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 2022 4.0»

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

Слушатель Соколовский Болеслав Леонидович

Москва, 2023

Содержание

Введение

Данная работа выполнена в рамках курса Data Science.

Тема выпускной квалификационной работы: Прогнозирование конечных свойств новых материалов (композиционных материалов).

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, то есть 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 известны 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).

1 Аналитическая часть

1.1 Постановка задачи

Цель исследовательской работы – разработать модели для прогноза модуля упругости при растяжении, прочности при растяжении и соотношения «матрица-наполнитель».

В качестве входных данных были приняты данные о начальных свойствах и параметрах компонентов композиционных материалов:

- соотношение матрица-наполнитель;
- плотность;
- модуль упругости;
- количество отвердителя;
- содержание эпоксидных групп;
- температура вспышки;
- поверхностная плотность;
- модуль упругости при растяжении;
- прочность при растяжении;
- потребление смолы;
- угол нашивки;
- шаг нашивки;
- плотность нашивки.

На выходе необходимо спрогнозировать ряд конечных свойств получаемых композиционных материалов.

Кейс основан на реальных производственных задачах Центра НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана).

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

1.2 Описание используемых методо

Так как поставленные задачи прогнозирования параметров модуля упругости при растяжении и прочности при растяжении, а также прогноза рекомендованного значения соотношения «матрица-наполнитель» являются задачей регрессии, для их решения были выбраны следующие методы машинного обучения с учителем:

1) Линейная регрессия;
2) Полиномиальная регрессия;
3) KNeighborsRegressor;
4) RandomForestRegressor;
5) GradientBoostingRegressor;
6) Полносвязная нейронная сеть (для прогноза рекомендованного значения соотношения «матрица-наполнитель»).

Линейные модели представляют собой класс моделей, которые давно и широко используются на практике. Линейные модели делают прогноз, используя линейную функцию входных признаков.

Линейная регрессия — регрессионная модель зависимости одной (целевой, таргета, зависимой) переменной y от другой или нескольких других переменных (факторов, предикторов, регрессоров, независимых переменных) x с линейной функцией зависимости.

Существует различные виды линейных моделей для регрессии, основное различие между которыми заключается в способе оценивания параметров модели по обучающим данным и контроле сложности модели. В библиотеке sklearn есть несколько классов, реализующих линейную регрессию:

1) LinearRegression — "классическая" линейная регрессия с оптимизацией MSE;
2) Ridge — регрессия с оптимизацией MSE и L2 – регуляризацией;
3) LASSO — регрессия с оптимизацией MSE и L1 – регуляризацией.

Необходимо заметить, что метод регрессии лассо (LASSO, Least Absolute Shrinkage and Selection Operator) — это вариация линейной регрессии, специально адаптированная для данных, которые демонстрируют сильную корреляцию признаков друг с другом. Очевидно ее применение необоснованно к нашему датасету, ввиду отсутствия корреляции признаков друг с другом. Аналогично и Ridge-регрессия.

Линейная регрессия это один из самых простых и эффективных инструментов статистического анализа и машинного обучения. Она определяет зависимость переменных с помощью линии наилучшего соответствия. Модель регрессии создаёт несколько метрик. R2, или коэффициент детерминации, позволяет измерить, насколько модель может объяснить дисперсию данных, какая доля изменчивости целевой переменной объясняется с помощью модели. Если R-квадрат равен 1, это значит, что модель описывает все данные. Если же R-квадрат равен 0.5, модель объясняет лишь 50 процентов дисперсии данных. Оставшиеся отклонения не имеют объяснения. Чем ближе R2 к единице, тем лучше.

Линейная регрессия находит параметры, которые минимизируют среднеквадратическую ошибку (mean squared error) между спрогнозированными и фактическими ответами в обучающем наборе. Среднеквадратичная ошибка равна сумме квадратов разностей между спрогнозированными и фактическими значениями.

Преимущества линейных моделей:

1) очень быстро обучаются;
2) быстро прогнозируют;
3) масштабируются на очень большие наборы данных;
4) хорошо интерпретируются;
5) по полученным коэффициентам регрессии можно судить о том, как тот или иной фактор влияет на результат, сделать на этой основе дополнительные полезные выводы;

6) широкая применимость.

Недостатки линейных моделей:

1) в низкоразмерном пространстве альтернативные модели могут показать более высокую обобщающую способность;
2) у простой линейной регрессии нет инструментов, позволяющих контролировать сложность модели;
3) линейная регрессия очень чувствительна к выбросам;
4) колинеарность предикторов негативно влияет на точность модели.

Данный алгоритм для решения поставленной задачи был выбран потому, что хорошо работает в условиях низкой корреляции между предикторами и отсутствия резко отклоняющихся значений. Значимых корреляций между предикторами в исследуемых данных выявлено не было, а выбросы были удалены на этапе обработки данных.

Метод k-ближайших соседей. В случае использования метода k-ближайших соседей для задачи регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны. Данный метод был выбран потому, что является хорошим базовым алгоритмом, который имеет смысл попробовать, прежде чем рассматривать альтернативные, более сложные методы. Важной составляющей предварительной обработки данных для этого метода является нормализация, которая и была проведена, так как значения расстояния могут сильно зависеть от атрибутов с бóльшими диапазонами.

Преимущества метода k-ближайших соседей:

1) простота реализации;
2) легко интерпретируются;
3) метод дает приемлемое качество без необходимости использования большого количества настроек;
4) быстро работает на небольших объемах данных;
5) не чувствителен к выбросам.

Недостатки метода k-ближайших соседей:

1) требует обязательной предварительной обработки данных, в частности нормализации;
2) при увеличении объема выборки алгоритм работает значительно медленнее;
3) не слишком хорошо работает на наборах данных с большим числом признаков;
4) плохо работает с разреженными наборами данных, когда большинство признаков в значительной части наблюдений имеют пропуски и нулевые значения;
5) для корректной работы требует обязательного определения оптимального значение k – количества ближайших соседей;
6) зависимость от выбранной метрики расстояния между примерами.

«Случайный лес» (Random Forest). Случайный лес — алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев. Основным недостатком одиночных деревьев решений является их склонность к переобучению. Случайный лес является одним из путей решения данной проблемы. Каждое из используемых в ансамбле деревьев само по себе даёт невысокое качество, но за счёт их большого количества получается необходимый результат.

Алгоритм случайного леса сочетает в себе две основные идеи: метод бэггинга и метод случайных подпространств.

Это такой метод построения ансамбля моделей, в котором обучение базовых моделей производится параллельно. При этом каждая модель обучается на отдельной выборке, сформированной из исходного набора данных, а выход ансамбля деревьев определяется посредством усреднения выходов базовых моделей.

Смысл случайного леса заключается в том, что каждое отдельное дерево может давать неплохой прогноз, но скорее всего переобучится на части данных. При объединении деревьев, которые хорошо работают и переобучаются с разной степенью, можно уменьшить переобучение путем усреднения их результатов.

Преимущества метода «Случайный лес» (Random Forest):

1) способность эффективно обрабатывать данные с большим числом признаков;
2) метод обладает высокой прогнозной силой;
3) не требуется масштабирование данных;
4) одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки;
5) гибкость;
6) очень высокая точность.

Недостатки метода «Случайный лес» (Random Forest):

1) большой размер получающихся моделей;
2) построение леса занимает много времени;
3) случайный лес требует больше памяти и медленнее прогнозирует, чем линейные модели;
4) чем больше объем, тем меньше интерпретируемость;
5) случайный лес плохо работает на разреженных данных.

Данный метод был выбран в силу своей высокой прогнозной способности, гибкой настройке параметров и отсутствия разреженности в исходных данных.

Градиентный бустинг деревьев регрессии. Градиентный бустинг это метод построения ансамбля моделей, при котором базовые модели обучаются последовательно, и каждая последующая модель ансамбля применяется к результатам на выходе предыдущей.

Каждое отдельное дерево решений может сделать хорошие прогнозы только для части данных, таким образом, для итеративного улучшения качества добавляется все большее количество деревьев. Метод градиентного бустинга реализует последовательную композицию алгоритмов обучения, в которой каждый алгоритм должен компенсировать ошибки, допущенные ансамблем предыдущих алгоритмов.

По умолчанию в градиентном бустинге деревьев регрессии отсутствует случайность, вместо этого применяется строгая предварительная обрезка.

В отличие от случайного леса градиентный бустинг чуть более чувствителен к настройке гиперпараметров.

Преимущества метода градиентный бустинг:

1) высокая обобщающая способность;
2) гибкость;
3) универсальность;
4) не требует предварительного масштабирования.

