

**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»

**Прогнозирование физико-механических свойств
композиционных материалов на основе углеродных волокон**

Слушатель

Денисов А.Э.

Москва, 2023

СОДЕРЖАНИЕ

1. Аналитическая часть

- 1.1 Постановка задачи
- 1.2 Описание используемых методов
- 1.3 Разведочный анализ данных

2. Практическая часть

- 2.1 Предобработка данных
- 2.2 Разработка и обучение модели
- 2.3 Тестирование модели
- 2.4 Разработка нейронной сети
- 2.5 Разработка приложения
- 2.6 Создание удаленного репозитория
и загрузка результатов работы на него

1. Аналитическая часть

1.1 Введение

Композиты — это материалы, состоящие минимум из двух компонентов с разными свойствами. На сегодняшний день, исследования их свойств стали особенно актуальны, так как эти материалы находятся в списке государственных приоритетных направлений развития на пути к достижению технологического суверенитета. В ближайшие годы стоит ожидать открытия новых производств малотоннажной химии, например компонентов для эпоксидных связующих, волокон термопластичных полимеров. Наиболее популярные из композитов — полимерные. Они включают в себя армирующий наполнитель (углеродное или стеклянное волокно с определенным плетением и технологией производства) и связующее вещество — например, полиэфирную или эпоксидную смолу. Так, изделия из углеродных полимерных композиционных материалов на эпоксидной смоле по прочности можно сравнить со сталью, но при этом их масса примерно в три раза меньше. За счет способности к самовосстановлению и стойкости к высоким температурам применение композитов возможно во всех отраслях тяжелой промышленности. Композиционные материалы отлично справятся с поставленными задачами как в космосе, так и в судостроении, и в авиации.

1.2 Постановка задачи

В настоящей работе изучаются методы прогнозирования физико-механических свойств композиционных материалов на основе углеродных волокон. Практическая польза от успешной разработки методов прогнозирования заключается в повышении эффективности исследовательской работы за счет сокращения объема рутинных операций.

В работу поступили данные, полученные лабораторией функциональных полимерных материалов Института новых материалов и нанотехнологий Университета науки и технологий МИСИС. Данные представлены результатами лабораторного исследования деформационного поведения при растяжении образцов непрерывных высокопрочных и высокомодульных одиночных углеродных волокон, пропитанных раствором полисульфона.

Имеются экспериментальные данные физико-механических испытаний образцов композиционных материалов на основе углеродных волокон. Эксперименты проводились над образцами с различным соотношением компонентов (полимер/углеродное волокно) при различных скоростях деформации.

Эксперименты проводились на универсальной машине для испытания на растяжение Zwick/Roell Z020 с максимальным приложенным усилием 20 кН, оснащенной высокоточной системой измерения контактной деформации MultiXtens. Изучались прочностные и деформационные характеристики углеродных нитей, пропитанных термопласт-полимером с учетом требований ASTM D4018 и ISO 10618.

Исходные данные: каталог файлов формате MS Excel с результатами механических испытаний образцов материала. В таблицах представлена зависимость Усилие (МПа)/Деформация(мм). Общее число экспериментов – 168, из них в серии 2 - 35 экспериментов, в серии 3 - , в серии 6 - 6, в серии 4 – 30. Число параметров – 3, общее число строк – 93720 (папка RAWdata).

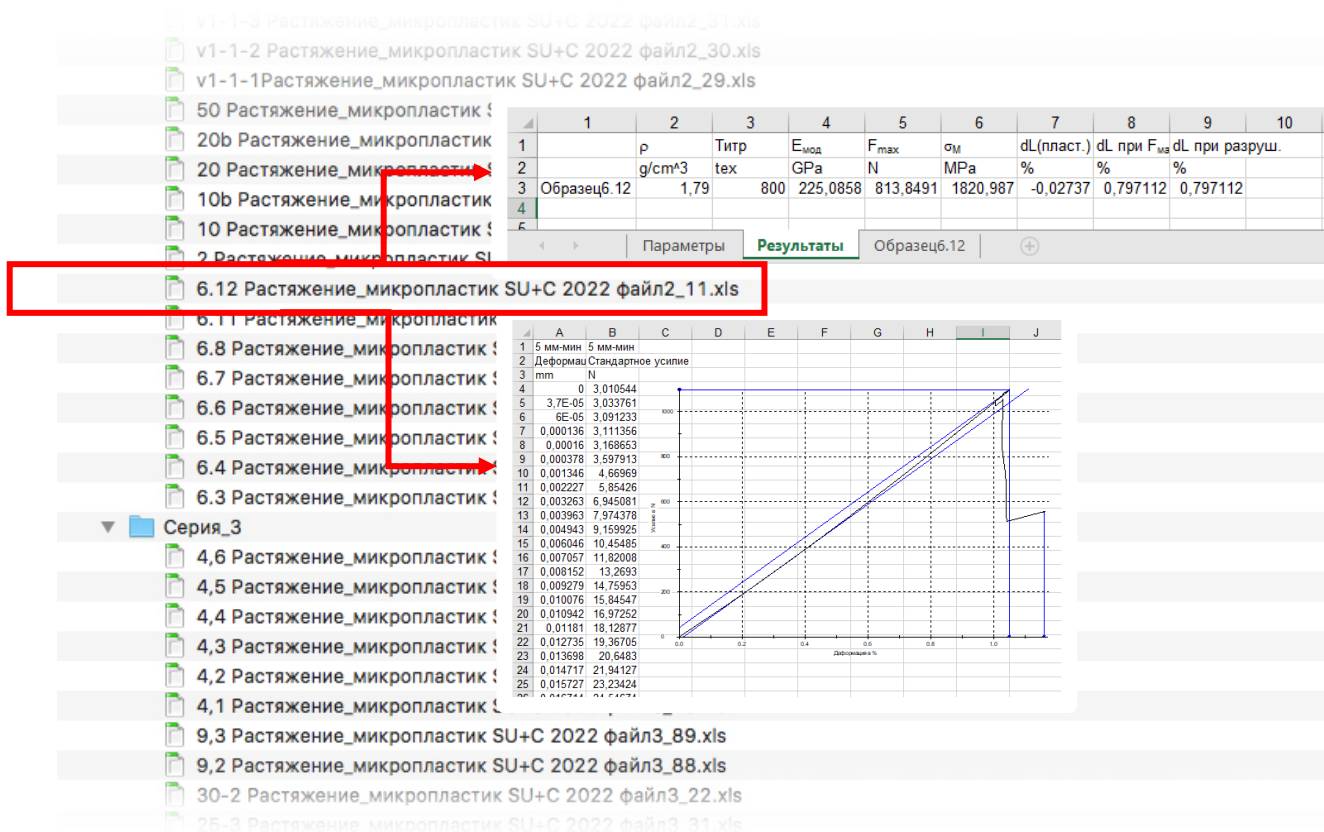


Рисунок 1 – Вид сырых данных и снимки экрана с содержимым одного из файлов xls.

Данные, представленные а таком виде (Рис. 1), требуют предварительной обработки, а именно, работы с файлами и каталогами:

1.3 Предварительная работа с файлами

1.3.1 Унификация названий файлов.

Приведение названий фалов к стандартному виду и нумерации, к латинице с заменой нежелательных символов.

1.3.2 Расчет содержания углерода в образце C_{wt}.

В экспериментах используется углеродное волокно с параметром линейной плотности T=800 текс. Следовательно, 1000m углеродного волокна (100% углерода) имеет массу $800g = 8 \cdot 10^5 \text{ mg}$. Значит, масса образца с содержанием углерода 100% и длиной 1мм составит $a_c = 0.8 \text{ mg/mm}$

Рассчитаем содержание углерода для всех испытанных образцов,
данные внесем в таблицу (Рис. 2):

$$\text{Доля углерода в стержне wt, \%} = \frac{m_{\text{углерода в образце}}}{m_{\text{образца}}} \cdot 100\% = \frac{\rho_L \cdot L}{m_{\text{образца}}} \cdot 100\% \quad (1)$$

ρ_L – линейная плотность 100% углеродного волокна,
 L – длина образца

Таким образом, например, для образца длиной 311мм и массой 295,7mg (Серия 3),

$$\text{получим wt, \%} = \frac{0.8 \cdot 211}{222.56} \cdot 100\% = 75.845$$

Серия 3 (98 экспериментов)	No. of lines	Комментарий, L, mm	Комментарий, m, mg	база, mm	Рабочая длина образца, mm	d ₀ , mm	Wt, %
1 Растяжение_микропластик SU+C 2022 файл3_4.xls	2423	211	222.56	-	100	0.85	

