МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**по курсу**

«Data Science»

Тема: прогнозирование конечных свойств новых материалов

(композиционных материалов)

Слушатель Ядов Николай Александрович

Москва, 2023

# **Содержание**

1. Введение. Аналитическая часть…………………………………………….. 3
   1. Предмет и актуальность исследования……………………………………... 3
   2. Описание входных данных………………………………………………...… 4
   3. Постановка задачи…………………………………………………………… 5
   4. Методы разведочного анализа данных (EDA)……………………………… 5
   5. Методы предобработки данных (ETL)……………………………………… 6
   6. Методы прогнозирования…………………………………………………… 6
   7. Метрики оценки качества прогнозирования…………………………….... 10
   8. Pipline машинного обучения……………………………………………….. 12
2. Практическая часть…………………………………………………………. 13

2.1. EDA…………………………………………………………………………... 13

2.2. ETL…………………………………………………………………………… 20

2.3. Разработка, обучение и тестирование моделей………………………….… 21

2.3.1. Построение моделей для прогнозирования значений модуля упругости при растяжении………………………………………………………………………… 21

2.3.2. Построение моделей для прогнозирования значений прочности при растяжении………………………………………………………………………… 22

2.3.3. Построение нейронной сети для прогнозирования значений соотношения матрица-наполнитель……………………………………………………………... 26

2.3.3.1. ETL ………………………………………………………………………… 26

2.3.3.2. Последовательная нейронная сеть……………………………………..… 26

2.3.3.3. Keras Tuner………………………………………………………………… 28

2.4. Flask-приложение……………………………………………………………... 31

3. Заключение……………………………………………………………………… 33

Список литературы……………………………………………………………...… 35

Приложение 1. Код машинного обучения (ВКР.ipynb)

Приложение 2. Код flask-приложения (аpp.py).

# **Введение. Аналитическая часть**

## **Предмет и актуальность исследования**

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 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.

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

Актуальность исследования состоит в том, что созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов и цифровыми двойниками новых композитов.

## **Описание входных данных**

Информация о свойствах композитов представлена в виде двух датасетов в формате MS Excel (.xlsx). Датасеты расположены по следующему адресу: <https://drive.google.com/file/d/1B1s5gBlvgU81H9GGolLQVw_SOivyNf2/view?usp=sharing>.

Загрузку и обработку данных осуществляем с помощью ПО Jupyter Notebook (библиотека pandas, pd.read\_excel). Проводим объединение по индексу тип объединения INNER.

После объединения исходных таблиц для получен датасет размером 1023 х 13 с отсутствующими пустыми значениями и типами данных, представленных числовыми целыми значениями (int64) и числовыми значениями с плавающей точкой (float64).

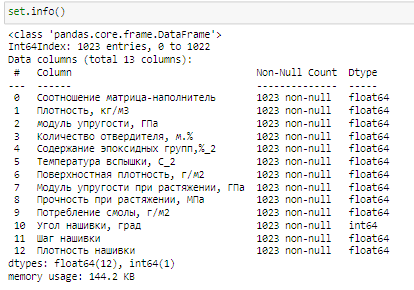


Рисунок 1 – Информация об исходных данных

## **Постановка задачи**

В рамках настоящей работы по прогнозированию конечных свойств композиционных материалов требуется выполнить следующее:

1. Изучить теоретические основы и методы решения поставленной задачи;

2. Выполнить разведочный анализ данных. Нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек. Для каждой переменной получить среднее и медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков;

3. Провести предобработку данных: удаление шумов, нормализацию и т.д.;

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

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

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

7. Оценить точность модели на тренировочном и тестовом датасетах;

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

## **Методы разведочного анализа данных (EDA)**

Аббревиатура процесса разведочного анализа данных EDA расшифровывается как «Exploratory Data Analysis».

Для разведочного анализа исходных данных с целью формирования представления об их составе и структуре, а также определения необходимых преобразовании в настоящей работе используются следующие методы из библиотек Numpy, Pandas и Seaborn:

- pd.read\_excel – загрузка исходных таблиц в Juputer Notebook;

- .join – объединение исходных таблиц в один датасет;

- .shape() – размерность датасета;

- .info() –наименования столбцов и размерность датасета, тип данных в датасете (int, float, str и т.д.);

- .describe() – минимальные, средние, максимальные значения, стандартные отклонения, процентные перцентили в разрезе столбцов датасета;

- .isnull().sum() – анализ пустых NuN ячеек в датасете;

- sns.heatmap – визуализация корреляции переменных между собой;

- .corr() – список корреляционных значений переменных;

- sns.boxplot – визуализация и анализ наличия выбросов;

- sns.histplot – визуализация и анализ распределения значений переменных;

- sns.pairplot – попарная визуализация и анализ распределения значений переменных.

## **Методы предобработки данных (ETL)**

Аббревиатура процесса предобработки данных ETL расшифровывается как «Extract, Transformer, Load».

В настоящей работе для приведения исходных данных к формату, удобному для их дальнейшей обработки моделями машинного обучения, применяется метод стандартизации StandartScaler из библиотеки Sklearn.preprocessing – масштабирование данных путем их приведения к интервалу со средним нулевым значением и стандартным отклонением равным единице с целью соблюдения единого диапазона значений и масштаба.

Для удобства работы с переменными выполнено частичное переименование переменных посредством применения метода .rename().

Также для разделения массива данных на тестовую и обучающую выборки для процесса обучения, тестирования и оценки результатов применяется метод train\_test\_split из библиотеки Sklearn.model\_selection.

## **Методы прогнозирования**

Прогнозирование конечных свойств композиционных материалов является задачей регрессии в машинном обучении, поэтому для прогнозирования значений модуля упругости при растяжении применяется модель Linear Regression из библиотеки Sklearn.linear\_model и ансамблевая модель Random Forest Regression из библиотеки Sklearn.ensemble.

Для прогнозирования значений прочности при растяжении применяется модель Support Vector Regression из библиотеки Sklearn и модель Decision Tree Regression из библиотеки Sklearn.tree.

Для прогнозирования соотношения матрица-наполнитель выполняется построение искусственной последовательной нейронной сети.

Для поиска гиперпараметров моделей используются методы поиска по сетке Grid Search и случайного поиска Random Search, а для поиска гиперпараметров нейронной сети используется Keras Tuner.

Рассмотрим описание, достоинства и недостатки принятых к использованию методов прогнозирования.

1. Linear Regression.

Множественная линейная регрессия - это модель линейной регрессии, которая оценивает взаимосвязь между несколькими независимыми переменными (признаками) и одной зависимой переменной.

Используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной «y» от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) «x» с линейной функцией зависимости.

Линейная регрессия использует метод, известный как обычный метод наименьших квадратов, чтобы найти наиболее подходящее уравнение регрессии.

1. Decision Tree Regression.

Регрессия дерева решений строит регрессионную модель в виде древовидной структуры. Структура дерева представляет собой «листья» и «ветки». На рёбрах («ветках») дерева решения записаны признаки, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах — признаки, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе. Каждый лист представляет собой значение целевой переменной, изменённой в ходе движения от корня по рёбрам дерева до листа. Каждый внутренний узел сопоставляется с одной из входных переменных. Алгоритм вычисляет информационный прирост для каждой характеристики и выбирает ту, которая дает наивысшее значение.

1. Random Forest Regression.

Алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга и метод случайных подпространств. Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество решения, но за счёт их большого количества результат получается хорошим. Прогнозирование конечного значения выполняется за счет расчета среднего значения по всем построенным деревьям.

1. Support Vector Regression.

Метод классификации опорных векторов может быть расширен для решения задач регрессии. Этот метод называется регрессией опорных векторов.

Модель, созданная с помощью классификации опорных векторов (зависит только от подмножества обучающих данных, поскольку функция затрат для построения модели не заботится о точках обучения, которые лежат за пределами поля. Аналогично, модель, созданная с помощью регрессии опорных векторов, зависит только от подмножества обучающих данных, поскольку функция стоимости игнорирует выборки, прогнозирование которых близко к их цели.

1. Нейронная сеть.

Нейронная сеть состоит из взаимосвязанных групп узлов, называемых нейронами. Входные данные передаются в эти нейроны в виде линейной комбинации со множеством переменных. Значение, умножаемое на каждую функциональную переменную, называется весом. Затем к этой линейной комбинации применяется нелинейность, что даёт нейронной сети возможность моделировать сложные нелинейные отношения. Чаще всего нейросети бывают многослойными: выход одного слоя передается следующему так, как описано выше. На выходе нелинейность не применяется.

Нейронные сети тренируются с помощью метода стохастического градиента и алгоритма обратного распространения ошибки.

