5. Разработка приложения.	23
2.6. Создание удаленного репозитория.	25
Заключение.....	26
Библиографический список.....	27

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).

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.

В процессе исследовательской работы планируется создание нескольких моделей, способных с высокой вероятностью предсказывать необходимые параметры.

Актуальность данной работы – это 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 выполнения работы были предоставлены два датасета: «X_br.xlsx» и «X_pur.xlsx». Они содержали следующие значения:

- 1) Порядковый номер записи;
- 2) Соотношение матрица-наполнитель;
- 3) Плотность, кг/м³;
- 4) Модуль упругости, ГПа;
- 5) Количество отвердителя, м.%;
- 6) Содержание эпоксидных групп, %₂;
- 7) Температура вспышки, С₂;
- 8) Поверхностная плотность, г/м²;
- 9) Модуль упругости при растяжении, ГПа;
- 10) Прочность при растяжении, МПа;
- 11) Потребление смолы, г/м²;
- 12) Порядковый номер записи;
- 13) Угол нашивки, град;
- 14) Шаг нашивки;
- 15) Плотность нашивки.

Планируется, что из данного перечня на выходе, после реализации моделей, будут рассчитываться следующие показатели:

- 1) Модуль упругости при растяжении, ГПа;
- 2) Прочность при растяжении, МПа;
- 3) Соотношение матрица-наполнитель.

При этом, для показателей «Модуль упругости при растяжении, ГПа» и «Прочность при растяжении, МПа» прогнозирование должно быть реализовано с помощью моделей машинного обучения. Для показателя «Соотношение матрица-наполнитель» прогнозирование предполагается с помощью нейронной сети.

По итогам объединения файлов размер выборки составил 1023 строки. Все значения являются числовыми. В большинстве случаев значения показателей практически уникальны. Исключение составляет показатель «Угол нашивки, град». Он содержит только 2 значения. Полный перечень значений перечислен на рисунке 1.

	Тип данных	Количество_пропусков	Количество_уникальных_значений	Процент_пропусков
id_x	int64	0	1023	0.0
Соотношение матрица-наполнитель	float64	0	1014	0.0
Плотность, кг/м3	float64	0	1013	0.0
модуль упругости, ГПа	float64	0	1020	0.0
Количество отвердителя, м.%	float64	0	1005	0.0
Содержание эпоксидных групп, %_2	float64	0	1004	0.0
Температура вспышки, C_2	float64	0	1003	0.0
Поверхностная плотность, г/м2	float64	0	1004	0.0
Модуль упругости при растяжении, ГПа	float64	0	1004	0.0
Прочность при растяжении, МПа	float64	0	1004	0.0
Потребление смолы, г/м2	float64	0	1003	0.0
id_y	int64	0	1023	0.0
Угол нашивки, град	int64	0	2	0.0
Шаг нашивки	float64	0	989	0.0
Плотность нашивки	float64	0	988	0.0

Рисунок 1 – Характеристики выборки

Пропуски в данной выборке также отсутствуют, в соответствии с рисунком 2.

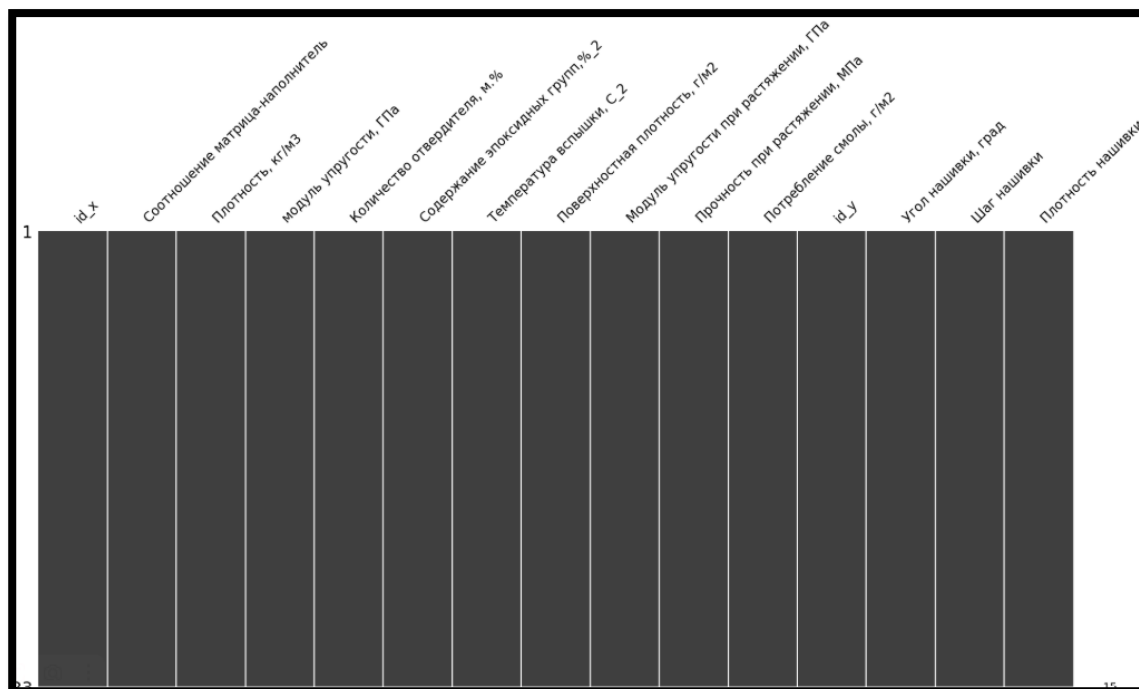


Рисунок 2 – Визуализация отсутствия пропусков.

При просмотре информации на предмет выбросов, было выявлено 87 записей. Относительно общего количества записей, их немного. Позже они будут удалены. Также датасет содержит 2 столбца id, которые были в изначальных файлах. Они не несут в себе полезной информации, т.к. просто дублируют порядковый номер строки. Эти столбцы будут удалены.

1.2 Описание используемых методов.

Поставленная задача относится к разделу машинного обучения «с учителем». Это задачи регрессии. Такие модели учатся на размеченных данных, где для каждого входного экземпляра известен желаемый выходной результат.

В процессе исследования были применены следующие методы:

1. Линейная регрессия (Linear Regression). Это математическая модель, которая описывает связь нескольких переменных. Модели линейной

регрессии представляют собой статистическую процедуру, помогающую прогнозировать будущее. Она применяется в научных сферах и в бизнесе, а в последние десятилетия используется в машинном обучении. Простой и широко применяемый метод регрессии, основанный на поиске линейной связи между признаками и целевой переменной.