ЭКСПЕРИМЕНТАЛЬНЫЕ ДАННЫЕ							
Деформация, mm X Стандартное усилие, N							
Серия 2 (35 экспериментов)	No. of lines	L x кол-во образцов	Комментарий, L, mm	Комментарий, m, mg	Рабочая длина образца, mm	Плотность, tex	Плотность материала образца,
6.3 Растяжение_микропластик SU+C 2022 файл2_2.xls	144		312	304.73	100	800	1.79
6.4 Растяжение_микропластик SU+C 2022 файл2_3.xls	171		312	304.73	100	800	1.79
6.5 Растяжение_микропластик SU+C 2022 файл2_4.xls	134		312	304.73	100	800	1.79
6.6 Растяжение_микропластик SU+C 2022 файл2_5.xls	173		312	304.73	100	800	1.79
6.7 Растяжение_микропластик SU+C 2022 файл2_6.xls	139		312	304.73	100	800	1.79
6.8 Растяжение_микропластик SU+C 2022 файл2_7.xls	133		312	304.73	100	800	1.79
6.10 Растяжение_микропластик SU+C 2022 файл2_9.xls	127		312	304.73	100	800	1.79
6.11 Растяжение_микропластик SU+C 2022 файл2_10.xls	151		312	304.73	100	800	1.79
6.12 Растяжение_микропластик SU+C 2022 файл2_11.xls	169		312	304.73	100	800	1.79
1 Растяжение_микропластик SU+C 2022 файл2_13.xls	1068	10x260	2600	3326.77	100	800	1.79
2 Растяжение_микропластик SU+C 2022 файл2_15.xls	987	10x260	2600	3326.77	100	800	1.79
5 Растяжение_микропластик SU+C 2022 файл2_16.xls	1018	10x260	2600	3326.77	100	800	1.79
10 Растяжение_микропластик SU+C 2022 файл2_20.xls	762	10x260	2600	3326.77	100	800	1.79
10b Растяжение_микропластик SU+C 2022 файл2_23.xls	969	10x260	2600	3326.77	100	800	1.79
20 Растяжение_микропластик SU+C 2022 файл2_21.xls	384	10x260	2600	3326.77	100	800	1.79
20 Растяжение_микропластик SU+C 2022 файл2_21.xls	969	10x260	2600	3326.77	100	800	1.79
20b Растяжение_микропластик SU+C 2022 файл2_25.xls	308	10x260	2600	3326.77	100	800	1.79
50 Растяжение_микропластик SU+C 2022 файл2_27.xls	164	10x260	2600	3326.77	100	800	1.79
v1-1-1 Растяжение_микропластик SU+C 2022 файл2_29.xls	764	35x248	8680	8658.84	100	800	1.79
v1-1-2 Растяжение_микропластик SU+C 2022 файл2_30.xls	948	35x248	8680	8658.84	100	800	1.79
v1-1-3 Растяжение_микропластик SU+C 2022 файл2_31.xls	677	35x248	8680	8658.84	100	800	1.79
v1-1-4 Растяжение_микропластик SU+C 2022 файл2_32.xls	790	35x248	8680	8658.84	100	800	1.79
v1-1-5 Растяжение_микропластик SU+C 2022 файл2_33.xls	654	35x248	8680	8658.84	100	800	1.79
v1-1-6 Растяжение_микропластик SU+C 2022 файл2_35.xls	743	35x248	8680	8658.84	100	800	1.79
v1-1-7 Растяжение_микропластик SU+C 2022 файл2_38.xls	605	35x248	8680	8658.84	100	800	1.79
v1-1-8 Растяжение_микропластик SU+C 2022 файл2_37.xls	839	35x248	8680	8658.84	100	800	1.79
v1-1-9 Растяжение_микропластик SU+C 2022 файл2_39.xls	737	35x248	8680	8658.84	100	800	1.79
v1-5-1 Растяжение_микропластик SU+C 2022 файл2_40.xls	594	35x248	8680	8658.84	100	800	1.79
v1-5-2 Растяжение_микропластик SU+C 2022 файл2_41.xls	796	35x248	8680	8658.84	100	800	1.79
v1-5-3 Растяжение_микропластик SU+C 2022 файл2_42.xls	785	35x248	8680	8658.84	100	800	1.79
v1-5-4 Растяжение_микропластик SU+C 2022 файл2_44.xls	688	35x248	8680	8658.84	100	800	1.79
v1-5-5 Растяжение_микропластик SU+C 2022 файл2_45.xls	666	35x248	8680	8658.84	100	800	1.79
v1-5-6 Растяжение_микропластик SU+C 2022 файл2_46.xls	617	35x248	8680	8658.84	100	800	1.79
v1-5-7 Растяжение_микропластик SU+C 2022 файл2_47.xls	591	35x248	8680	8658.84	100	800	1.79
v1-5-8 Растяжение_микропластик SU+C 2022 файл2_48.xls	695	35x248	8680	8658.84	100	800	1.79
v1-5-8red Растяжение_микропластик SU+C 2022 файл2_49.xls	591	35x248	8680	8658.84	100	800	1.79
Серия 3 (98 экспериментов)	No. of lines	Комментарий, L, mm	Комментарий, m, mg	база, mm	Рабочая длина образца, mm	d ₀ , mm	Wt, %
1 Растяжение_микропластик SU+C 2022 файл3_4.xls	2423		211	222.56	-	100	0.85
1-2 Растяжение_микропластик SU+C 2022 файл3_7.xls	843		215	222.39	-	100	0.88
1-3 Растяжение_микропластик SU+C 2022 файл3_8.xls	652		208	200.30	-	100	0.83

Рисунок 2 – Вид сводной таблицы после добавления столбца с расчетными данными по поддержанию углерода.

1.3.3 Расчет содержания углерода в образце C_{wt}

1.3.4 Предварительная грубая очистка данных.

Необходимо удалить ту часть данных, которая продолжает поступать с датчика уже после разрушения образца (критерий – снижение напряжения F, рис. 3)

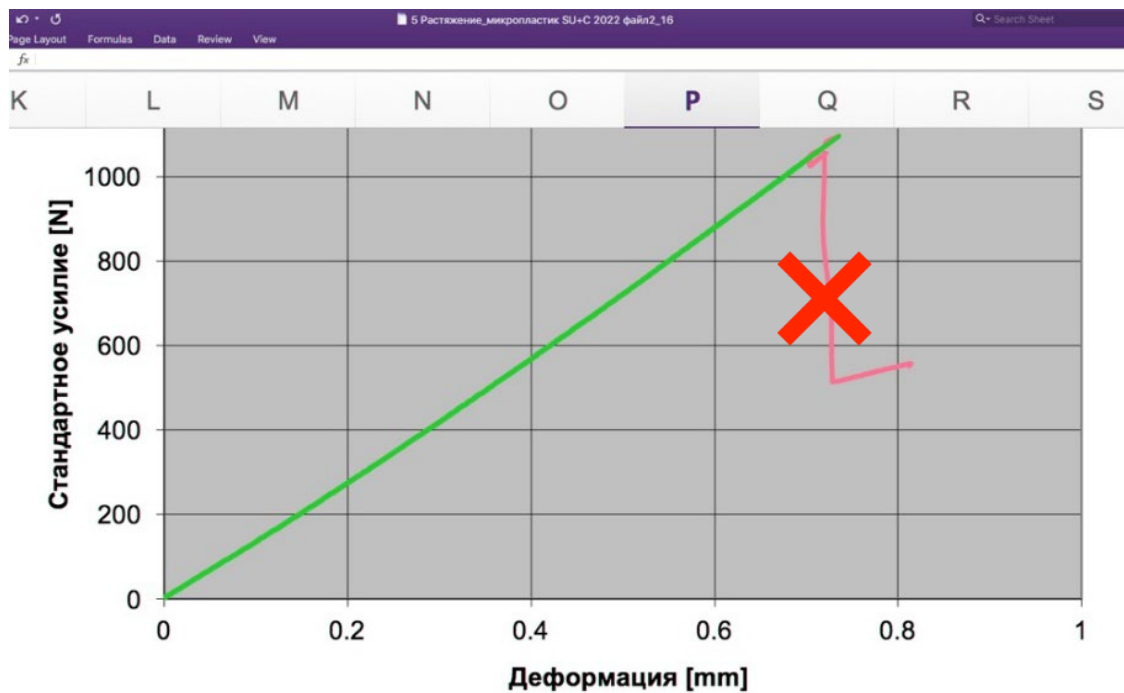


Рисунок 3 – Пример обнаруженных артефакта в данных, подлежащего чистке на начальном этапе обработки.

```
In [5]: df.shape
Out[5]: (1015, 2)

In [6]: # Посмотрим, как выглядит график зависимости деформации от усилия F
sns.lineplot(data=df, x="L", y="F")
Out[6]: <AxesSubplot:xlabel='L', ylabel='F'>
```

The figure is a line plot showing the relationship between Force (F) and Deformation (L). The x-axis is labeled 'L' and ranges from 0.0 to 0.8 with major ticks every 0.1. The y-axis is labeled 'F' and ranges from 0 to 1000 with major ticks every 200. A blue line represents the data, starting at (0.0, 0) and increasing linearly to approximately (0.7, 1000). At this point, the line drops sharply to approximately (0.75, 500), where a large red 'X' is drawn. The line then continues horizontally to the right edge of the plot area at x=0.8.

```
In [7]: # Необходимо исключить из рассмотрения данные, поступившие после разрушения образца,
# начиная с момента, когда F внезапно уменьшается
# Найдём разницу между каждой предыдущей строкой в колонке F
df['F_diff'] = df['F'].diff ()

In [8]: # Выведем номера строк, начиная с момента, когда прекращается рост F
df.query('F_diff < 0').index
Out[8]: Int64Index([ 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890,
891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901,
929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939,
940, 1013, 1014],
dtype='int64')
```

```

In [9]: # Введем переменную 'a' - индекс первой строки датафрейма, где F перестает возрастать
# (это соответствует моменту разрушения образца, либо изменению условий эксперимента.
# Данные, полученные после этого момента, исключаем из датасета)
a = df.query('F_diff < 0').index[0]
a

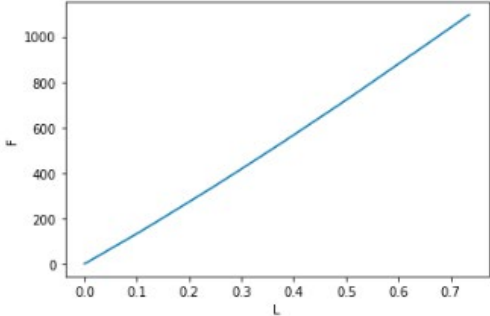
Out[9]: 880

In [10]: # Удалим лишние строки
df=df.drop(df.index[a:])

In [11]: # Контроль - что получилось
sns.lineplot(data=df, x="L", y="F")

Out[11]: <AxesSubplot:xlabel='L', ylabel='F'>

```



Дополнение малого датасета расчетным значением C_wt

```

In [12]: #Дополняем данные расчетным значением содержания углерода в процентах C_wt.
# Для данного эксперимента C_wt=65.523 %
df = df.assign(C_wt='62.523')

In [13]: # Удалим столбец F_diff с промежуточными вычислениями.
df = df.drop(['F_diff'], axis=1)

In [51]: # Объединим результаты пяти экспериментов в один датафрейм
df = pd.concat(
    map(pd.read_csv, ['5_2_16.csv',
                     '6_10_2_9.csv',
                     '4_1_4_18.csv',
                     '3_5_6_3.csv',
                     'v1-5-1_2_40.csv']))

df.head(5)

Out[51]:
   L      F  C_wt
0  0.000000  3.010544  62.523
1  0.000037  3.033761  62.523
2  0.000060  3.091233  62.523
3  0.000136  3.111356  62.523
4  0.000160  3.168653  62.523

In [52]: df.shape

Out[52]: (3217, 3)

In [53]: # Сохраним полученный датафрейм
df.to_csv('df1_5experiments.csv', index=False)

```

Рисунок 4 – Алгоритм обработки данных, позволяющий избавиться от артефактов в полуавтоматическом режиме. Реализация на Python.


```

In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

In [2]: # загружаем данные из файла xls,
# берем данные из третьей вкладки, пропуская первые 3 строки, берем первые 2 столбца
df = pd.read_excel(r"C:\Users\piken\anaconda3\DS\LOADtest\5_2__16.xls",
                  sheet_name = 2,
                  skiprows=2,
                  usecols='A:B')

In [15]: # Сохраним полученный датафрейм в формате CSV без индекса
df.to_csv('5_2__16.csv', index=False)

```

Рисунок 5 – Алгоритм обработки данных, позволяющий избавиться от артефактов в полуавтоматическом режиме. Результат работы алгоритма.