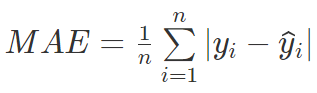
Таблица 1 - Преимущества и недостатки используемых методов прогнозирования.

|  |  |  |
| --- | --- | --- |
| Регрессионный метод | Преимущества | Недостатки |
| Linear Regression | * быстрое моделирование (особенно если отсутствует большой объём данных); * легко объясняема и интерпретируема; * достаточно хорошо работает на большом наборе данных. | * сложности проектирования в случае нелинейных данных, необходимость наличия информации о структуре данных и взаимосвязи между переменными. |
| Decision Tree Regression | * подходит для изучения сложных линейных и нелинейных отношений; * основные алгоритмы просты в понимании и реализации. Границы решений, которые создаются во время обучения, легко понять; * отсутствие требования масштабирования данных. | * склонность к переобучению. Завершенная модель дерева решений может быть чрезмерно сложной и содержать ненужную структуру; * плохие результаты на небольших наборах данных. |
| Random Forest Regression | * подходит для изучения сложных линейных и нелинейных отношений; * основные алгоритмы просты в понимании и реализации. Границы решений, которые создаются во время обучения, легко понять; * отсутствие требования масштабирования данных. | * склонность к переобучению. Завершенная модель дерева решений может быть чрезмерно сложной и содержать ненужную структуру. * используя большие случайные леса для достижения более высокой производительности расходуется память и время. |
| Support Vector Regression | * эффективен в пространствах с высокой размерностью. * эффективен в случаях, когда количество измерений больше, чем количество выборок. * использует подмножество обучающих точек в функции принятия решений (называемых опорными векторами), поэтому он также экономит память. * универсальность: для функции принятия решения могут быть указаны различные [функции ядра](https://scikit-learn.org/stable/modules/svm.html#svm-kernels). Предоставляются общие ядра, но также возможно указать пользовательские ядра. * хорошо адаптируется; * хорошо работает для нелинейных задач; * отсутствие предвзятости к объекту, отличающемуся от других. | * если количество функций намного больше, чем количество выборок чувствителен к чрезмерной подгонке при выборе [функций ядра](https://scikit-learn.org/stable/modules/svm.html#svm-kernels), термин регуляризации имеет решающее значение. * SVM напрямую не предоставляют оценки вероятности, они рассчитываются с использованием дорогостоящей пятикратной перекрестной проверки * требуется масштабирование данных; * труден для интерпретации. |
| Нейронная сеть | • эффективна при моделировании сложных нелинейных отношений ввиду многослойности;  • гибкость: не нужно беспокоиться о структуре данных в нейронных сетях;  • производительность растет с увеличением тренировочных данных. | * возможная сложность архитектуры и модели в целом; * требование тщательной настройки гиперпараметров и скорости обучения. * для достижения высокой производительности нейронным сетям необходимо огромное количество данных, и в результате, как правило, нейросети уступают другим ML алгоритмам в тех случаях, когда данных мало. |

## **Метрики оценки качества прогнозирования**

Для оценки точности и качества работы выбранных моделей прогнозирования и нейронных сетей применяются следующие метрики:

* + MAE (средняя абсолютная ошибка) - определяет среднее абсолютное расстояние между прогнозируемыми и целевыми значениями – то, насколько число в прогнозе разошлось с реальным числом. Данную ошибку удобно трактовать – погрешность измеряется в тех же единицах, что и значения целевой переменной.

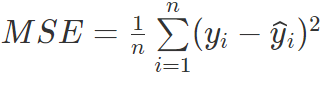
 (1)

где n – кол-во элементов;

у – реальное целевое значение;

ŷ – предсказанной целевое значение.

* + МSE (средняя квадратичная ошибка) - определяет среднеквадратичную ошибку между прогнозируемыми и целевыми значениями. Настроена на отражение влияния именно больших ошибок на качество модели. Менее удобна для понимания ввиду измерения в квадратных единицах. Данная метрика обычно применяется для сравнения моделей между собой.

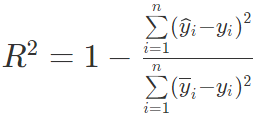
 (2)

где n – кол-во элементов;

у – реальное целевое значение;

ŷ – предсказанной целевое значение.

* + R2 (коэффициент детерминации) - показывает, какую долю разнообразия данных модель смогла объяснить. Метрика просто интерпретируема: модель, для которой R2 больше 0,5, является удовлетворительной. Если R2 больше 0,8, то модель рассматривается как очень хорошая. Значения, меньшие 0,5, говорят о том, что модель некачественна.

 (3)

где n – кол-во элементов;

у – реальное целевое значение;

ŷ – предсказанной целевое значение.

Кроме того, для оценки общей точности алгоритмов прогнозирования используется оценка отношения средней абсолютной ошибки к среднему значению фактической целевой переменной:

accuracy = 100 – (MAE / y.mean() \*100) (4)

где y.mean() – арифметическое среднее реального целевое значения

## **Pipline машинного обучения**

Обобщая вышесказанное, можно визуализировать последовательность и шаги процесса машинного обучения, реализованного в настоящей работе.

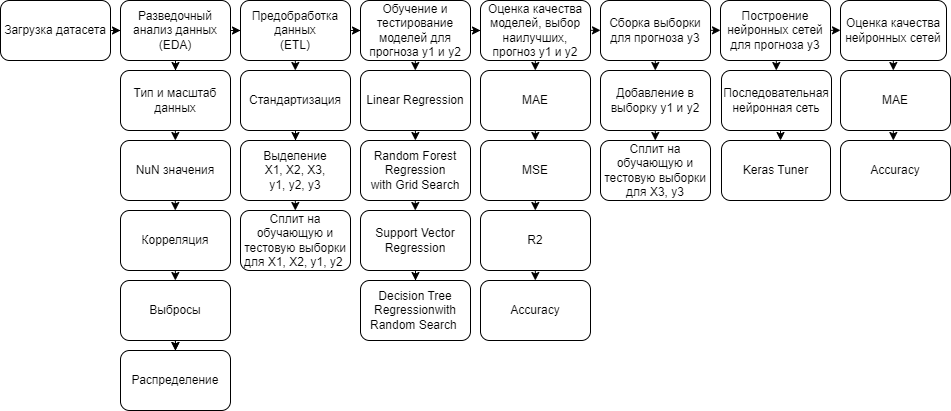
****

Рисунок 2 - Pipline машинного обучения

# **Практическая часть**

## **Разведочный анализ данных**

Разведочный анализ данных показал, что в датасете отсутствуют пропущенные пустые NuN значения, признаки представлены числовыми целыми (int64) и числовыми дробными (float64) значениями, строчные значения (str) отсутствуют, соответственно, отсутствует необходимость их категорирования.

Диапазон разброса минимальных и максимальных значений переменных достаточно велик, минимальное значение в датасете равно «0», максимальное равно «3848».

Матрица корреляции, а также числовые вычисления показывают слабую взаимосвязь переменных между собой, минимальные и максимальные значения корреляций признаков по отношению к целевым переменным исчисляются сотыми долями.