- a. К плюсам данного метода можно отнести быстрое обучение и предсказание. Так же метод хорошо изучен и прост в реализации. Метод требует минимум ресурсов и работает быстро даже на больших наборах данных. Может применяться в разных областях и для решения широкого спектра задач
 - b. К минусам можно отнести то, что модель ограничивается линейными отношениями в данных. Также метод чувствителен к выбросам. Когда независимые переменные сильно коррелируют между собой, коэффициенты модели становятся нестабильными.
2. Гамма-регрессор (GammaRegressor). Специальный тип GLM (Generalized Linear Model), предназначенный для регрессии с положительными целевыми переменными и предполагающим гамма-распределение остатков.
 - a. Преимущества данного метода: подходит для позитивных целевых переменных с асимметричным распределением. Устойчивость к влиянию выбросов. Позволяет моделировать широкий диапазон распределений данных благодаря гибкости гамма-распределения.
 - b. Недостатки метода: требуется дополнительная настройка и диагностика модели. Меньше распространённость и поддержка

в инструментах. Предполагает постоянный параметр формы для всех наблюдений, что может не выполняться для всех наборов данных.

3. **ARD Регрессия (Automatic Relevance Determination).** Метод байесовской регрессии, позволяющий определять важность признаков путем автоматического определения релевантности признаков. Метод подходит, когда данных очень мало, и обычные методы уже не работают. Метод сложен с вычислительной точки зрения. Ему необходим огромный объем оперативной памяти.

- а. К плюсам метода относится: возможность отбора значимых признаков. Может использоваться для больших объемов данных. Метод использует только наиболее важные признаки, что может улучшить производительность модели и снизить риск переобучения
- б. К минусам метода можно отнести более сложную реализацию и настройку. Медленное обучение относительно классической линейной регрессии.

4. **AdaBoostRegressor.** Ансамблевый метод, который начинает с обучения регрессора на исходном датасете, а затем обучает дополнительные копии регрессора на том же датасете, но с учётом того, что веса экземпляров корректируются в соответствии с ошибкой текущего предсказания. Метод использует последовательность слабых моделей (обычно деревья решений), которые постепенно улучшают общую модель.

- а. Преимущества метода. Метод объединяет сильные стороны нескольких слабых моделей для создания сильной модели, которая хорошо обобщает данные на невидимые примеры.

- b. К минусам метода относится чувствительность к шуму и выбросам. Так же длительная работа при большом количестве деревьев.
- 5. ExtraTreesRegressor. Один из вариантов ансамбля решающих деревьев, где деревья выращиваются случайно, а потом голосуют для предсказания. Метод создаёт дополнительные деревья случайным образом в подвыборках датасета, чтобы улучшить предсказательность модели. Для этого он усредняет выходы деревьев решений.
 - a. Плюсы: быстрая скорость обучения и предсказания. Устойчивость к шумам и выбросам.
 - b. Минусы: склонность к переобучению при недостаточной глубине дерева. Необходимость правильной настройки гиперпараметров.
- 6. GradientBoostingRegressor. Эффективный ансамбль, построенный на последовательном обучении моделей, каждая следующая из которых улучшает предыдущую. Он работает путём итеративного обучения деревьев решений на остатках предыдущего дерева, эффективно исправляя ошибки предыдущей модели. Этот процесс продолжается до тех пор, пока не будет достигнуто заданное количество деревьев, в результате чего получается высокоточная и надёжная модель.
 - a. Достоинства метода: очень высокая точность на многих наборах данных. Способность обрабатывать большие объёмы данных. Также устойчив к выбросам.
 - b. Недостатки метода: долгое время обучения при большом числе деревьев. Повышенная склонность к переобучению без надлежащей настройки.

7. **BaggingRegressor**. Ансамбль-метод, использующий бутстрэппинг (выборку с возвратом) для уменьшения вариации и повышения точности. Принцип работы: создаётся несколько копий исходного обучающего набора данных с помощью бутстрэпа, на каждой копии обучается отдельная модель, а затем результаты этих моделей объединяются в одну предсказательную модель.
- а. Достоинства метода: устойчивость к переобучению. Работоспособность на высокомерных пространствах. Снижает ошибки базовых алгоритмов, обученных по различным подвыборкам, взаимно компенсируются при голосовании. Эффективен на малых выборках. Исключение даже малой части обучающих объектов приводит к построению существенно различных базовых классификаторов.
 - б. Недостатки метода: сложность параллельного обучения моделей. Увеличенное потребление памяти при большом ансамбле.
8. **CatBoostRegressor**. Это эмулятор для работы с наборами данных регрессии. Он подходит для задач, где требуется предсказать непрерывную величину на основе набора признаков, например, прогнозирование цен на недвижимость, анализ временных рядов, оценка рисков.
- а. Преимущества метода: лучшее обращение с категориальными признаками. Высокие скорости обучения и точности.
 - б. Недостатки метода: потребляет большое количество оперативной памяти. Необходимо аккуратно подбирать гиперпараметры. Также чувствителен к выбросам и имеет склонность к переобучению.

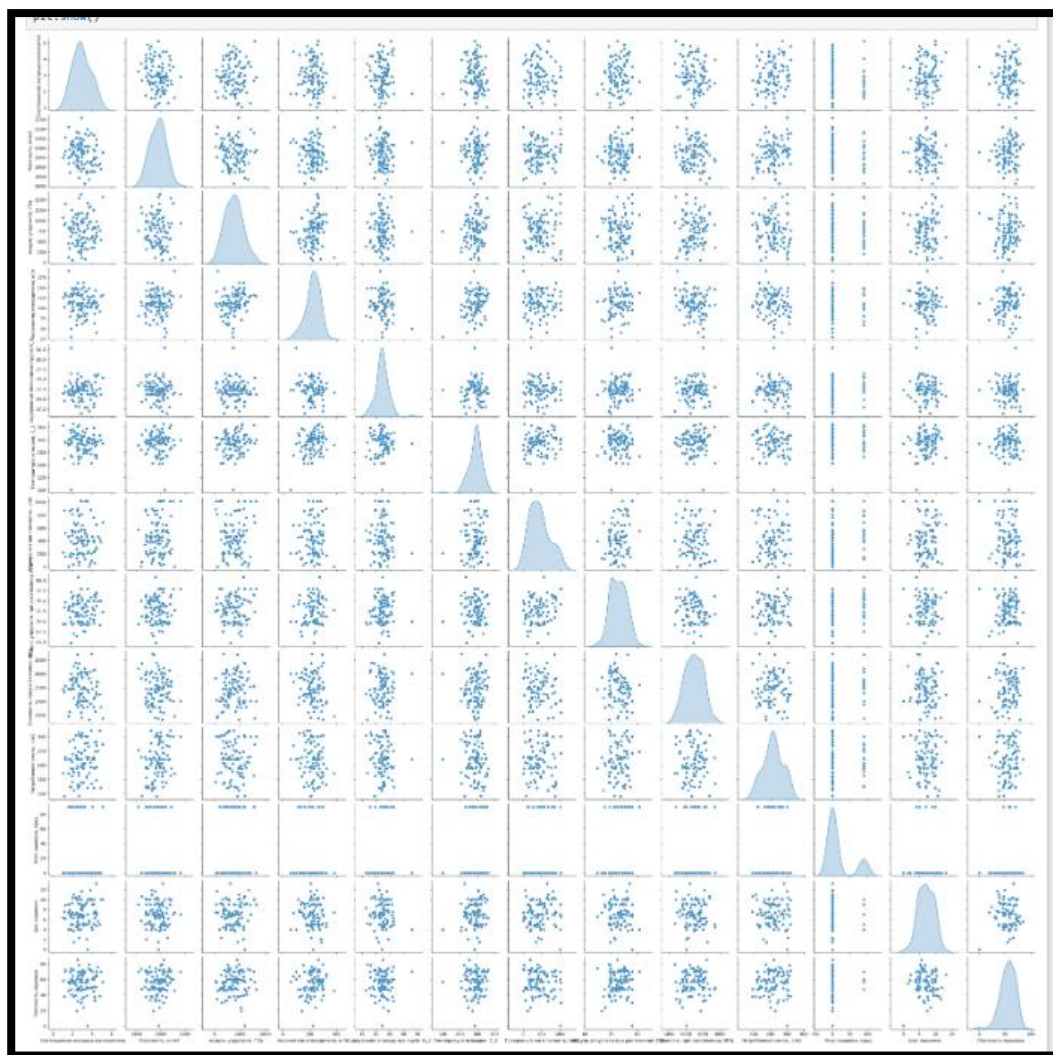
Все перечисленные методы относятся к классу алгоритмов машинного обучения, используемых преимущественно для задач регрессии (за исключением некоторых случаев, когда их можно использовать и для классификации).

