

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»

ЦЫГАНКОВ ПАВЕЛ ИГОРЕВИЧ

Начало работы

- ▶ Постановка задачи:
- ▶ Для исследовательской работы были даны 2 файла: с данными о параметрах базальтопластика и данными нашивок углепластика.

Загружаем исходные данные из обеих excel таблиц и удаляем колонку с индексом

```
#Загружаем первый датасет (базальтопластик) и посмотрим на названия столбцов
df_bp = pd.read_excel(r"C:\Users\Пользователь\Desktop\ВКР\X_bp.xlsx")
df_bp.shape
```

(1023, 11)

```
#Удаляем первый неинформативный столбец
df_bp.drop(['Unnamed: 0'], axis=1, inplace=True)
df_bp.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
0	1.857143	2030.0	738.736842	30.00	22.267857	100.000000	210.0	70.0	3000.0	220.0
1	1.857143	2030.0	738.736842	50.00	23.750000	284.615385	210.0	70.0	3000.0	220.0
2	1.857143	2030.0	738.736842	49.90	33.000000	284.615385	210.0	70.0	3000.0	220.0
3	1.857143	2030.0	738.736842	129.00	21.250000	300.000000	210.0	70.0	3000.0	220.0
4	2.771331	2030.0	753.000000	111.86	22.267857	284.615385	210.0	70.0	3000.0	220.0

```
#Удаляем первый неинформативный столбец
df_nup.drop(['Unnamed: 0'], axis=1, inplace=True)
df_nup.head()
```

	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0	0	4.0	57.0
1	0	4.0	60.0
2	0	4.0	70.0
3	0	5.0	47.0
4	0	5.0	57.0

```
# Проверим размерность второго файла
df_nup.shape
```

(1040, 3)

Цель - разработать модели для прогноза модуля упругости при растяжении, прочности при растяжении и соотношения «матрица-наполнитель». Для этого нужно объединить 2 файла. Часть информации (17 строк таблицы способов компоновки композитов) не имеют соответствующих строк в таблице соотношений и свойств используемых компонентов композитов, поэтому были удалены.

Объединим по индексу, тип объединения INNER, смотрим итоговый датасет

```
# По условию задачи объединяем их по типу INNER.  
df = df_bp.merge(df_nup, left_index = True, right_index = True, how = 'inner')  
df.head().T
```

	0	1	2	3	4
Соотношение матрица-наполнитель	1.857143	1.857143	1.857143	1.857143	2.771331
Плотность, кг/м3	2030.000000	2030.000000	2030.000000	2030.000000	2030.000000
модуль упругости, ГПа	738.736842	738.736842	738.736842	738.736842	753.000000
Количество отвердителя, м.%	30.000000	50.000000	49.900000	129.000000	111.860000
Содержание эпоксидных групп,%_2	22.267857	23.750000	33.000000	21.250000	22.267857
Температура вспышки, С_2	100.000000	284.615385	284.615385	300.000000	284.615385
Поверхностная плотность, г/м2	210.000000	210.000000	210.000000	210.000000	210.000000
Модуль упругости при растяжении, ГПа	70.000000	70.000000	70.000000	70.000000	70.000000
Прочность при растяжении, МПа	3000.000000	3000.000000	3000.000000	3000.000000	3000.000000
Потребление смолы, г/м2	220.000000	220.000000	220.000000	220.000000	220.000000
Угол нашивки, град	0.000000	0.000000	0.000000	0.000000	0.000000
Шаг нашивки	4.000000	4.000000	4.000000	5.000000	5.000000
Плотность нашивки	57.000000	60.000000	70.000000	47.000000	57.000000

Посмотрим количество колонок и столбцов, просмотрим информацию о датасете, проверим тип данных в каждом столбце (типы признаков), все переменные, кроме угла нашивки содержат значения float64, угол нашивки же имеет тип int64, качественные характеристики отсутствуют. Пропусков не имеется. Ни одна из записей не является NaN, очистка не требуется. Объединенный файл имеет всего 1023 строки. Видим в основном общее число уникальных значений в каждом столбце, но в столбце "Угол нашивки" всего 2 значения. Можем привести данные в этой колонке к значениям 0 и 1.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1023 entries, 0 to 1022
Data columns (total 13 columns):
 #   Column                                     Non-Null Count  Dtype
---  -
 0   Соотношение матрица-наполнитель         1023 non-null   float64
 1   Плотность, кг/м3                         1023 non-null   float64
 2   модуль упругости, ГПа                    1023 non-null   float64
 3   Количество отвердителя, м.%              1023 non-null   float64
 4   Содержание эпоксидных групп,%_2         1023 non-null   float64
 5   Температура вспышки, С_2                 1023 non-null   float64
 6   Поверхностная плотность, г/м2           1023 non-null   float64
 7   Модуль упругости при растяжении, ГПа    1023 non-null   float64
 8   Прочность при растяжении, МПа           1023 non-null   float64
 9   Потребление смолы, г/м2                  1023 non-null   float64
10   Угол нашивки, град                       1023 non-null   int64
11   Шаг нашивки                             1023 non-null   float64
12   Плотность нашивки                        1023 non-null   float64
dtypes: float64(12), int64(1)
memory usage: 111.9 KB
```

```
Соотношение матрица-наполнитель    1014
Плотность, кг/м3                    1013
модуль упругости, ГПа                1020
Количество отвердителя, м.%         1005
Содержание эпоксидных групп,%_2     1004
Температура вспышки, С_2            1003
Поверхностная плотность, г/м2       1004
Модуль упругости при растяжении, ГПа 1004
Прочность при растяжении, МПа        1004
Потребление смолы, г/м2              1003
Угол нашивки, град                   2
Шаг нашивки                          989
Плотность нашивки                    988
dtype: int64
```

Поработаем со столбцом "Угол нашивки"

Изучим описательную статистику наших данных (максимальное, минимальное, квартили, медиана, стандартное отклонение, среднее значение и т.д.), посмотрим на основные параметры анализа данных. Описательная статистика содержит по каждому столбцу (по каждой переменной): count - количество значений, mean - среднее значение, std - стандартное отклонение, min - минимум 25% - верхнее значение первого квартиля, 50% - медиана, 75% - верхнее значение третьего квартиля, max - максимум

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660	5.591742
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375	2207.773481
модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	500.047452	739.664328	961.812526	1911.536477
Количество отвердителя, м.%	1023.0	110.570769	28.295911	17.740275	92.443497	110.564840	129.730366	198.953207
Содержание эпоксидных групп,%_2	1023.0	22.244390	2.406301	14.254985	20.608034	22.230744	23.961934	33.000000
Температура вспышки, С_2	1023.0	285.882151	40.943260	100.000000	259.066528	285.896812	313.002106	413.273418
Поверхностная плотность, г/м2	1023.0	482.731833	281.314690	0.603740	266.816645	451.864365	693.225017	1399.542362
Модуль упругости при растяжении, ГПа	1023.0	73.328571	3.118983	64.054061	71.245018	73.268805	75.356612	82.682051
Прочность при растяжении, МПа	1023.0	2466.922843	485.628006	1036.856605	2135.850448	2459.524526	2767.193119	3848.436732
Потребление смолы, г/м2	1023.0	218.423144	59.735931	33.803026	179.627520	219.198882	257.481724	414.590628
Угол нашивки	1023.0	0.491691	0.500175	0.000000	0.000000	0.000000	1.000000	1.000000
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

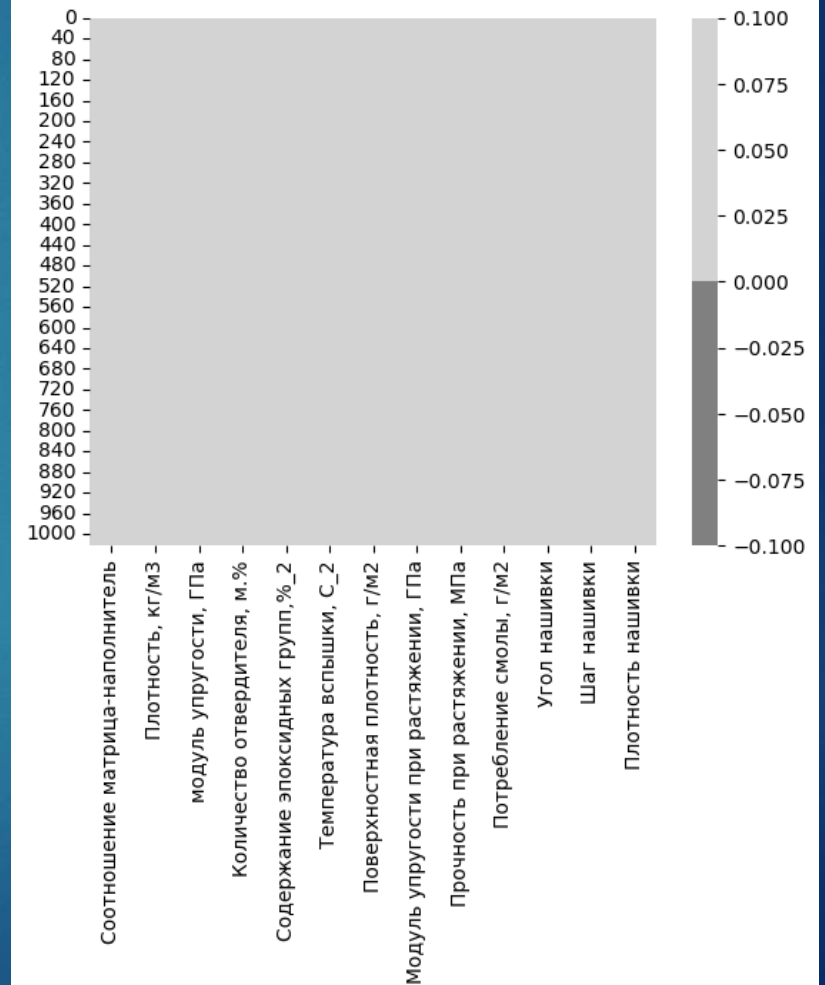
Проверим датасет на пропущенные и продублированные данные, для наглядности построим «тепловую карту» распределения пропусков.

```
# Проверим на пропущенные данные
df.isnull().sum()
# Пропущенных данных нет = нулевых значений нет, очистка не требуется
```

