



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Prognosing final properties of composite materials

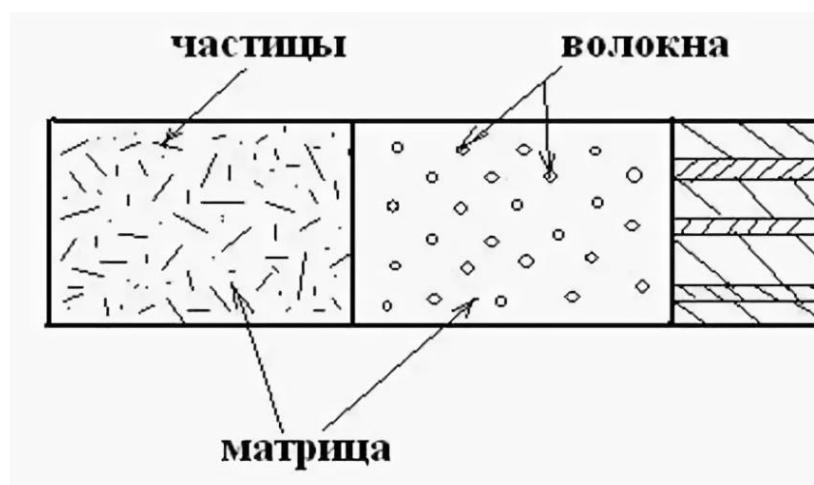
«Data Science pro»

Anna Pogorelova



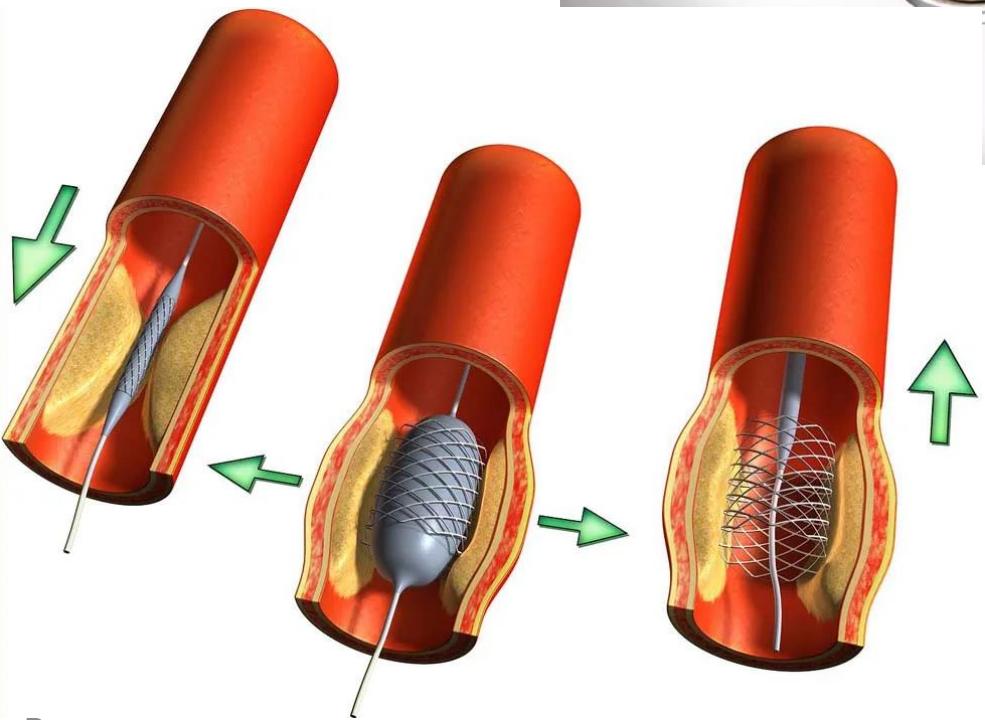
Introduction

- 1 The advantages of its components or extremely high property indicators manifest in the composite.
- 2 Some examples of composite materials include reinforced concrete, carbon, and glass ceramics.





ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана





The goal of the project

Predicting the final properties of new composite materials based on the characteristics of their components and certain production parameters.

Objectives of the study:

1. Train a machine learning algorithm to predict tensile modulus.
2. Train a machine learning algorithm to predict tensile strength.
3. Develop a neural network that recommends the matrix-filler ratio.
4. Create an application that provides a forecast of the matrix-filler ratio upon entering a set of parameters.



Compilation of the final dataset

- import libraries
- load files

```
# загружаем файл с характеристиками материала
path = '/Users/kusita_1/Desktop/dataset compositum/X_bp.xlsx'
df_bp = pd.read_excel(path, index_col=0)
```

```
# объединяем две таблицы по индексу, тип объединения согласно условию - 'inner'
df = df_bp.merge(df_nup, left_index=True, right_index=True, how='inner')
df.head(3)
```

Python

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	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0	1.857143	2030.0	738.736842	30.0	22.267857	100.000000	210.0	70.0	3000.0	220.0	0	4.0
1	1.857143	2030.0	738.736842	50.0	23.750000	284.615385	210.0	70.0	3000.0	220.0	0	4.0
2	1.857143	2030.0	738.736842	49.9	33.000000	284.615385	210.0	70.0	3000.0	220.0	0	4.0

```
# проверяем размерность итогового датасета
df.shape
```

Python

(1023, 13)



Initial data analysis датасета

```
# посмотрим на типы данных и их количество в наборе  
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
Index: 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
```

```
df.isna().sum().sum()
```

0

```
# проверим датасет на наличие дубликатов  
any(df.duplicated())
```

False

- df['Угол нашивки, град'].value_counts()