1.3 Разведочный анализ данных.

Проведение разведывательного анализа является основным этапом для корректной подготовки данных перед началом машинного обучения. Для этого были использованы диаграммы ящика с усами, рисунок 5, попарные графики рассеяния точек гистограммы распределения каждой из переменной, рисунок 4. Для каждой колонки были получены среднее, медианное значение, рисунок 3.

	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
count	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000
mean	2.930366	1975.734888	739.923233	110.570769	22.244390	285.882151	482.731833	73.328571	2466.922843	218.423144	44.252199
std	0.913222	73.729231	330.231581	28.295911	2.406301	40.943260	281.314690	3.118983	485.628006	59.735931	45.015793
min	0.389403	1731.764635	2.436909	17.740275	14.254985	100.000000	0.603740	64.054061	1036.856605	33.803026	0.000000
25%	2.317887	1924.155467	500.047452	92.443497	20.608034	259.066528	266.816645	71.245018	2135.850448	179.627520	0.000000
50%	2.906878	1977.621657	739.664328	110.564840	22.230744	285.896812	451.864365	73.268805	2459.524526	219.198882	0.000000
75%	3.552660	2021.374375	961.812526	129.730366	23.961934	313.002106	693.225017	75.356612	2767.193119	257.481724	90.000000
max	5.591742	2207.773481	1911.536477	198.953207	33.000000	413.273418	1399.542362	82.682051	3848.436732	414.590628	90.000000

Рисунок 3 – Расчет среднего и медианных значений



4. Рассеивание точек и распределение каждой из переменной

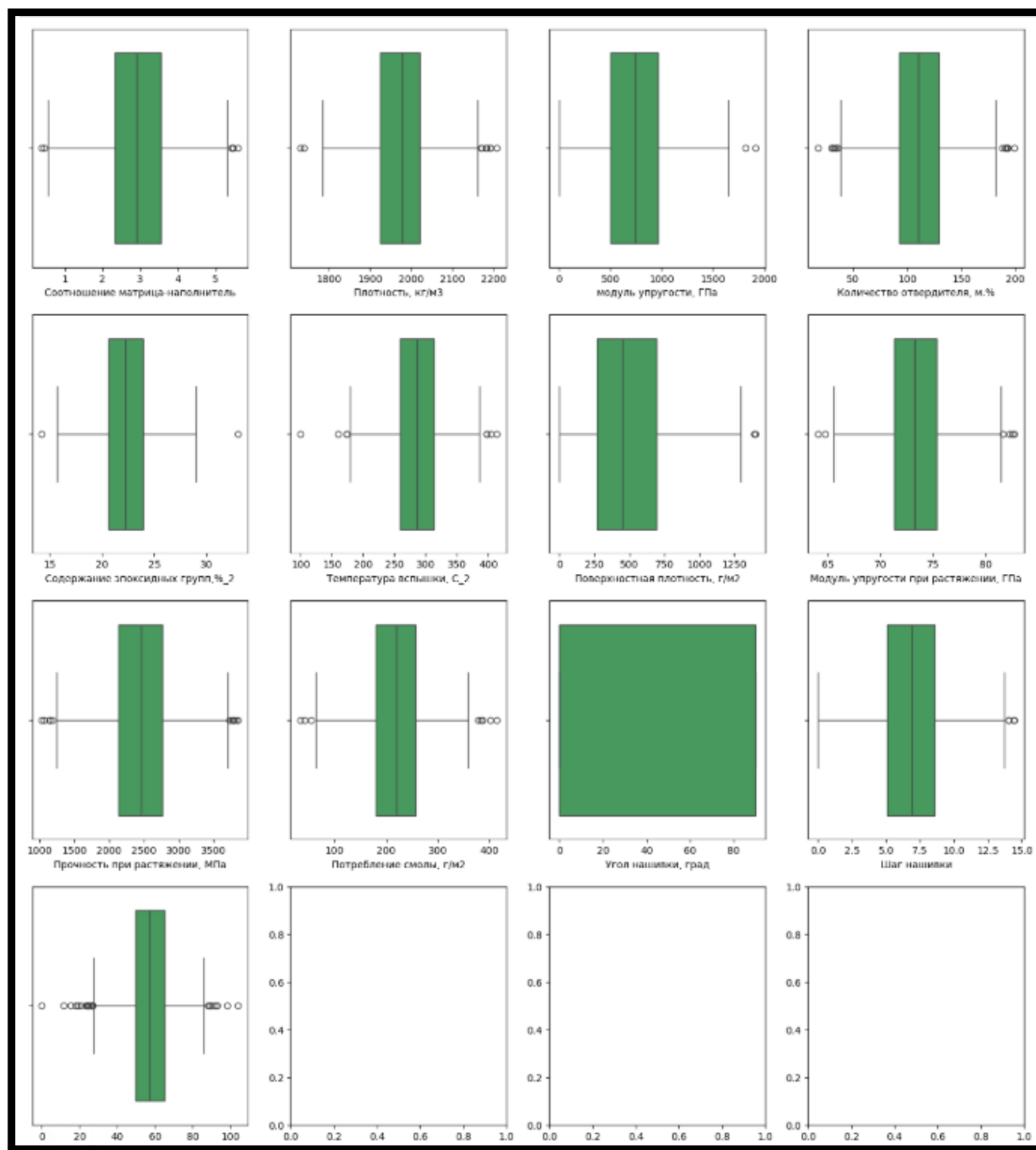


Рисунок 5 – Выбросы

Все выявленные выбросы будут удалены.

Данные не имеют чётко выраженной зависимости, что подтверждает тепловая карта с матрицей корреляции, рисунок 6.