Недостатки метода градиентный бустинг:

1) требует тщательной настройки параметров;
2) требует много времени для обучения;
3) плохо работает на высокоразмерных разреженных данных;
4) последние деревья в ансамбле обучаются в основном на самых проблемных примерах, содержащих ошибки, что снижает точность ансамбля.

Данный метод также был выбран из-за своей высокой прогнозной силы, гибкости в настройке гиперпараметров, способности давать результаты там, где не справляются другие модели.

Полносвязная нейронная сеть. Обучение нейронной сети заключается в нахождении весов — коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными параметрами и выходными, а также выполнять обобщение. В случае успешного обучения нейросеть способна выдать правильный результат на основании данных, которые отсутствовали в обучающей выборке, а также на неполных или «зашумленных» данных.

Преимущества нейронных сетей:

1) мощная прогнозная способность;
2) способность обрабатывать информацию, содержащуюся в больших объемах данных, и строить сложные модели;
3) зачастую превосходят другие алгоритмы машинного обучения.

Недостатки нейронных сетей

1) длительное время обучения, особенно для крупных нейросетей;
2) необходимость тщательной предобработки данных;
3) обязательность масштабирования для корректной работы;
4) необходимость тонкой настройки гиперпараметров;
5) требовательность к вычислительным ресурсам.

Таблица 1 – Итоговая сравнительная таблица используемых методов.

Метод	Достоинства	Недостатки
1	2	3
Линейные модели	• быстро обучаются и прогнозируют; • масштабируются на очень большие наборы данных; • хорошо интерпретируются; • по полученным коэффициентам регрессии можно судить о том, как тот или иной фактор влияет на результат; • широкая применимость.	• в низкоразмерном пространстве альтернативные модели могут показать более высокую обобщающую способность; • у простой линейной регрессии нет инструментов для контроля сложности модели; • чувствительна к выбросам; • колинеарность предикторов негативно влияет на точность модели.

Продолжение таблицы 1

1	2	3
Метод k-ближайших соседей	• прост в реализации; • легко интерпретируются; • нет необходимости в использовании	• требует обязательной предобработки данных; • при увеличении объема выборки работает значительно медленнее;

	большого количества настроек; • быстро работает на небольших объемах данных; • нечувствителен к выбросам.	• плохо работает с высокоразмерными и разреженными наборами данных; • требует обязательного определения оптимального значение k – количества ближайших соседей; • сильно зависит от выбранной метрики расстояния между примерами.
«Случайный лес»	• эффективно обрабатывает данные с большим числом признаков; • обладает высокой прогнозной силой; • не требует масштабирования данных; • хорошо обрабатывает как непрерывные, так и дискретные признаки.	• большой размер получающихся моделей; • построение леса занимает много времени; • требует больше памяти и медленнее прогнозирует, чем линейные модели; • чем больше объем, тем меньше интерпретируемость. • плохо работает на разреженных данных.

Продолжение таблицы 1

1	2	3
Градиентный бустинг	• высокая обобщающая способность; • гибкость; • универсальность;	• требует тщательной настройки параметров; • требует много времени для обучения;

	• не требуется предварительного масштабирования.	• плохо работает на высокоразмерных и разреженных данных; • последние деревья в ансамбле обучаются в основном на самых проблемных примерах, содержащих ошибки, что снижает точность ансамбля.

1	2	3
Полносвязная нейронная сеть	• мощная прогнозная способность; • способность обрабатывать информацию, содержащуюся в больших объемах данных, и строить сложные модели; • зачастую превосходят другие алгоритмы машинного обучения.	• длительное время обучения, особенно для крупных нейросетей; • необходимость тщательной предобработки данных; • обязательность масштабирования для корректной работы; • необходимость тонкой настройки гиперпараметров; • требовательность к вычислительным ресурсам.

1.3 Разведочный анализ данных

Разведочный анализ — это предварительное исследование данных с целью определения их основных характеристик, взаимосвязей между признаками,

наиболее общих зависимостей, закономерностей, законов распределения анализируемых данных, нахождения аномалий, построения для этого различных визуализаций.

Для исследовательской работы были даны два файла с датасетами: X_bp, состоящий из 1024 строки и 11 колонок и X_nup, состоящий из 1041 строки и 4 колонок. Соответственно, датасет df_bp имеет 10 признаков, 1023 записи. Датасет df_nup имеет 3 признака, 1040 записей (в соответствии с рисунком 1).

```
df1 = pd.read_excel(r'Датасет для ВКР_композиты\X_bp.xlsx')
df2 = pd.read_excel(r'Датасет для ВКР_композиты\X_nup.xlsx')

df1.info()

 <class 'pandas.core.frame.DataFrame'>
 RangeIndex: 1023 entries, 0 to 1022
 Data columns (total 11 columns):
  #   Column                             Non-Null Count  Dtype
 ---  ------                             --------------  -----
  0   Unnamed: 0                         1023 non-null   int64
  1   Соотношение матрица-наполнитель    1023 non-null   float64
  2   Плотность, кг/м3                   1023 non-null   float64
  3   модуль упругости, ГПа              1023 non-null   float64
  4   Количество отвердителя, м.%        1023 non-null   float64
  5   Содержание эпоксидных групп,%_2    1023 non-null   float64
  6   Температура вспышки, С_2           1023 non-null   float64
  7   Поверхностная плотность, г/м2      1023 non-null   float64
  8   Модуль упругости при растяжении, ГПа  1023 non-null   float64
  9   Прочность при растяжении, МПа      1023 non-null   float64
  10  Потребление смолы, г/м2            1023 non-null   float64
 dtypes: float64(10), int64(1)
 memory usage: 88.0 KB
```

```
df2.info()

 <class 'pandas.core.frame.DataFrame'>
 RangeIndex: 1040 entries, 0 to 1039
 Data columns (total 4 columns):
  #   Column              Non-Null Count  Dtype
 ---  ------              --------------  -----
  0   Unnamed: 0          1040 non-null   int64
  1   Угол нашивки, град  1040 non-null   int64
  2   Шаг нашивки         1040 non-null   float64
  3   Плотность нашивки   1040 non-null   float64
 dtypes: float64(2), int64(2)
 memory usage: 32.6 KB
```

Рисунок 1 – Размерность исходных датасетов.

В результате предобработки данных, датасеты df_bp и df_nup были объединены по индексу, тип объединения INNER. Часть данных (17 записей), была удалена исходя из метода объединения. Также была удалена колонка «Unnamed: 0».

Объединенный датасет имеет 1023 записи. Общее количество признаков составило 12 признаков.

```
df = pd.merge(df1,df2,how = 'inner',on=['Unnamed: 0', 'Unnamed: 0'])
df.drop(['Unnamed: 0'], axis=1, inplace=True)

df.info()
```

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

Рисунок 2 – Размерность итогового датасета.

В качестве методов и инструментов разведочного анализа данных были использованы:

1) Описательные статистики (с помощью команды .describe().T):

```
df.describe().T
```

1-11 13 rows × 8 columns

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660	5.591742
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375	2207.773481
модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	500.047452	739.664328	961.812526	1911.536477
Количество отвердителя, м.%	1023.0	110.570769	28.295911	17.740275	92.443497	110.564840	129.730366	198.953207
Содержание эпоксидных групп,%_2	1023.0	22.244390	2.406301	14.254985	20.608034	22.230744	23.961934	33.000000
Температура вспышки, С_2	1023.0	285.882151	40.943260	100.000000	259.066528	285.896812	313.002106	413.273418
Поверхностная плотность, г/м2	1023.0	482.731833	281.314690	0.603740	266.816645	451.864365	693.225017	1399.542362
Модуль упругости при растяжении, ГПа	1023.0	73.328571	3.118983	64.054061	71.245018	73.268805	75.356612	82.682051
Прочность при растяжении, МПа	1023.0	2466.922843	485.628006	1036.856605	2135.850448	2459.524526	2767.193119	3848.436732
Потребление смолы, г/м2	1023.0	218.423144	59.735931	33.803026	179.627520	219.198882	257.481724	414.590628
Угол нашивки, град	1023.0	44.252199	45.015793	0.000000	0.000000	0.000000	90.000000	90.000000

Рисунок 3 – Описательные статистики датасета.

В результате проверки были получены основные описательные статистики для каждой переменной: количество, среднее значение, стандартное отклонение, минимальное и максимальное значения, медиана, 75-й и 25-й перцентили. Кроме того, была выявлена величина, принимающая только два значения: 0 и 90 градусов (параметр «Угол нашивки»). Так как данный параметр принимает только два значения, по сути он может считаться категориальным, или бинарной переменной, принимающей ровно два значения.