Соотношение матрица-наполнитель	0
Плотность, кг/м3	0
модуль упругости, ГПа	0
Количество отвердителя, м.%	0
Содержание эпоксидных групп,%_2	0
Температура вспышки, C_2	0
Поверхностная плотность, г/м2	0
Модуль упругости при растяжении, ГПа	0
Прочность при растяжении, МПа	0
Потребление смолы, г/м2	0
Угол нашивки	0
Шаг нашивки	0
Плотность нашивки	0

```
dtype: int64
```

```
#светлый - не пропущенные, темный - пропущенные данные
cols = df.columns
colours = ['gray', 'lightgray']
sns.heatmap(df[cols].isnull(), cmap = sns.color_palette(colours))
#Тепловая карта, так же как info() и функция ISNULL() показывает, что пропусков нет.
```

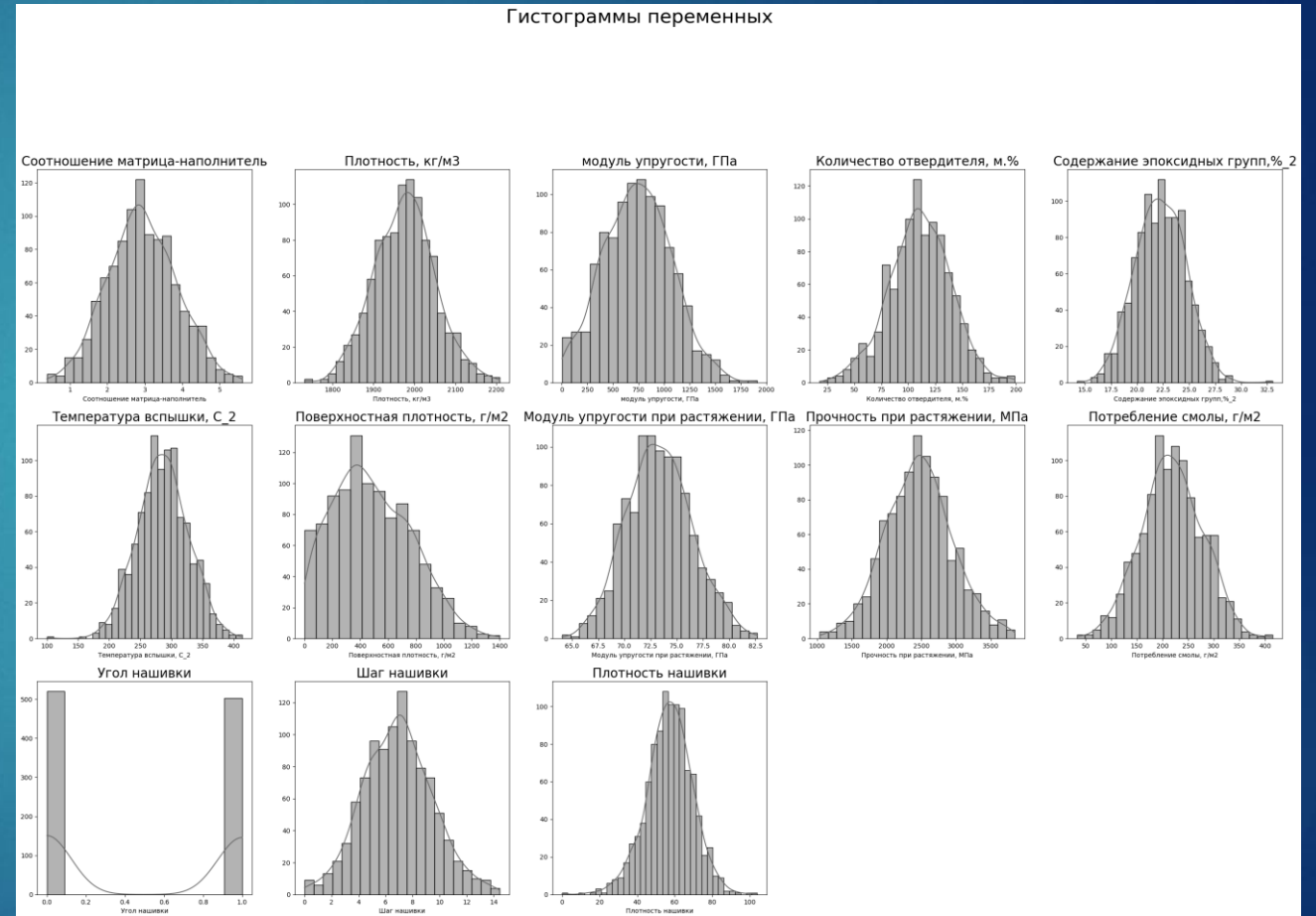
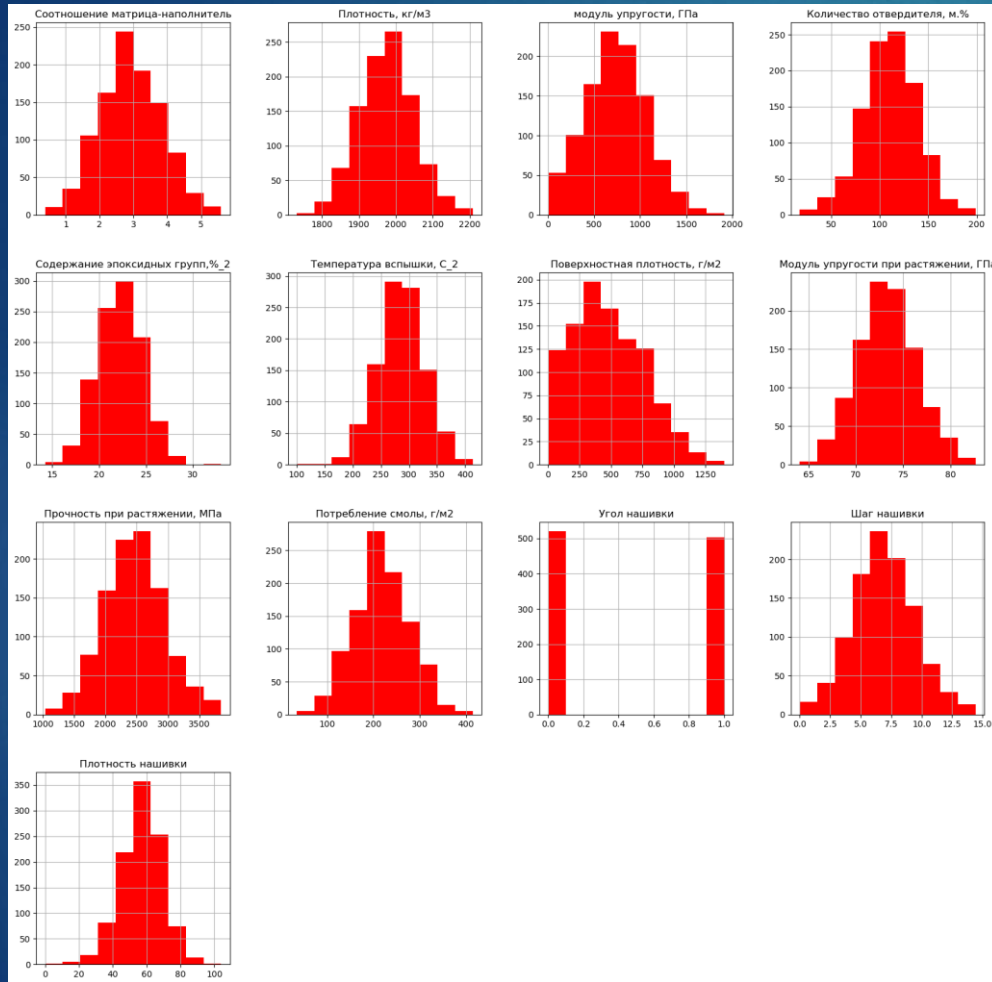


Определим наличие попарной корреляции столбцов. Вычисляем коэффициенты ранговой корреляции Кендалла, Пирсона, Спирмана. Видим что коэффициенты корреляции там крайне малы.