Угол нашивки, град	count	dtype
0	520	
90	503	

Name: count, dtype: int64



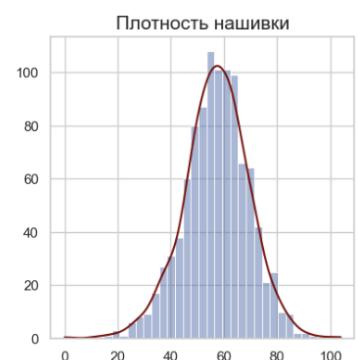
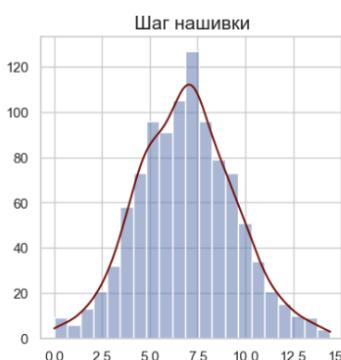
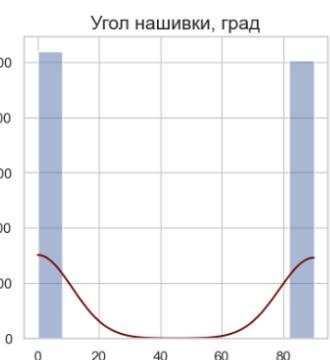
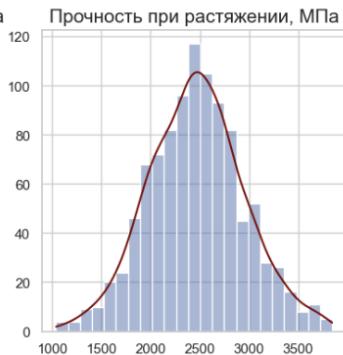
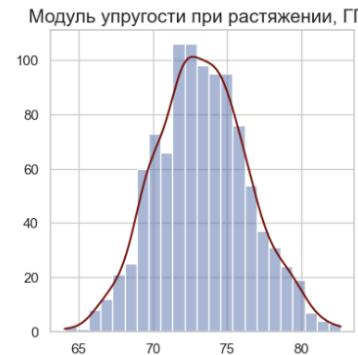
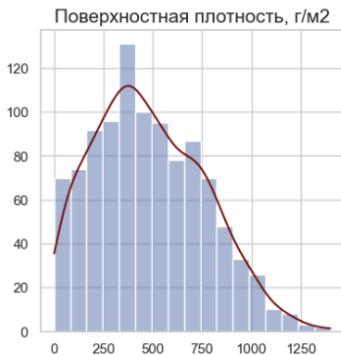
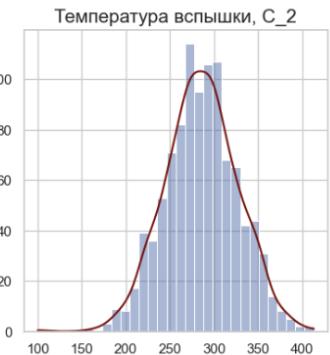
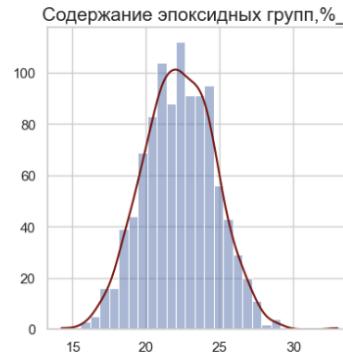
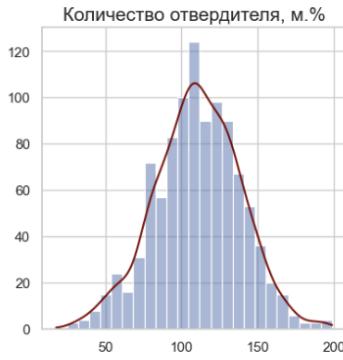
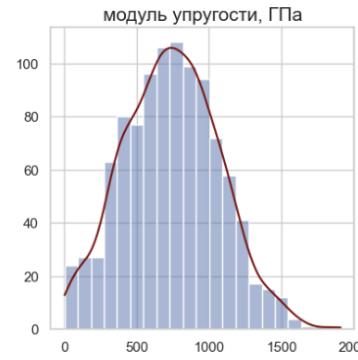
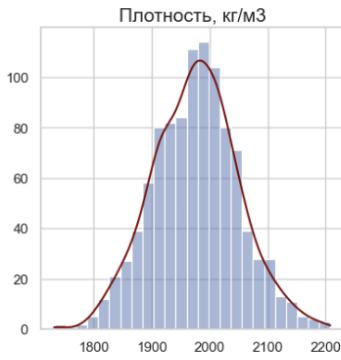
Descriptive statistics

df.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, С_2	1023.0	285.882151	40.943260	100.000000	259.066528	285.896812	313.002106	413.273418
Поверхностная плотность, г/м2	1023.0	482.731833	281.314690	0.603740	266.816645	451.864365	693.225017	1399.542362
Модуль упругости при растяжении, ГПа	1023.0	73.328571	3.118983	64.054061	71.245018	73.268805	75.356612	82.682051
Прочность при растяжении, МПа	1023.0	2466.922843	485.628006	1036.856605	2135.850448	2459.524526	2767.193119	3848.436732
Потребление смолы, г/м2	1023.0	218.423144	59.735931	33.803026	179.627520	219.198882	257.481724	414.590628
Угол нашивки, град	1023.0	44.252199	45.015793	0.000000	0.000000	0.000000	90.000000	90.000000
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



Exploratory data analysis





Для параметра Поверхностная плотность, г/м²

Максимальное значение: 1399.54236233989

Минимальное значение 0.603739925153945

Среднее: 482.73183303841853

Стандартное отклонение: 281.17716165829404

Коэф. асимметрии (skewness): 0.394885855345264

Коэф. эксцесса (kurtosis): -0.44281878519445605

Критерий Шапиро–Уилка (Shapiro–Wilk): statistic=0.9775959849357605, pvalue= 1.848812249127807e-