2) Проверка, используя библиотеку ydata-profiling для удобства:

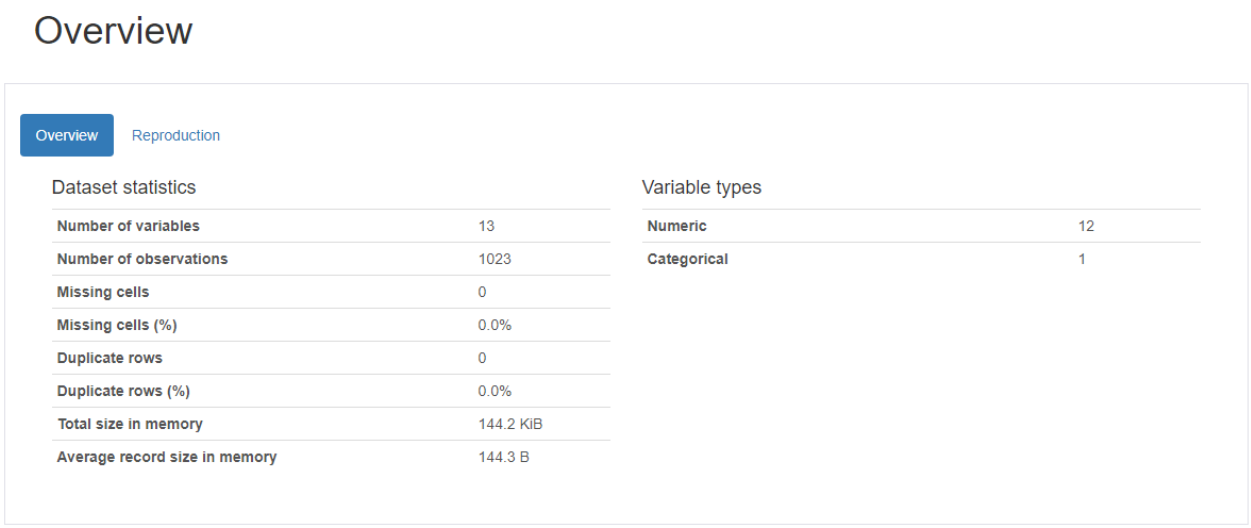


Рисунок 4 – Отчет ydata-profiling (команда RpofileReport).

В результате проверки пропусков и дубликатов в данных не выявлено. Кроме того, была выявлена величина, принимающая только два значения: 0 и 90 градусов (параметр «Угол нашивки»). Можно было бы предположить, что это категориальный признак, так как в предоставленном датасете он имеет всего 2 уникальных значения. Вероятно, в реальных условиях встречаются и другие углы использования нашивки, поэтому будем считать его вещественным числом. Тем более, что при дальнейшей стандартизации этот признак и так примет значения 0 и 1. Отчет ydata-profiling предоставляет исчерпывающую информацию по датасету и можно было бы ограничиться только им, но в угоду последовательности и академичности изложения в дальнейшем будем

использовать классические методы. Но отметим, что «под капотом» у ydata-profiling именно они и используются.(duplicated().sum(); isna().sum(); и.т.д)

3) Определение закона распределения каждой переменной

Для определения соответствия распределения нормальному закону были построены гистограммы

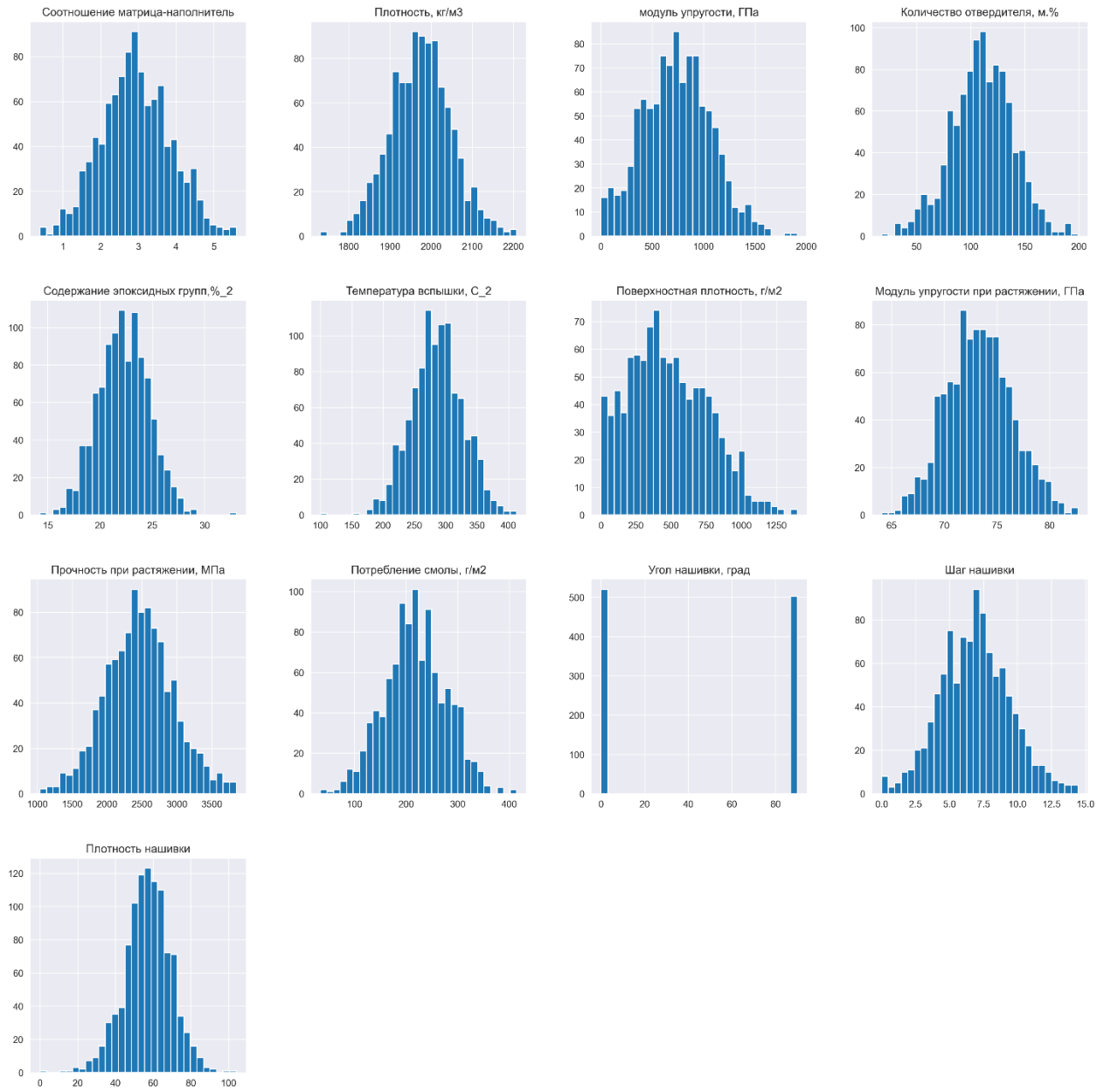


Рисунок 5 – Гистограмма распределения каждой переменной

В результате визуализации с помощью гистограмм было выявлено распределение величин, близкое к нормальному для большей части переменных, за исключением переменной «Угол нашивки», которая имеет только два значения. Для данной переменной гистограмма оказалась не показательна. Кроме того, для признаков «Поверхностная плотность, г/м2» и "Модуль упругости, ГПа" распределение оказалось скошено влево.

4) Попарные графики рассеяния точек – диаграммы Pairplot.