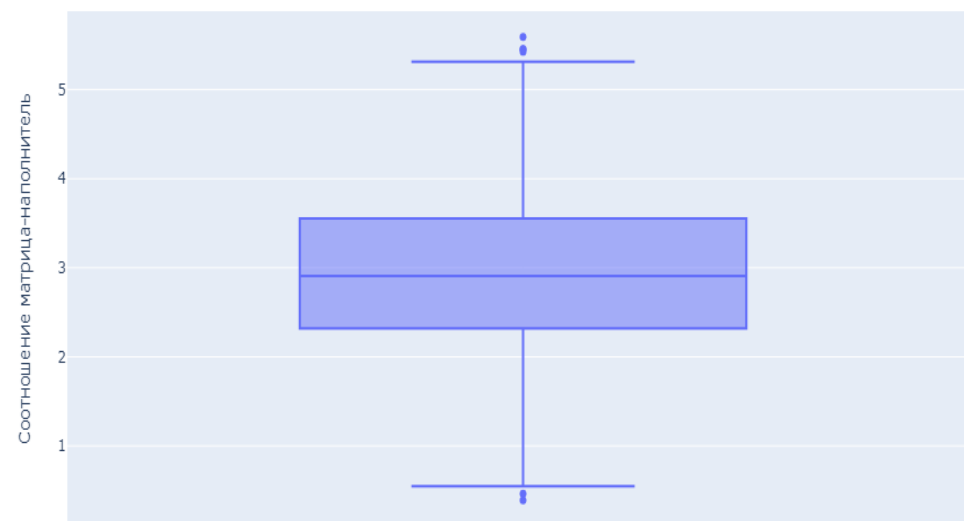
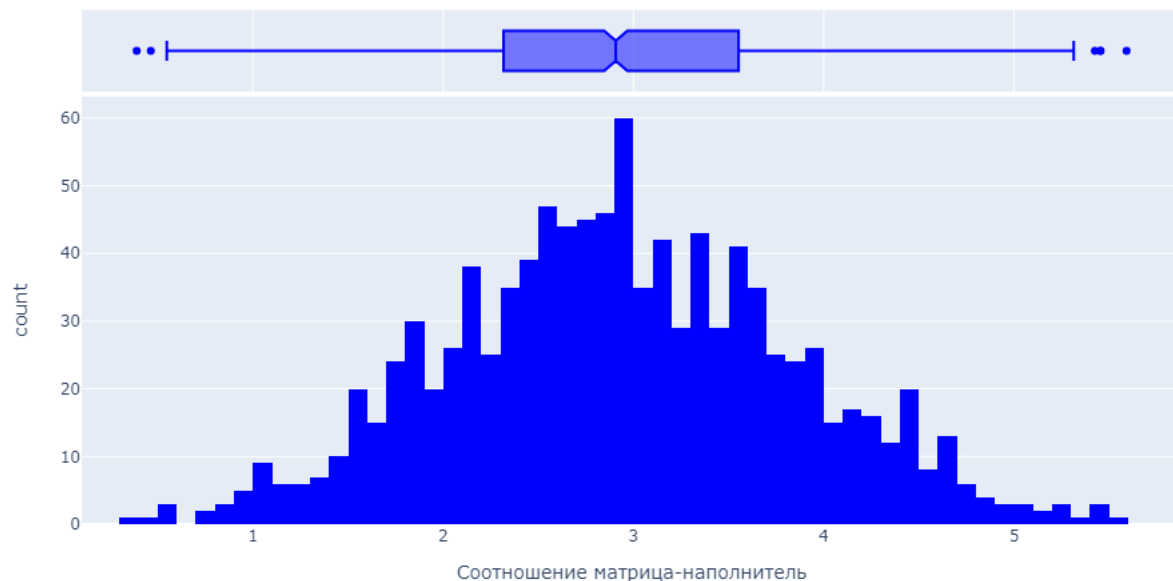
	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/ м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2
Соотношение матрица-наполнитель	1.000000	-0.003135	0.021247	0.001410	0.010180	-0.009480	-0.002060	-0.004157	0.011614	0.035145
Плотность, кг/ м3	-0.003135	1.000000	-0.008059	-0.021963	-0.007758	-0.019947	0.037302	-0.021151	-0.047426	-0.017079
модуль упругости, ГПа	0.021247	-0.008059	1.000000	0.022382	0.002351	0.021028	-0.000442	0.005458	0.022959	0.005169
Количество отвердителя, м.%	0.001410	-0.021963	0.022382	1.000000	0.000010	0.059034	0.033110	-0.043140	-0.046507	-0.003677
Содержание эпоксидных групп,%_2	0.010180	-0.007758	0.002351	0.000010	1.000000	-0.002170	-0.006859	0.041994	-0.013441	0.009756
Температура вспышки, С_2	-0.009480	-0.019947	0.021028	0.059034	-0.002170	1.000000	0.017196	0.016481	-0.019106	0.035313
Поверхностная плотность, г/ м2	-0.002060	0.037302	-0.000442	0.033110	-0.006859	0.017196	1.000000	0.024051	-0.005099	-0.004446
Модуль упругости при растяжении, ГПа	-0.004157	-0.021151	0.005458	-0.043140	0.041994	0.016481	0.024051	1.000000	-0.006599	0.034814
Прочность при растяжении, МПа	0.011614	-0.047426	0.022959	-0.046507	-0.013441	-0.019106	-0.005099	-0.006599	1.000000	0.013580
Потребление смолы, г/м2	0.035145	-0.017079	0.005169	-0.003677	0.009756	0.035313	-0.004446	0.034814	0.013580	1.000000
Угол нашивки	-0.021395	-0.051525	-0.031695	0.024690	0.004668	0.017880	0.045452	0.022431	0.020609	-0.002402
Шаг нашивки	0.022723	-0.031220	-0.008305	0.006232	-0.004539	0.029552	0.025514	-0.010024	-0.048049	0.005962
Плотность нашивки	0.002788	0.052935	0.049347	0.016607	-0.021968	0.005268	-0.022320	-0.002600	0.009821	0.010792

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/ м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2
Соотношение матрица-наполнитель	1.000000	0.003841	0.031700	-0.006445	0.019766	-0.004776	-0.006272	-0.008411	0.024148	0.072531
Плотность, кг/ м3	0.003841	1.000000	-0.009647	-0.035911	-0.008278	-0.020695	0.044930	-0.017602	-0.069981	-0.015937
модуль упругости, ГПа	0.031700	-0.009647	1.000000	0.024049	-0.006804	0.031174	-0.005306	0.023267	0.041868	0.001840
Количество отвердителя, м.%	-0.006445	-0.035911	0.024049	1.000000	-0.000684	0.095193	0.055198	-0.065929	-0.075375	0.007446
Содержание эпоксидных групп,%_2	0.019766	-0.008278	-0.006804	-0.000684	1.000000	-0.009769	-0.012940	0.056828	-0.023899	0.015165
Температура вспышки, С_2	-0.004776	-0.020695	0.031174	0.095193	-0.009769	1.000000	0.020121	0.028414	-0.031763	0.059954
Поверхностная плотность, г/ м2	-0.006272	0.044930	-0.005306	0.055198	-0.012940	0.020121	1.000000	0.036702	-0.003210	0.015692
Модуль упругости при растяжении, ГПа	-0.008411	-0.017602	0.023267	-0.065929	0.056828	0.028414	0.036702	1.000000	-0.009009	0.050938
Прочность при растяжении, МПа	0.024148	-0.069981	0.041868	-0.075375	-0.023899	-0.031763	-0.003210	-0.009009	1.000000	0.028602
Потребление смолы, г/м2	0.072531	-0.015937	0.001840	0.007446	0.015165	0.059954	0.015692	0.050938	0.028602	1.000000
Угол нашивки	-0.031073	-0.068474	-0.025417	0.038570	0.008052	0.020695	0.052299	0.023003	0.023398	-0.015334
Шаг нашивки	0.036437	-0.061015	-0.009875	0.014887	0.003022	0.025795	0.038332	-0.029468	-0.059547	0.013394
Плотность нашивки	-0.004652	0.080304	0.056346	0.017248	-0.039073	0.011391	-0.049923	0.006476	0.019604	0.012239