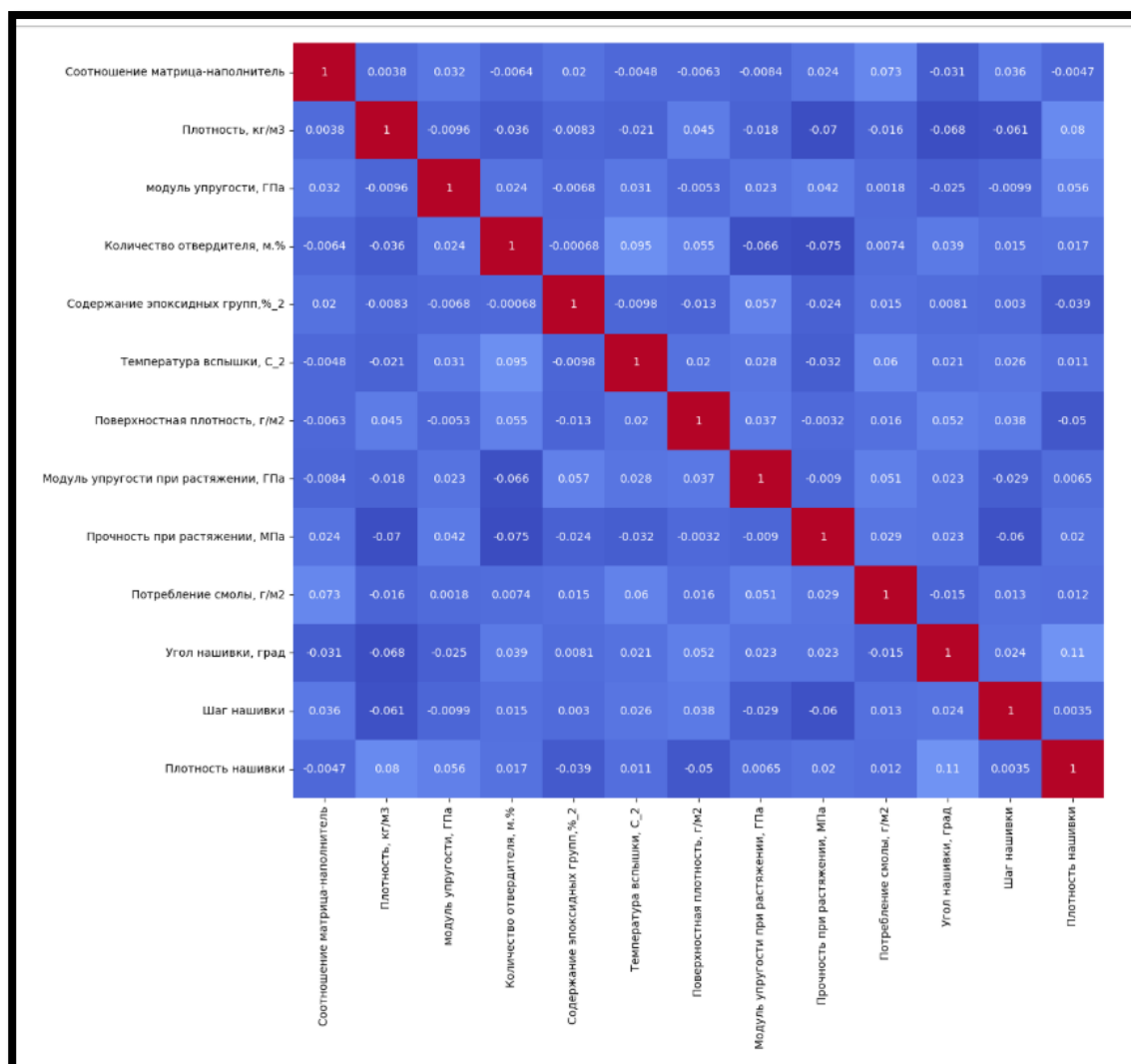


Рисунок 6 – Тепловая карта с матрицей корреляции

Максимальная корреляция, которая была найдена составляет 0.11. Это говорит о том, что нет зависимости между этими данными. Корреляция между всеми параметрами очень близка к 0, корреляционные связи между переменными не наблюдаются.

В рамках уточнений была получена информация, что в датасете есть синтетические данные.

Синтетические данные в машинном обучении — это искусственно созданные данные, которые сохраняют статистические и структурные

свойства реальной информации, но при этом не содержат фактических сведений из окружающего мира.

Проведем работы в части оценки количества синтетических данных в датасете. В датасете очень много нецелочисленных данных. Проверим, какое их количество, и какие метрики покажет датасет без них. Нецелочисленных данных в датасете очень много, практически все, что показано на рисунке 7.

```
def has_zero_fraction(value):  
    decimal_part = str(float(value)).split('.')[1]  
    return int(decimal_part) == 0  
  
zero_decimal_count = {}  
  
for column in Compozit.columns:  
    zero_decimal_count[column] = sum(map(has_zero_fraction, Compozit[column]))  
  
for column, count in zero_decimal_count.items():  
    print(f"В столбце '{column}' найдено {count} строк.")
```

В столбце 'Соотношение матрица-наполнитель' найдено 0 строк.
В столбце 'Плотность, кг/м3' найдено 20 строк.
В столбце 'модуль упругости, ГПа' найдено 18 строк.
В столбце 'Количество отвердителя, м.%' найдено 13 строк.
В столбце 'Содержание эпоксидных групп,%_2' найдено 0 строк.
В столбце 'Температура вспышки, C_2' найдено 12 строк.
В столбце 'Поверхностная плотность, г/м2' найдено 20 строк.
В столбце 'Модуль упругости при растяжении, ГПа' найдено 15 строк.
В столбце 'Прочность при растяжении, МПа' найдено 15 строк.
В столбце 'Потребление смолы, г/м2' найдено 20 строк.
В столбце 'Угол нашивки, град' найдено 936 строк.
В столбце 'Шаг нашивки' найдено 37 строк.
В столбце 'Плотность нашивки' найдено 37 строк.

Рисунок 7 – Количество нецелочисленных записей в столбцах

Создадим датасет с целочисленными значениями. Т.к. таких записей очень мало, то отбор будет по столбцу 'Плотность нашивки', где будут все целочисленные значения данного столбца.

По отобранным данным была создана дополнительная тепловая карта с матрицей корреляции, рисунок 8.