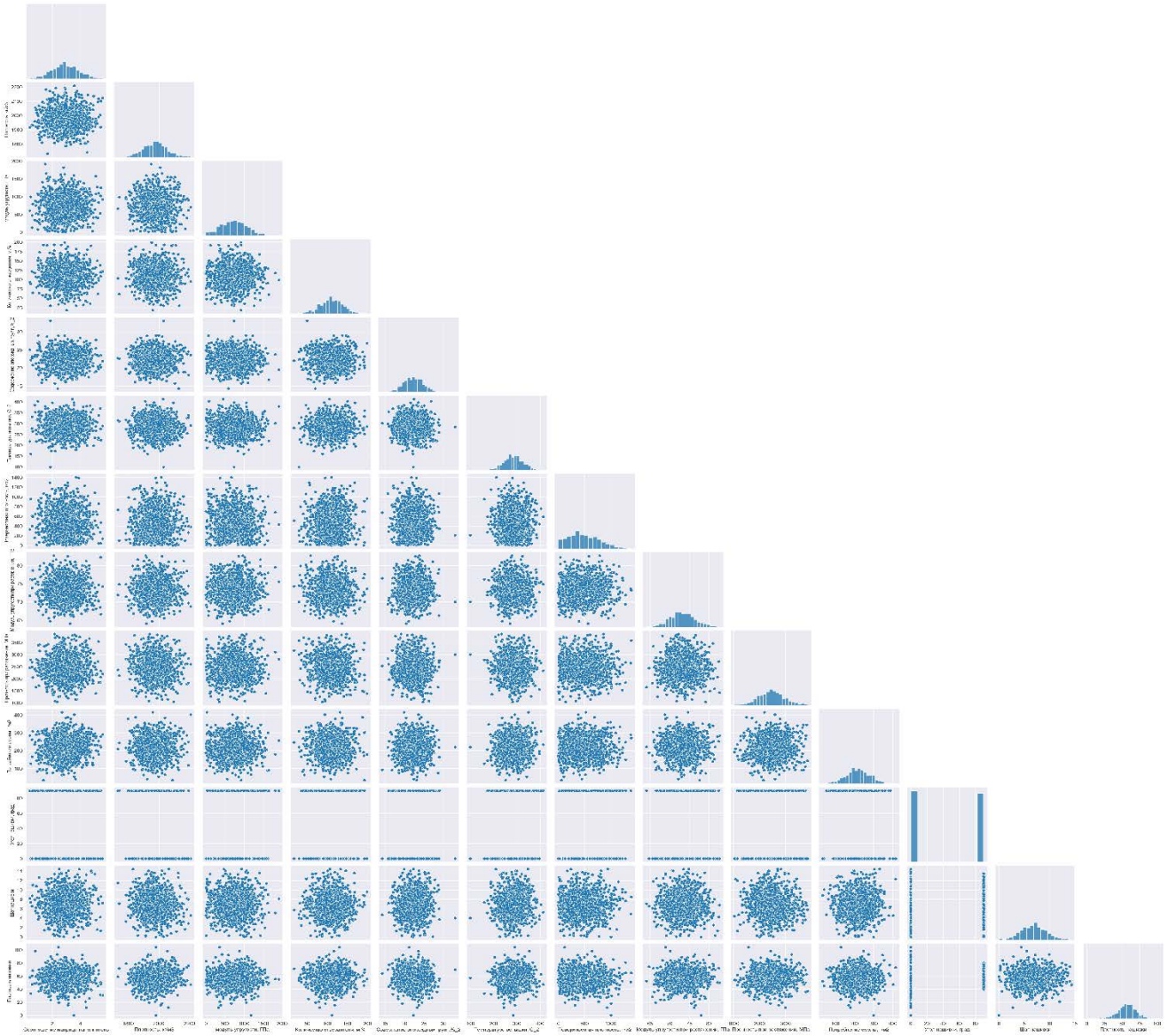


Рисунок 6 – Попарные графики рассеяния точек диаграммы Pairplot.

Данные графики показывают отношения между всеми парами переменных и на них также наблюдается отсутствие значимых взаимосвязей между ними .

5) Выявление выбросов и аномалий.

С помощью графиков Boxplot («Ящик с усами», «Диаграмма размаха») по всем переменным и по каждому признаку для лучшей визуализации. Индивидуальные графики построены с помощью библиотека Plotly - очень информативного и интерактивного инструмента.

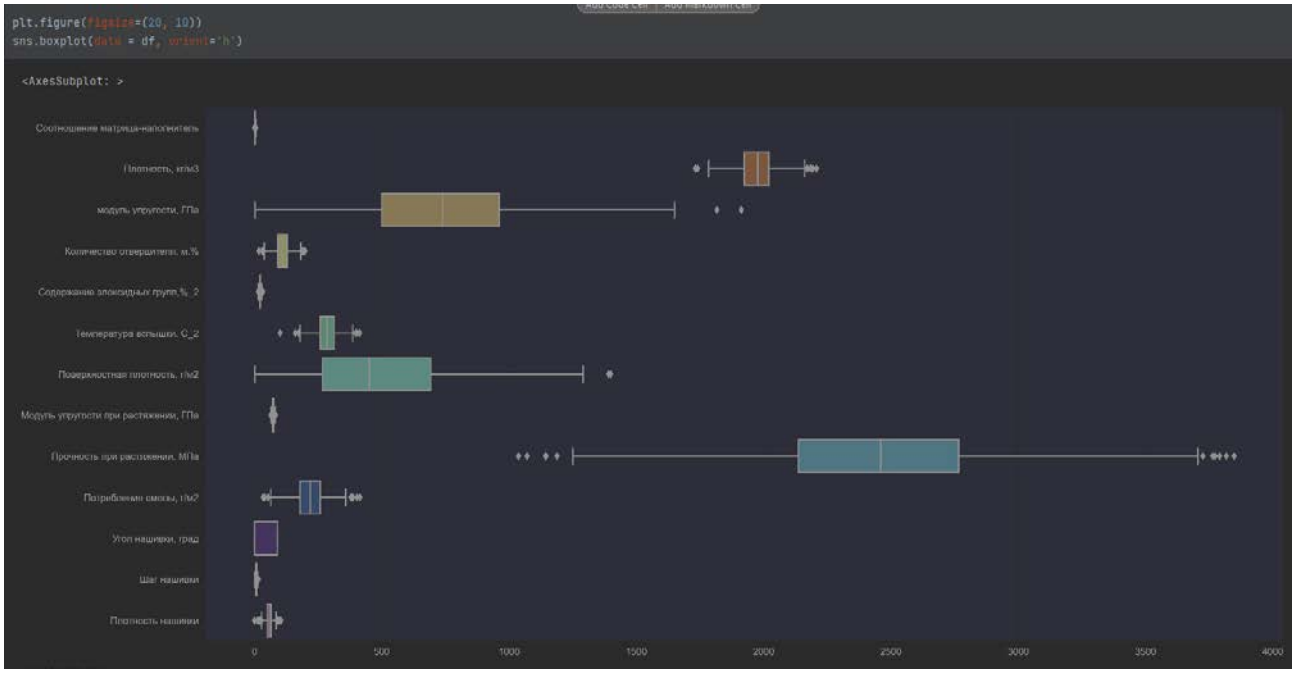


Рисунок 7 – Визуализация выбросов с помощью графиков Boxplot («Ящик с усами») по всем переменным.

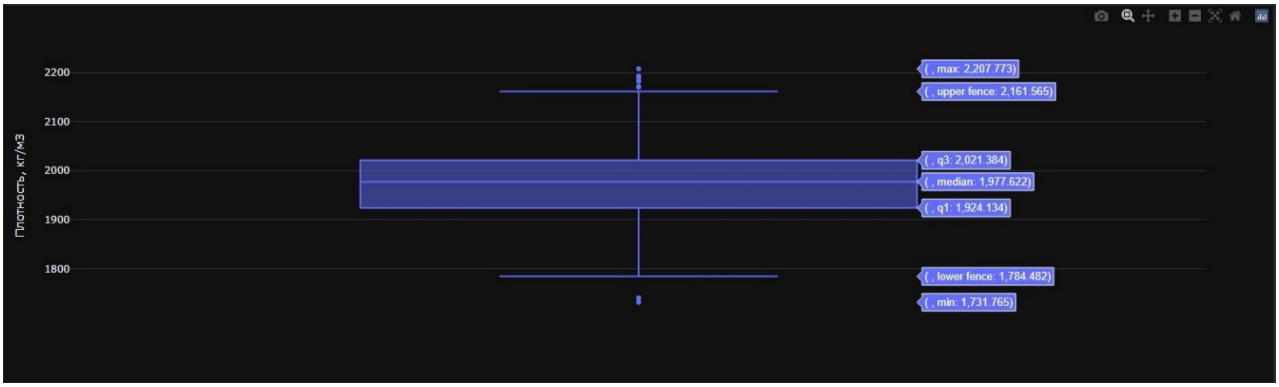


Рисунок 8 – Визуализация выбросов с помощью графиков Plotly («Ящик с усами») по отдельному признаку («Плотность, кг/м3»).

В результате визуализации с помощью графиков Boxplot («Ящик с усами») были выявлены выбросы по всем признакам.

6) Анализ корреляций между признаками.