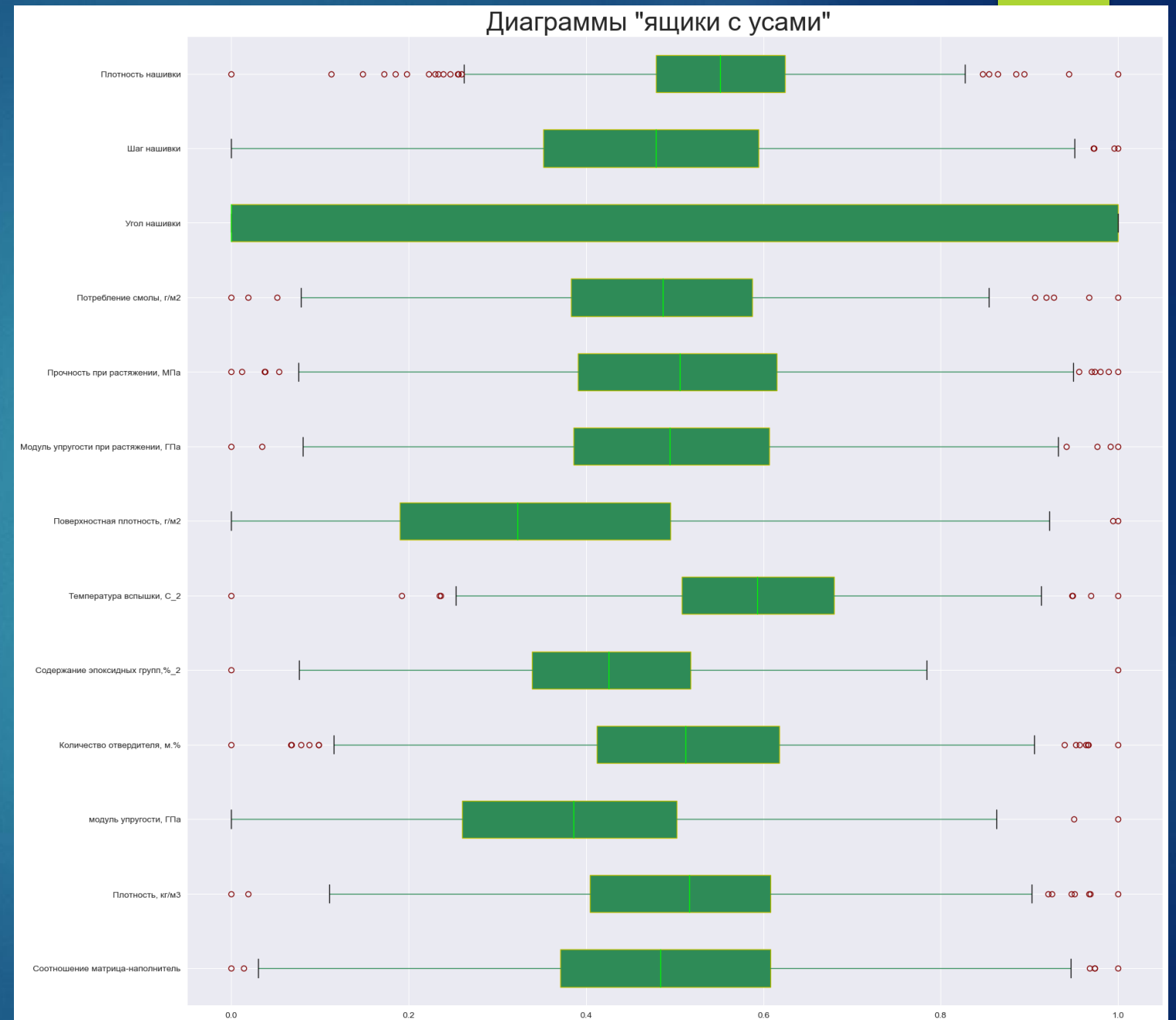
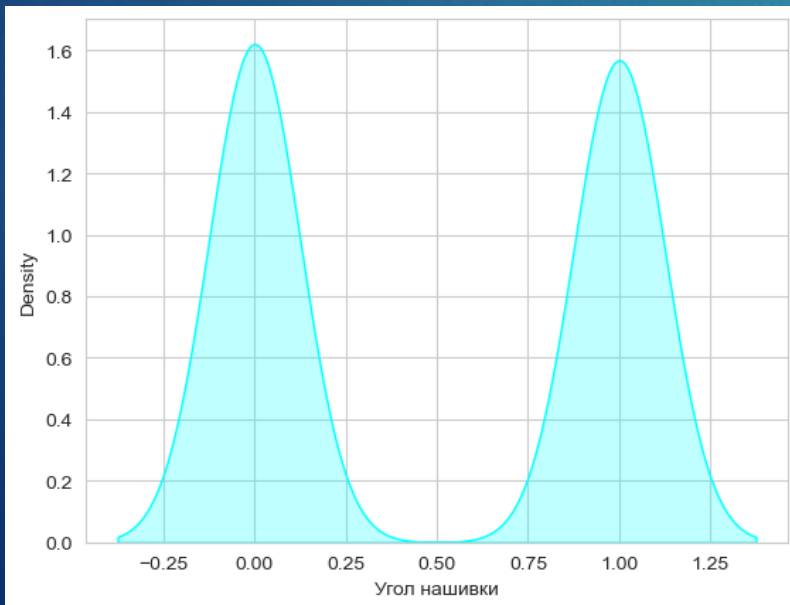
Проведем визуализацию и анализ. Построим гистограммы распределения каждой из переменных



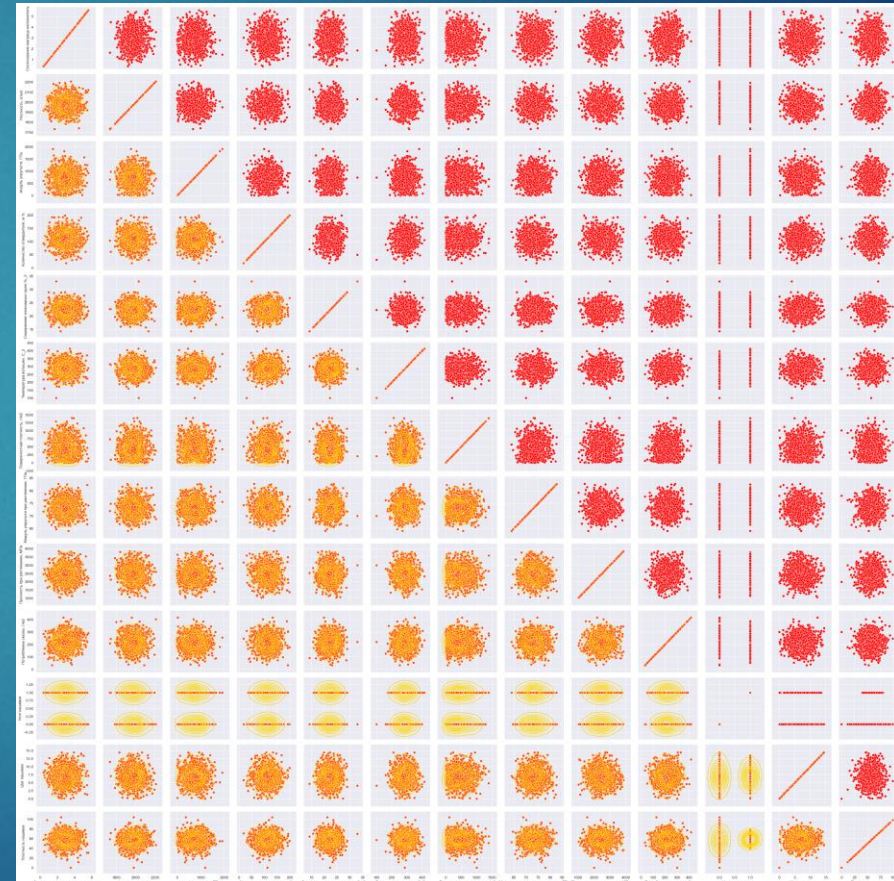
боксплоты (несколько разных способов визуализации), диаграммы "ящиков с усами" (несколько вариантов), попарные графики рассеяния точек (несколько вариантов) графики квантиль-квантиль без нормализации и исключения шумов. Т.к. беглый взгляд на общий файл и дополнительный анализ в excel не дал каких-то явных и бросающихся в глаза закономерностей, то используем разные варианты визуализации в надежде, что получится увидеть какую-то корреляцию. Показатели описательной статистики и визуализация гистограмм и/или диаграмм размаха («ящик с усами») позволяют получить наглядное представление о характерах распределений переменных. Такое частотное распределение показывает, какие именно конкретные значения или диапазоны значений исследуемой переменной встречаются наиболее часто, насколько различаются эти значения, расположено ли большинство наблюдений около среднего значения, является ли распределение симметричным или асимметричным, многомодальным (т.е. имеет две или более вершины) или одномодальным и т.д.



По форме распределения можно судить о природе исследуемой переменной (например, бимодальное распределение позволяет предположить, что выборка не является однородной и содержит наблюдения, принадлежащие двум различным множествам, которые в свою очередь нормально распределены).

