

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

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  
**по курсу**  
**«Data Science»**

Слушатель

Зиянгирова Зинаида Михайловна

Москва, 2022

## Содержание

Введение. ....	3
1. Аналитическая часть .....	4
1.1 Постановка задачи .....	4
1.2 Описание используемых методов .....	4
1.3 Разведочный анализ данных .....	6
2 Практическая часть .....	8
2.1 Предобработка данных .....	8
2.2 Разработка и обучение модели .....	13
2.3 Тестирование модели .....	14
2.4 Нейронная сеть, которая будет рекомендовать соотношение матри- ца-наполнитель. ....	15
2.5 Разработка приложения .....	17
2.6 Создание удаленного репозитория и загрузка результатов работы. ....	18
Заключение .....	19
Библиографический список .....	20

## 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, 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. 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 Аналитическая часть.**

### **1.1 Постановка задачи**

Для решения задачи прогнозирования конечных свойств новых материалов (композиционных материалов) предоставлено 2 датасета: «X\_br» и «X\_pur», в формате `xlsx`.

Необходимо выполнить следующие действия:

- Объединить имеющиеся датасеты по индексу тип объединения INNER;
- обучить алгоритм машинного обучения, который будет определять значения Модуль упругости при растяжении, ГПа и Прочность при растяжении, МПа;
- написать нейронную сеть, которая будет рекомендовать Соотношение матрица-наполнитель;
- написать приложение, которое будет выдавать прогноз, полученный ранее, при работе с алгоритмами машинного обучения или нейронной сетью;
- создать профиль на `github.com`;
- сделать `commit` приложения на `github.com`.

Для выполнения задач использовалась среда Google Colaboratory, версия Python 3.9.16, версия TensorFlow 2.12.0.

### **1.2 Описание используемых методов**

Для прогнозирования конечных свойств композиционных материалов необходимо решить задачу регрессии. Задача регрессии в машинном обучении — это предсказание одного параметра  $Y$  по известному параметру  $X$ , где  $X$  — набор параметров, характеризующий наблюдение.

Для решения поставленной задачи предполагается использовать методы, представленные в Таблице 1.

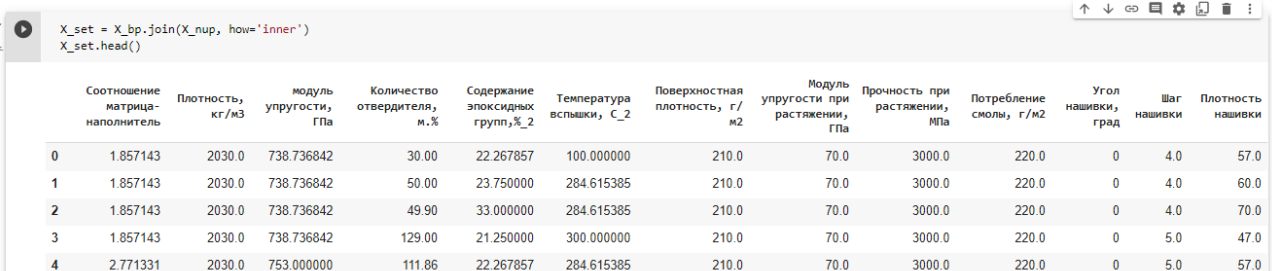
Таблица 1 – Описание используемых методов.

Метод	Описание	Достоинства	Недостатки
Линейная регрессия (англ. Linear regression)	Используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной $Y$ от другой или нескольких других переменных $X$ с линейной функцией зависимости. Целью обучения является поиск линии наилучшего соответствия.	Прост в реализации, легко интерпретируем.	Может моделировать только прямые линейные зависимости, в то время как часто возникает необходимость создания модели других типов отношений между данными; чувствителен к выбросам
Случайный лес (англ. — Random forest)	Алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев. Каждое дерево строится на случайном подмножестве обучающих данных и случайном подмножестве признаков. В результате каждое дерево в ансамбле отличается от других, что повышает качество предсказаний. При решении задачи регрессии результаты всех деревьев усредняются, и это значение становится предсказанием.	Способность обрабатывать данные с большим количеством признаков. Имеет возможность обрабатывать отсутствующие значения. Способность обрабатывать шумные и нелинейные данные. Модель является достаточно устойчивой к переобучению. Способность измерять важность каждого признака для задачи.	Модель может быть сложной для интерпретации, так как она объединяет множество деревьев решений. Потребность в настройке гиперпараметров, таких как количество деревьев и глубина каждого дерева. Для обучения модели может потребоваться большое количество времени и вычислительных ресурсов, особенно при использовании большого количества деревьев.
К-ближайших соседей (англ. - k-nearest neighbors)	Алгоритм машинного обучения, в котором объекту присваивается среднее значение по $k$ ближайшим к нему объектам, значения которых уже известны.	Прост и легко реализуем. Не чувствителен к выбросам. Нет необходимости строить модель, настраивать несколько параметров или делать дополнительные допущения.	При увеличении объема выборки, предикторов или независимых переменных требуется большое количество времени и вычислительных ресурсов. Необходимо определять оптимальное значение $k$ .

### 1.3 Разведочный анализ данных

Разведочный анализ данных (англ. Exploratory data analysis, EDA) — анализ основных свойств данных, нахождение в них общих закономерностей, распределений и аномалий, построение начальных моделей, зачастую с использованием инструментов визуализации.