Просмотренные и подготовленные данные записываем в формате **csv**. Реализация на Python, а также результат работы алгоритма представлен на рис. 4-5.

1.3.5 Расчет относительного удлинения (%) dL

Обращаемся к закладке ‘Результаты’ каждого исходного файла *.xls (Рис. 1) и определяем отношение L к длине образца.

$$dL=L*1.42857 \text{ или } dL=L*2 \quad (2)$$

для образцов с длиной рабочей части 70 и 50мм соответственно.

1.3.6 Добавляем колонку с расчетным значением НАПРЯЖЕНИЯ (МПа) Stress

Площадь сечения образца s определяется отношением σ_M к F_{max}

$$\text{Stress} = F * s \quad (3)$$

Расчетные значения для экспериментов внесены в таблицу

Data summary1503.xls (колонки К, М, О) и будут добавлены в подготавливаемый для работы большой датафрейм.

После отсеивания обнаруженных сбойных файлов и файлов, практически дублирующих друг друга, общее число предобработанных документов (Папка **CSV_preprocessed**) – 117шт,

Процесс предобработки данных представлен в ноутбуке

Q_Primary_Processing1503.ipynb

Полученные предобработанные данные методом конкатенации объединяем в единый большой датафрейм **Full_Stress_df.csv**

1.4 Описание используемых методов.

Для решения поставленной задачи предполагается использовать следующие методы.

Линейная регрессия – метод машинного обучения, который позволяет определить линейное уравнение, лучше всего описывающее корреляцию между зависимой (целевой) и независимыми переменными. Коэффициенты линейной модели минимизируют остаточную сумму квадратов между наблюдаемыми целями в датасете и целевой переменной, предсказанной линейным приближением.

ElasticNet – это модель линейной регрессии, обученная как с регуляризацией коэффициентов L1 и L2. Данная комбинация позволяет наиболее эффективно изучать разреженную модель, в которой лишь немногие веса отличны от нуля, как у Лассо, сохраняя при этом свойства регуляризации Ridge.

Lasso (Least absolute shrinkage and selection operator) – это вариация линейной регрессии, которая оценивает коэффициенты линейной регрессионной модели. Это метод выполняет как выбор переменных, так и регуляризацию с целью повышения точности прогнозирования и интерпретируемости результирующей статистической модели.

KNeighborsRegressor – метод решения задач классификации и задач регрессии, основанный на поиске ближайших объектов с известными значениями целевой переменной. Значения целевой переменной прогнозируются путем локальной интерполяции целей, связанных с ближайшими соседями в обучающем наборе. В данном методе реализовано обучение на основе k ближайших соседей каждой точки запроса, где k – целочисленное значение, указанное пользователем.

Random Forest regressor – метод случайного леса – алгоритм машинного обучения, использующий ансамбли решающих деревьев. Его можно применять как для классификации, так и для регрессии. Он создает деревья решений для случайно выбранных семплов данных, получает прогноз от каждого дерева и выбирает наилучшее решение посредством голосования.

1.5 Разведочный анализ данных

Этап разведочного анализа (EDA, Exploratory Data Analysis) играет важнейшую роль после получения набора данных и ставит своей целью обобщение основных характеристик и описание структуры данных с использованием статистической графики, и, как следствие, лучшее понимание данных. Загружаем объединенный датафрейм **Full_Stress_df.csv** и приступаем к разведочному ана-

лизу данных. Анализ будем проводить на языке Python, используя инструмент Jupyter Notebook.

Загружаем данные в файле **DataStress_28MAR.ipynb**

Будем использовать **pandas** – основной стандарт исследования данных, **matplotlib** – пакет для визуализации, **seaborn** – библиотеку для производства продвинутых графиков и диаграмм.

Планируем исследовать признаки отдельно и совместно, построить соответствующие гистограммы, парные диаграммы распределения признаков, **Boxplot** для всех параметров датафрейма, рассмотрим матрицу корреляции и тепловую карту.

Тип данных - числа с плавающей запятой, двойной точности (**float64**).
Размерность массива - (**49622, 3**). Пропуски отсутствуют. Описательная статистика и размерность данных представлена на Рис. 6.

	C_wt	dL	Stress
count	49622.000000	49622.000000	49622.000000
mean	77.791706	0.381332	729.508951
std	6.351242	0.249812	547.576213
min	62.523000	-0.000009	0.160228
25%	76.555000	0.178455	302.526312
50%	80.195000	0.359614	631.638535
75%	81.405000	0.539306	1009.816534
max	85.743000	1.203548	2645.988464

Рисунок 6 – Описательная статистика и размерность исходных данных после грубой очистки.

1.5.1 Гистограммы

На рисунке 7 представлены столбчатые диаграммы для напряжения (МПа), концентрации углерода (%) и относительной деформации образца (%). Они отражают распределение частоты значений физических величин.

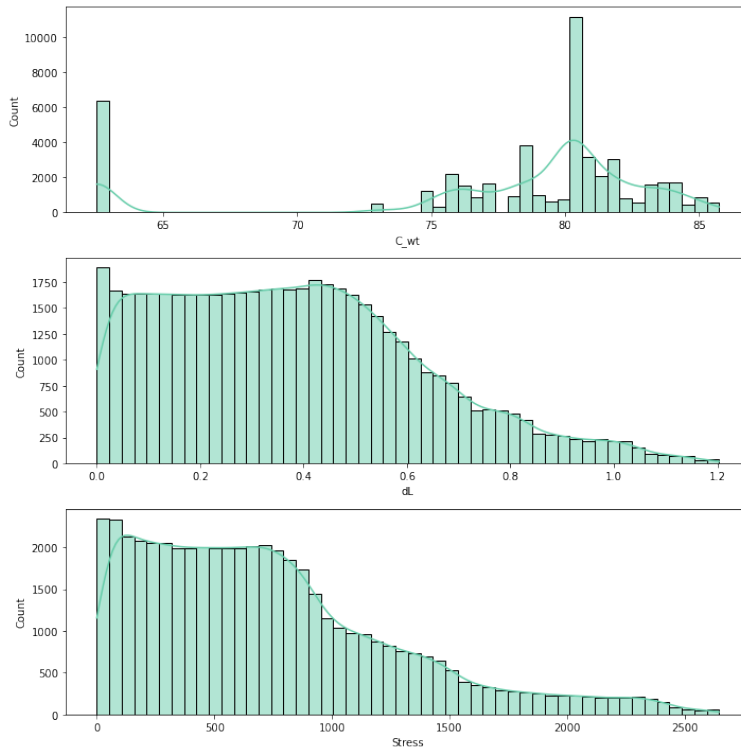


Рисунок 7 – Гистограммы признаков для напряжения, концентрации углерода и относительной деформации.

1.5.2 Матрица корреляции и тепловая карта

Рассмотрим матрицу корреляции и тепловую карту.

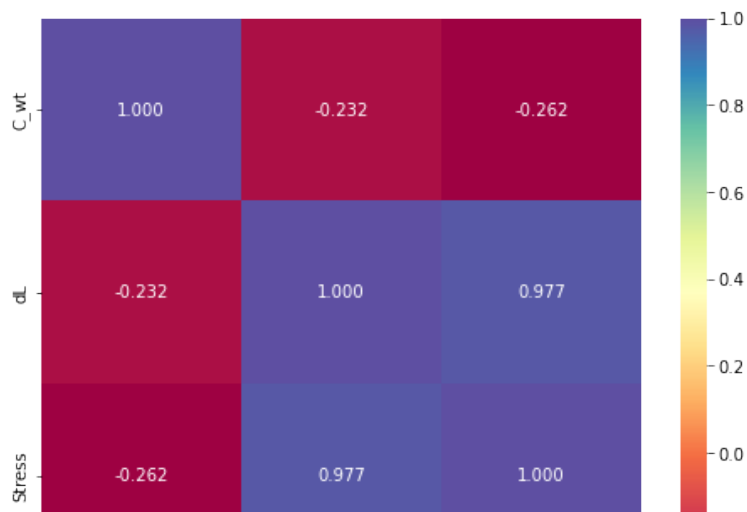


Рисунок 8 – Тепловая карта.