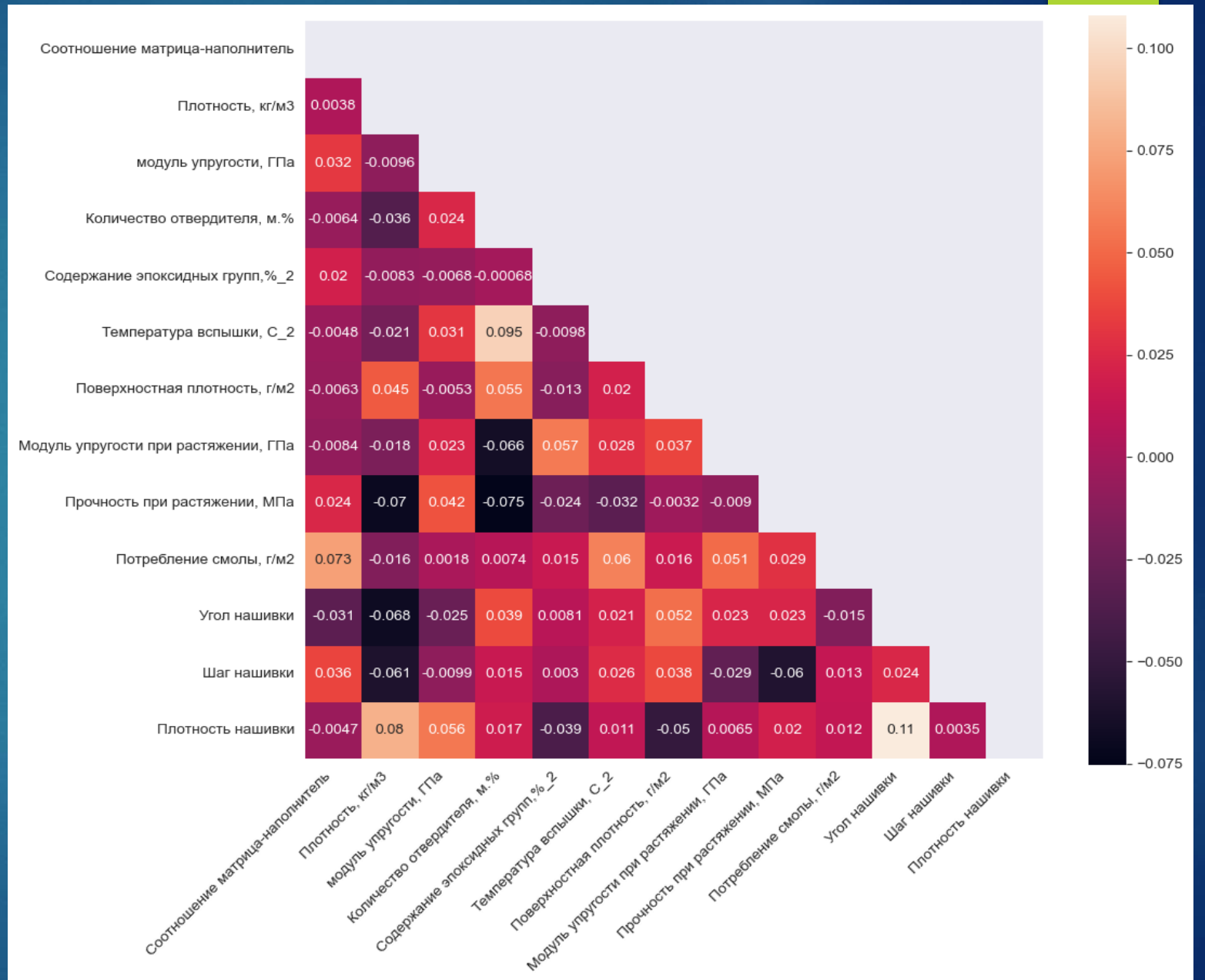


11



Максимальная корреляция между Плотностью нашивки и углом нашивки и составляет 0.11, что говорит об отсутствии зависимости между этими данными.

Корреляция между всеми параметрами очень близка к 0, что говорит об отсутствии корреляционных связей между переменными.



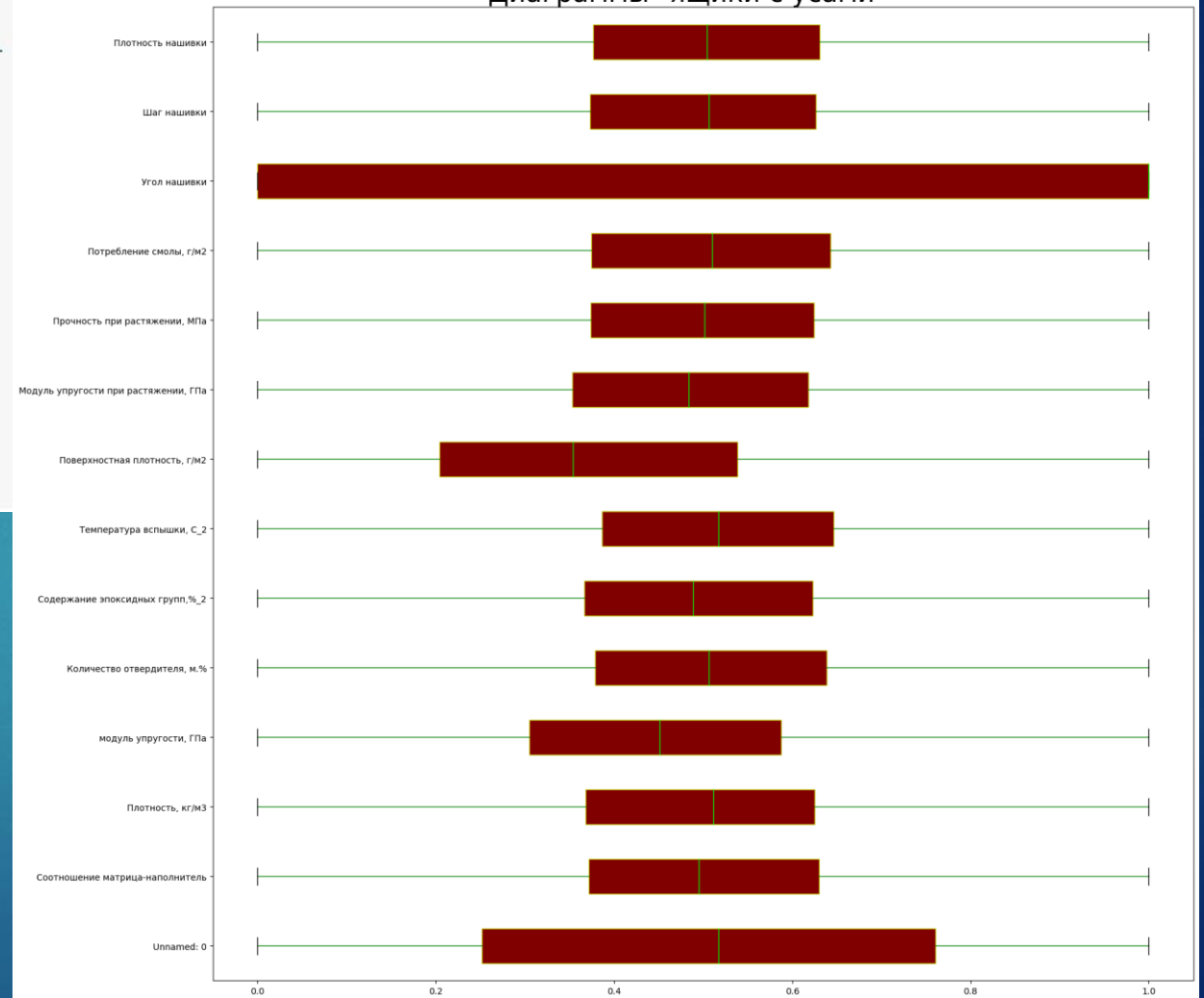
Проведем предобработку данных. Вспомним, что пропуски в данных отсутствуют. Значит, сразу приступаем к работе с выбросами. Для удаления выбросов существует 2 основных метода - метод 3-х сигм и межквартильных расстояний. Сравним эти 2 метода. Для достижения отсутствия выбросов используем методы многократно.

```
metod_3s = 0
metod_iq = 0
count_iq = [] # Список, куда записывается количество выбросов по каждой колонке датафрейма методом 3х сигм.
count_3s = [] # Список, куда записывается количество выбросов по каждой колонке датафрейма методом межквартильных расстояний.
for column in df:
    d = df.loc[:, [column]]
    # методом 3-х сигм
    zscore = (df[column] - df[column].mean()) / df[column].std()
    d['3s'] = zscore.abs() > 3
    metod_3s += d['3s'].sum()
    count_3s.append(d['3s'].sum())
    print(column, '3s', ': ', d['3s'].sum())

    # методом межквартильных расстояний
    q1 = np.quantile(df[column], 0.25)
    q3 = np.quantile(df[column], 0.75)
    iqr = q3 - q1
    lower = q1 - 1.5 * iqr
    upper = q3 + 1.5 * iqr
    d['iq'] = (df[column] <= lower) | (df[column] >= upper)
    metod_iq += d['iq'].sum()
    count_iq.append(d['iq'].sum())
    print(column, ': ', d['iq'].sum())
print('Метод 3-х сигм, выбросов:', metod_3s)
print('Метод межквартильных расстояний, выбросов:', metod_iq)
```

```
Unnamed: 0 3s : 0
Unnamed: 0 : 0
Соотношение матрица-наполнитель 3s : 0
Соотношение матрица-наполнитель : 6
Плотность, кг/м3 3s : 3
Плотность, кг/м3 : 9
модуль упругости, ГПа 3s : 2
модуль упругости, ГПа : 2
Количество отвердителя, м.% 3s : 2
Количество отвердителя, м.% : 14
Содержание эпоксидных групп,% 2 3s : 2
Содержание эпоксидных групп,% 2 : 2
Температура вспышки, C_2 3s : 3
Температура вспышки, C_2 : 8
Поверхностная плотность, г/м2 3s : 2
Поверхностная плотность, г/м2 : 2
Модуль упругости при растяжении, ГПа 3s : 0
Модуль упругости при растяжении, ГПа : 6
Прочность при растяжении, МПа 3s : 0
Прочность при растяжении, МПа : 11
Потребление смолы, г/м2 3s : 3
Потребление смолы, г/м2 : 8
Угол нашивки 3s : 0
Угол нашивки : 0
Шаг нашивки 3s : 0
Шаг нашивки : 4
Плотность нашивки 3s : 7
Плотность нашивки : 21
Метод 3-х сигм, выбросов: 24
Метод межквартильных расстояний, выбросов: 93
```