После объединения предоставленных данных по типу INNER датасет приобрел вид в соответствии с рисунком 1.



```
X_set = X_bp.join(X_nup, how='inner')
X_set.head()
```

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп, %_2	Температура вспышки, C_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0	1.857143	2030.0	738.736842	30.00	22.267857	100.000000	210.0	70.0	3000.0	220.0	0	4.0	57.0
1	1.857143	2030.0	738.736842	50.00	23.750000	284.615385	210.0	70.0	3000.0	220.0	0	4.0	60.0
2	1.857143	2030.0	738.736842	49.90	33.000000	284.615385	210.0	70.0	3000.0	220.0	0	4.0	70.0
3	1.857143	2030.0	738.736842	129.00	21.250000	300.000000	210.0	70.0	3000.0	220.0	0	5.0	47.0
4	2.771331	2030.0	753.000000	111.86	22.267857	284.615385	210.0	70.0	3000.0	220.0	0	5.0	57.0

Рисунок 1 – первые пять строк объединенного датасета

Для получения статистики по набору данных использовались следующие команды библиотеки pandas:

- `info()` - вывод информации о количестве непустых значений и типах переменных;
- `nunique()` – количество уникальных значений по каждой переменной;
- `isnull()` - вывод информации о количестве пропусков;
- `duplicated()` - количество полностью совпадающих строк;
- `describe()` - вывод описательной статистики с информацией о количестве значений, среднем значении, стандартном отклонении, минимальном и максимальном значениях, квартилях.

В исследуемом датасете (`X_set`) содержится 13 столбцов и 1023 строки. Анализ данных показал, что в наборе данных отсутствуют пропуски и строки-дубликаты. Описательная статистика датасета показана на рисунке 2. Информация о количестве непустых значений, типах переменных и количестве уникальных значений представлена в Таблице 2.

X\_set.describe().T

	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, C_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
Шаг нашивки	1023.0	6.899222	2.563467	0.000000	5.080033	6.916144	8.586293	14.440522
Плотность нашивки	1023.0	57.153929	12.350969	0.000000	49.799212	57.341920	64.944961	103.988901

Рисунок 2 – описательная статистика датасета

Таблица 2 – характеристика датасета

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

Для визуализации данных использовались библиотеки matplotlib и seaborn. В проекте применены следующие команды для первоначального анализа:

- `histplot ()`, для вывода гистограмм распределения значений в столбцах;
- `boxplot ()`, для вывода ящиков с усами;
- `pairplot ()`, для вывода графиков попарного распределения переменных;
- `heatmap ()`, для вывода тепловой карты корреляции.

## 2 Практическая часть.

### 2.1 Предобработка данных

Для каждого параметра были построены гистограммы распределения переменных (в соответствии с рисунком 3), ящики с усами (в соответствии с рисунком 4) и попарные графики рассеивания точек (в соответствии с рисунком 5).