Для выявления корреляций между признаками и вычисления коэффициентов корреляции Пирсона была построена корреляционная матрица с

помощью команды .corr(). Для удобства рассмотрения была сделана визуализация корреляций с помощью тепловой карты heatmap

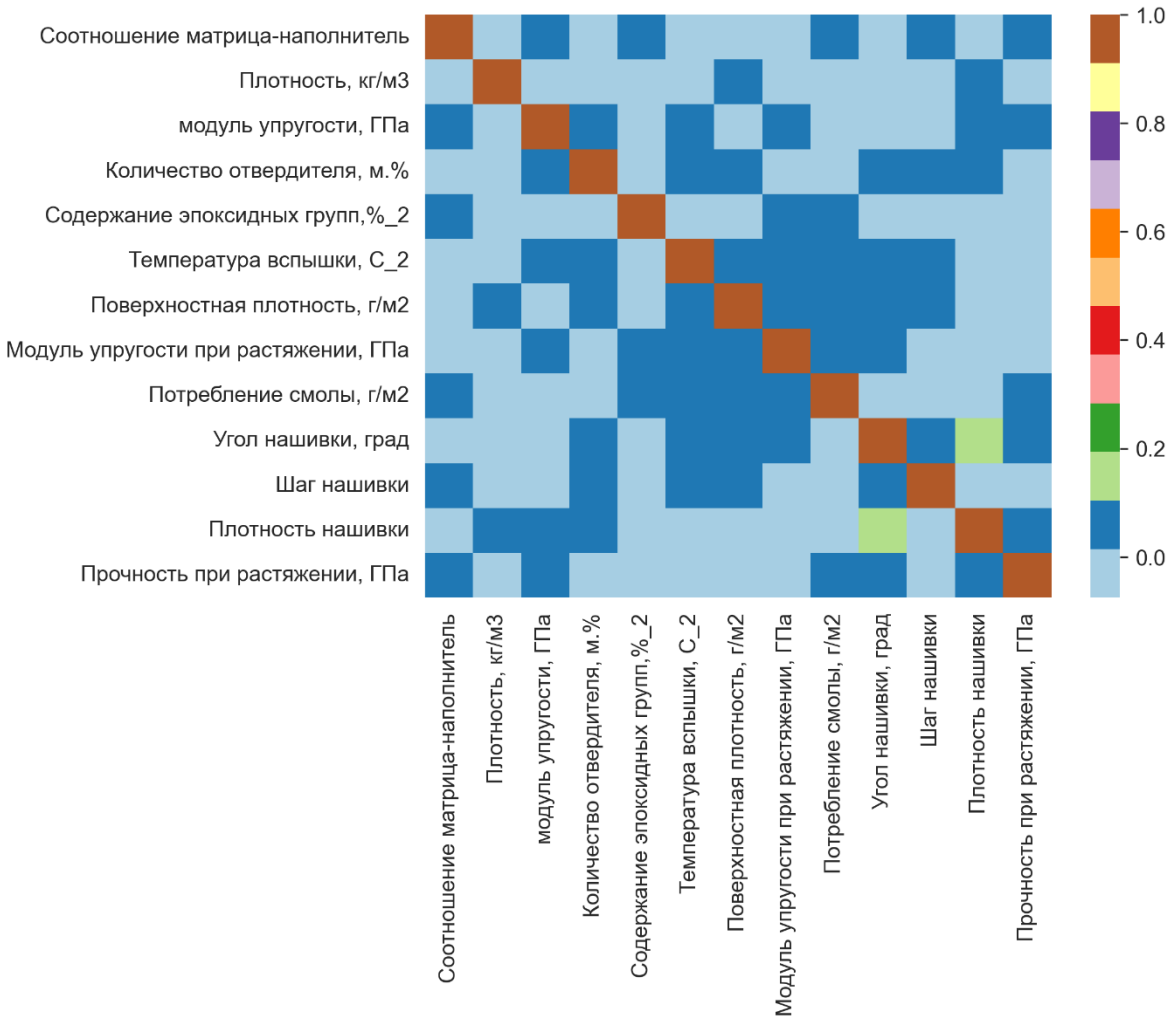


Рисунок 9 – Тепловая карта.

Построения корреляционной матрицы, визуализации корреляционных взаимосвязей указывают на отсутствие значимой линейной взаимосвязи между переменными. Корреляции отсутствуют как между независимыми переменными, что имеет положительное значение, так как нет мультиколинеарности, так и между зависимой и независимыми переменными, что может означать, что большинство линейных регрессоров не смогут корректно предсказать искомый результат.

2 Практическая часть

2.1 Предобработка данных

Предобработка данных включала в себя:

1) удаление выявленных с помощью разведочного анализа данных выбросов;
2) нормализацию данных с помощью MinMaxScaler() для обучения моделей машинного обучения прогноза модуля упругости при растяжении и прочности при растяжении

Так как раннее мы наблюдали скошенное распределение у признака "Поверхностная плотность, г/м2","модуль упругости, ГПа", для устранения выбросов справедливо будет применить метод на основе межквартильного размаха. Выбросы были удалены у всех признаков.

После удаления выбросов размер итогового датасета составил 936 записей, 13 колонок.

После нормализации данных с помощью MinMaxScaler() видно, что значения признаков были приведены к единому масштабу и распределились в диапазоне от 0 до 1.

```
mms=MinMaxScaler()
df_std =mms.fit_transform(df)
df = pd.DataFrame(df_std, columns = df.columns)
```

```
df.describe().T
```

13 rows × 8 columns

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	936.0	0.498933	0.187489	0.0	0.372274	0.494538	0.629204	1.0
Плотность, кг/м3	936.0	0.502695	0.187779	0.0	0.368517	0.511229	0.624999	1.0
модуль упругости, ГПа	936.0	0.446764	0.199583	0.0	0.301243	0.447061	0.580446	1.0
Количество отвердителя, м.%	936.0	0.504664	0.188865	0.0	0.376190	0.506040	0.637978	1.0
Содержание эпоксидных групп,%_2	936.0	0.491216	0.180620	0.0	0.367716	0.489382	0.623410	1.0
Температура вспышки, С_2	936.0	0.516059	0.190624	0.0	0.386128	0.515980	0.646450	1.0
Поверхностная плотность, г/м2	936.0	0.373733	0.217078	0.0	0.205619	0.354161	0.538683	1.0
Модуль упругости при растяжении, ГПа	936.0	0.488647	0.191466	0.0	0.359024	0.485754	0.615077	1.0
Потребление смолы, г/м2	936.0	0.521141	0.195781	0.0	0.392067	0.523766	0.652447	1.0
Угол нашивки, град	936.0	0.511752	0.500129	0.0	0.000000	1.000000	1.000000	1.0
Шаг нашивки	936.0	0.502232	0.183258	0.0	0.372211	0.504258	0.624604	1.0
Плотность нашивки	936.0	0.513776	0.191342	0.0	0.390482	0.516029	0.638842	1.0
Прочность при растяжении, ГПа	936.0	0.495706	0.188915	0.0	0.365149	0.491825	0.612874	1.0

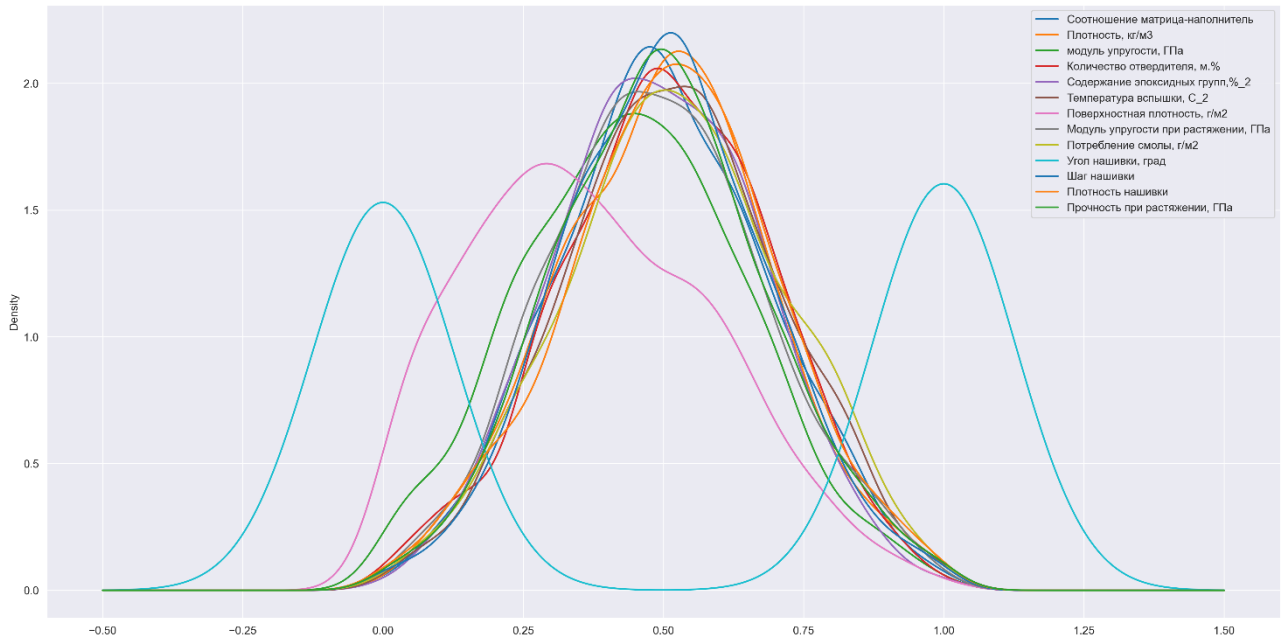


Рисунок 10 – Описание датасета и Графики распределения ядра каждой из переменных после нормализации.