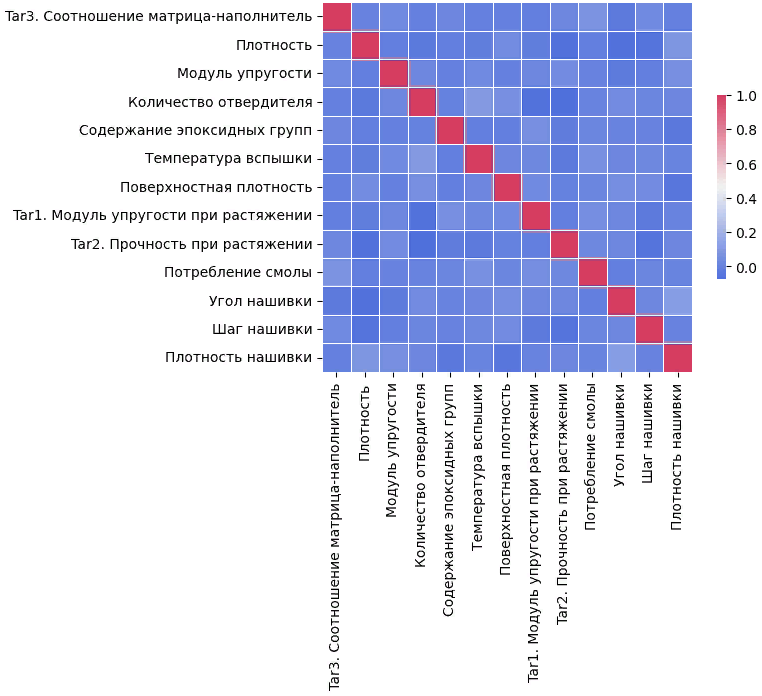
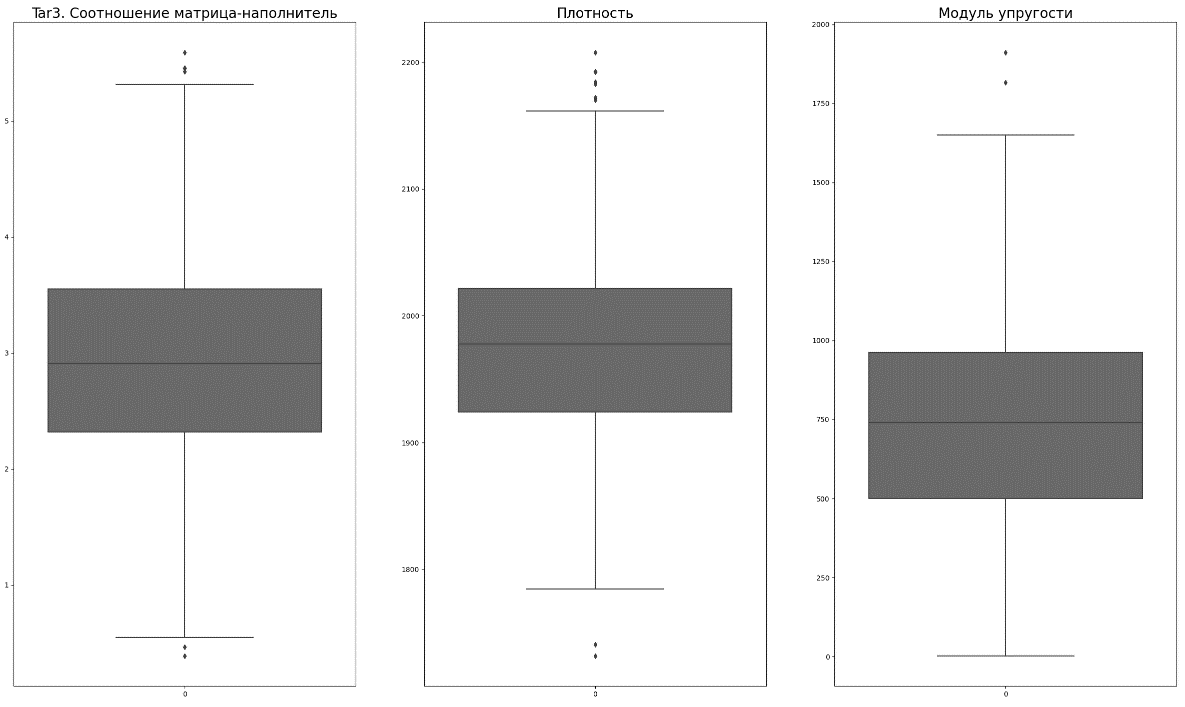
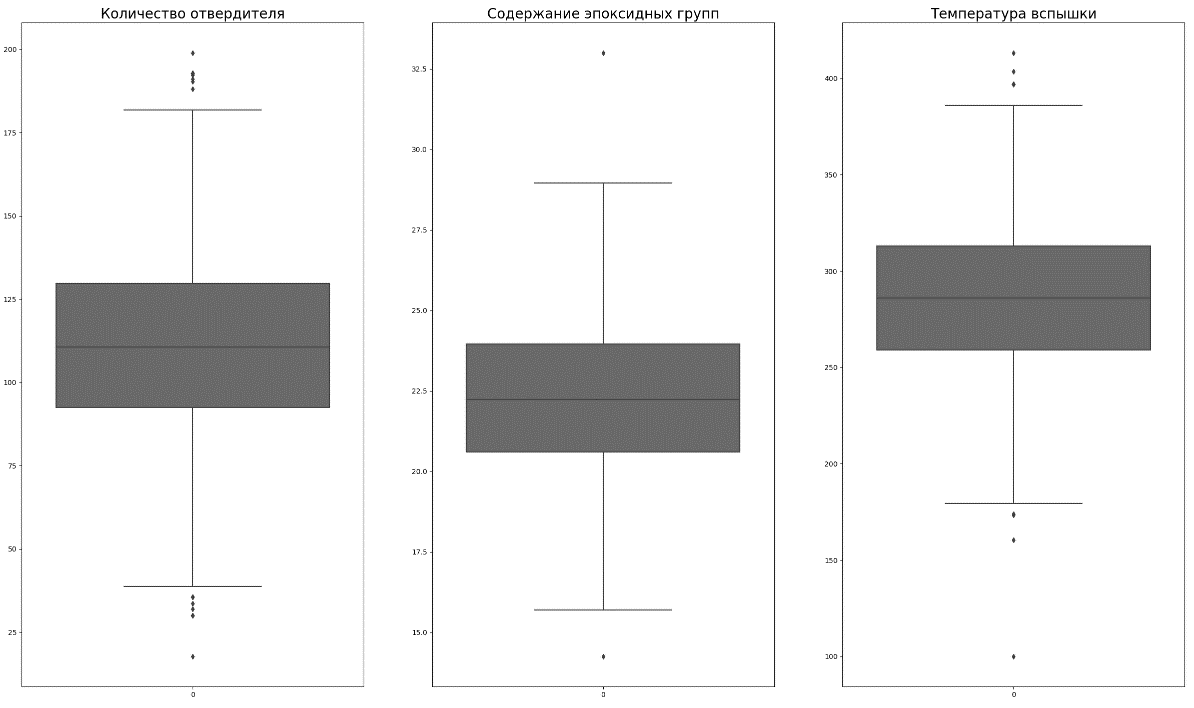
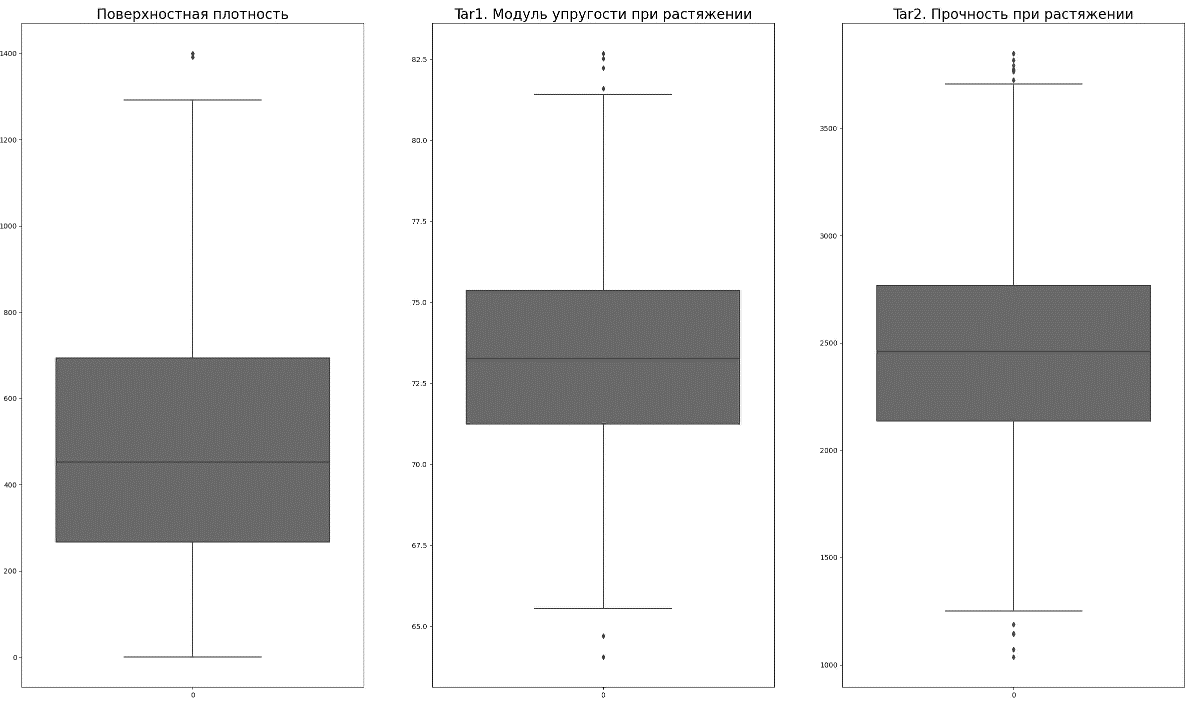