1.5.3 Парные диаграммы распределения признаков

Для проверки гипотезы о наличии корреляционной связи между двумя признаками построим парные диаграммы распределения признаков

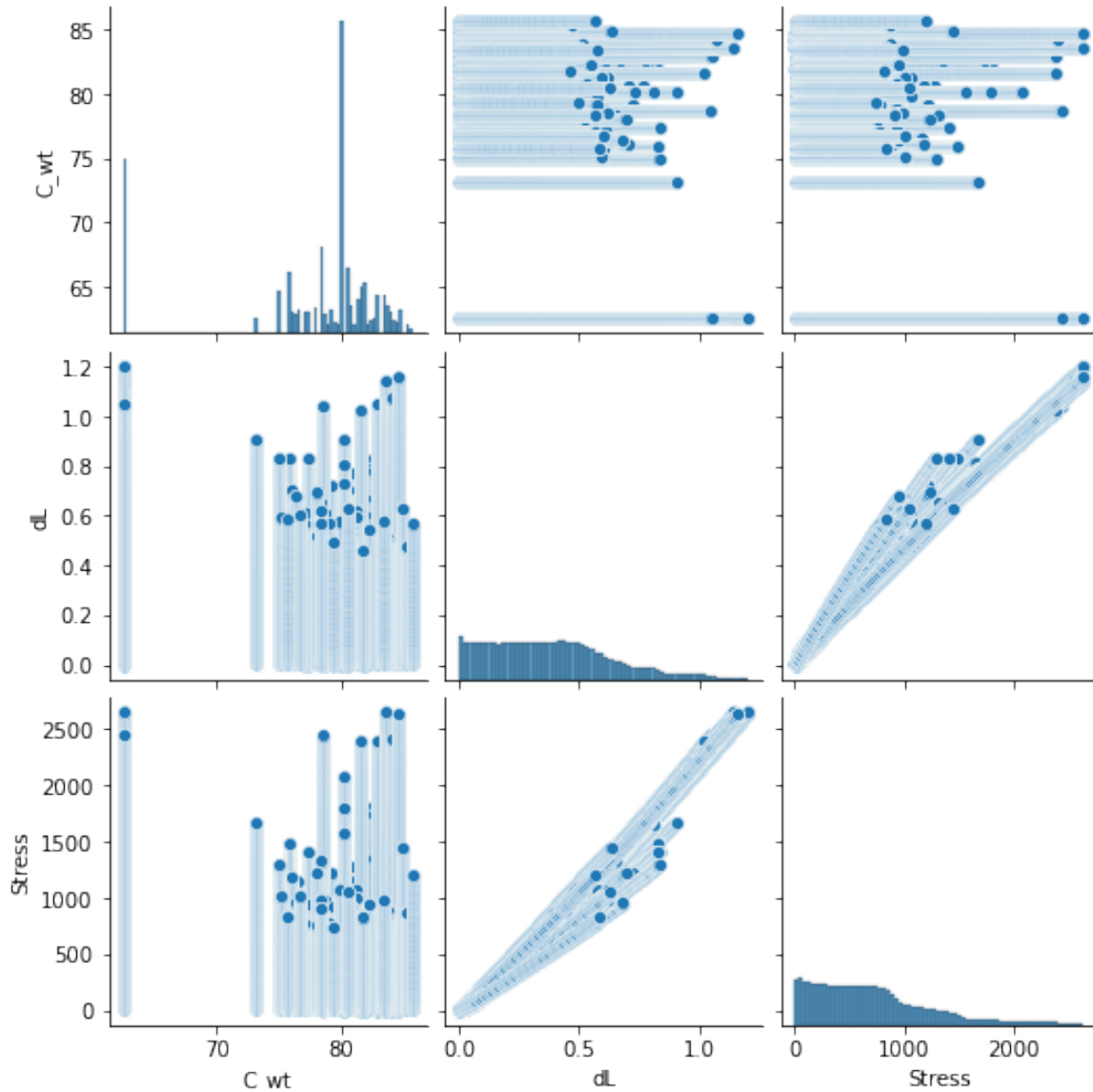


Рисунок 9 – Парные диаграммы распределения всех признаков.

1.5.4 Ящик с усами (Boxplot)

Данный вид диаграммы компактно изображает одномерное распределение вероятностей и позволяет визуально оценить медиану, нижний и верхний квартили, минимальное и максимальное значение выборки и выбросы. Для фильтрации выбросов воспользуемся методом межквартильных расстояний. Результат представлен на Рис. 10.

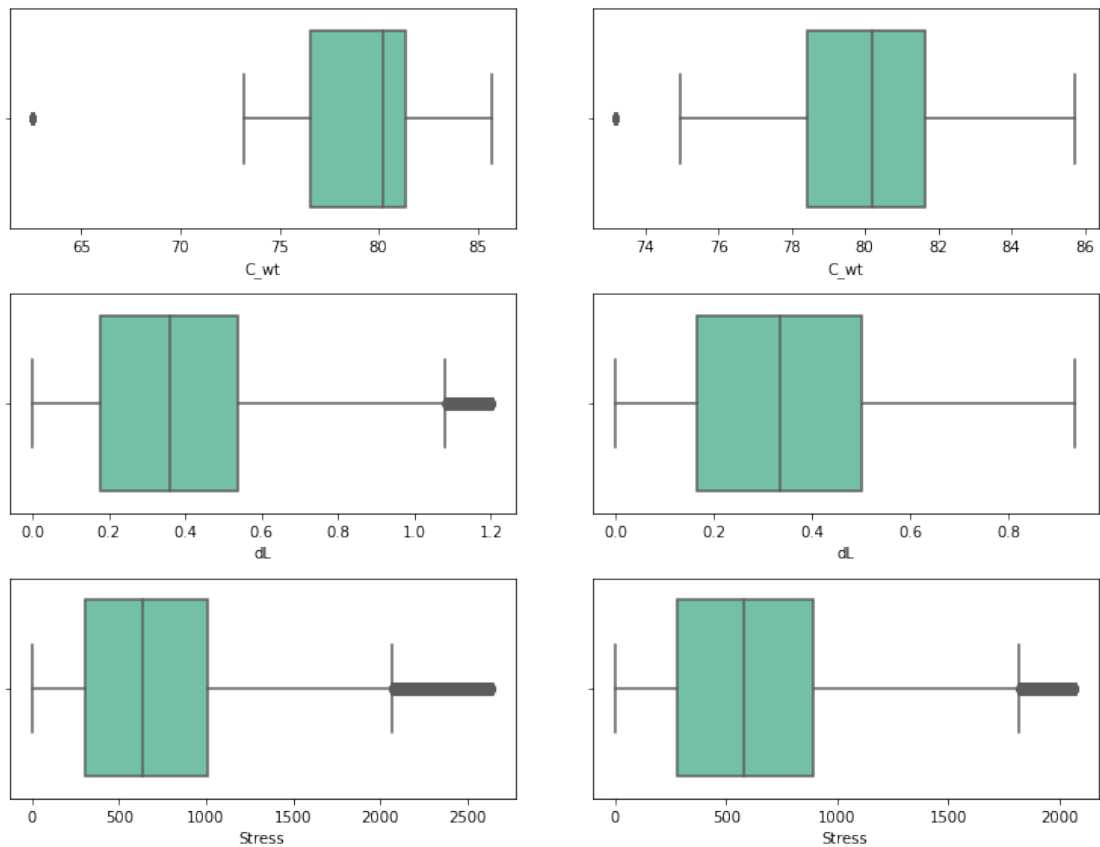


Рисунок 10 – Boxplot до и после фильтрации выбросов.

1.5.5 Плотность ядра

Оценка плотности ядра (KDE) – это непараметрический метод оценки плотности случайной величины. Ядерная оценка плотности является задачей сглаживания данных, когда делается заключение о совокупности, основываясь на конечных выборках данных.

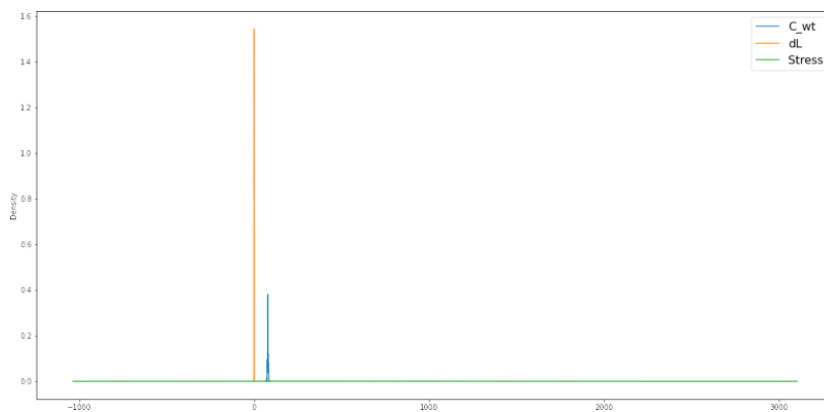


Рисунок 11 – Оценка ядерной плотности.

2. Практическая часть

2.1 Предобработка данных

Для эффективного моделирования необходимо провести нормализацию – процедуру предобработки входной информации (обучающих, тестовых и валидационных выборок, а также реальных данных), при которой значения признаков во входном векторе приводятся к некоторому заданному диапазону. Воспользуемся PowerTransformer - семейством параметрических монотонных преобразований, которые применяются для придания данным большего гауссовского сходства.