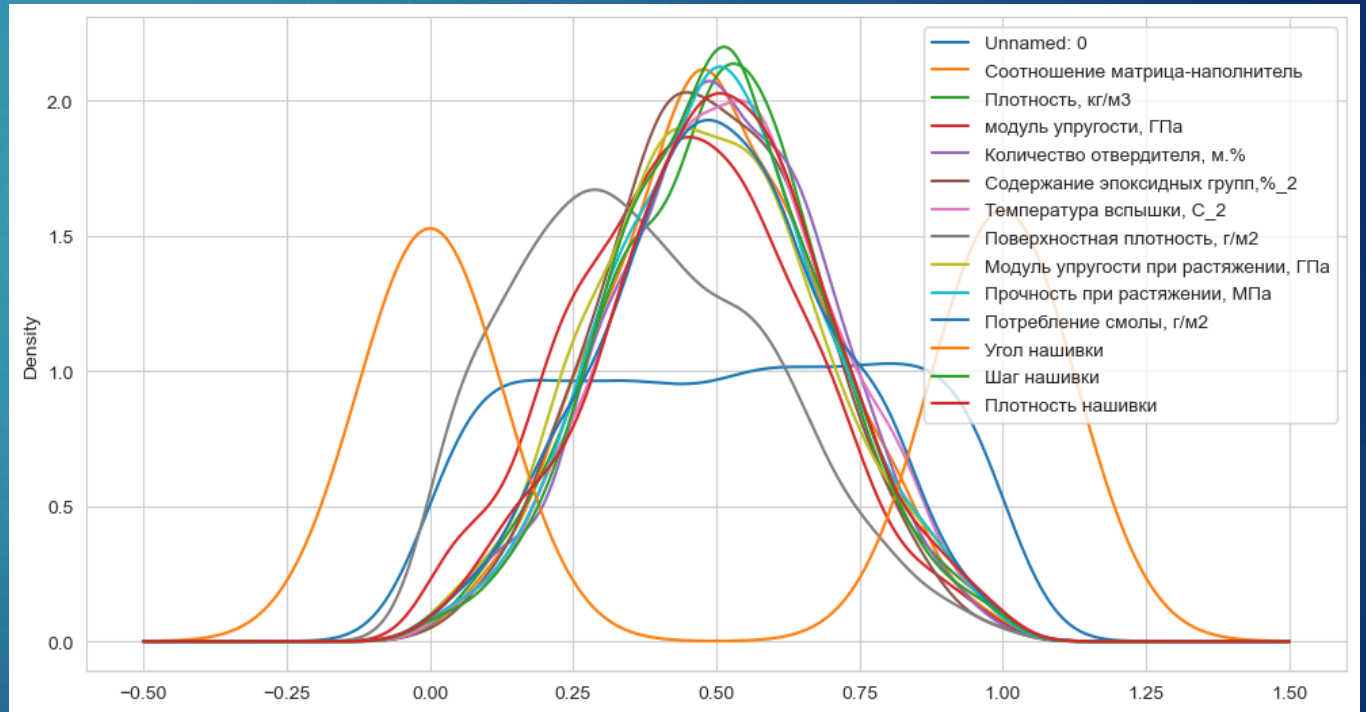
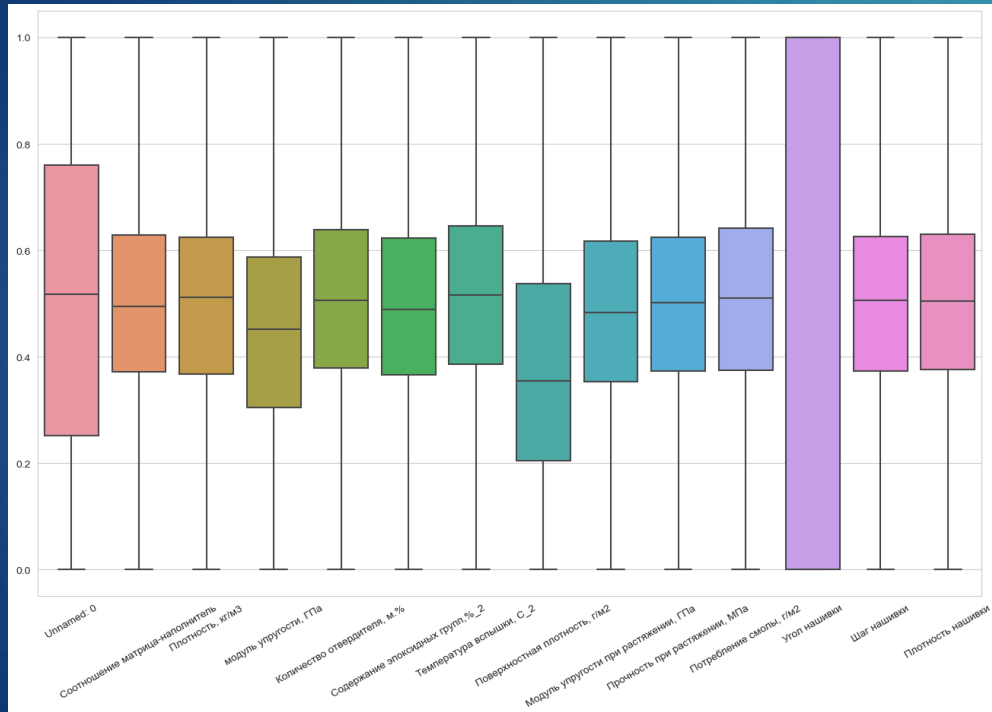
Диаграммы "ящики с усами"



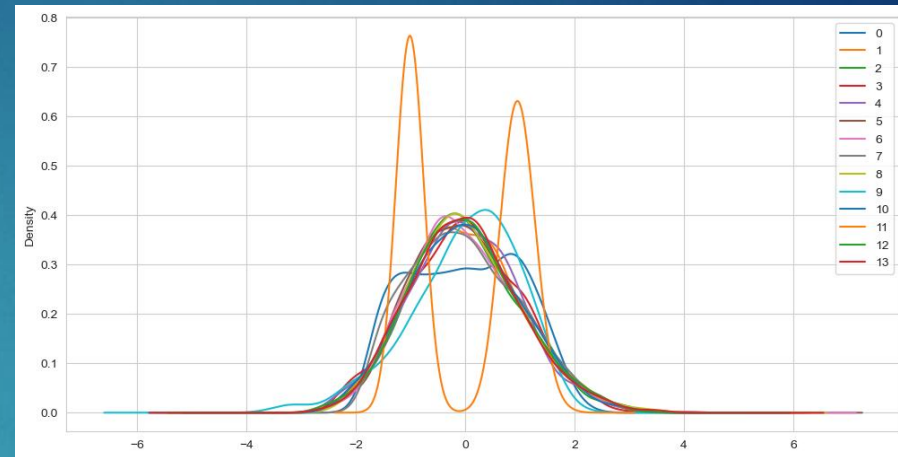
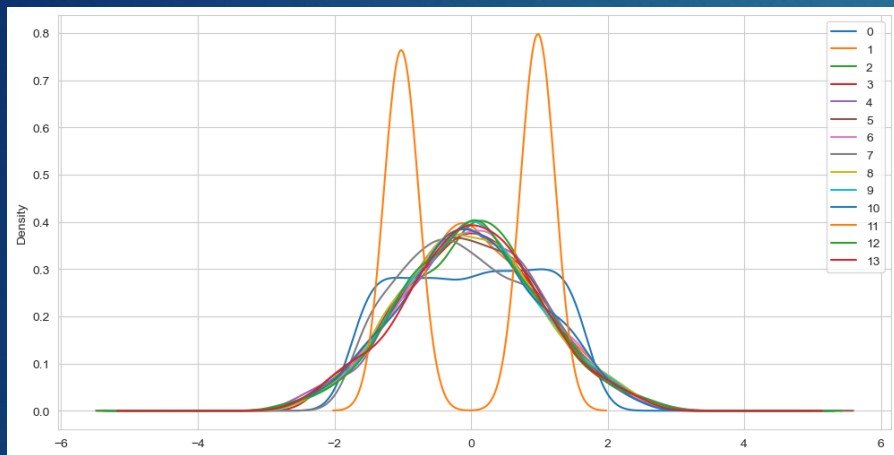
Предобработка данных.

Нормализуем данные. У нас в основном количественные признаки, поэтому можно применить нормализацию (приведение в диапазон от 0 до 1) или стандартизацию (приведение к матожиданию 0, стандартному отклонению 1). Т.к. это в том числе учебная работа, то используем и нормализацию, и стандартизацию.

Этап предобработки данных нужен нам и для введенных данных в будущем приложении.



Стандартизируем данные с помощью `StandardScaler()`;
Оценим результат, в том числе, на графиках