2.2 Разработка и обучение модели

В качестве моделей для прогноза модуля упругости при растяжении и прочности при растяжении были выбраны:

1) Линейная регрессия;
2) Полиномиальная регрессия;
3) KNeighborsRegressor;
4) RandomForestRegressor;
5) GradientBoostingRegressor;

При построении моделей был осуществлен поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10.

При построении модели в соответствии с поставленной задачей 30% данных было оставлено на тестирование модели, на остальных 70% происходило обучение моделей.

Размер обучающей выборки: 655.

Размер тестовой выборки: 281.

2.3 Тестирование модели

После обучения выбранных моделей для прогноза упругости при растяжении и прочности при растяжении была проведена оценка точности моделей. В данном разделе показаны ошибка каждой модели на тестирующей части выборок. Результат консолидирован в таблице:

ИТОГИ ПО МОДЕЛЯМ

model_loss

10 rows | 10 rows × 5 columns

	целевая переменная	модель	MAE	MSE	R2
0	Модуль упругости при растяжении	Linear Regression	0.145773	0.032470	-0.003036
1	Модуль упругости при растяжении	Polynomial Regression	0.156588	0.037613	-0.161910
2	Модуль упругости при растяжении	KNeighborsRegressor	0.148200	0.032793	-0.013020
3	Модуль упругости при растяжении	RandomForestRegressor	0.146204	0.032330	0.001306
4	Модуль упругости при растяжении	GradientBoostingRegressor	0.146091	0.032810	-0.013526
5	Прочность при растяжении	Linear Regression	0.141190	0.033028	-0.032068
6	Прочность при растяжении	Polynomial Regression	0.149222	0.036791	-0.149642
7	Прочность при растяжении	KNeighborsRegressor	0.141451	0.033193	-0.037208
8	Прочность при растяжении	RandomForestRegressor	0.142092	0.033202	-0.037486
9	Прочность при растяжении	GradientBoostingRegressor	0.140149	0.032722	-0.022485

Рисунок 11 – Ошибки каждой модели

По итогам оценки лучшие результаты почти по всем выбранным метрикам для прогноза модуля упругости при растяжении показал метод RandomForestRegressor.

Лучшие результаты для прочности при растяжении показал метод линейной регрессии. Хотя и параметры R2 в большинстве случаев отрицательны – что говорит о неудовлетворительной описательной способности моделей.

2.4 Нейронная сеть

Первая модель нейронной сети :

- количество скрытых слоев – 2;
- количество нейронов на входном слое – 12;
- количество нейронов на 1 и 2 скрытых слоях – 25;

- количество нейронов на выходном слое – 1;
- активационная функция «Relu», на выходном слое «Sigmoid»;
- оптимизатор «Adam»;
- функция потерь – MSE;
- метрика – MAE;

Обучение модели происходило за 250 эпох.

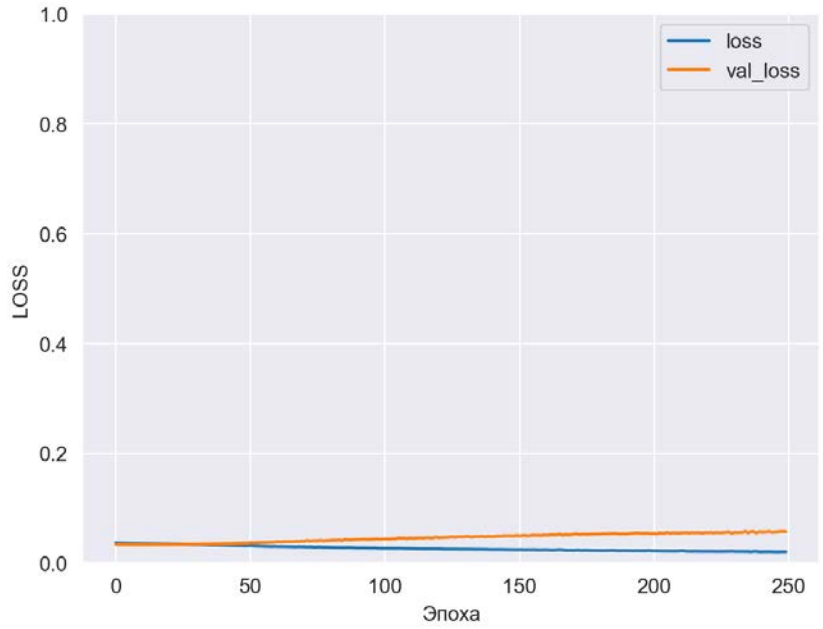


Рисунок 12 – Изменени MAE за время обучения модели

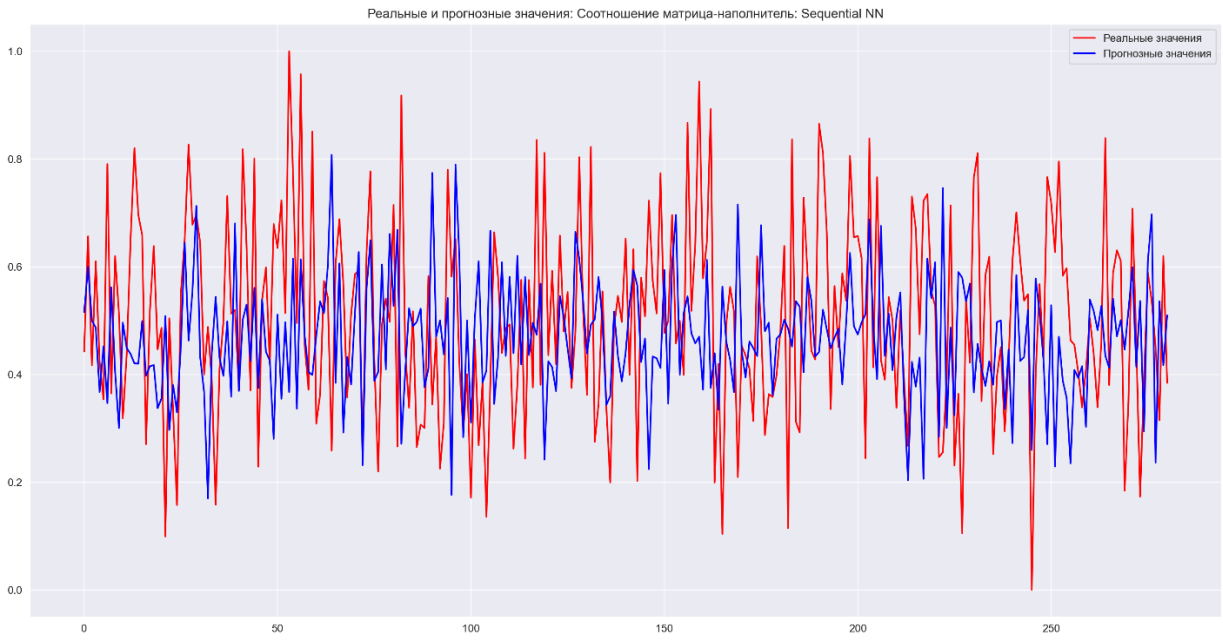


Рисунок 13 –Графическое представление реальных и прогнозных значений на тестовой выборке модели Sequential NN

Архитектура и гиперпараметры модели Yet_another_NN:

- количество скрытых слоев – 4;
- количество нейронов на входном слое – 120;
- количество нейронов на 1 и 2 скрытых слоях – 1440;
- количество нейронов на 3 скрытом слое – 60;
- количество нейронов на 4 скрытом слое – 20;
- количество нейронов на выходном слое – 1;
- активационная функция «Relu», на выходном слое «Sigmoid»;
- оптимизатор «rmsproг»;
- функция потерь – MAE;
- метрика – MSE;

.

Данная архитектура нейронной сети показала лучшие результаты по сравнению с другими ранее опробованными архитектурами.

Обучение модели происходило за 250 эпох.