2.1.1 Гистограмма признаков до и после нормализации

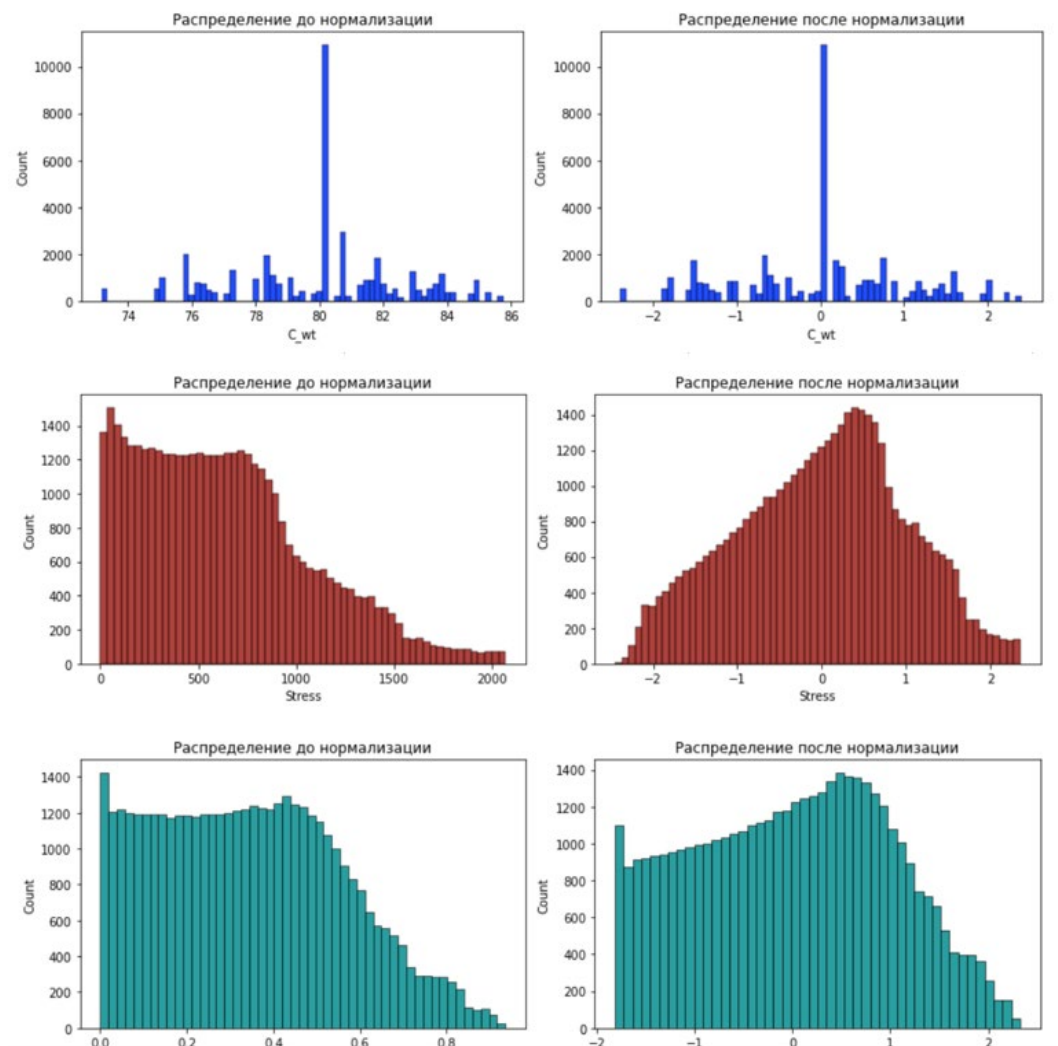


Рисунок 12 – Гистограммы признаков до и после нормализации.

2.1.2 Вохplot до и после нормализации

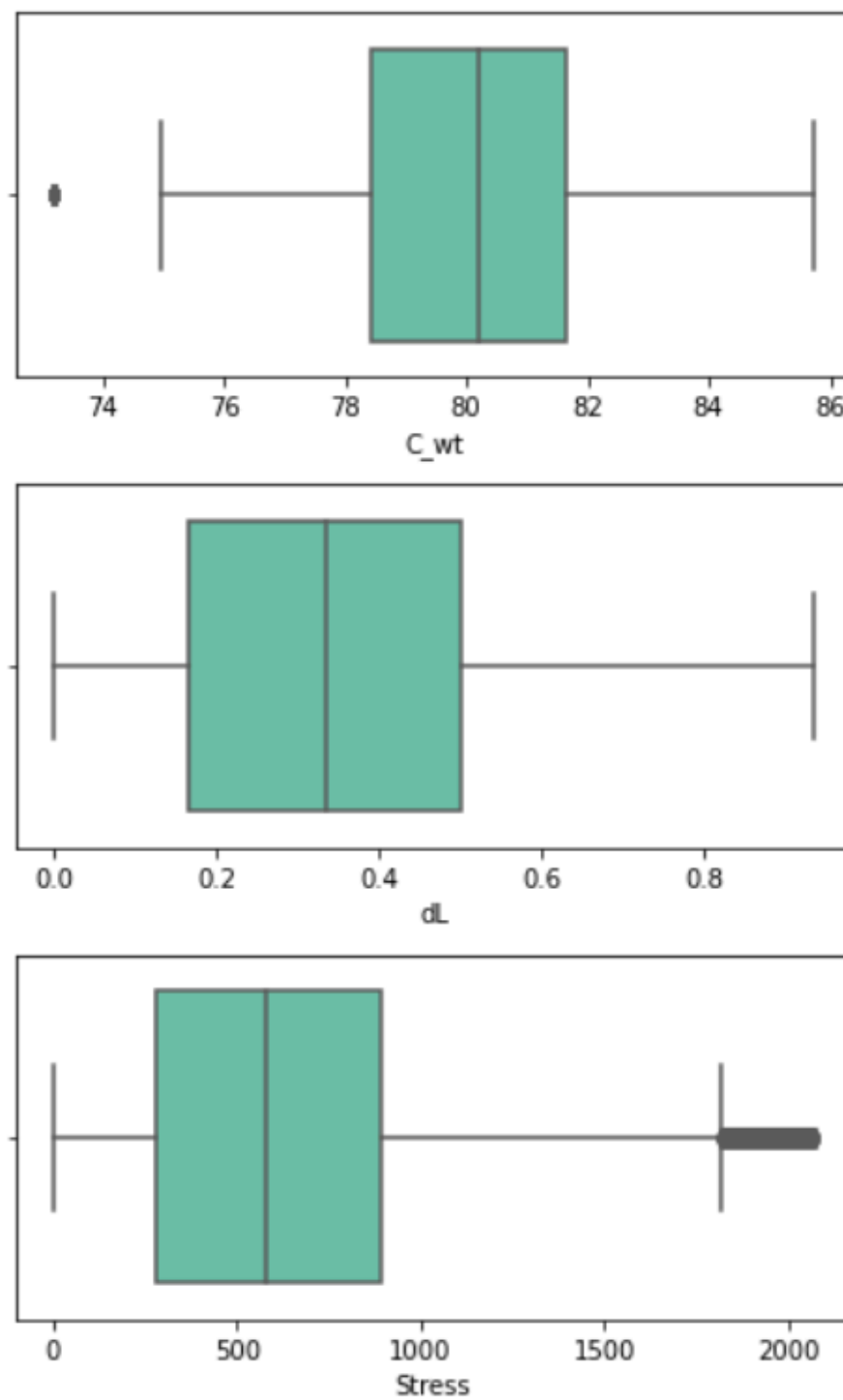


Рисунок 13 – Вохplot до нормализации.