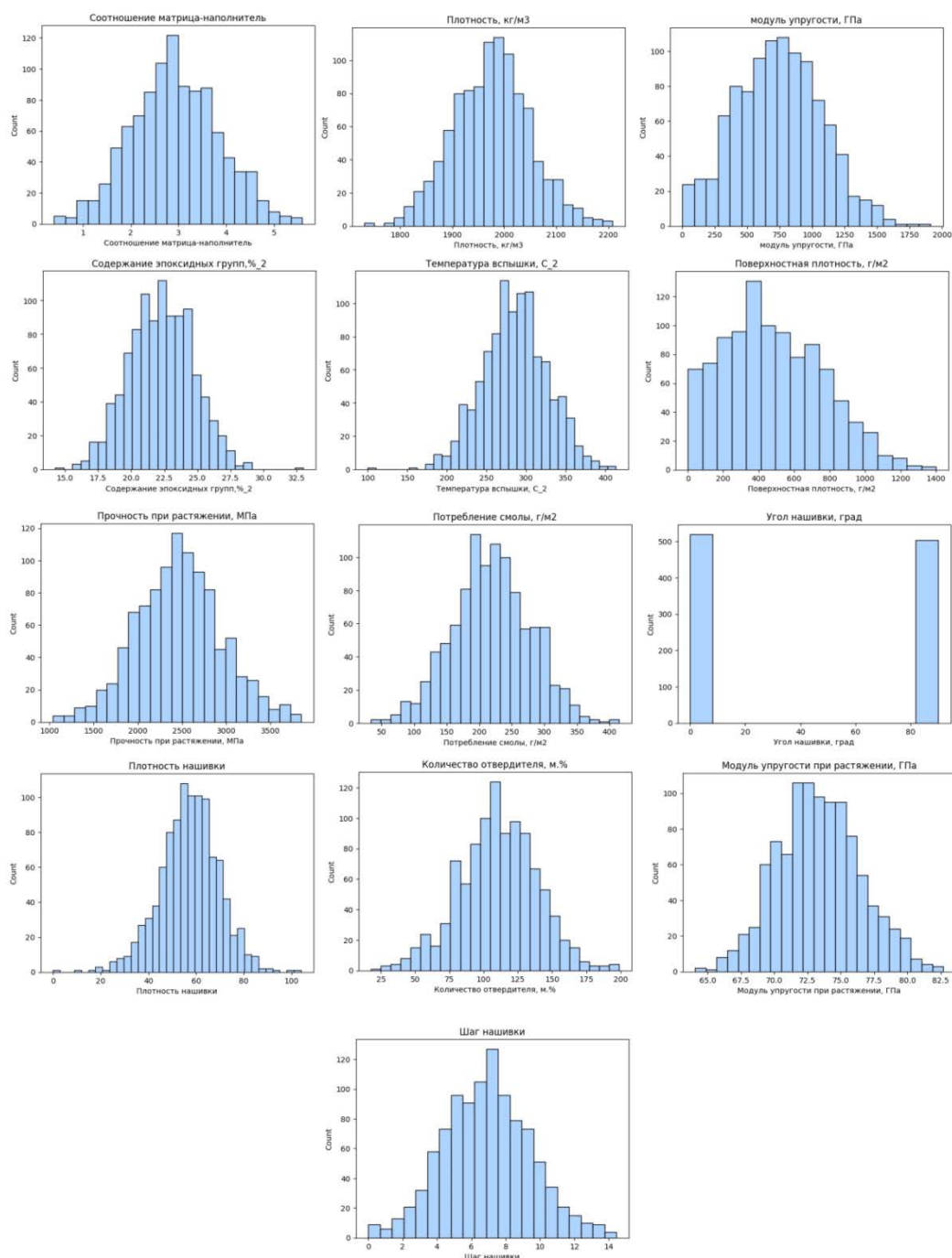


Рисунок 3 - гистограммы распределения переменных



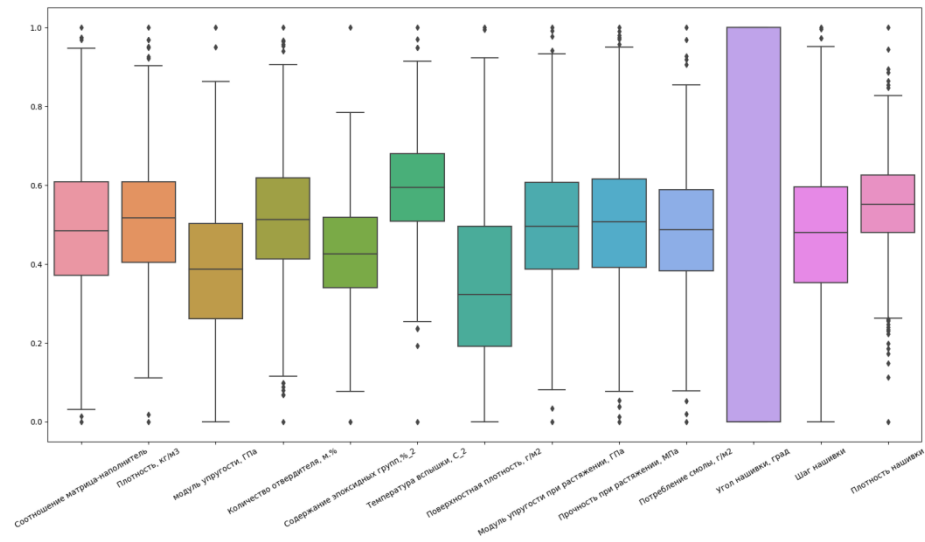


Рисунок 4 – ящики с усами

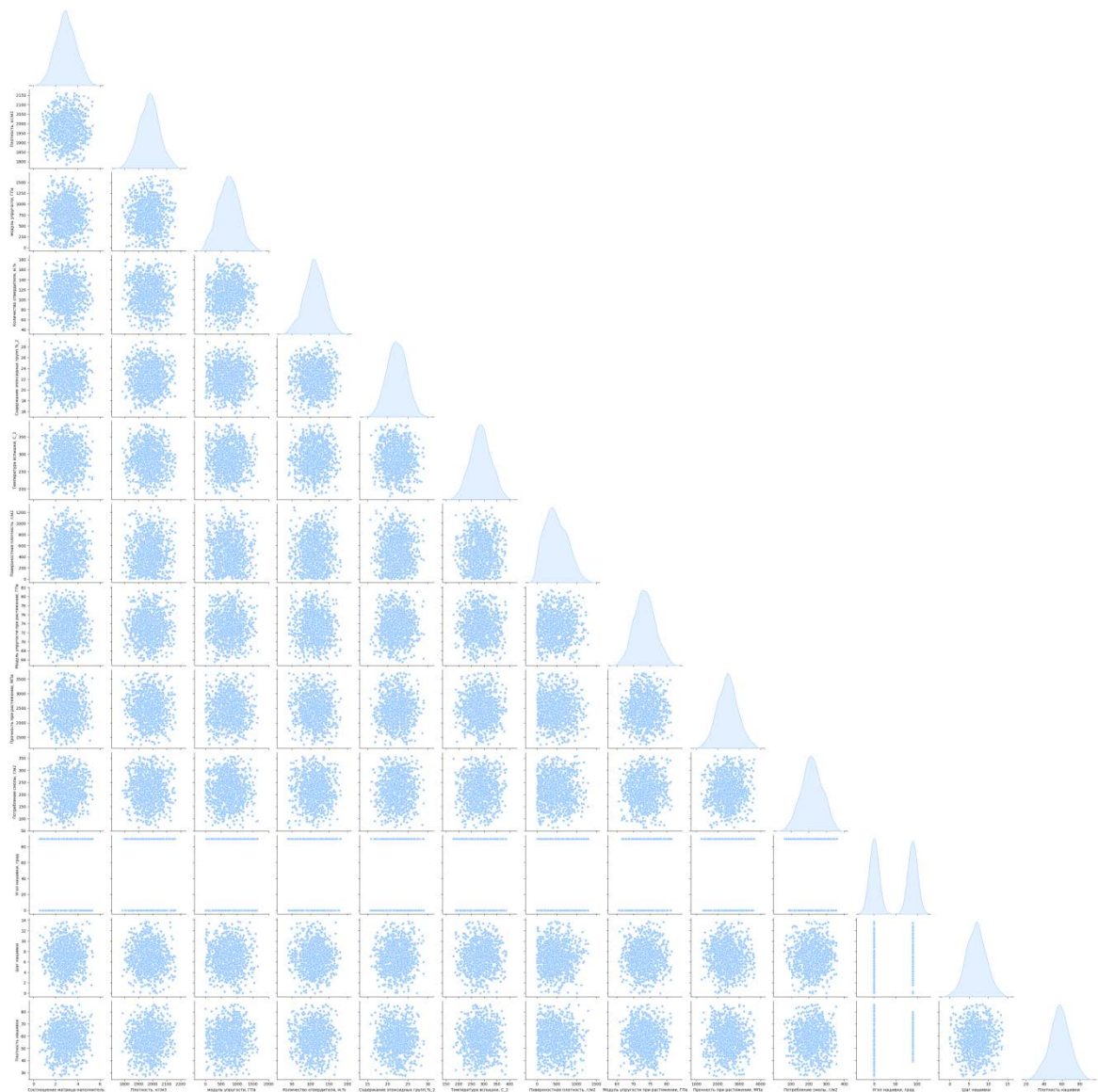


Рисунок 5 – попарные графики рассеивания

Гистограммы распределения используются для визуального изучения частот значений переменных. Попарные графики рассеяния переменных – для выяснения отношения между двумя числовыми наборами данных. Ящики с усами (англ. – Boxplot) позволяют визуально установить наличие выбросов.

Выбросы – это результаты измерений, выделяющиеся из общей выборки. Для их нахождения был использован метод межквартильных интервалов (InterQuartile Range, IQR). Он определяется через квантили, а именно принимается равным разнице между 75-м и 25-м процентилями, то есть между третьим и первым квартилями.

В рабочем наборе данных выбросы были обнаружены в 12 столбцах. Угол нашивки принимает только 2 значения, 0 и 90 град, поэтому он был исключен из списка колонок для удаления выбросов. Код для определения и замены выбросов представлен на рисунке 6. Аномальные значения в датасете были заменены на Nan.

```
[16] #создание переменной со списком всех параметров, в которых есть выбросы
X_set.columns
column_list_outliers = ["Соотношение матрица-наполнитель",
                        "Плотность, кг/м3",
                        "Модуль упругости, ГПа",
                        "Количество отвердителя, м.%",
                        "Содержание эпоксидных групп, %_2",
                        "Температура вспышки, C_2",
                        "Поверхностная плотность, г/м2",
                        "Модуль упругости при растяжении, ГПа",
                        "Прочность при растяжении, МПа",
                        "Потребление смолы, г/м2",