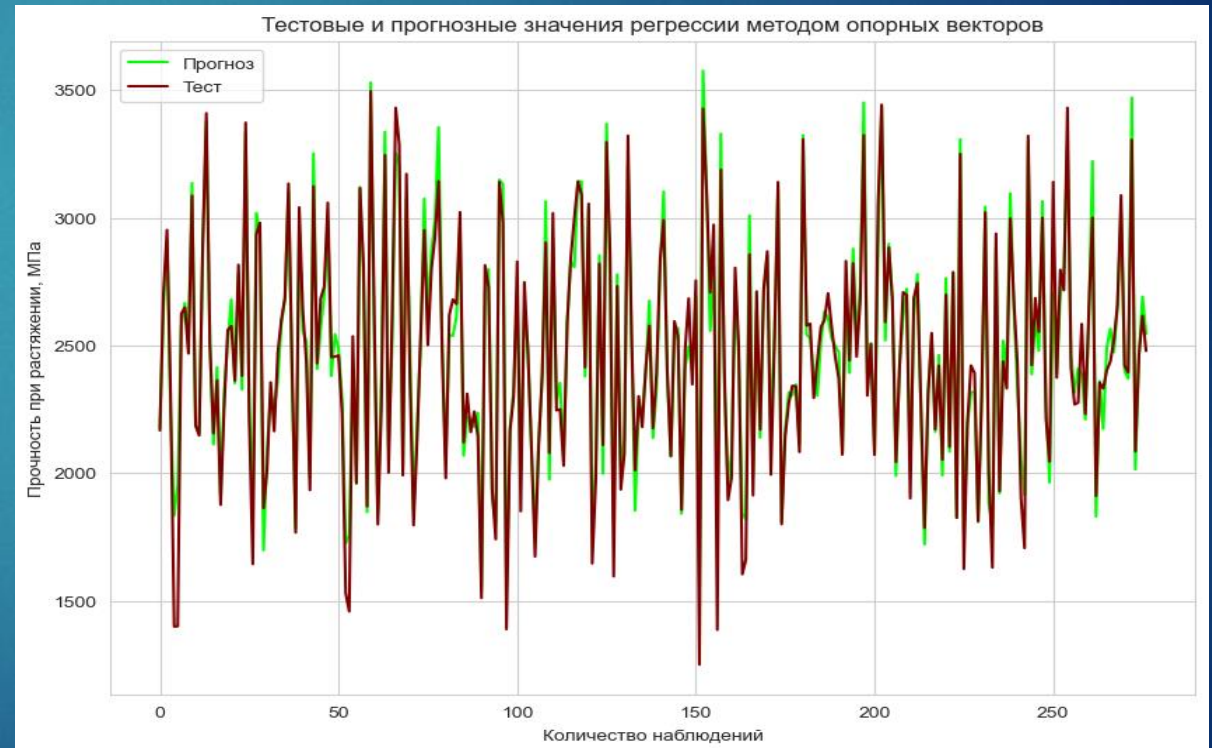
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	-1.739385	-1.324290	-0.673310	-0.192709	-2.238400	-0.229669	-0.592779	-1.043223	-1.252979	1.233256	-0.315993	-1.011359	-1.249682	-0.268579
1	-1.733271	-1.325677	-0.680131	-0.194349	0.172936	-0.913623	-0.315598	-1.043756	-1.259092	1.224275	-0.318535	-1.011359	-0.912495	-1.212890
2	-1.730217	-0.452381	-0.683041	-0.154526	-0.351629	-0.639223	-0.598576	-1.043983	-1.261699	1.220444	-0.319619	-1.011359	-0.913058	-0.490481
3	-1.727089	-0.443357	-0.772113	-0.158863	-0.335771	-0.611150	-0.574422	-1.040815	-1.225372	1.273817	-0.304515	-1.011359	-0.905207	-0.253258
4	-1.723842	-0.610240	-1.082192	0.031521	-0.303027	-0.553184	-0.524549	-1.034272	-1.150363	1.384023	-0.273329	-1.011359	-0.888996	0.522208
...
917	1.667873	-0.722175	-0.213300	0.548260	-0.856201	-0.697708	0.704321	-0.981427	-0.133222	-0.250095	-1.546787	0.967697	0.759512	-0.886443
918	1.719164	0.556940	0.384241	-0.889752	1.185882	-0.771553	-0.637253	-0.460225	-0.028640	-0.223856	-1.637904	0.995491	1.366327	-0.292881
919	1.494237	0.153711	-0.608268	-1.064542	-0.263789	0.066835	-1.087322	0.769892	-0.414989	0.317007	0.058926	0.863251	-1.143186	0.464726
920	1.810786	0.930807	0.705665	0.104358	1.147689	-0.735985	-0.062347	0.646362	0.335566	-1.237139	-0.254341	1.044573	-0.173505	0.184334
921	1.404339	0.564912	-1.224577	-1.099651	0.205722	0.828228	-0.260270	0.756697	-0.720070	0.749105	-0.672037	0.807617	-0.533394	1.028527

922 rows × 14 columns

Разработка и обучение моделей для прогноза прочности при растяжении:
После всех подготовительных работ переходим к процессу создания, обучения моделей. Мы будем использовать в Python библиотеку Scikit-Learn. В качестве базового уровня предскажем медианное значение цели на обучающем наборе для всех примеров в тестовом наборе. В качестве метрики возьмём среднюю абсолютную ошибку (mae) в прогнозах. Для обучения используем 70 % данных, а для тестирования — 30 %. Первый прогноз построим методом опорных векторов.

```
svr = make_pipeline(StandardScaler(), SVR(kernel = 'rbf', C = 500.0, epsilon = 1.0))
svr.fit(x_train_1, np.ravel(y_train_1))
y_pred_svr=svr.predict(x_test_1)
mae_svr = mean_absolute_error(y_pred_svr, y_test_1)
mse_svr_str = mean_squared_error(y_test_1,y_pred_svr)
print('Support Vector Regression Results Train:')
print("Test score train: {:.2f}".format(svr.score(x_train_1, y_train_1)))
print('Support Vector Regression Results:')
print('SVR_MAE:', round(mean_absolute_error(y_test_1, y_pred_svr)))
print('SVR_MAPE: {:.2f}'.format(mean_absolute_percentage_error(y_test_1, y_pred_svr)))
print('SVR_MSE: {:.2f}'.format(mse_svr_str))
print("SVR_RMSE: {:.2f}".format(np.sqrt(mse_svr_str)))
print("Test score test: {:.2f}".format(svr.score(x_test_1, y_test_1)))
```

```
Support Vector Regression Results Train:
Test score train: 0.99
Support Vector Regression Results:
SVR_MAE: 76
SVR_MAPE: 0.03
SVR_MSE: 11159.19
SVR_RMSE: 105.64
Test score test: 0.95
```



Второй метод – линейной регрессии. Третий – К ближайших соседей. Далее находим гиперпараметры методом GridSearchCV с перекрёстной проверкой с количеством блоков 10. Выводим гиперпараметры для оптимальной модели, подставляем оптимальные гиперпараметры в модель К ближайших соседей, обучаем модель. Сравниваем результаты предсказания моделей по метрикам MAE, MSE. Аналогично делаем с Модулем упругости при растяжении.

```
# Метод К ближайших соседей - K Neighbors Regressor - 3
knn = KNeighborsRegressor(n_neighbors=5)
knn.fit(x_train_1, y_train_1)
y_pred_knn = knn.predict(x_test_1)
mae_knn = mean_absolute_error(y_pred_knn, y_test_1)
mse_knn_str = mean_squared_error(y_test_1, y_pred_knn)
print('K Neighbors Regressor Results Train:')
print("Test score: {:.2f}".format(knn.score(x_train_1, y_train_1)))
print('K Neighbors Regressor Results:')
print('KNN_MAE: ', round(mean_absolute_error(y_test_1, y_pred_knn)))
print('KNN_MAPE: {:.2f}'.format(mean_absolute_percentage_error(y_test_1, y_pred_knn)))
print('KNN_MSE: {:.2f}'.format(mse_knn_str))
print("KNN_RMSE: {:.2f}".format(np.sqrt(mse_knn_str)))
print("Test score: {:.2f}".format(knn.score(x_test_1, y_test_1)))
```

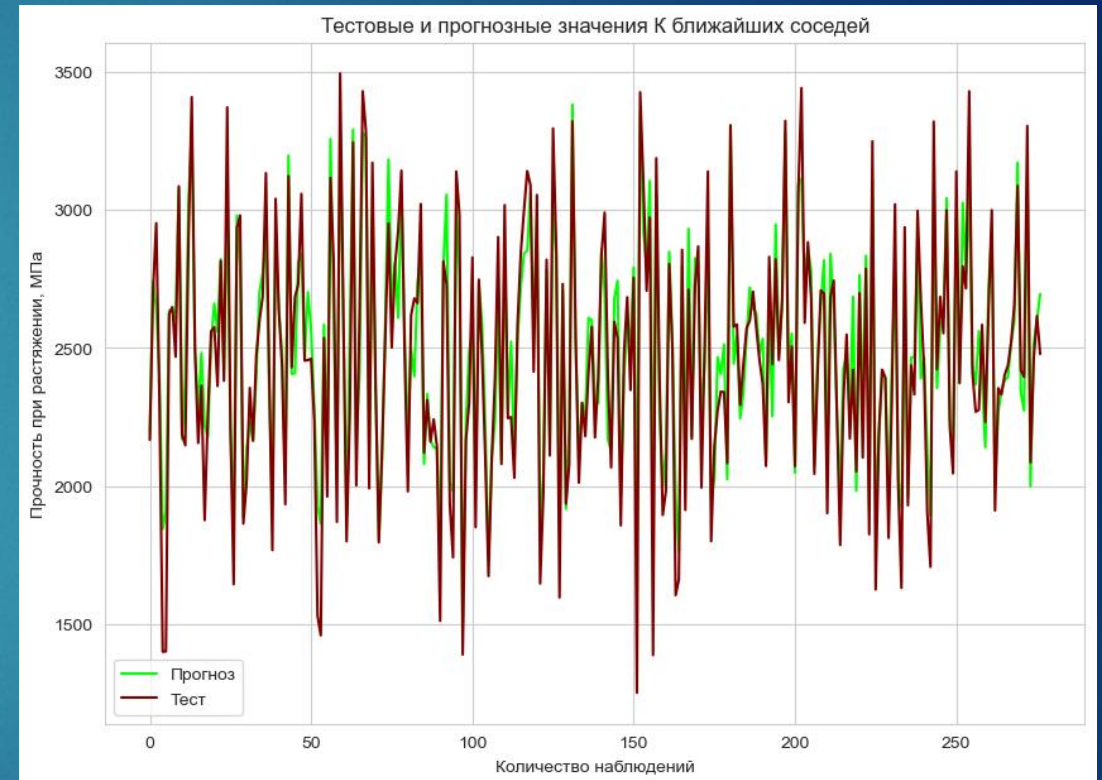
```
K Neighbors Regressor Results Train:
Test score: 0.92
K Neighbors Regressor Results:
KNN_MAE: 120
KNN_MAPE: 0.05
KNN_MSE: 24594.48
KNN_RMSE: 156.83
Test score: 0.89
```

mae_df

	Перепсcop	MAE	MSE
0	Support Vector	76.326686	11159.192078
1	Linear Regression	64.302274	6491.797584
2	KNeighbors	120.264255	24594.475902