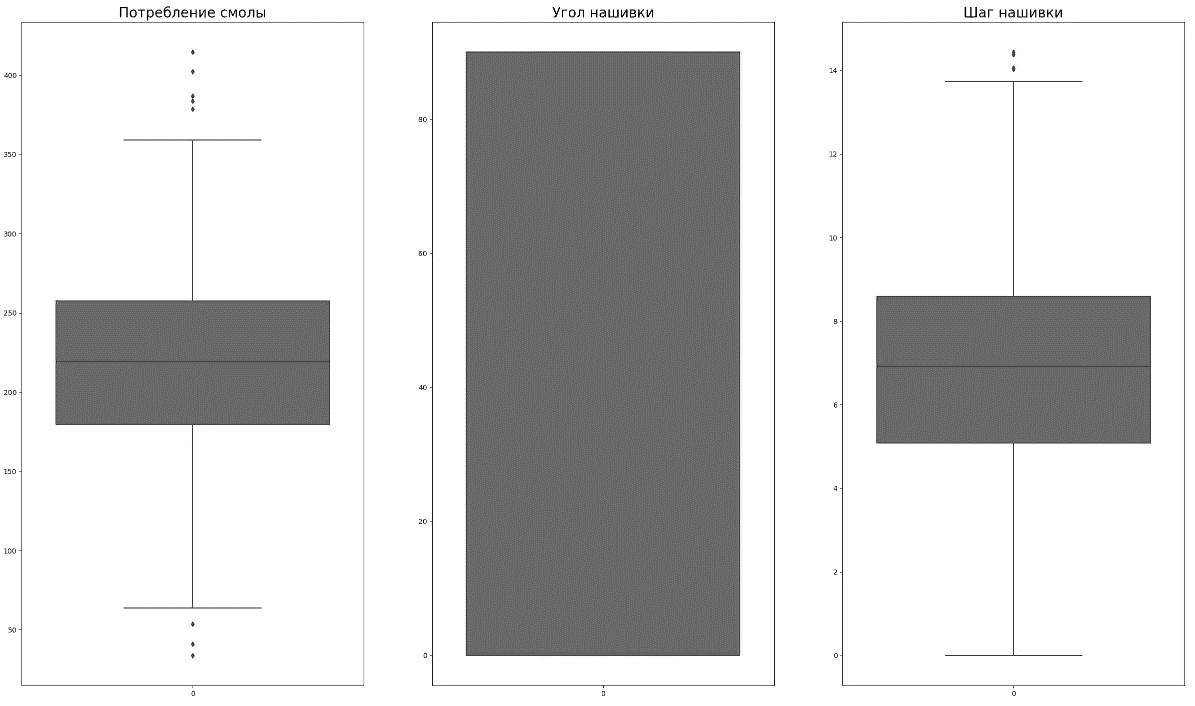
Рисунок 3 – Корреляционная матрица

Анализ выбросов показал наличие значений вне минимума и максимума диапазона распределения по всем признакам, однако, выполнив более детальный анализ данных значений и в целом распределения значений по каждому признаку, можно сделать вывод, что данные значения не являются выбросами. В связи с чем их исключение из набора данных не требуется (что подтверждается экспериментом по исключению данных из выборки и последующему прогнозу – исключение данных не повлияло существенным образом на качество моделей).









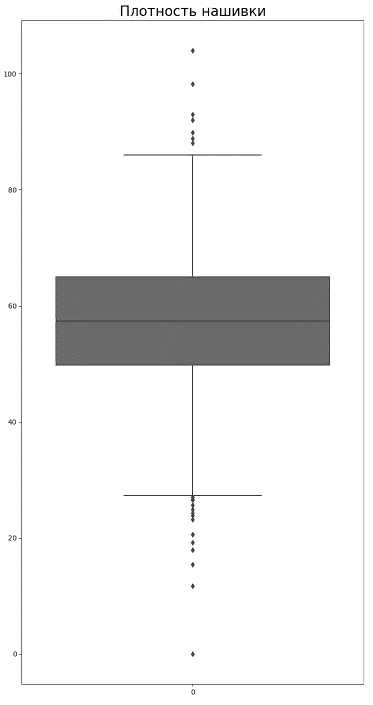
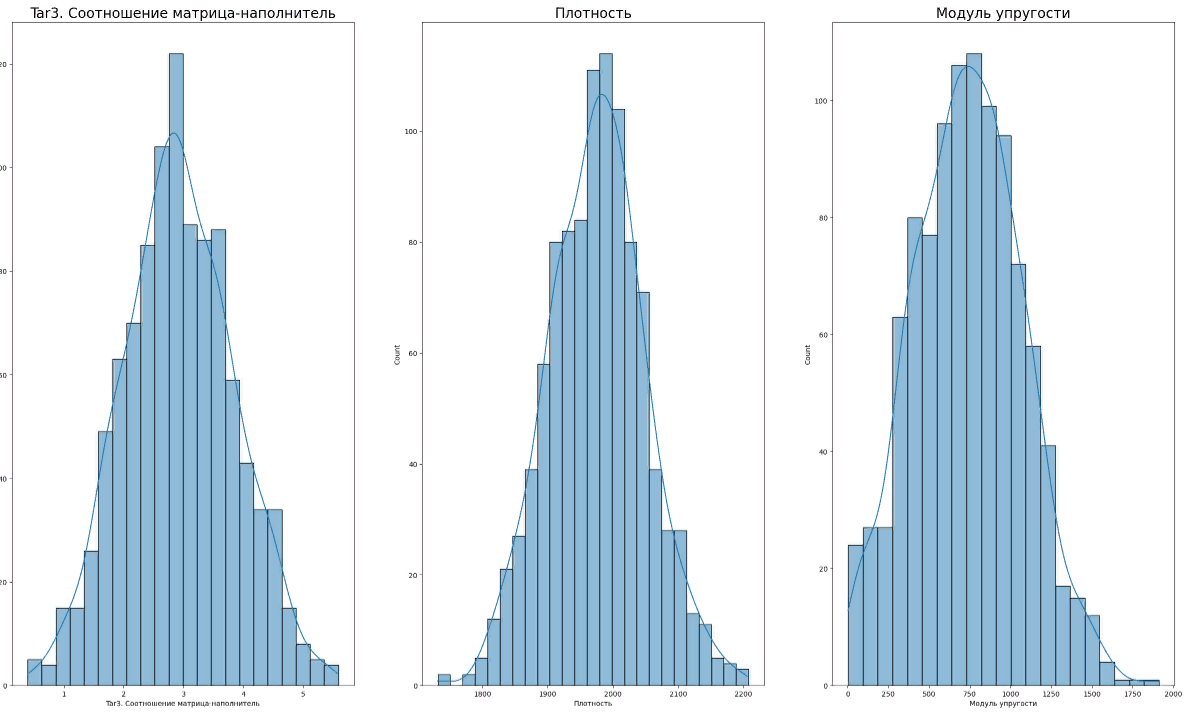
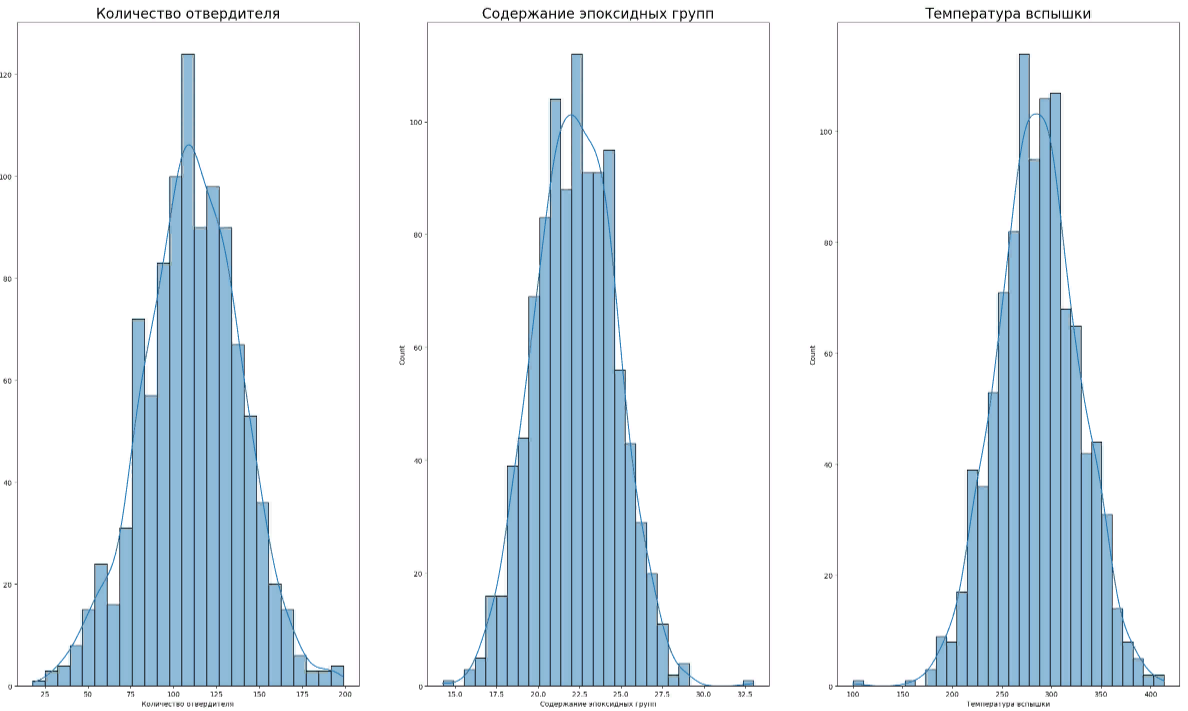
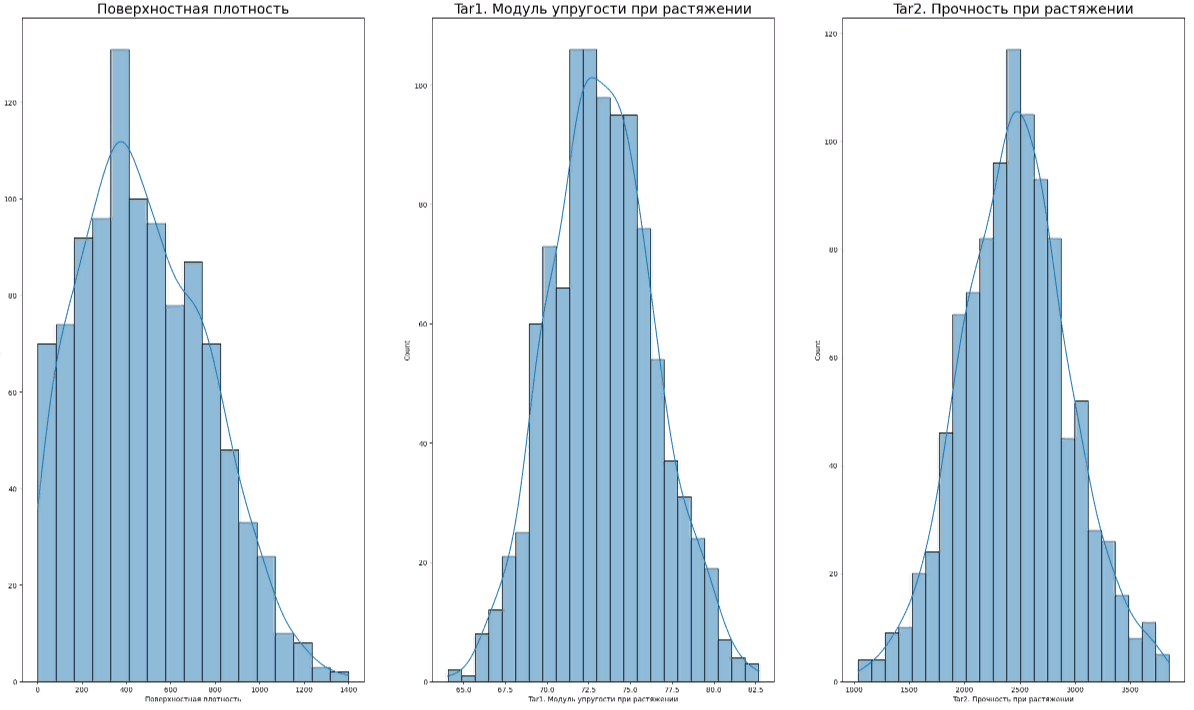