                        "Шаг нашивки",
                        "Плотность нашивки"]

for i in column_list_outliers:
    Q3, Q1 = np.percentile(X_set.loc[:,i],[75, 25])
    IQR = Q3 - Q1
    max = Q3 + (1.5*IQR)
    min = Q1 - (1.5*IQR)
    X_set.loc[X_set[i] < min,i] = np.nan
    X_set.loc[X_set[i] > max,i] = np.nan
```

Рисунок 6 – код для определения и замены выбросов

Для работы с выбросами использовалось 2 метода:

- Удаление значений Nan с помощью функции dropna (). 87 строк, содержащих выбросы, были удалены. В новом датасете (X\_set\_clean) осталось 936 строк. Краткий анализ характеристик очищенного от выбросов датасета представлен на рисунке 7.

[20] X\_set\_clean = X\_set.dropna(axis=0, how='any')

[21] X\_set\_clean.head()

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	количество отвердителя, м.%	Содержание эпоксидных групп, %_2	Температура вспышки, C_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
1	1.857143	2030.0	738.736842	50.00	23.750000	284.615385	210.0	70.0	3000.0	220.0	0	4.0	60.0
3	1.857143	2030.0	738.736842	129.00	21.250000	300.000000	210.0	70.0	3000.0	220.0	0	5.0	47.0
4	2.771331	2030.0	753.000000	111.86	22.267857	284.615385	210.0	70.0	3000.0	220.0	0	5.0	57.0
5	2.767918	2000.0	748.000000	111.86	22.267857	284.615385	210.0	70.0	3000.0	220.0	0	5.0	60.0
6	2.569620	1910.0	807.000000	111.86	22.267857	284.615385	210.0	70.0	3000.0	220.0	0	5.0	70.0

[23] X\_set\_clean.info()

[22] X\_set\_clean.shape

(936, 13)

Рисунок 7 – анализ характеристик датасета X\_set\_clean.

- Замена Nan на средние значения во всех столбцах с помощью функции fillna (). При применении такого решения в новом датасете (X\_set\_mean) сохраняются 1023 строки. Краткий анализ характеристик датасета, в котором выбросы заменены на средние значения, представлен на рисунке 8.