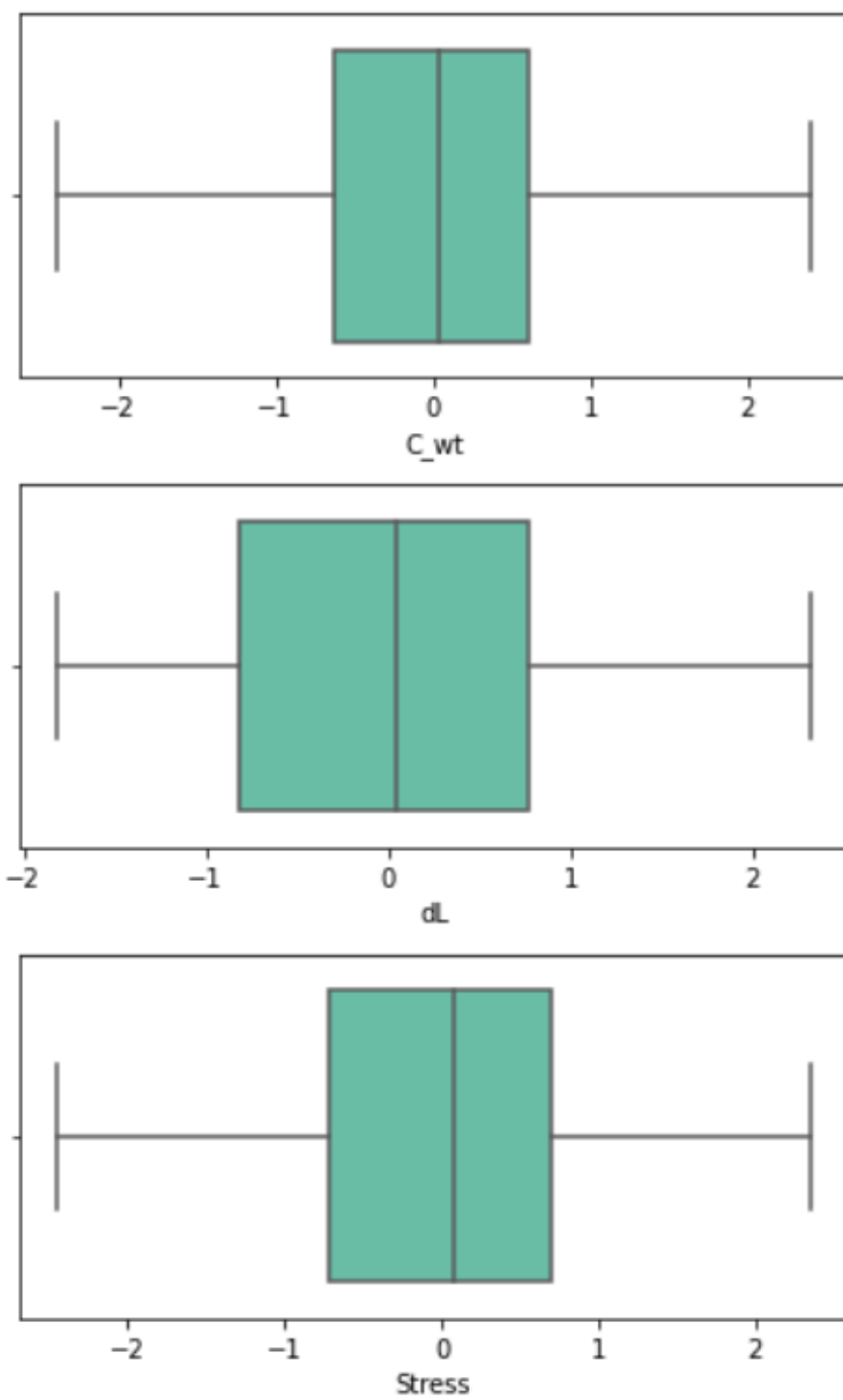


Рисунок 14 – Boxplot после нормализации

2.1.3 Ядерная плотность до и после нормализации

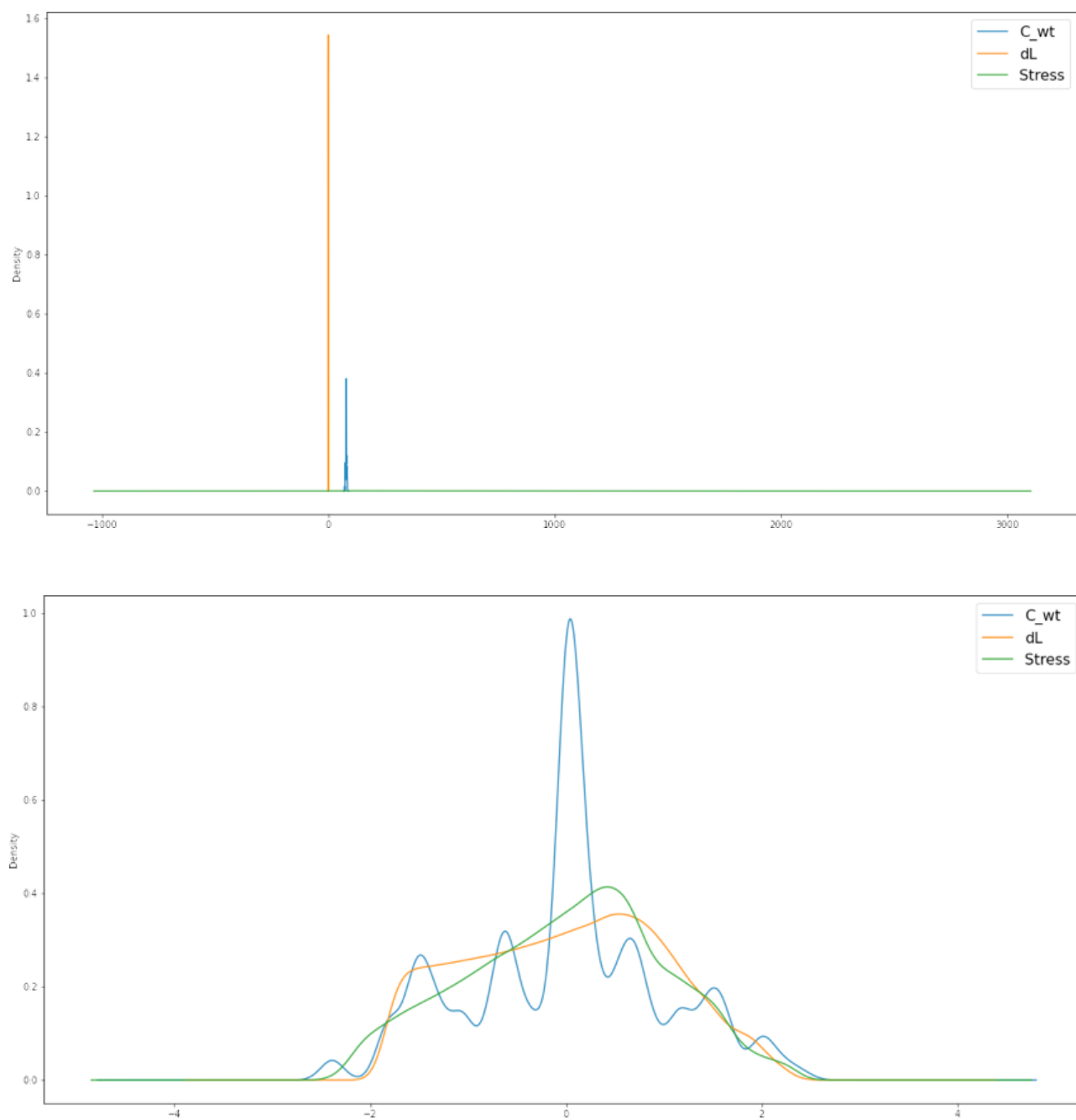


Рисунок 15 – Ядерная плотность до и после нормализации.

2.1.4 Probability-график (график вероятности)

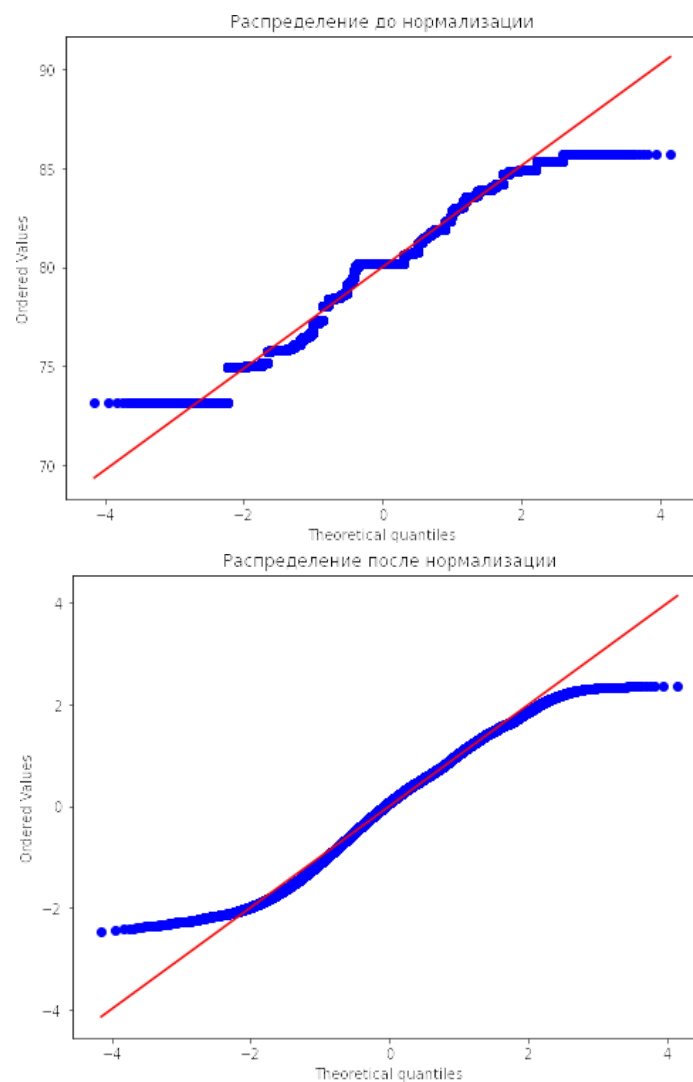


Рисунок 16 – Probability-график (график вероятности) до и после нормализации.

2.1.5 Визуализация переменных в виде столбчатой гистограммы до и после нормализации

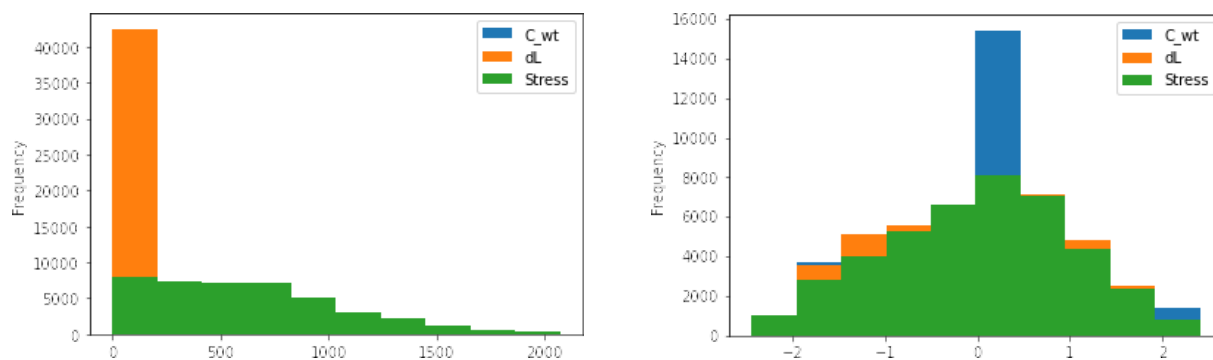


Рисунок 17 – Визуализация переменных в виде столбчатой гистограммы до и после нормализации.

2.1.6 Тепловая карта после нормализации

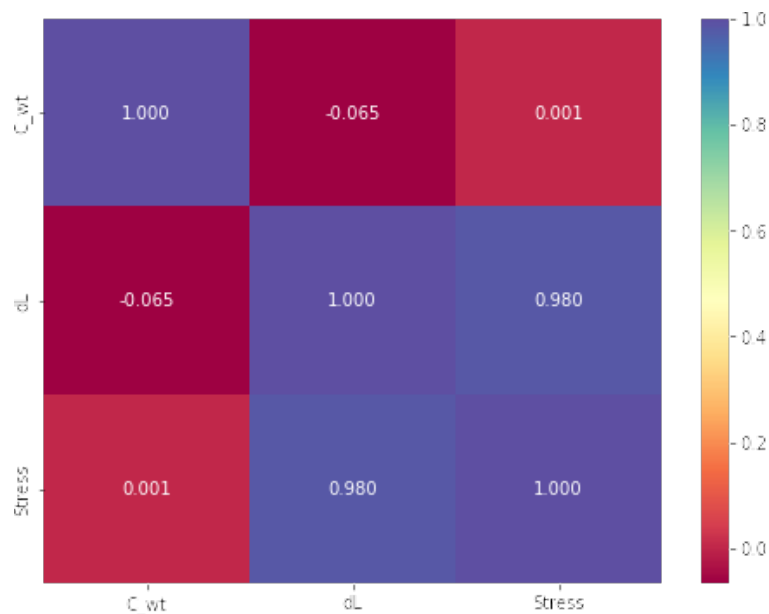


Рисунок 17 – Матрица корреляции и тепловая карта после нормализации.

Результат – подготовленные и нормализованные данные запишем в файл **dfnorm.csv**, с которым будем работать далее. Выведем первые строки дата-фрейма и описательную статистику нормализованных данных.

Размерность - (42485, 3).