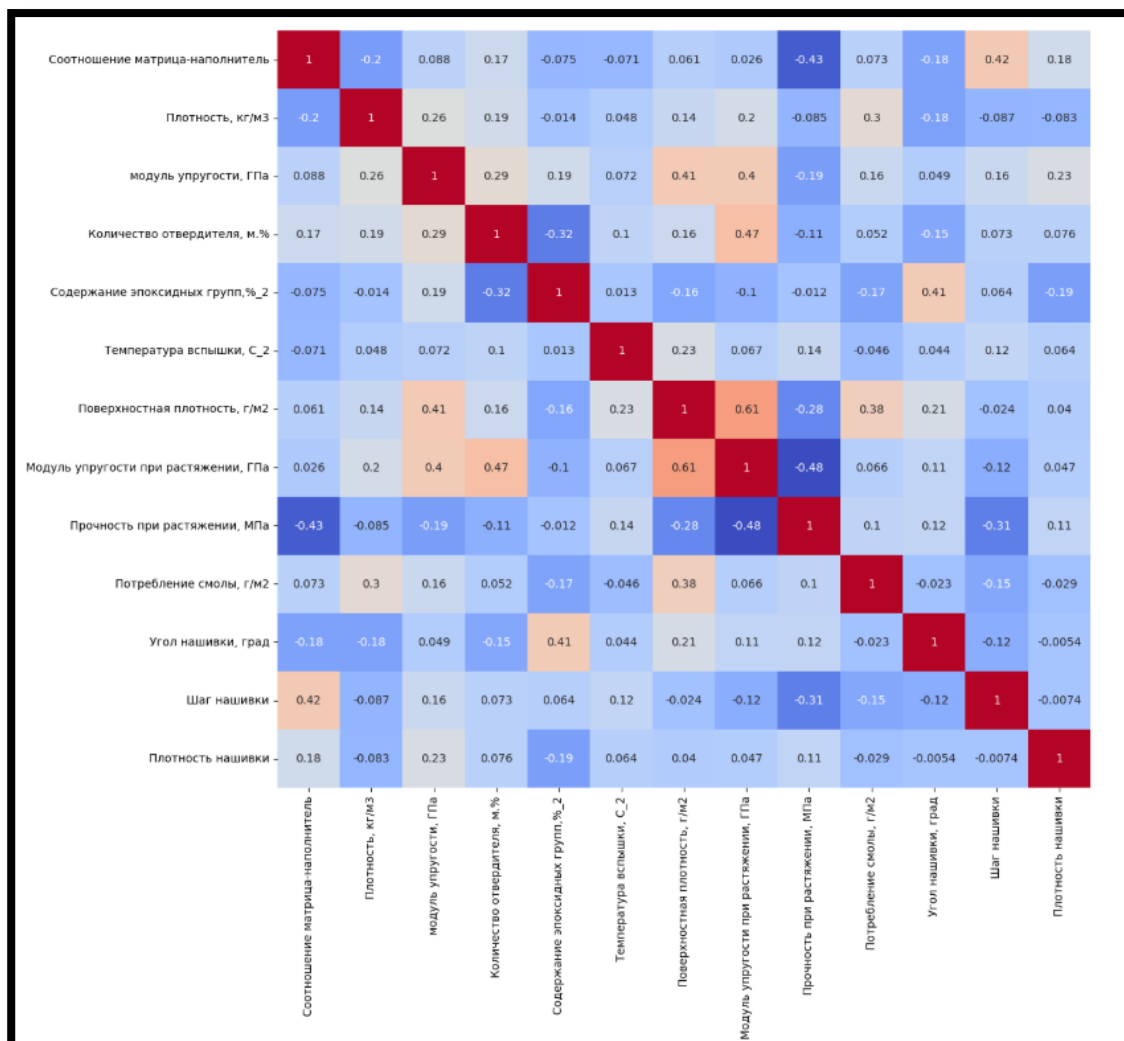


Рисунок 8 – Дополнительная тепловая карта с матрицей корреляции
 Корреляция улучшилась. Это говорит о наличии зависимости между
 этими данными. Дальнейшие работы будут производиться с новым
 датасетом. Текущую выборку сохраняем.

2. Практическая часть

2.1 Предобработка данных.

В ходе проведенного анализа:

1. Значение столбца "Угол нашивки" приводим к виду «0» и «1». Использовался LabelEncoder ().
2. Столбцы «id_x» и «id_y» были удалены из датасета.
3. Удалены все выбросы.
4. Из обработки были исключены практически все нецелочисленные значения. Часть осталась, т.к. иначе в датасете останется мало данных.

2.2 Разработка и обучение модели.

Разработка и обучение моделей машинного обучения осуществляется для двух выходных параметров: «Модуль упругости при растяжении, ГПа» и «Прочность при растяжении, МПа» отдельно. Для решения применим все методы, описанные выше.

В рамках подбора и обучения моделей было выполнено:

- 1) Обучающая и тестовые выборки были разбиты в соотношении 70 на 30.
- 2) Выполнен поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10, рисунок 9;
- 3) Обучение моделей по отобранным гиперпараметрам;

- 4) Сравнение результатов модели с учетом нормализации данных и без.
- 5) В качестве параметра оценки выбран коэффициент детерминации (R2).

```
def my_best_model(X, y_1):
    algos = {
        'LinearRegression': {
            'model': LinearRegression(),
            'params': {}
        },
        'GammaRegressor': {
            'model': GammaRegressor(),
            'params': {}
        },
        'ARDRegression': {
            'model': ARDRegression(),
            'params': {'tol': [1e-3, 1e-2]}
        },
        'AdaBoostRegressor': {
            'model': AdaBoostRegressor(),
            'params': {'n_estimators': [5, 10, 15, 20, 30, 50], 'loss': ['linear', 'square', 'exponential']}
        },
        'BaggingRegressor': {
            'model': BaggingRegressor(),
            'params': {'n_estimators': [10, 50, 100], 'max_samples': [0.5, 1.0], 'max_features': [0.5, 1.0]}
        },
        'ExtraTreesRegressor': {
            'model': ExtraTreesRegressor(),
            'params': {'n_estimators': [20, 50, 100, 200], 'max_depth': [2, 5, 10, 20], 'min_samples_split': [2, 5, 10]}
        },
        'GradientBoostingRegressor': {
            'model': GradientBoostingRegressor(),
            'params': {'n_estimators': [20, 50, 100, 200], 'learning_rate': [0.01, 0.1, 0.5], 'max_depth': [3, 5, 7], 'loss': ['ls', 'lad', 'huber', 'quantile']}
        },
        'CatBoostRegressor': {
            'model': CatBoostRegressor(verbose=False),
            'params': {'iterations': [100, 500], 'learning_rate': [0.0001, 0.001, 0.01, 0.1, 1.0], 'subsample': [0.5, 0.7, 1.0], 'depth': [3, 5, 7]}
        }
    }

    scores = []
    for algo_name, config in algos.items(): # Используем стандартный цикл for
        gs = GridSearchCV(config['model'], config['params'], cv=10, return_train_score=False, scoring='r2')
        gs.fit(X, y_1)
        scores.append({
            'model': algo_name,
            'best_score': gs.best_score_,
            'best_params': gs.best_params_
        })

    return pd.DataFrame(scores, columns=['model', 'best_score', 'best_params'])
```

Рисунок 9 – поиск гиперпараметров модели

2.3 Тестирование модели.

Результат для показателя «Модуль упругости при растяжении, ГПа».