[25] X\_set\_mean = X\_set.fillna(X\_set.mean())

[26] X\_set\_mean.info()

[27] X\_set\_mean.head()

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	количество отвердителя, м.%	Содержание эпоксидных групп, %_2	Температура вспышки, C_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0	1.857143	2030.0	738.736842	110.558048	22.267857	285.949508	210.0	70.0	3000.0	220.0	0	4.0	57.0
1	1.857143	2030.0	738.736842	50.000000	23.750000	284.615385	210.0	70.0	3000.0	220.0	0	4.0	60.0
2	1.857143	2030.0	738.736842	49.900000	22.241680	284.615385	210.0	70.0	3000.0	220.0	0	4.0	70.0
3	1.857143	2030.0	738.736842	129.000000	21.250000	300.000000	210.0	70.0	3000.0	220.0	0	5.0	47.0
4	2.771331	2030.0	753.000000	111.860000	22.267857	284.615385	210.0	70.0	3000.0	220.0	0	5.0	57.0

X\_set\_mean.shape

(1023, 13)

Рисунок 8 – анализ характеристик датасета X\_set\_mean.

Для визуализации определения взаимосвязей между переменными было решено построить тепловую карту корреляции, используя функцию heatmap библиотеки seaborn. Исследовались корреляции в трёх датасетах: X\_set (в соответствии с рисунком 8.1), X\_set\_clean (в соответствии с рисунком 8.2) и X\_set\_drop (в соответствии с рисунком 8.3).

Корреляция между всеми параметрами очень близка к 0, чётко выраженных зависимостей между параметрами не наблюдается. После сравнения значений на тепловой карте корреляции было принято решение дальнейшую работу проводить с датасетом X\_set\_mean, в котором сохранен исходный объем данных.