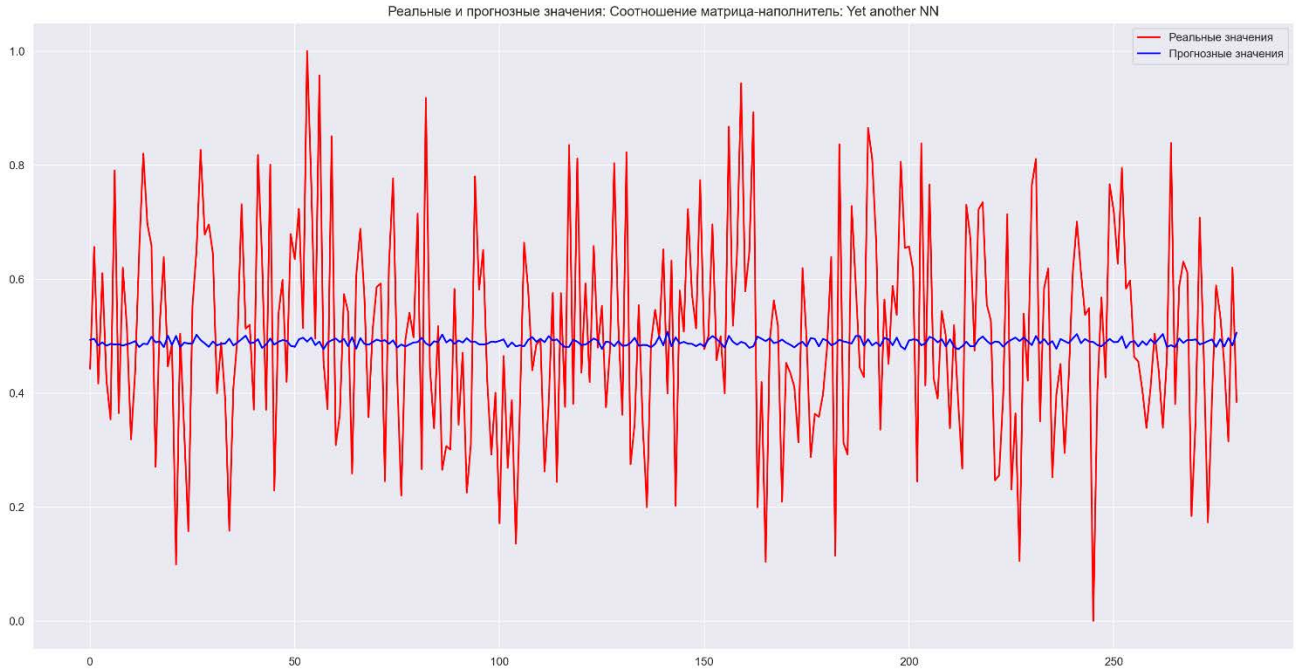


Рисунок 14 – Графическое представление реальных и прогнозных значений на тестовой выборке модели Yet_another_NN

В модели Yet_another_NN использовались точки фиксации минимального значения ошибки на тестовой выборке – чекпоинты. Согласно истории обучения, оптимальные параметры получены на 8 эпохе обучения.(MAE

=0.14713). В дальнейшем сохраним эту модель для использования в приложении.

2.5 Разработка приложения

Приложение было разработано с помощью Flask, HTML. В интерфейсе использованы наименования параметров из датасета. Скриншот приложения приведен на рисунке :

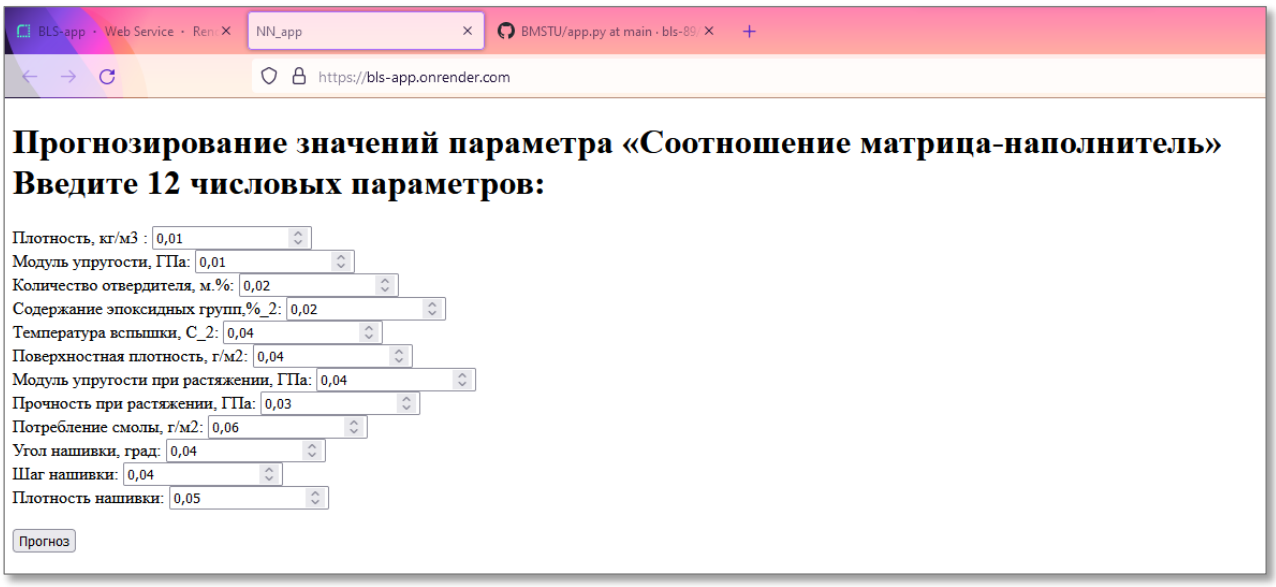


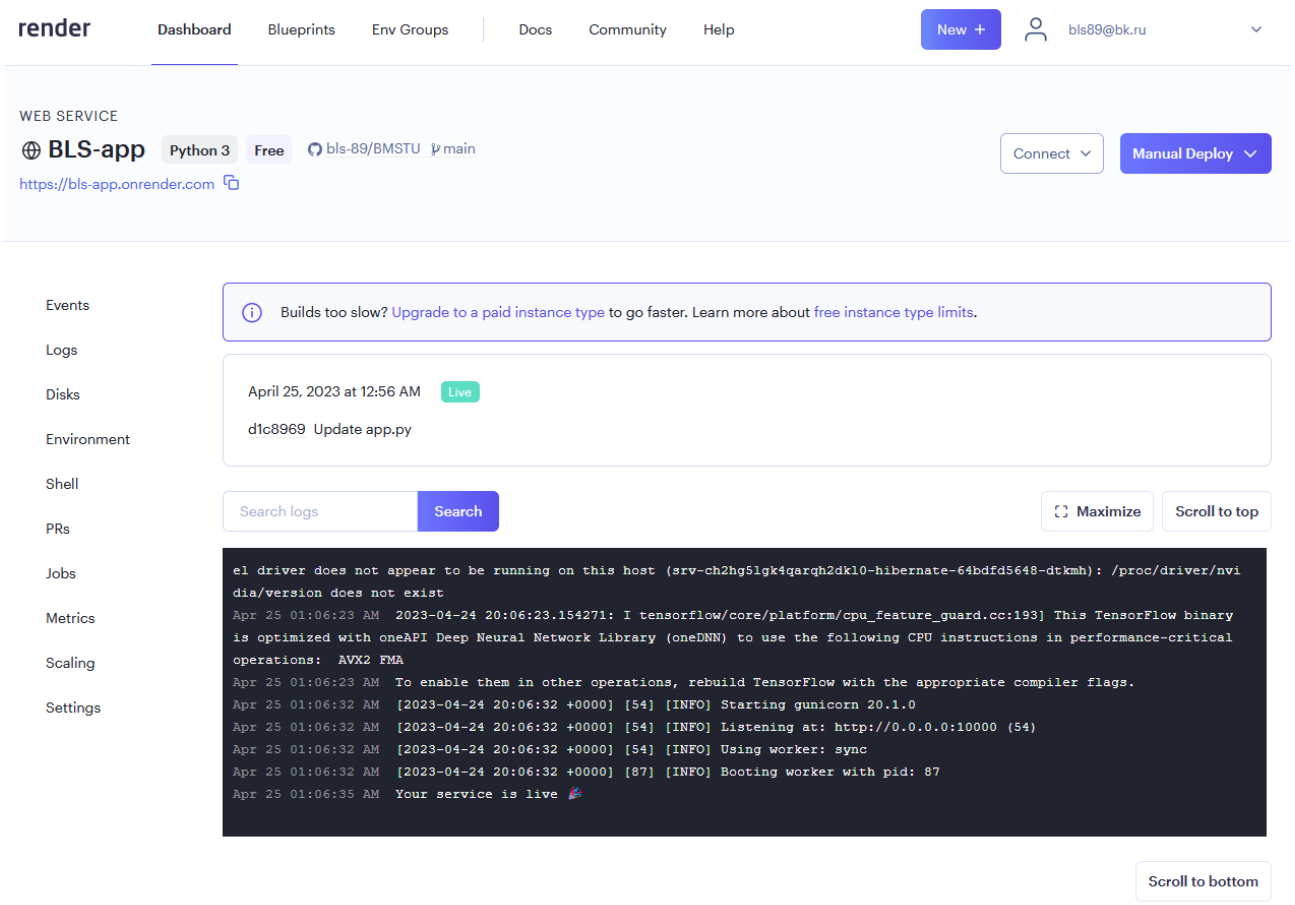
Рисунок 15 – Интерфейс приложения.