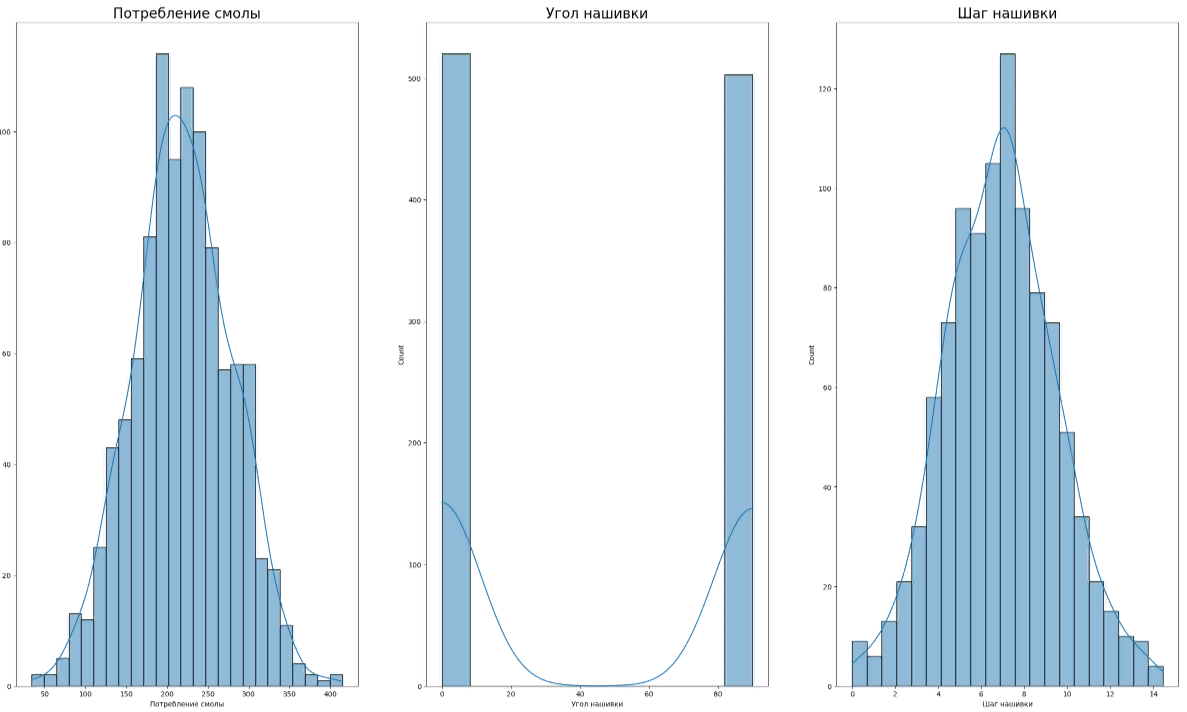
Рисунок 4. Визуализированный анализ выбросов (ящик с усами boxplot)

Анализ распределения переменных показал наличие в целом распределения переменных по каждому признаку, близкому к нормальному распределению Гаусса, при этом у признака «Поверхностная плотность» наблюдается небольшое смещение значений, а распределение значение признака «Угол нашивки» полностью не соответствует нормальному. Однако экспериментальное приведение к гауссову распределению признака «Поверхностная плотность» и полное исключение из набора данных признака «Угол нашивки» существенно не повысили качество прогнозирующих моделей, в связи с чем его выполнение данных операций излишне.









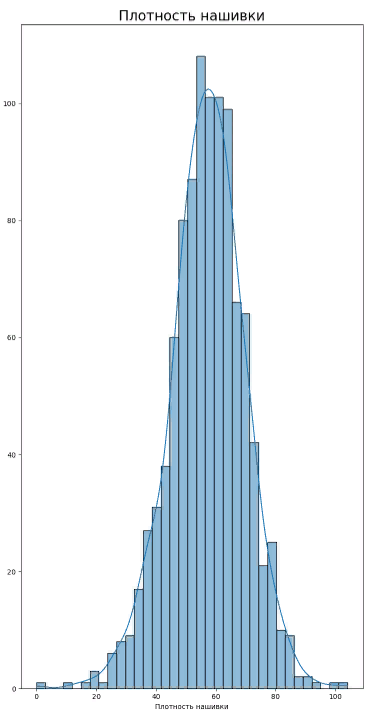
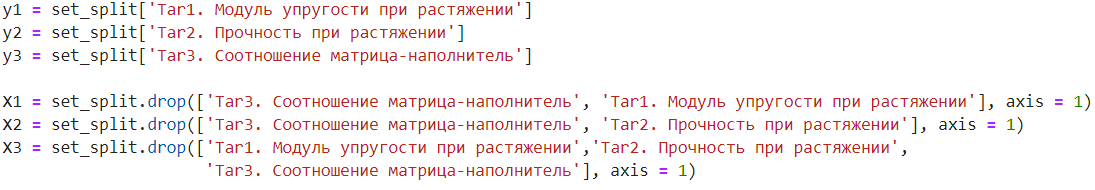


Рисунок 5 - Анализ распределения значений переменных

## **Предобработка данных**

Поставленная задача предполагает прогнозирование трех переменных: «Модуль упругости при растяжении», «Прочность при растяжении» и «Соотношение матрица-наполнитель». Причем прогноз переменной «Соотношение матрица-наполнитель» должен основываться на двух первых спрогнозированных переменных.

Разделяем исходный датасет на «X» (feature-переменные) и «у» (target-переменные):



Следующим шагом применяем к X-данным метод масштабирования StandartScaler. На примере переменных «Плотность нашивки» и «Шаг нашивки» видно, что значения стандартизировались.