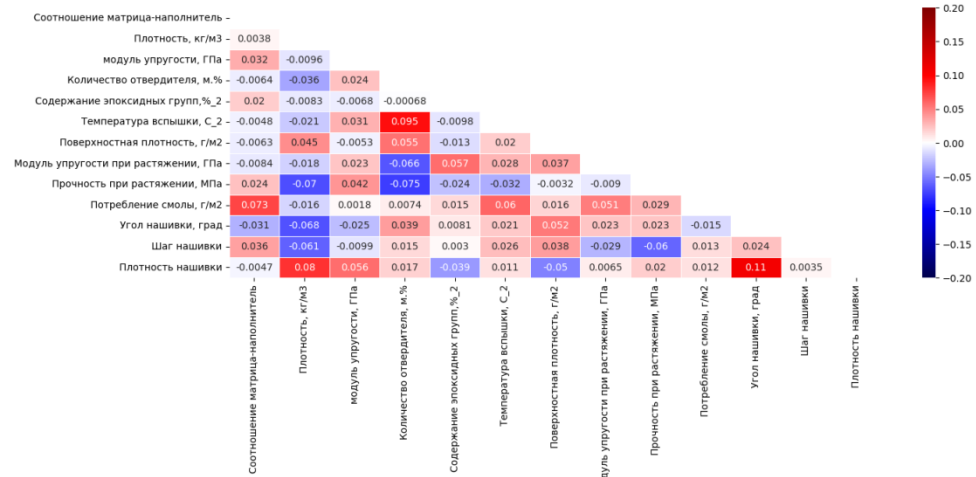


Рисунок 8.1 – Тепловая карта датасета X\_set

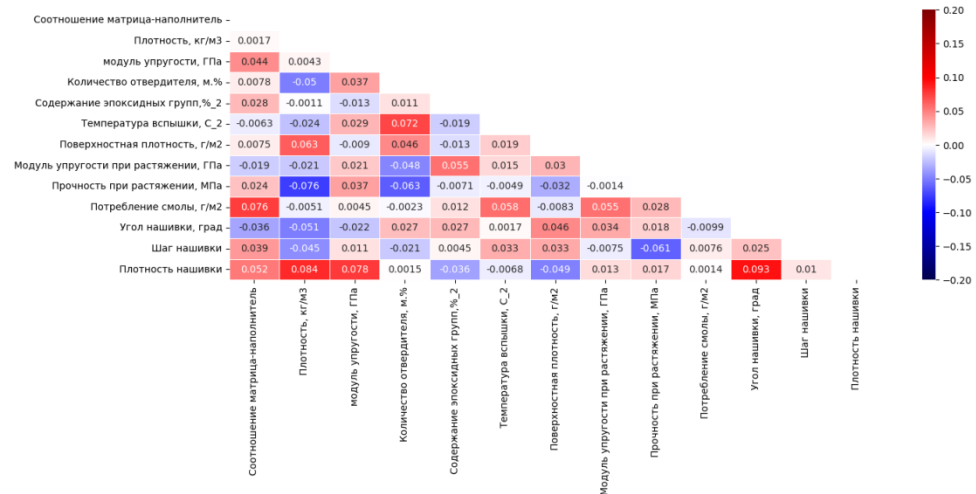


Рисунок 8.2 – Тепловая карта датасета X\_set\_clean

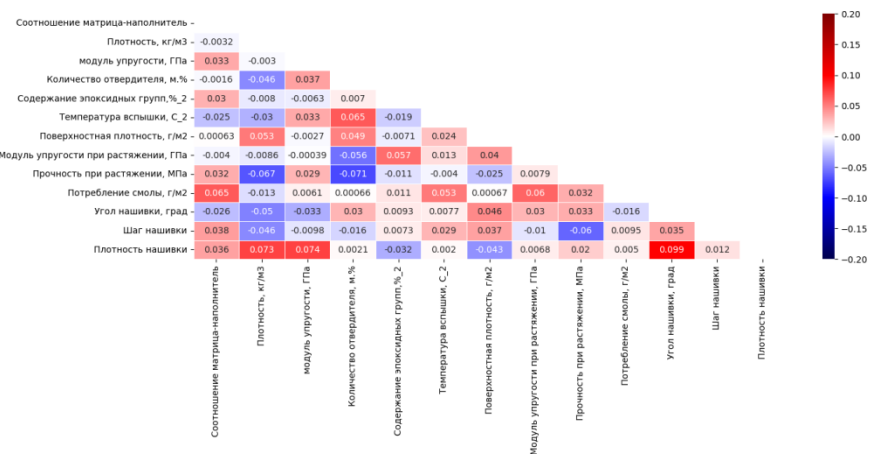


Рисунок 8.3 – Тепловая карта датасета X\_set\_mean

При выполнении разведочного анализа данных было замечено, что значения параметров данных находятся в разных диапазонах. Это может привести к некорректной работе моделей машинного обучения – большой дисбаланс между значениями признаков может ухудшать результаты обучения и замедлять процесс моделирования. Поэтому данные были нормализованы с использованием метода MinMaxScaler из библиотеки Sklearn. Процесс нормализации и описательная статистика нормализованного датасета представлены на рисунке 9.

```
#MinMaxScaler
minmax = preprocessing.MinMaxScaler()
col = df.columns
df_minmax_n = minmax.fit_transform(np.array(df[col]))
df_minmax = pd.DataFrame(data=df_minmax_n, columns=[col])
df_minmax.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	0.498844	0.187191	0.0	0.372092	0.495399	0.629650	1.0
Плотность, кг/м3	1023.0	0.504592	0.187870	0.0	0.371411	0.510844	0.626245	1.0
модуль упругости, ГПа	1023.0	0.446445	0.198214	0.0	0.302135	0.447061	0.579819	1.0
Количество отвердителя, м.%	1023.0	0.502162	0.186237	0.0	0.378699	0.502162	0.632613	1.0
Содержание эпоксидных групп, %_2	1023.0	0.493679	0.178708	0.0	0.371013	0.492857	0.623384	1.0
Температура вспышки, С_2	1023.0	0.515619	0.190405	0.0	0.387036	0.515619	0.644492	1.0
Поверхностная плотность, г/м2	1023.0	0.372145	0.215689	0.0	0.206249	0.349615	0.535487	1.0
Модуль упругости при растяжении, ГПа	1023.0	0.489020	0.191799	0.0	0.361445	0.487331	0.615795	1.0
Прочность при растяжении, МПа	1023.0	0.495005	0.189583	0.0	0.364888	0.493717	0.613140	1.0
Потребление смолы, г/м2	1023.0	0.522725	0.195289	0.0	0.393568	0.523036	0.653680	1.0
Угол нашивки, град	1023.0	0.491691	0.500175	0.0	0.000000	0.000000	1.000000	1.0
Шаг нашивки	1023.0	0.500310	0.183651	0.0	0.369930	0.502022	0.624505	1.0
Плотность нашивки	1023.0	0.513109	0.189698	0.0	0.391631	0.513109	0.635267	1.0

Рисунок 9 – нормализация датасета

## 2.2 Разработка и обучение модели

Процесс подготовки моделей к обучению начинается с определения входных и выходных параметров. Для определения модуля упругости при растяжении (значение которого нужно определить на выходе) было решено использовать значения параметров Количество отвердителя, Содержание эпоксидных групп и Потребление смолы, так как именно они показали наилучшую корреляцию. Предполагается, что остальные параметры усложняют процесс обучения модели и ухудшают точность результата, поэтому они не учитывались в исследовании. Аналогичным образом были определены наилучшие входные параметры для значения прочности при растяжении. Это Плотность, Количество отверди-

теля, Шаг нашивки, Угол нашивки, Потребление смолы и Соотношение матрица-наполнитель. Выборки были разделены на обучающее и тестовое множество. Для прогнозирования модуля упругости при растяжении и прочности при растяжении использовались модели линейной регрессии, случайного леса и метода К- ближайших соседей.

### 2.3 Тестирование модели

Для оценки качества моделей были рассчитаны средняя абсолютная ошибка (MAE), Среднеквадратичная ошибка (MSE), Коэффициент детерминации (R2), ошибка модели и точность модели. Результаты оценки моделей для определения модуля упругости при растяжении представлены на рисунке 10.1, результаты оценки моделей для определения прочности при растяжении представлены на рисунке 10.2.

```
accuracy1 = pd.DataFrame({'Model': ['LinearRegression', 'RandomForestRegressor', 'KNeighborsRegressor'],
                          'MAE': [MAE_linear1, MAE_forest1, MAE_kn1],
                          'MSE': [MSE_linear1, MSE_forest1, MSE_kn1],
                          'R2': [R2_linear1, R2_forest1, R2_kn1],
                          'error': [ep_linear1, ep_forest1, ep_kn1],
                          'accuracy': [ac_linear1, ac_forest1, ac_kn1]})

accuracy1
```

	Model	MAE	MSE	R2	error	accuracy
0	LinearRegression	2.504425	9.603236	-0.028179	3.420074	96.579926
1	RandomForestRegressor	2.566119	10.393928	-0.112835	3.504324	96.495676
2	KNeighborsRegressor	2.489067	9.493850	-0.016468	3.399100	96.600900

Рисунок 10. 1 – оценка моделей