При введении параметров и нажатии кнопки «Прогноз», приложение, на основании расчетов нейросети, будет выводить внизу формы рекомендуемое значение соотношения «матрица – наполнитель».

Краткая инструкция использования приложения для прогноза соотношения «Матрица-наполнитель»:

1) ввести необходимые параметры в поля формы;
2) нажать кнопку «Прогноз»;
3) внизу формы появится прогнозное рекомендованное значение для требуемых параметров.

При размещении на сервере в работе приложения возникают ошибки. Хотя все требования по размещению выполнены. Исходя из найденной информации, могу предположить ограничение функционала самого сервиса размещения.



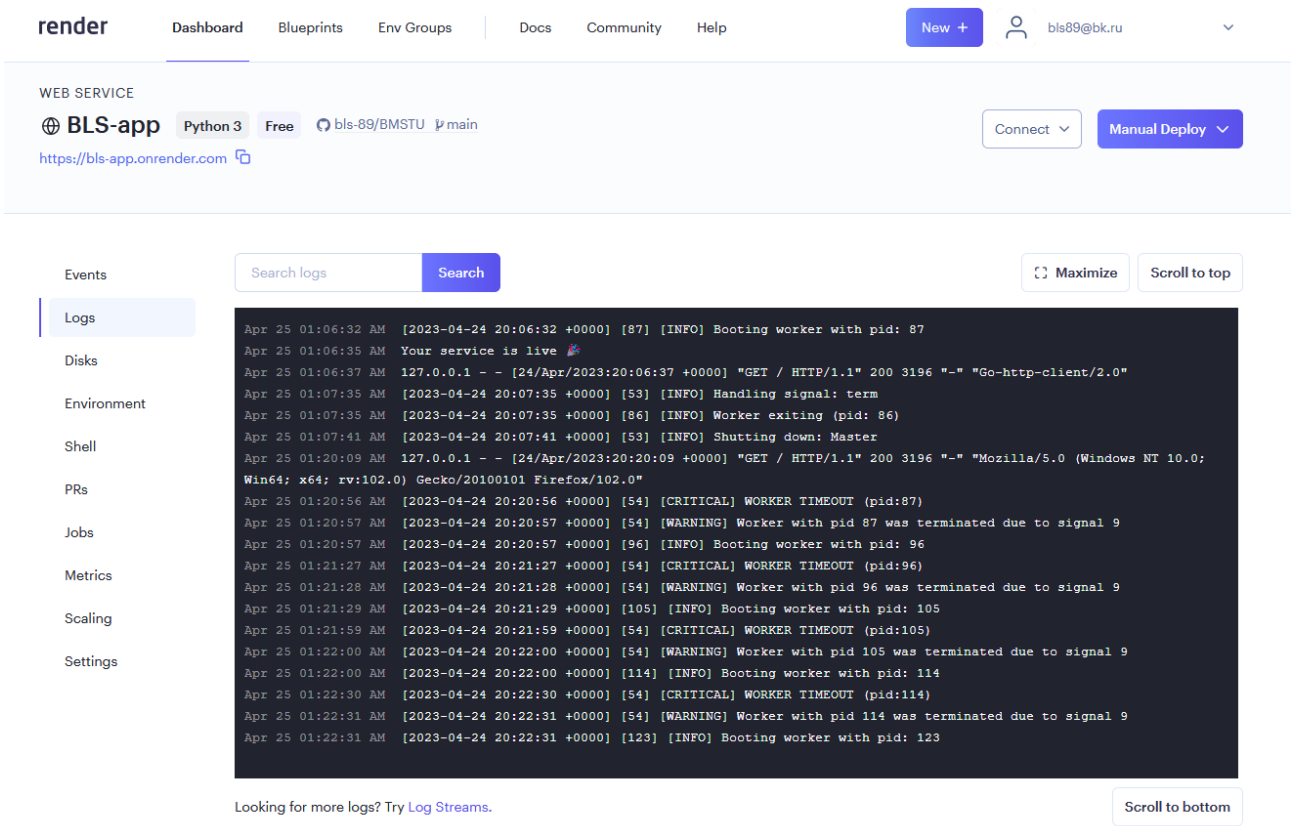


Рисунок 16 – Размещение приложения на сервере и отображение лога.

2.6 Создание удаленного репозитория

Страница слушателя на GitHub: https://github.com/bls-89

Созданный репозиторий: https://github.com/bls-89/BMSTU

Ссылка на приложение на render.com: https://bls-app.onrender.com/

Ссылка на страницу Kaggle.com: https://www.kaggle.com/boleslav89

Заключение

На основании проведённого исследования можно сделать следующие выводы.

В ходе разведочного анализа данных выяснилось, что корреляция между всеми переменными стремится к нулю, то есть практически отсутствует. С одной стороны, отсутствие корреляций между независимыми переменными имеет положительное значение, но, с другой стороны их отсутствие между целевой переменной и предикторами не позволяет построить эффективные модели обучения, особенно линейные. Кроме того, объем предоставленных данных не слишком велик, что также оказывает негативное влияние на результаты обучения моделей и получения достоверных прогнозов.

Коэффициенты детерминации для всех моделей на тестирующей части выборке отрицательные или близкие к нулю, что свидетельствует о том, что прогноз сопоставим по качеству с константным предсказанием и модель никак не объясняет (обобщает) данные. Для регрессий, которые справляются с предсказанием хуже, чем базовая регрессия, коэффициент детерминации выдает отрицательный результат. Поэтому обобщающая способность данных моделей не удовлетворительна и не позволяет рекомендовать их для получения достоверных прогнозов.

Полученная модель нейронной сети также не совсем удовлетворительна, но позволяет предсказывать некоторые значения, близкие к средним значениям параметров и, возможно, при дальнейшем поиске и подборе гиперпараметров оптимальной архитектуры, будет способна выдавать более приемлемый результат.

В целом, для получения более корректных результатов и эффективности получаемых прогнозов видится необходимость увеличения объема доступных данных, возможно, оптимизации подхода к их сбору, поиск и выделение дополнительных признаков, имеющих более выраженную взаимосвязь с целевыми переменными, а также консультации экспертов предметной области.

Библиографический список

1. Жерон О Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. / О Жерон. – Санкт-Петербург: ООО “Альфа-книга”, 2018. – 688 с.
2. Силен Дэви, Мейсман Арно, Али Мохамед. Основы Data Science и Big Data. Python и наука о данных. – СПб.: Питер, 2017. – 336 с.: ил.
3. Траск Эндрю. Грокаем глубокое обучение. – СПб.: Питер, 2019. – 352 с.: ил.
4. Документация по работе с библиотекой Matplotlib: сайт. – URL: https://matplotlib.org/stable/tutorials/introductory/pyplot.html (дата обращения: 22.04.2023). – Режим доступа: свободный
5. Документация по библиотеке scikit-learn: – Режим доступа: https://scikit-learn.org/stable/user_guide.html. (дата обращения: 20.04.2023).
6. Документация по библиотеке seaborn: – Режим доступа: https://seaborn.pydata.org/tutorial.html. (дата обращения: 19.04.2023).
7. Линейная регрессия: примеры и вычисление функции потерь: сайт. – URL: https://neurohive.io/ru/osnovy-data-science/linejnaja-regressija/ (дата обращения: 14.10.2022)
8. Основы нейронных сетей, алгоритмы обучения, функции активации и потери: сайт. – URL: https://neurohive.io/ru/osnovy-data-science/osnovy-nejronnyh-setej-algoritmy-obuchenie-funkcii-aktivacii-i-poteri/ (дата обращения: 15.04.2023).
9. Реализация и разбор алгоритма «случайный лес» на Python // Tproger : сайт. – URL: https://tproger.ru/translations/python-random-forest-implementation/ (дата обращения: 18.04.2023)
10. Случайный лес (Random Forest) // КвазиНаучный блог Александра Дьяконова: сайт. – URL: https://alexanderdyakonov.wordpress.com/2016/11/14/случайный-лес-random-forest/ (дата обращения: 03.11.2022)