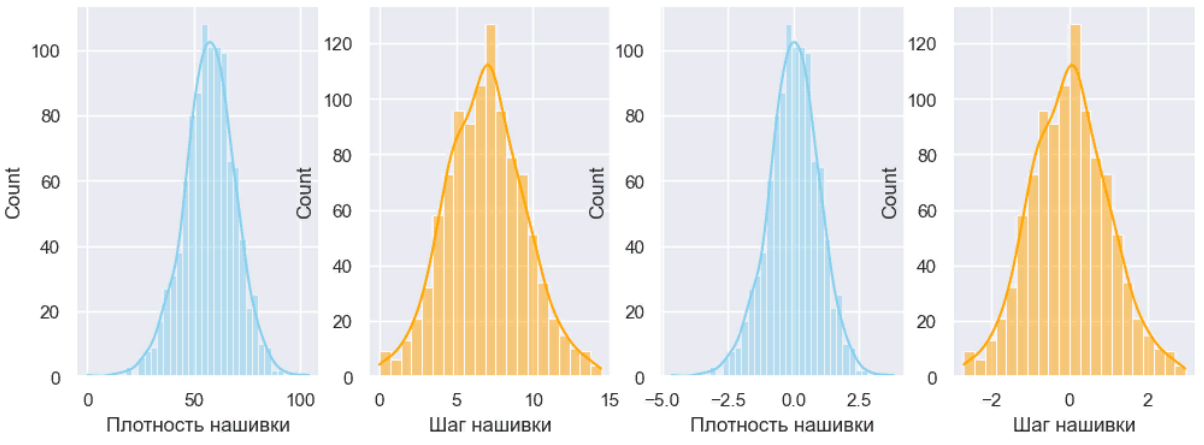


Рисунок 6 - Сравнение данных до масштабирования и после

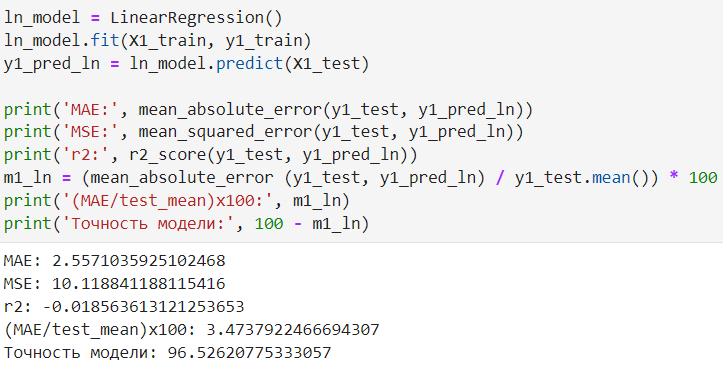
Завершающим этапом препроцессинга данных является разделение массивов «Х» и «у» на обучающие и тестовые выборки методом train\_test\_split, размер тестовой выборки принят равным 30% от общей, включено перемешивание shuffle данных для исключения последовательности и каких-либо закономерностей при составлении изначального датасета, а также путем указания значения random\_state зафиксирован набор данных в выборках для создания единых условий для разных моделей прогнозирования.



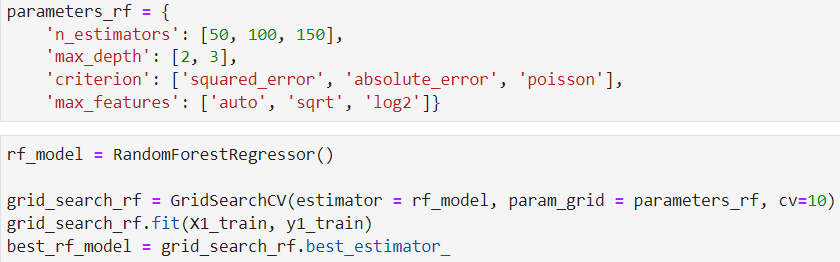
## **Разработка, обучение и тестирование моделей**

### **Построение моделей для прогнозирования модуля упругости при растяжении**

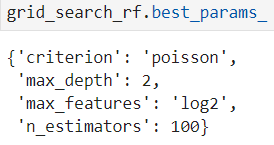
* 1. Применяем модель Linear Regression без указания конкретных гиперпараметров, на вход модели подается X1 и у1.

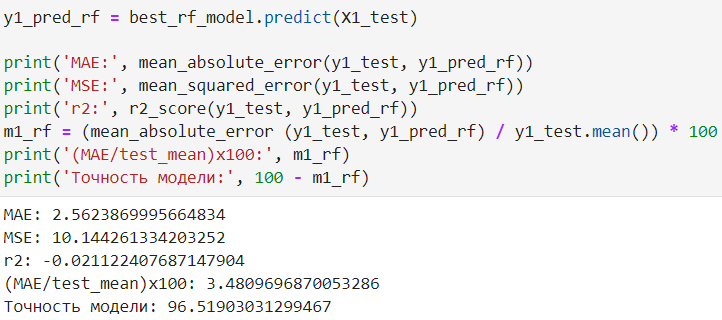


* 1. Применяем модель Random Forest Regression c апгрейдом в виде алгоритма поиска гиперпараметров по сетке Grid Search с перекрестной кросс-валидацией, кол-во итераций валидации принимается равным 10, на вход подается X1 и у1. В качестве параметров указывается кол-во деревьев в выстраиваемом лесу, максимальная глубина деревьев, функция для оценки качества разделения деревьев и функции, применяемые при поиске наилучшего разделения.



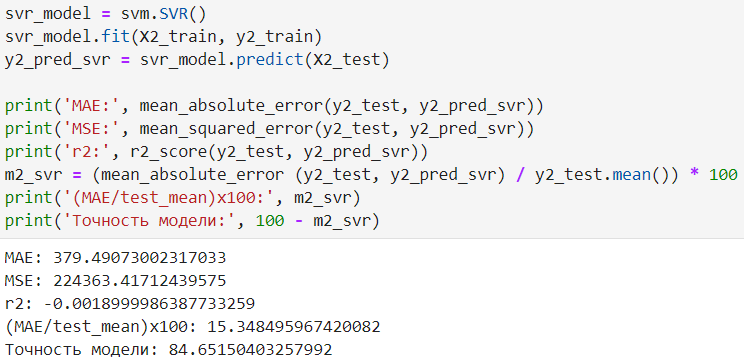
Поиск по сетке определил наилучшие параметры для модели: 100 деревьев с максимальной глубиной, равной 2, функцию оценки качества разделения, которая использует уменьшение в отклонении Пуассона для поиска разделений, и функцию логарифмирования сэмплов.



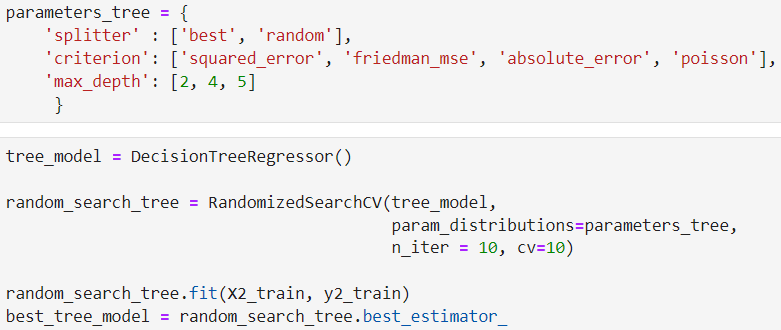


### **Построение моделей для прогнозирования прочности при растяжении**

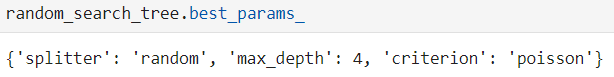
* 1. Применяем модель «Support Vector Regression» без указания конкретных гиперпараметров, на вход модели подается X2 и у2.

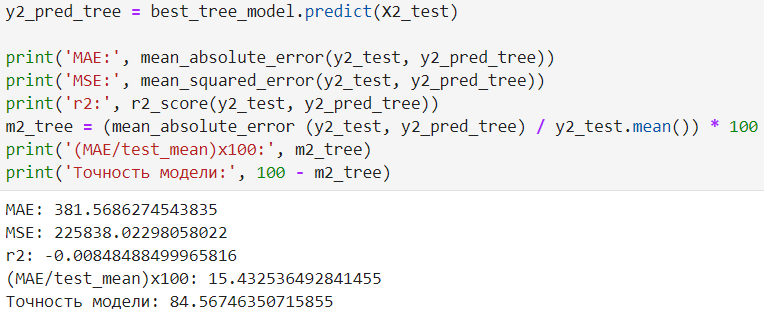


* 1. Применяем модель Decision Tree Regression c апгрейдом в виде алгоритма случайного поиска гиперпараметров Random Search с перекрестной кросс-валидацией, кол-во итераций валидации принимается равным 10, на вход подается X2 и у2. В качестве параметров указывается максимальная глубина деревьев, функция для оценки качества разделения деревьев и функции, и стратегия, используемая для выбора разделения на каждом узле.



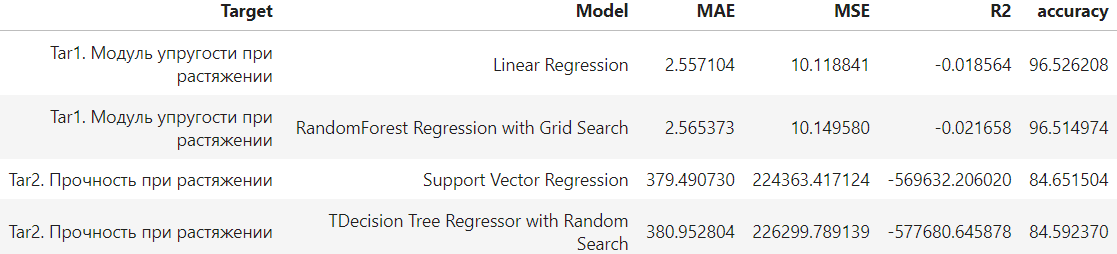
Случайный поиск определил наилучшие параметры для модели: максимальная глубина деревьев, равная 4, функция оценки качества разделения, которая использует уменьшение в отклонении Пуассона для поиска разделений, и стратегия «random» для выбора наилучшего случайного разделения.