	C_wt	dL	Stress
count	4.248500e+04	4.248500e+04	4.248500e+04
mean	1.425611e-14	6.722278e-16	3.385547e-16
std	1.000012e+00	1.000012e+00	1.000012e+00
min	-2.389737e+00	-1.814995e+00	-2.449682e+00
25%	-6.304612e-01	-8.204129e-01	-7.205830e-01
50%	3.221095e-02	4.534098e-02	7.788915e-02
75%	6.063542e-01	7.707815e-01	6.977880e-01
max	2.407360e+00	2.324179e+00	2.354996e+00

	C_wt	dL	Stress
0	-1.026045	-1.814900	-2.247307
1	-1.026045	-1.814860	-2.247055
2	-1.026045	-1.814207	-2.239318
3	-1.026045	-1.811071	-2.227549
4	-1.026045	-1.805147	-2.210352

Рисунок 18 – Вид и описательная статистика нормализованных данных.

2.2 Разработка и обучение моделей

Для прогнозирования значения напряжения при растяжении обучим несколько моделей (объем тестовой выборки 30%, обучающей – 70%). Рассчитаем ошибку MAE, RMSE, R2 и определим модель, лидирующую по точности предсказания. Выполним разработку следующих моделей:

- Модель линейной регрессии
- Модель ElasticNet - линейная регрессия с регуляризацией
- Модель Lasso - метод оценки коэффициентов
- Модель регрессии К ближайших соседей
- Модель регрессии случайного леса
- Нейронная сеть, многослойный персептрон

2.3 Тестирование моделей

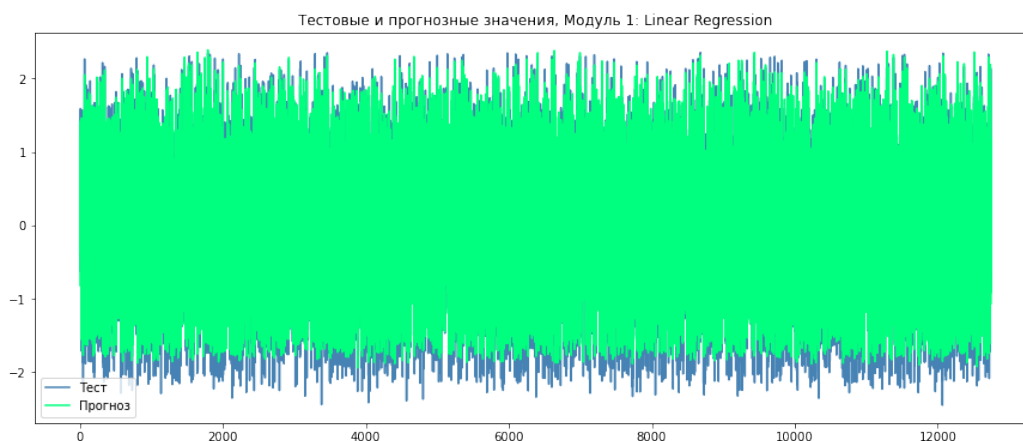


Рисунок 19 – Результат работы модели линейной регрессии, сравнение прогноза и теста.

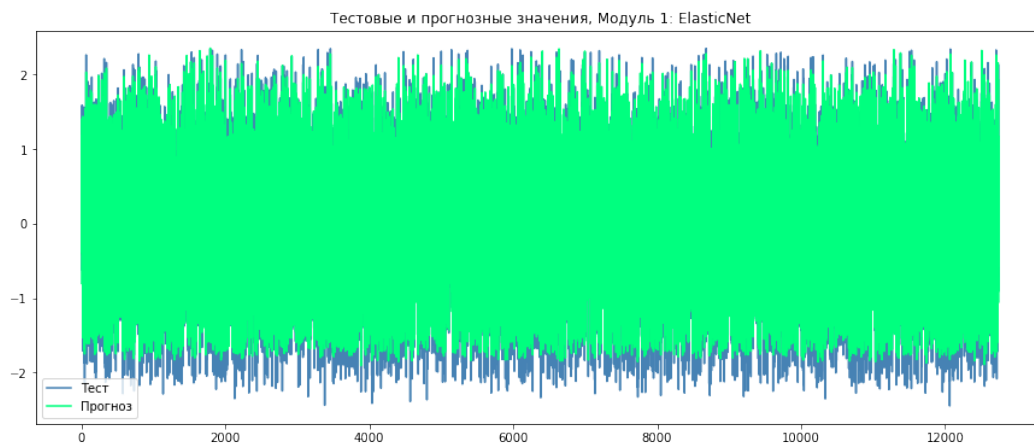


Рисунок 20 – Результат работы модели ElasticNet, сравнение прогноза и теста.

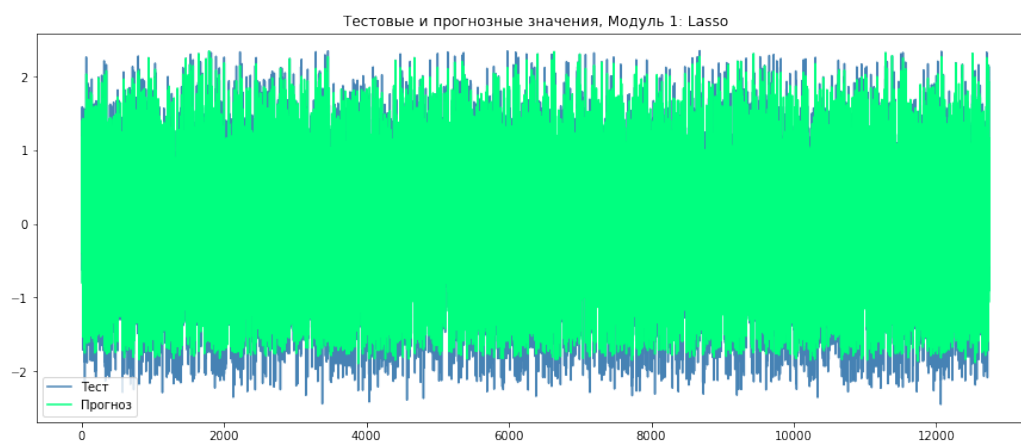
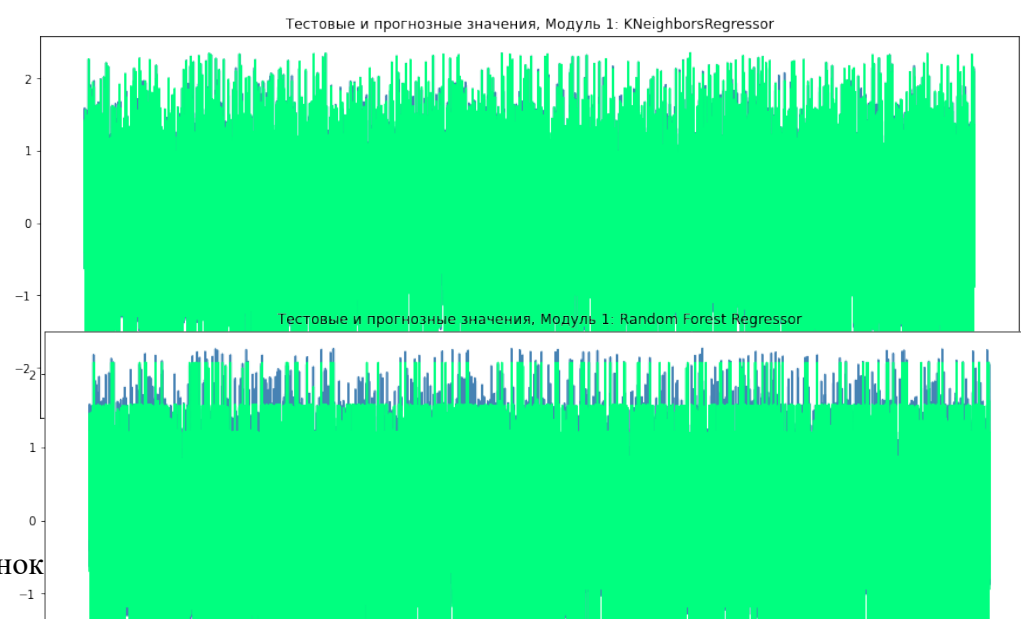


Рисунок 21 – Результат работы модели Lasso, сравнение прогноза и теста.



Рисунок

Рисунок 23 – Результат работы модели Random Forest Regressor, сравнение прогноза и теста.

Для оценки эффективности сравним метрики протестированных моделей. Лучшие результаты показал метод регрессии К ближайших соседей.

	model	target	MAE	RSME	R2
0	Linear Regression	Stress	0.155514	0.036080	0.963962
1	ElasticNet	Stress	0.155909	0.036211	0.963831
2	Lasso	Stress	0.156013	0.036300	0.963743
3	KNeighborsRegressor	Stress	0.042737	0.005516	0.994491
4	Random Forest Regressor	Stress	0.145526	0.033752	0.966288

Рисунок 24 – Сравнение точности работы протестированных моделей

2.4 Создание нейронной сети

Для создания нейронной сети применяем Tensorflow.Keras – открытую нейросетевую надстройку над фреймворками, нацеленную на оперативную работу с сетями глубокого обучения. Общая структура сети разрабатывается с помощью объекта модели keras Sequential(), который создает последовательную модель с пошаговым добавлением слоев в нее. Dense слой является базовым. Он отвечает за соединение нейронов из предыдущего и следующего слоя и у него есть настраиваемые гиперпараметры: количество нейронов, тип активации, инициализация типа ядра. Dropout Layer помогает избавиться от переобучения модели. Таким образом, некоторые нейроны становятся равными 0, и это сокращает вычисления в процессе обучения.