```
[ ] accuracy2 = pd.DataFrame({'Model': ['LinearRegression', 'RandomForestRegressor', 'KNeighborsRegressor'],
                             'MAE': [MAE_linear2, MAE_forest2, MAE_kn2],
                             'MSE': [MSE_linear2, MSE_forest2, MSE_kn2],
                             'R2': [R2_linear2, R2_forest2, R2_kn2],
                             'error': [ep_linear2, ep_forest2, ep_kn2],
                             'accuracy': [ac_linear2, ac_forest2, ac_kn2]})

accuracy2
```

	Model	MAE	MSE	R2	error	accuracy
0	LinearRegression	374.136895	222258.062113	-0.006573	15.239563	84.760437
1	RandomForestRegressor	391.188940	245345.126459	-0.111130	15.934137	84.065863
2	KNeighborsRegressor	373.442818	222241.455938	-0.006497	15.211292	84.788708

Рисунок 10. 2 – оценка моделей



Выбранные модели плохо справились с задачей обучения для имеющихся данных. У всех моделей коэффициент детерминации имеет отрицательные значения. Таким образом, модели не дают прогнозов, которые были бы лучше расчета базовой модели. Свойства композитных материалов в первую очередь зависят от используемых материалов.

Изменение параметров модели случайного леса, таких как количество деревьев и глубина каждого дерева был связан с затратами времени и вычислительных ресурсов. К сожалению, это не позволило улучшить результат. Не удалось подобрать модель, которая могла бы оказать помощь в принятии решений специалисту предметной области.

#### **2.4 Нейронная сеть, которая будет рекомендовать соотношение матрица-наполнитель**

Нейронная сеть — это последовательность нейронов, соединенных между собой связями. Структура нейронной сети пришла в мир программирования из биологии. Вычислительная единица нейронной сети — нейрон или персептрон.

Нейронные сети применяются для решения задач регрессии, классификации, распознавания образов и речи, компьютерного зрения и других. На настоящий момент это самый мощный, гибкий и широко применяемый инструмент в машинном обучении.

Обучение нейронной сети — это процесс, при котором происходит подбор оптимальных параметров модели с точки зрения минимизации функционала ошибки.

Для соотношения матрица-наполнитель было принято решение создавать нейронную сеть с помощью Sequential - модели в библиотеке Keras, позволяющей создать нейронную сеть прямого распространения путем последовательного добавления слоев.

Модель состоит из двух скрытых Dense слоев, количество нейронов в которых равно 128 и 64, и выходного слоя с одним нейроном. Функция актив-

вадии слоев – `relu`. Она возвращает 0, если принимает отрицательный аргумент, в случае же положительного аргумента, функция возвращает само число. На выходе использовалась функция активации `tanh`. Её природа нелинейна, она хорошо подходит для комбинации слоёв, а диапазон значений функции - (-1, 1), что позволяет избежать перегрузки от больших значений.

В качестве оптимизатора использовался `adam`. Количество эпох – 50, его оказалось достаточно для выполнения поставленной задачи. Для борьбы с переобучением были добавлены `Dropout`-слои. Добавлен параметр `EarlyStopping`, который позволяет остановить обучение, когда контролируемая метрика перестает улучшаться.

Структура нейронной сети представлена на рисунке 11.

```
[59] model = tf.keras.models.Sequential()
      model.add(layers.Dense(128, input_dim=X_3.shape[1], activation='relu'))
      model.add(layers.Dropout(0.12))
      model.add(layers.Dense(64, activation='relu'))
      model.add(layers.Dropout(0.12))
      model.add(layers.Dense(1))
      model.add(layers.Dense(64, activation='tanh'))
      model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	1664
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65
dense_3 (Dense)	(None, 64)	128
Total params: 10,113		
Trainable params: 10,113		
Non-trainable params: 0		

Рисунок 11 – структура нейронной сети

На рисунке 12 показан процесс обучения модели. На графике видно, что модель не претерпевает существенных изменений после 10 эпохи.



```
def plot_loss(history):
    plt.plot(history.history['loss'], label='loss')
    plt.plot(history.history['val_loss'], label='val_loss')
    plt.ylim([1, 3])
    plt.xlabel('Эпоха')
    plt.ylabel('MAE')
    plt.legend()
    plt.grid(True)
```

```
plot_loss(history)
```