Модели с нормализованными данными показали схожий результат с ненормализованными данными. Результат можно считать положительным, с учетом минимального набора данных. Наилучший результат показала модель

AdaBoostRegressor. Результат по оценке r^2 равен 0,67 и 0,66 соответственно, таблица 1.

Таблица 1. Результаты для «Модуль упругости при растяжении, ГПа»

Модели	Без нормализации			С нормализацией		
	MAE	MSE	r^2	MAE	MSE	r^2
1	2	3	4	5	6	7
1. LinearRegression	2,08	5,75	0,4	2,08	5,75	0,4
2. GammaRegressor	1,84	4,29	0,55	2,62	9	0,06
3. ARDRegression	2,6	8,3	0,13	1,9	4,28	0,55
4. AdaBoostRegressor	1,38	3,20	0,67	1,37	3,23	0,66
5. ExtraTreesRegressor	1,57	4,45	0,54	1,7	4,88	0,49
6. GradientBoostingRegressor	1,34	3,62	0,62	1,44	3,85	0,6
7. BaggingRegressor	1,74	4,93	0,48	1,63	5,92	0,38
8. CatBoostRegressor	2,01	6,98	0,27	2,01	6,98	0,27

Модель будет сохранена и добавлена в разработанное приложение.

Результат для показателя «Прочность при растяжении, МПа».

Модели с нормализованными данными показали схожий результат с ненормализованными данными. Результаты ниже, чем для переменной выше. Наилучший результат показала также модель AdaBoostRegressor. Результат по оценке r^2 равен 0,34 и 0,31 соответственно, таблица 2.

Таблица 2. Результаты для «Прочность при растяжении, МПа».

Модели	Без нормализации			С нормализацией		
	MSE	MAE	r^2	MSE	MAE	r^2
1	2	3	4	5	6	7
1. LinearRegression	623,77	473638,04	-1,05	623,77	473638,04	-1,05
2. GammaRegressor	456,87	262844,88	-0,14	450,56	257126,75	-0,11
3. ARDRegression	449,03	276128,64	-0,19	460,21	278366,25	-0,2
4. AdaBoostRegressor	296,21	152585,1	0,34	330,68	159189,23	0,31
5. ExtraTreesRegressor	337,34	173246,8	0,25	376,45	184493,12	0,2
6. GradientBoostingRegressor	431,24	243393,79	-0,05	430,28	242669,69	-0,05
7. BaggingRegressor	345,91	180277,16	0,22	343,56	194889,73	0,18
8. CatBoostRegressor	346,87	173018,99	0,25	346,87	173018,99	0,25

Модель будет сохранена и добавлена в разработанное приложение.

2.4. Создание нейронной сети.

Нейронная сеть создана для показателя «Соотношение матрица-наполнитель».

Нейронная сеть построена на классе Sequential из библиотеки Keras/Tensorflow, что означает последовательное подключение слоев друг за другом. Входные данные проходят через всю цепочку слоев без обратных связей. Также в нейронной сети использовались скрытые слои. Архитектура полносвязанной сети проста и универсальна.

Нейронная сеть ориентирована на задачу регрессии, где целью является предсказание непрерывного значения. Такая структура с увеличением глубины и последующим сужением (bottleneck-like architecture) эффективна для выявления сложных нелинейных взаимосвязей между признаками и зависимыми переменными.

В слоях используются активации: тангенс гиперболического (tanh), и линейная (linear).

Весовые матрицы инициализируются согласно нормальному распределению Хе (kernel_initializer='he_normal'), а смещённые члены (bias) также инициализируются нормально распределёнными значениями (bias_initializer='he_normal'). Такой подход помогает ускорить обучение и минимизировать проблемы начальной фазы.

Активации:

- 1) Активация tanh сжимает выход в диапазоне $[-1, +1]$. Используется в скрытых слоях.
- 2) Последний слой использует линейную активацию (linear), что актуально для задач регрессии, где ожидается непрерывный выход.

Используется оптимизатор ADAM (Adam()), который сочетает достоинства моментов и адаптивного шага обучения. Он хорошо справляется с задачей ускорения обучения и стабилизации сходимости.

Критерий ошибки измеряется по среднему квадратичному отклонению (mean_squared_error), что характерно для задач регрессии. В качестве основного параметра оценки выбран коэффициент детерминации (R2).

Для исключения возможности переобучения был применен гиперпараметр callbacks.

Была создана нейросеть. Пример модели отражен на рисунке 10.

```
def baseline_model():
    model = Sequential()
    normalizer
    model.add(Dense(20, input_dim=10, activation='tanh', bias_initializer='he_normal', kernel_initializer='he_normal'))
    model.add(Dense(150, input_dim=20, activation='tanh', bias_initializer='he_normal', kernel_initializer='he_normal'))
    model.add(Dense(150, input_dim=150, activation='tanh', bias_initializer='he_normal', kernel_initializer='he_normal'))
    model.add(Dense(100, input_dim=150, activation='tanh', bias_initializer='he_normal', kernel_initializer='he_normal'))
    model.add(Dense(20, input_dim=100, activation='tanh', bias_initializer='he_normal', kernel_initializer='he_normal'))
    model.add(Dense(1, input_dim=20, activation='linear', bias_initializer='he_normal', kernel_initializer='he_normal'))

    adam = Adam()
    model.compile(loss='mean_squared_error', optimizer='adam')
    return model

model = baseline_model()
```

Рисунок 10 - Модель нейронной сети.

Результат для показателя «Соотношение матрица-наполнитель». Модель с нормализованными данными не дала результата. Значение по оценке r2 равно -0,48.

Другие оценки модели:

- 1) MSE – 0.53;
- 2) MAE – 0.57.

С использованием гиперпараметра callbacks, процесс обучения нейронной сети остановился на 12 эпохе, рисунок 11.