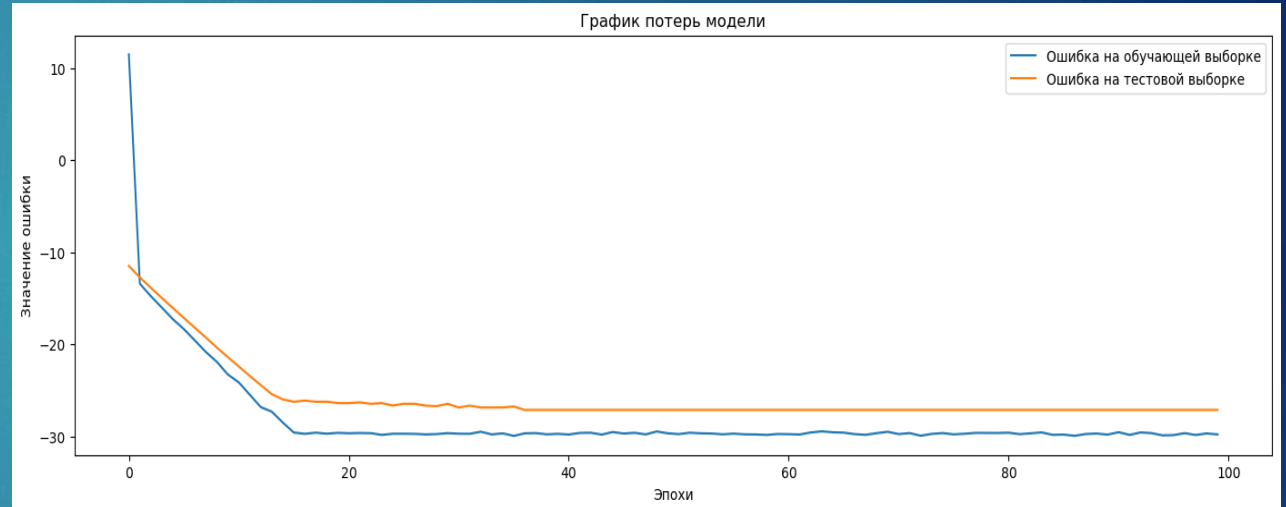
mae2_df

	Перепсcop	MAE	MSE
0	Support Vector	3.450178	18.380163
1	Linear Regression	2.612429	6491.797584
2	KNeighbors	2414.374790	12.047130



Написать нейронную сеть, которая будет рекомендовать соотношение «матрица – наполнитель». Обучение нейронной сети— это такой процесс, при котором происходит подбор оптимальных параметров модели, с точки зрения минимизации функционала ошибки. Начнём строить нейронную сеть с помощью класса `keras.Sequential`. Определим оптимальные параметры, посмотрим на результаты. С помощью `KerasClassifier` выйдем на наилучшие параметры для нашей нейронной сети и построим окончательную нейросеть. Обучим и оценим модель, посмотрим на потери, зададим функцию для визуализации факт/прогноз для результатов моделей.

```
def create_model(lyrs=[32], act='softmax', opt='SGD', dr=0.1):  
  
    seed = 7  
    np.random.seed(seed)  
    tf.random.set_seed(seed)  
  
    model = Sequential()  
    model.add(Dense(lyrs[0], input_dim=x_train.shape[1], activation=act))  
    for i in range(1, len(lyrs)):  
        model.add(Dense(lyrs[i], activation=act))  
  
    model.add(Dropout(dr))  
    model.add(Dense(3, activation='tanh')) # выходной слой  
  
    model.compile(loss='binary_crossentropy', optimizer=opt, metrics=['mae', 'accuracy'])  
  
    return model
```

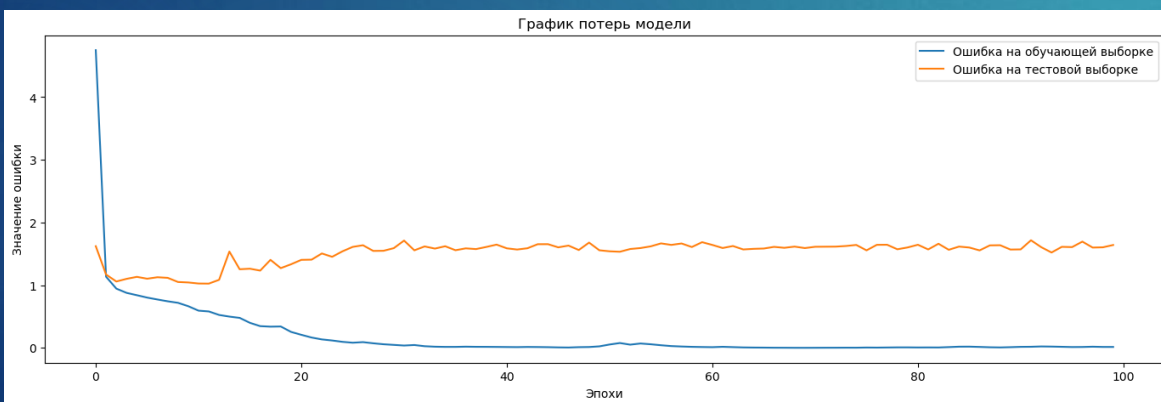
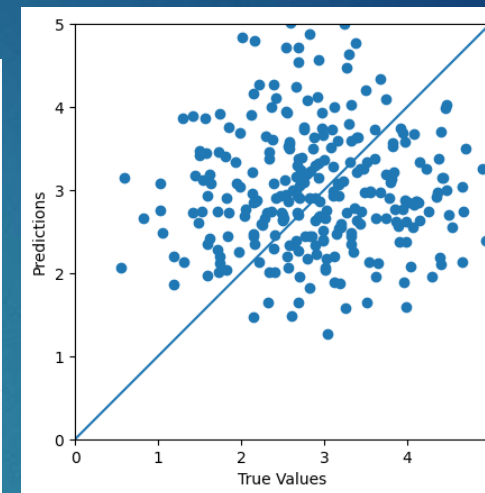


Построение второй модели нейросети: Сформируем входы и выход для модели, разобьем на обучающую и тестовую выборки, нормализуем данные. Построим модель, задав слои, обучим ее. Посмотрим на метрики и потери модели, построим графики потерь тренировочной и тестовой выборок. Построим график результата модели.

```
# Сконфигурируем другую модель, зададим слои
from tensorflow.keras.layers import Dense
model1 = tf.keras.Sequential([x_train_n, layers.Dense(128, activation='relu'),
                             layers.Dense(128, activation='relu'),
                             layers.Dense(128, activation='relu'),
                             layers.Dense(64, activation='relu'),
                             layers.Dense(64, activation='relu'),
                             layers.Dense(32, activation='relu'),
                             layers.Dense(16, activation='relu'),
                             layers.Dense(1)

                             ])

model1.compile(optimizer = tf.keras.optimizers.Adam(0.001), loss = 'mean_squared_error', metrics = [tf.keras.metrics.RootMeanSquareError])
# Посмотрим на архитектуру модели
model1.summary()
```



```
# оценка модели MSE
model1.evaluate(x_test, y_test, verbose = 1)

9/9 [=====] - 0s 6ms/step - loss: 1.3339 - root_mean_squared_error: 1.1549
[1.3338944911956787, 1.1549434661865234]
```

Приложение:

сохранив модель, используем ее для предсказания значения параметра «матрица – наполнитель». Предположительно работающий код приложения и шаблон страницы ввода и получения данных был создан, но рабочего варианта приложения не удалось добиться. Проблема, предположительно в настройках веб-сервера, не удается запустить службу Apache (я так понял, она не дает открыть локал-хост), что не позволяет открыть адрес <http://127.0.0.1:5000/>, устанавливаемый Flask по умолчанию для работы приложения.

Прогнозирование значения параметра «Соотношение матрица-наполнитель»	
Введите значения и нажмите "Рассчитать"	
Плотность, кг/м3	Введите значение плотности
Модуль упругости, ГПа	Введите значение модуля
Количество отвердителя, м. %	Введите значение количес
Содержание эпоксидных групп, % 2	Введите значение содержа
Температура вспышки, С 2	Введите значение темпера
Поверхностная плотность, г/м2	Введите значение поверхн
Модуль упругости при растяжении, ГПа	Введите значение модуля
Прочность при растяжении, МПа	Введите значение прочнос
Потребление смолы, г/м2	Введите значение потребл
Угол нашивки, град	Введите значение угла наш
Шаг нашивки	Введите значение шага на
Плотность нашивки	Введите значение плотнос
<input type="button" value="Рассчитать"/>	
Результат прогноза:	

Репозиторий был создан на github.com по адресу:

<https://github.com/Pavel-Tsygankov/VKR>

Распределение полученных данных в объединённом датасете близко к нормальному, но коэффициенты корреляции между парами признаков стремятся к нулю. Используемые при разработке моделей подходы не позволили получить сколько-нибудь достоверных прогнозов. Был сделан вывод, что невозможно определить из свойств материалов соотношение «матрица – наполнитель».