Соберем метрики в единую таблицу для бенчмаркинга результативности моделей.

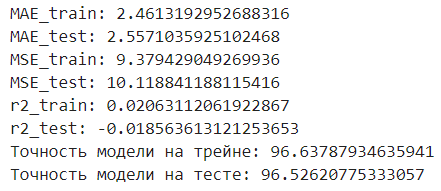
Таблица 2 – бенчмаркинг результативности моделей



Как видно из таблицы, модели показали практически одинаковую результативность при прогнозировании первой и второй целевых переменных соответственно. Модели, определявшие модуль упругости при растяжении, хорошо справились с задачей, точность составляет порядка 96%. Модели, определявшие прочность при растяжении, показали худший результат, точность составляет порядка 85%.

При этом модели с параметрами по умолчанию показали несколько лучший результат, чем модели с подбираемыми по сетке или случайным способом параметрами.

Дополнительно для модели Linear Regression был выполнен расчет метрик для обучающих данных.



Из приведенных данных видно, что на тестовой выборке ошибки немного больше, а точность меньше, чем на обучающей, что является нормальным. Аналогичное соотношение наблюдается у всех моделей прогнозирования, поэтому подробные расчеты метрик для других моделей в пояснительной записке отражаться не будут.

Стоит отметить, что у всех моделей коэффициент детерминации R2 отрицательный, что, согласно теории, говорит о некачественной модели и несоответствии спрогнозированных данных реальным. Скорее всего данные значения обусловлены спецификой исходного набора данных.

### **Построение нейронной сети для прогнозирования соотношения матрица-наполнитель**

#### **Предобработка данных**

Для построения нейронной сети, прогнозирующей соотношение матрица-наполнитель, по условиям задания предварительно необходимо спрогнозировать значения модуля упругости при растяжении и прочности при растяжении. После выполнения указанных прогнозов, их масштабирования и конкатенации к набору данных Х3 получим следующий датасет с feature-переменными.

Таблица 3 – набор данных Х3 (первые 5 строчек)



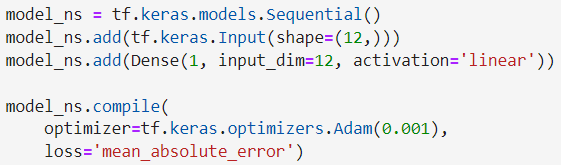
Выполним разделение на обучающие и тестовые выборки с сохранением условий, принятых при разделении при прогнозировании y1 и y2.



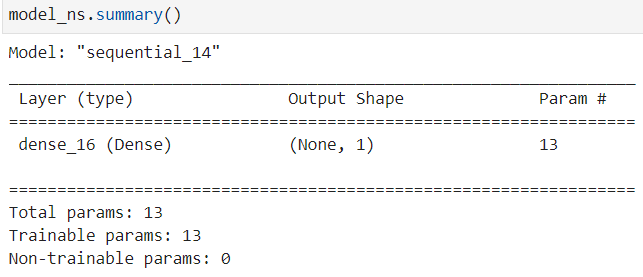
#### **Последовательная нейронная сеть**

Для построения модели нейронной сети использована библиотека keras tensorflow.

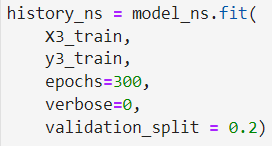
Построена простейшая последовательная нейронная сеть без скрытых слоев с 1 нейроном на выходе, оптимизатором Adam и метрикой оценки ошибки MAE.



Архитектура нейронной сети будет следующей:



По результатам обучения сети на 300 эпохах построим график снижения MAE на обучающей и валидационной выборках и оценим точность сети.



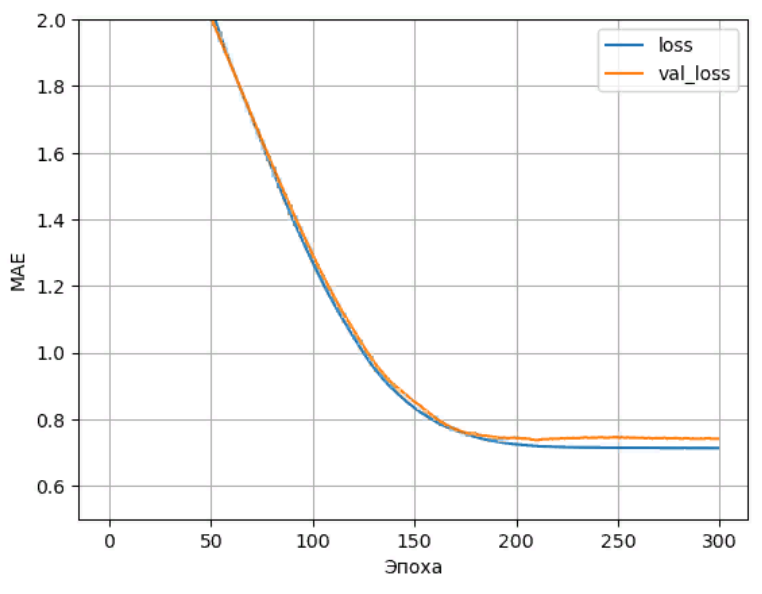
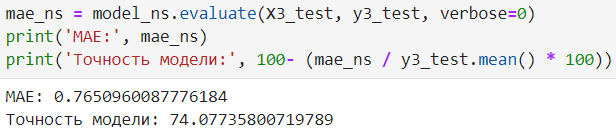
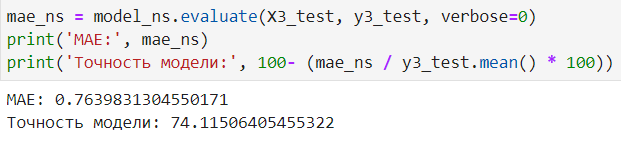


Рисунок 7 – снижение МАЕ на обучающей и валидационной выборках



Повторное обучение сети на 300 эпохах немного улучшило её результативность.



Далее были выполнены эксперименты с добавлением скрытых слоев, варьированием кол-ва нейронов в слоях, активационных функций, оптимизаторов и кол-ва эпох обучения, при этом наблюдалось переобучение модели, отражающееся в росте ошибки на валидационных данных при сохранении тенденции снижения ошибки на обучающих, и снижение её качества прогнозирования.

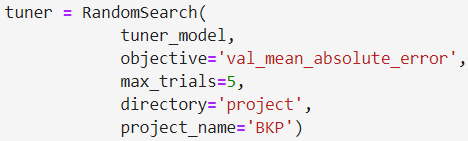
#### **Keras Tuner**

Keras Tuner представляет собой алгоритм автоматического подбора гипермараметров нейронной сети из библиотеки keras tensorflow аналогично алгоритмам Grid Search и Random Search.

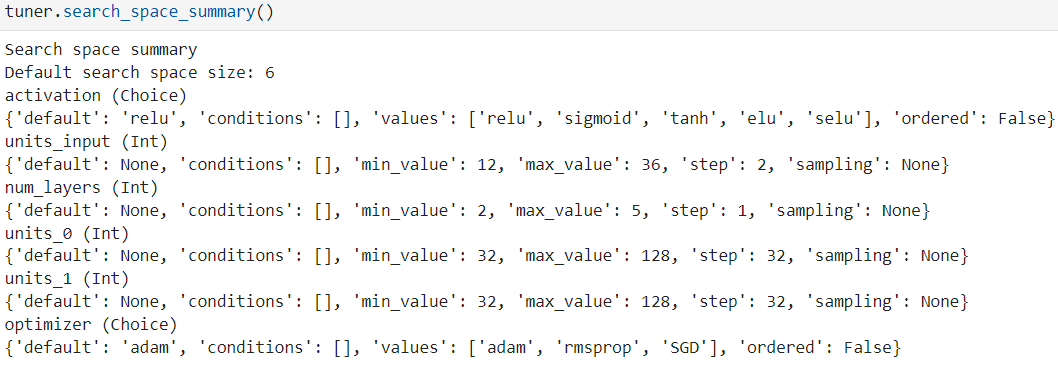
Для построения нейронной сети задаем диапазоны возможных активационных функций для входного и скрытых слоев: relu, sigmoid, tahr, elu, selu. Задаем минимальное 12 и максимальное 36 кол-во нейронов во входном слое с шагом подбора 2. Задаем диапазон возможных скрытых слоев от 2 до 5 и минимальное 32 и максимальное 128 кол-во нейронов в каждом слое с шагом подбора 32. Определяем выходной слой с одним нейроном и линейной функцией активации. Задаем возможные варианты функции оптимизации: adam, rmsprop, SGD.