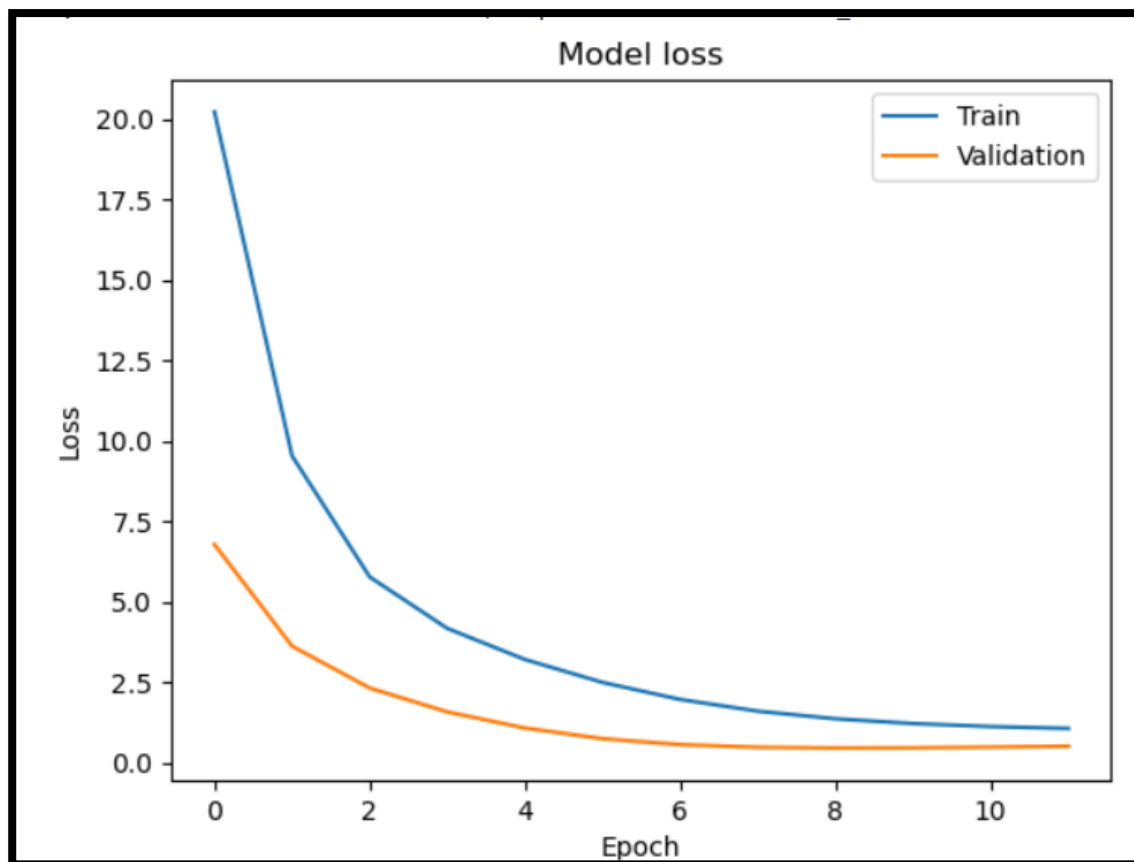


Рисунок 11 – Визуализация процесса обучения.

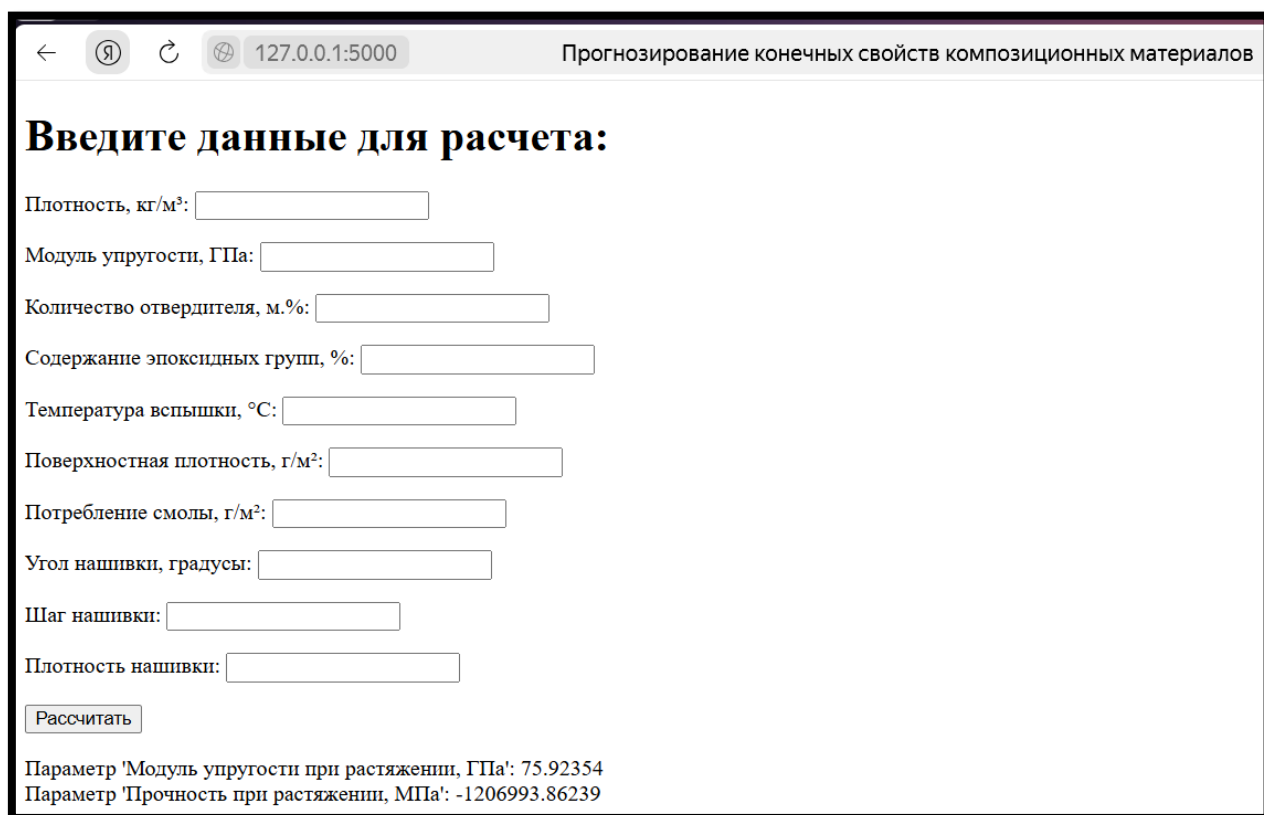
2.5. Разработка приложения.

Приложение было разработано на платформе Flask.

Flask — это web-фреймворк для разработки приложений на языке программирования Python. Он известен своей простотой и минимальным набором базовых компонентов, предоставляя разработчику свободу в выборе сторонних библиотек и расширений для удовлетворения специфичных требований проекта.

Flask предоставляет только базовые возможности, необходимые для запуска веб-приложения. Разработчик волен добавлять дополнительные компоненты по мере необходимости. Благодаря модульной структуре, Flask

легко интегрируется с внешними инструментами и библиотеками, такими как базы данных, системы аутентификации, шаблоны и многое другое. Изучение и начало работы с Flask занимают минимум времени, что делает его идеальным решением для прототипирования и небольших проектов. Результатом разработки приложения стала web-форма с получения информации на основании введенных данных, рисунок 12.



← 127.0.0.1:5000 Прогнозирование конечных свойств композиционных материалов

Введите данные для расчета:

Плотность, кг/м³:

Модуль упругости, ГПа:

Количество отвердителя, м. %:

Содержание эпоксидных групп, %:

Температура вспышки, °C:

Поверхностная плотность, г/м²:

Потребление смолы, г/м²:

Угол нашивки, градусы:

Шаг нашивки:

Плотность нашивки:

Параметр 'Модуль упругости при растяжении, ГПа': 75.92354
Параметр 'Прочность при растяжении, МПа': -1206993.86239

Рисунок 12 – форма для ввода и получения расчетных значений. На форме предусмотрена проверка на обязательность заполнения всех полей. Она отрабатывает по нажатию на кнопку «Рассчитать». Если хотя бы одно поле не заполнено, то появится информационно сообщение, рисунок 13. Для ввода доступны только числовые значения.