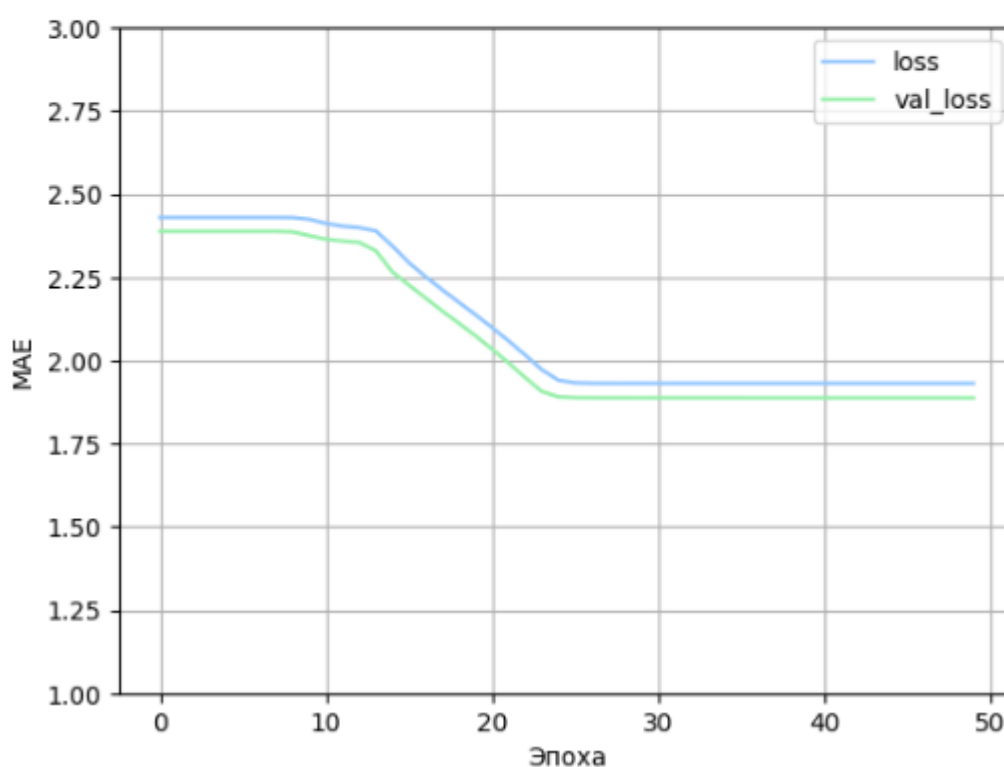


Рисунок 12 – график потерь модели

## 2.5 Разработка приложения

Приложение было разработано с помощью библиотеки `joblib`.

Для прогнозирования значения модуля упругости при растяжении на основании модели линейной регрессии приложение просит поэтапно ввести значения параметров Количество отвердителя, Содержание эпоксидных групп и Потребление смолы, на основании которых выдает итоговое значение. Код приложения представлен на рисунке 13. Пример запуска приложения представлен на рисунке 14.

```
def input_variable():

    x1 = float(input('Введите значение переменной Количество отвердителя, м.:%: '))
    x2 = float(input('Введите значение переменной Содержание эпоксидных групп,%_2: '))
    x3 = float(input('Введите значение переменной Потребление смолы, г/м2: '))

    return x1,x2,x3

def input_proc(X):
    print('Вызов модели')
    res = model_1.predict(X)
    return res

def app_model():
    job_x = load('/content/drive/MyDrive/Colab Notebooks/VKR/X_1.joblib')
    job_y = load('/content/drive/MyDrive/Colab Notebooks/VKR/y_1.joblib')
    model_1 = load('/content/drive/MyDrive/Colab Notebooks/VKR/filename.joblib')
    print('Приложение прогнозирует значения модуля упругости при растяжении')
    for i in range(110):
        try:
            print('введите 1 для прогноза, 2 для выхода')
            check = input()

            if check == '1':
                print('Введите данные')
                X = input_variable()
                X = job_x.transform(np.array(X_1).reshape(1,-1))
                print(['Модуль упругости при растяжении, ГПа'])
                print(job_y.inverse_transform(input_proc(X_1)))

            elif check == '2':
                break
            else:
                print('Повторите выбор')
        except:
            print('Неверные данные. Повторите операцию')
    app_model()
```

Рисунок 13 – код приложения

```
Приложение прогнозирует значения модуля упругости при растяжении
введите 1 для прогноза, 2 для выхода
1
Введите данные
Введите значение переменной Количество отвердителя, м.:%: 0.5
Введите значение переменной Содержание эпоксидных групп,%_2: 0.5
Введите значение переменной Потребление смолы, г/м2: 0.5
Неверные данные. Повторите операцию
введите 1 для прогноза, 2 для выхода

```

Рисунок 14 – пример работы приложения

## 2.6 Создание удаленного репозитория и загрузка результатов работы на него.

Репозиторий с загруженными результатами работы доступен по адресу:  
<https://github.com/Ziyangirova>

## **Заключение**

Данное исследование позволяет сделать основные выводы по теме. Используемые при разработке моделей подходы не позволили получить успешные результаты. Применённые модели регрессии не показали высокой эффективности в прогнозировании свойств композитов. 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 физико-химических свойств материалов учёными. Так как мы не являемся специалистами в области свойств композитных материалов, то можем опираться только на данные, полученные посредством машинного обучения.

## Библиографический список

1. Грас, Джоэл. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2021. - 416 с.
2. Жерон, Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. Пер. с англ. - СПб.: ООО "Альфа-книга": 2018. - 688 с.
3. Джулли, Пал: Библиотека Keras - инструмент глубокого обучения / пер. с англ. А. А. Слинкин.- ДМК Пресс, 2017. – 249 с.
4. Шитиков В.К., Мاستицкий С.Э. Классификация, регрессия и другие алгоритмы Data Mining с использованием R, 2017. 351 с.
5. Документация по библиотеке matplotlib: – Режим доступа: <https://matplotlib.org/stable/users/index.html>.
6. Документация по библиотеке numpy: – Режим доступа: <https://numpy.org/doc/1.22/user/index.html#user>.
7. Документация по библиотеке pandas: – Режим доступа: [https://pandas.pydata.org/docs/user\\_guide/index.html#user-guide](https://pandas.pydata.org/docs/user_guide/index.html#user-guide).
8. Документация по библиотеке scikit-learn: – Режим доступа: [https://scikit-learn.org/stable/user\\_guide.html](https://scikit-learn.org/stable/user_guide.html).
9. Документация по библиотеке seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>.
10. Документация по библиотеке Tensorflow: – Режим доступа: <https://www.tensorflow.org/overview>
11. Документация по языку программирования python: – Режим доступа: <https://docs.python.org/3.8/index.html>.