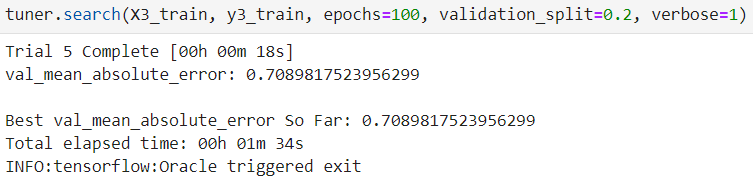
Задаем тюнер варианта RandomSearch, параметр, которую нужно оптимизировать - МАЕ на валидации, кол-во итераций подбора - 5, указываем директорию для сохранения модели и её наименование.



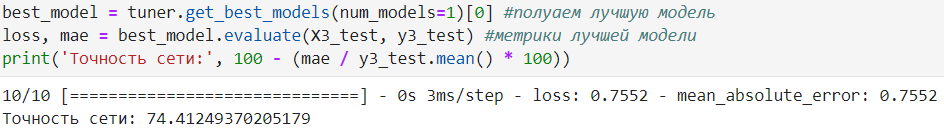
Методом tuner.search\_space\_summary() вызываем сетку всех возможных вариантов поиска гиперпараметров сети.



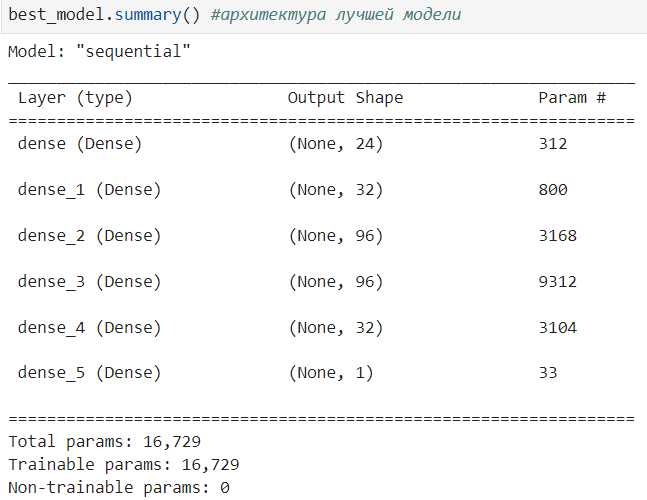
Производим поиск и обучение сети на 100 эпохах.



Определяем лучшую модель нейронной сети и её метрики.



Архитектура подобранной нейронной сети выглядит следующим образом. Сеть содержит помимо входного и выходного слоев 4 скрытых слоя с кол-вом нейронов от 32 до 96.



МАЕ построенной посредством подбора гиперпараметров нейронной сети составила 0,7552, точность accuracy составила порядка 74%. Стоит отметить, что метрика R2 так же как и во всех ранее построенных моделях и нейронных сетях показала отрицательный результат, что опять-таки может свидетельствовать об отсутствии взаимосвязей и низком качестве исходных данных, предоставленных для решения задач выпускной квалификационной работы.

## **Создание flask- приложения**

Flask — это небольшой и легкий веб-фреймворк, написанный на языке Python, предлагающий полезные инструменты и функции для облегчения процесса создания веб-приложений с использованием Python. Он обеспечивает гибкость и является более доступным фреймворком для новых разработчиков, так как позволяет создать веб-приложение быстро, используя только один файл Python.

Напишем flask-приложение для прогнозирования значений модуля упругости при растяжении путем ручного ввода значений параметров, предварительно сохранив модель Linear Regession и создав требуемый html-шаблон.



При запуске приложения и переходе по сгенерированной ссылке открывается форма ввода исходных значений параметров, на основании которых будет спрогнозировано искомое значение.



Рисунок 8 – форма ввода в web-приложении

Выполнив ввод значений и нажав на кнопку «Predict» получаем значение прогноза.

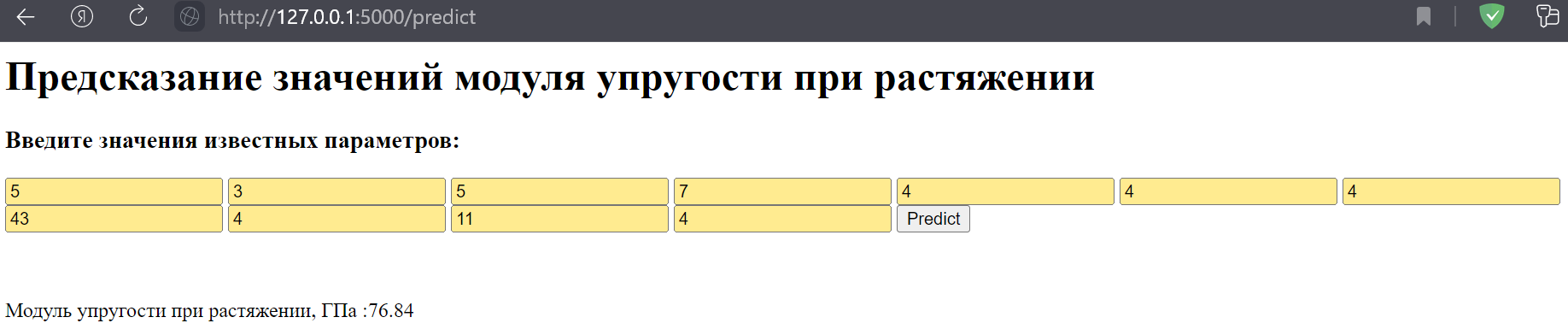
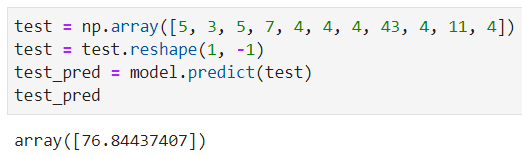


Рисунок 9 – исходные и спрогнозированное значение в web-приложении

Для проверки правильности прогноза создадим в ноутбуке массив с введенными в web-приложении значениями и выполним прогноз. Значение модуля упругости, полученное в результате обработки на языке Python, аналогичен результату, полученному в web-приложении.



# **Заключение**

В настоящей выпускной квалификационной работе были рассмотрены аспекты решения задач регрессии в машинном обучении, а именно: актуальность задачи, шаги и методы разведочного анализа и предобработки данных, некоторые существующие модели прогнозирования и искусственные нейронные сети, алгоритмы автоматического подбора гиперпараметров используемых для прогноза механизмов, а также метрики оценки работы таких механизмов.

Также были рассмотрены способы создания flask-приложения для развертывания модели для общего доступа, тонкости работы с Git для создания коммитов, был создан репозиторий с материалами проекта на сайте [www.github.com](http://www.github.com), расположенный по адресу: <https://github.com/NickSib/Composite>.

Рассмотренные алгоритмы предобработки данных и прогнозирования в целом подтверждают свою гибкость и эффективность, при этом некоторые полученные метрики результатов работы моделей и нейронных сетей лишний раз свидетельствуют о необходимости наличия достаточно объемного исходного массива данных с присутствием взаимосвязей и зависимостей элементов внутри для построения действительно близкого к реальным значениям прогноза.

# **Список литературы**

* 1. А.В. Протодьяконов, П.А. Пылов, В.Е. Садовников. Алгоритмы Data Science и их практическая реализация на Python: учебное пособие. Москва, Вологда, «Инфра-Инженерия», 2022, 392 с.;
  2. Хайкин Саймон. Нейронные сети: полный курс, 2-е издание. Издательский дом «Вильямс», 2006, 1104 с.;
  3. Ф.М. Гафаров, А.Ф. Галимянов. Искусственные нейронные сети и приложения: учеб. Пособие. Изд-во Казан. ун-та, 2018.
  4. Дюк В., Самойленко А. Data minig: учебный курс. Издательский дом «Питер», 2001, 368 с.;
  5. Грас Д. Data science: наука о данных с нуля. БВХ-Петербург, 2021, 416 с.;
  6. А. Джулли, с. Пал. Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow. ДМК пресс, 2018, 294 с.;
  7. Аллен Б. Дауни .Основы Python. Научитесь думать как программист. Манн, Иванов и Фербер, 2021, 304 с.;
  8. Б. Любанович. Простой Python. Современный стиль программирования. 2-е изд. СПб.: Питер, 2021, 592 с.;
  9. Рассел Стюарт, Норвиг Питер. Искусственный интеллект: современный подход, 2-е изд.. Издательский дом ‘‘Вильямс’’, 2007, 1408 с.;
  10. Жерон Орельен. Прикладное машинное обучение с помощью Scikit-Learn, Keras и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем, 2-е изд. СПб.: «Диалектика», 2020, 1040 с.;
  11. Справочник по библиотеке TensorFlow <https://www.tensorflow.org>.
  12. Справочник по библиотеке Matplotlib: <https://matplotlib.org>;
  13. Справочник по библиотеке Seaborn <https://seaborn.pydata.org>;
  14. Справочник по библиотеке Scikit-Learn <https://scikit-learn.org>;
  15. Справочник по Keras Tuner <https://keras.io/keras_tuner/>;