Рисунок 13 – Сообщение об обязательности заполнения поля.

Приложение доступно по пути: <https://ostroverkh-vkr-10.onrender.com/>

2.6. Создание удаленного репозитория.

Была создана учетная запись на ресурсе <https://github.com/>.

Все данные по проектной работе были размещены по пути:

https://github.com/VlslimirOstroverkh/Ostroverkh_VKR/

Заключение

Данная исследовательская работа позволяет сделать некоторые основные выводы по теме. Распределение полученных данных в объединённом датасете близко к нормальному, но коэффициенты корреляции между парами признаков стремятся к нулю. Если исключить все синтетические данные, то коэффициенты корреляции становятся лучше. Полностью уйти от синтетических данных не получилось из-за того, что количество таких значений минимально. В данном случае синтетические данные не способствовали повышению обучаемости моделей. Используемые при разработке моделей подходы позволили получить приемлемый результат только по одному выходному параметру из трех. Лучшие метрики для выходных параметров «Модуль упругости при растяжении, ГПа» и «Прочность при растяжении, МПа» показала модель AdaBoostRegressor.

Был сделан вывод, что невозможно определить из свойств материалов «Соотношение матрица-наполнитель». Данный факт не указывает на то, что прогнозирование характеристик композитных материалов на основании предоставленного набора данных невозможно, но может указывать на недостатки базы данных, подходов, использованных при прогнозе, необходимости пересмотра инструментов для прогнозирования.

Необходимы дополнительные вводные данные, получение новых результирующих признаков в результате математических преобразований, консультации экспертов предметной области, новые исследования.

Библиографический список

- 1) Вигерс, К. и Битти Д. Разработка требований к программному обеспечению. Практические приемы сбора требований и управления ими при разработке программных продуктов. – 3е изд., дополненное – СПб.: Русская редакция, 2014. – 718с.
- 2) Платформа Sicit-learn. GradientBoostingRegressor [Электронный ресурс] : – Режим доступа: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html>. (дата обращения: 02.05.2025).
- 3) Платформа Sicit-learn. ExtraTreesRegressor [Электронный ресурс] : – Режим доступа: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html>. (дата обращения: 02.05.2025).
- 4) Платформа Sicit-learn. AdaBoostRegressor [Электронный ресурс] : – Режим доступа: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html>. (дата обращения: 02.05.2025).
- 5) Платформа Sicit-learn. GammaRegressor [Электронный ресурс] : – Режим доступа: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.GammaRegressor.html. (дата обращения: 02.05.2025).
- 6) Платформа Sicit-learn. BaggingRegressor [Электронный ресурс] : – Режим доступа: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingRegressor.html> (дата обращения: 02.05.2025).

- 7) Платформа Sicit-learn. ARDRegression [Электронный ресурс] : – Режим доступа: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ARDRegression.html (дата обращения: 02.05.2025).
- 8) Гистограммы и графики распределения в Python [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/companies/skillfactory/articles/683738/> (дата обращения: 02.05.2025).
- 9) Scikit-Learn AdaBoostRegressor Model [Электронный ресурс] : – Режим доступа: <https://sklearn.com/scikit-learn-adaboostregressor/> (дата обращения: 02.05.2025).
- 10) Платформа Sicit-learn. Линейные модели [Электронный ресурс] : – Режим доступа: https://scikit-learn.ru/stable/modules/linear_model.html (дата обращения: 02.05.2025).
- 11) Регрессия с несколькими выводами в машинном обучении [Электронный ресурс] : – Режим доступа: <https://www.geeksforgeeks.org/multioutput-regression-in-machine-learning/> (дата обращения: 06.04.2025).
- 12) Keras: API высокого уровня для TensorFlow [Электронный ресурс] : – Режим доступа: <https://www.tensorflow.org/guide/keras?hl=ru> (дата обращения: 06.04.2025).
- 13) Учимся Flask: как написать своё веб-приложение [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/companies/otus/articles/886390/> (дата обращения: 20.05.2025).

- 14) Документация Flask [Электронный ресурс] : – Режим доступа: <https://flask.palletsprojects.com/en/stable/> (дата обращения: 20.05.2025).
- 15) Документация seaborn [Электронный ресурс] : – Режим доступа: <https://seaborn.pydata.org/generated/seaborn.histplot.html> (дата обращения: 20.04.2025).
- 16) Документация matplotlib [Электронный ресурс] : – Режим доступа:
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html
(дата обращения: 20.04.2025).
- 17) Документация NumPy [Электронный ресурс] : – Режим доступа: https://numpy.org/doc/stable/user/absolute_beginners.html (дата обращения: 20.04.2025).
- 18) Документация pandas [Электронный ресурс] : – Режим доступа:
<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html#pandas.DataFrame> (дата обращения: 20.04.2025).
- 19) Документация Missingno [Электронный ресурс] : – Режим доступа: <https://360digitmg.com/blog/missingno> (дата обращения: 20.04.2025).
- 20) Документация Python [Электронный ресурс] : – Режим доступа: <https://docs.python.org/3/tutorial/index.html> (дата обращения: 20.04.2025).
- 21) Документация Python [Электронный ресурс] : – Режим доступа: <https://metanit.com/python/tutorial/> (дата обращения: 20.04.2025).

- 22) Введение в модели и слои бэкэнда Keras [Электронный ресурс] : – Режим доступа:
https://proproprogs.ru/neural_network/keras-vvedenie-v-modeli-i-sloi-bekenda-keras (дата обращения: 20.04.2025).
- 23) Основы нейронных сетей [Электронный ресурс] : – Режим доступа: <https://www.dmitrymakarov.ru/intro/neural/> (дата обращения: 20.04.2025).
- 24) Как начать работу с Keras [Электронный ресурс] : – Режим доступа: <https://timeweb.com/ru/community/articles/kak-nachat-rabotu-s-keras> (дата обращения: 20.04.2025).
- 25) Как научиться писать на HTML [Электронный ресурс] : – Режим доступа: <https://dzen.ru/a/W73BAIMbRQCqIwbH> (дата обращения: 20.05.2025).
- 26) Основы HTML [Электронный ресурс] : – Режим доступа: <https://abuzov.com/osnovy-html/> (дата обращения: 20.05.2025).
- 27) Flask для начинающих [Электронный ресурс] : – Режим доступа: <https://habr.com/ru/articles/783574/> (дата обращения: 20.05.2025).
- 28) Основы Flask [Электронный ресурс] : – Режим доступа: <https://pythonru.com/uroki/3-osnovy-flask> (дата обращения: 20.05.2025).
- 29) Фреймворк Flask в Python [Электронный ресурс] : – Режим доступа: <https://pythonim.ru/libraries/freymvork-flask-v-python> (дата обращения: 20.05.2025).
- 30) Flask [Электронный ресурс] : – Режим доступа: <https://help.sprinthost.ru/framework/flask> (дата обращения: 20.05.2025).