Первый слой – входной, Dense, с 50 нейронами, затем добавляются два скрытых слоя по 128 и 19 нейронов со слоями Dropout между ними и выходной слой с одним нейроном. В первом и втором слое применяется активационная функция relu, в третьем – гиперболический тангенс. Количество эпох – 60.

```
from keras import layers
from keras import models
def plot_loss(history):
    plt.plot(history.history['loss'], label='loss')
    plt.plot(history.history['val_loss'], label='val_loss')
    plt.ylim([0, 1])
    plt.xlabel('Эпоха')

    plt.ylabel('MAE [MPG]')
    plt.legend()
    plt.grid(True)
model = Sequential()
model.add(layers.Dense(50, input_dim=X.shape[1], activation='relu', name="layer1"))
model.add(layers.Dropout(0.12))
model.add(layers.Dense(128, activation='relu', name="layer2"))
model.add(layers.Dropout(0.12))
model.add(layers.Dense(19, activation='tanh', name="layer3"))
model.add(layers.Dense(1))
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
layer1 (Dense)	(None, 50)	150
dropout (Dropout)	(None, 50)	0
layer2 (Dense)	(None, 128)	6528
dropout_1 (Dropout)	(None, 128)	0
layer3 (Dense)	(None, 19)	2451
dense (Dense)	(None, 1)	20
=====		
Total params: 9,149		
Trainable params: 9,149		

Рисунок 25 – Структура нейронной сети TF02.

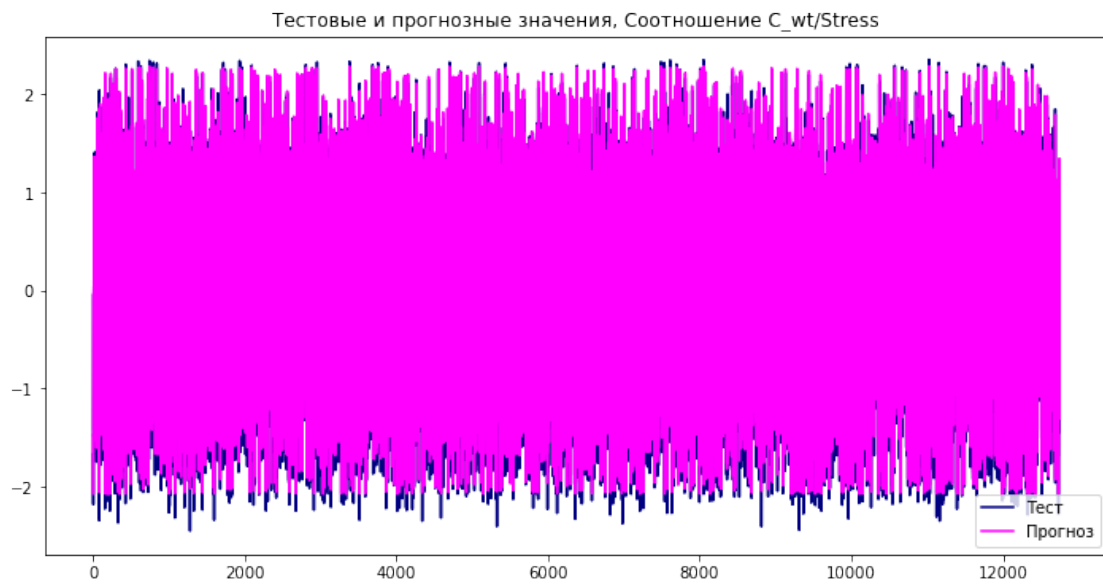


Рисунок 26 – Результат работы нейронной сети, сравнение прогноза и теста.

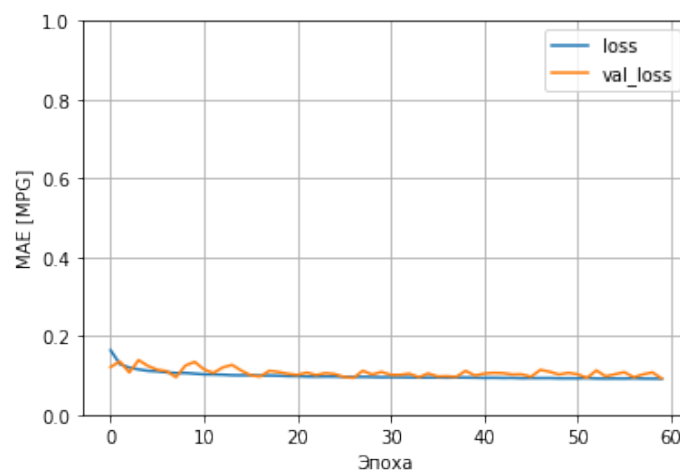


Рисунок 27 – График потерь.

	model	target	MAE	RSME	R2
0	Linear Regression	Stress	0.155514	0.036080	0.963962
1	ElasticNet	Stress	0.155909	0.036211	0.963831
2	Lasso	Stress	0.156013	0.036300	0.963743
3	KNeighborsRegressor	Stress	0.042737	0.005516	0.994491
4	Random Forest Regressor	Stress	0.145526	0.033752	0.966288
5	Нейросеть TF02	Stress	0.092696	0.017186	0.982698
6	Нейросеть TF0	Stress	0.085091	0.014932	0.984968

Рисунок 28 – Сравнение точности работы нейронной сети и протестированных моделей.

Для снижения ошибки следует оптимизировать параметры нейронной сети: output shape 1=50, output shape 2=128, output shape3=22. Dropout=0.05. Точность работы оптимизированной нейронной сети представлена на рисунке 28, позиция 6.

2.5 Разработка приложения

Консольная версия выдаёт прогноз значения напряжения (Stress, МПа) в зависимости от введённых с клавиатуры C_{wt} и dL:

```
Данное приложение прогнозирует значение напряжения (Stress)
Для прогноза введите 1, для выхода - 2
1
Введите данные:
Введите концентрацию углерода (Cwt), %: 81
Введите относительное удлинение (dL), %: 0.6
['Прогноз значения напряжения (Stress), МПа']
Вызов модели KNN.pkl
[[1053.72349874]]
```

Рисунок 29 – Результат работы консольного приложения. Код: App_Stress_xX.ipynb

Для создания приложения, предсказывающего значение напряжения в зависимости от введенных параметров концентрации углерода и относительного удлинения, создавалось с помощью инструмента **Flask** в Google Colaboratory и сервиса, предоставляющего возможность создания публичного адреса **ngrok**. Шаблон страницы представлен файлом в формате html.

```
!pip install flask-ngrok
!pip install pyngrok

import pandas as pd
import numpy as np
import copy
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, mean_absolute_percentage_error, r2_score
from sklearn.preprocessing import MinMaxScaler, Normalizer, StandardScaler
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import LinearRegression, LogisticRegression, SGDRegressor, Lasso
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import r2_score
from sklearn.multioutput import MultiOutputRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.preprocessing import PowerTransformer
import tensorflow as tf
from tensorflow import keras
from keras import layers
from keras import models
from keras.engine.sequential import Sequential
from keras.metrics import Precision, Recall
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn import metrics
from sklearn.metrics import mean_absolute_error
import pickle
```

```
[ ] !pip install pyngrok
```

```
ngrok.set_auth_token(auth)
public_url = ngrok.connect(5000).public_url
app = flask.Flask(__name__, template_folder = 'templates')

@app.route('/', methods = ['POST', 'GET'])
@app.route('/index', methods = ['POST', 'GET'])
def main():
    if flask.request.method == 'GET':
        return render_template('main.html')

    if flask.request.method == 'POST':
        with open('KNN_model.pkl', 'r') as f:
            loaded_model = pickle.load(f)

            exp = float(flask.request.form['Stress'])
            y_pred = loaded_model.predict([[exp]])
            # x1 = powertransformer_y_1.inverse_transform(np.array([0.34,0.77]).reshape(1,-1))
            # x1[0][0], x1[0][1]

            return render_template('main.html', result = y_pred)

print(f" Приложение по ссылке : {public_url}")
app.run(port = port_no)
```

```
[ ] from pyngrok import ngrok
from flask import Flask
import flask
import pickle
from flask import render_template
from joblib import load
import sklearn
from sklearn.linear_model import LinearRegression

# ключ для авторизации
auth = '2NewxdGSMpJeEdf94oPHTJbOhsj_4arPvNU1LaLmGvgsc9jjiu'

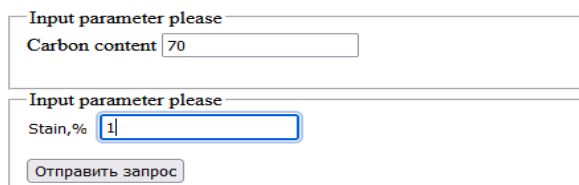
port_no = 5000
```

Рисунок 30 – Код разработки приложения в Google Colaboratory.

Приложение доступно по ссылке :

<http://8f07-34-141-210-133.ngrok.io>

Результат работы на данный момент:



2.6 Создание удаленного репозитория и загрузка на него результатов исследовательской работы

Код исследования размещён в созданном на GitHub репозитории:
https://github.com/AlexLivadius/VKR_Stress

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Композитный мир: чем уникальны и перспективны новые материалы. – Режим доступа:
<https://trends.rbc.ru/trends/industry/6380c0219a79474f0f5b2a35>
2. Stepashkin, A.A., Mohammad H., Makarova E.D., Odintsova Y.V., Laptev A.I., Tcherdyntsev V.V.. Deformation Behavior of Single Carbon Fibers Impregnated with Polysulfone by Polymer Solution Method. *Polymers* 2023, 15, 570.
3. Mohammad H., Stepashkin A.A., Laptev A.I., Tcherdyntsev V.V. Mechanical and Conductive Behavior of Graphite Filled Polysulfone-Based Composites. *Appl. Sci.* 2023, 13, 542.
4. Под ред. Васильева В.В., Тариопольского Ю.М. Композиционные материалы. Справочник – М.: Машиностроение, 1990
5. Худсон Д. Статистика для физиков – М.: Мир, 1970
6. Альтхоф К. #Сам себе программист — М.: Бомбора, 2022
7. Бурдуковский В. Г., Механические свойства металлов и модели разрушения — Екатеринбург: Издательство Уральского университета, 2020
8. Грас, Джоэл. Data Science. Наука о данных с нуля - СПб.: БХВ-Петербург, 2021
9. Николенко С., Кадуринов А., Архангельская Е., Глубокое обучение. Погружение в мир нейронных сетей. - СПб.: Питер. – 2020