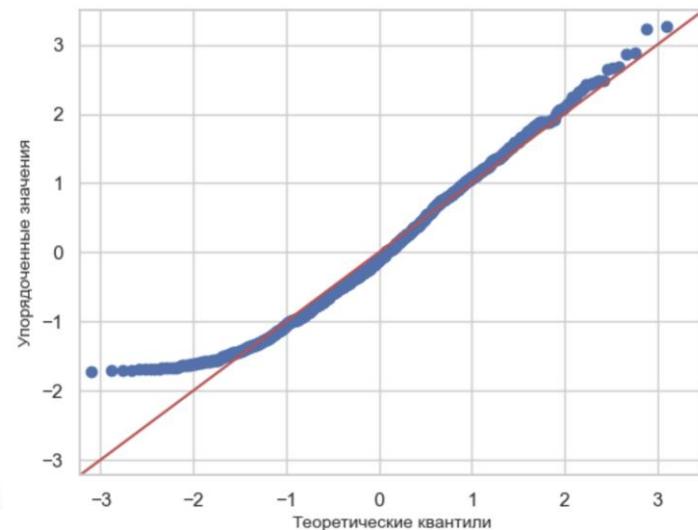
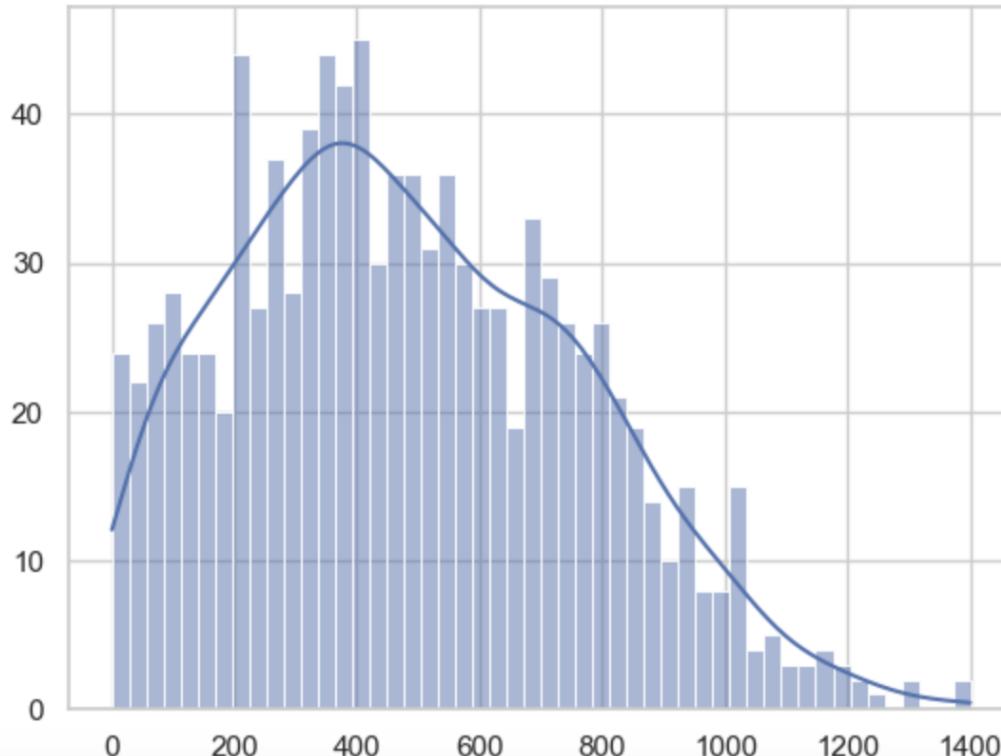
Нулевая гипотеза может быть отвергнута

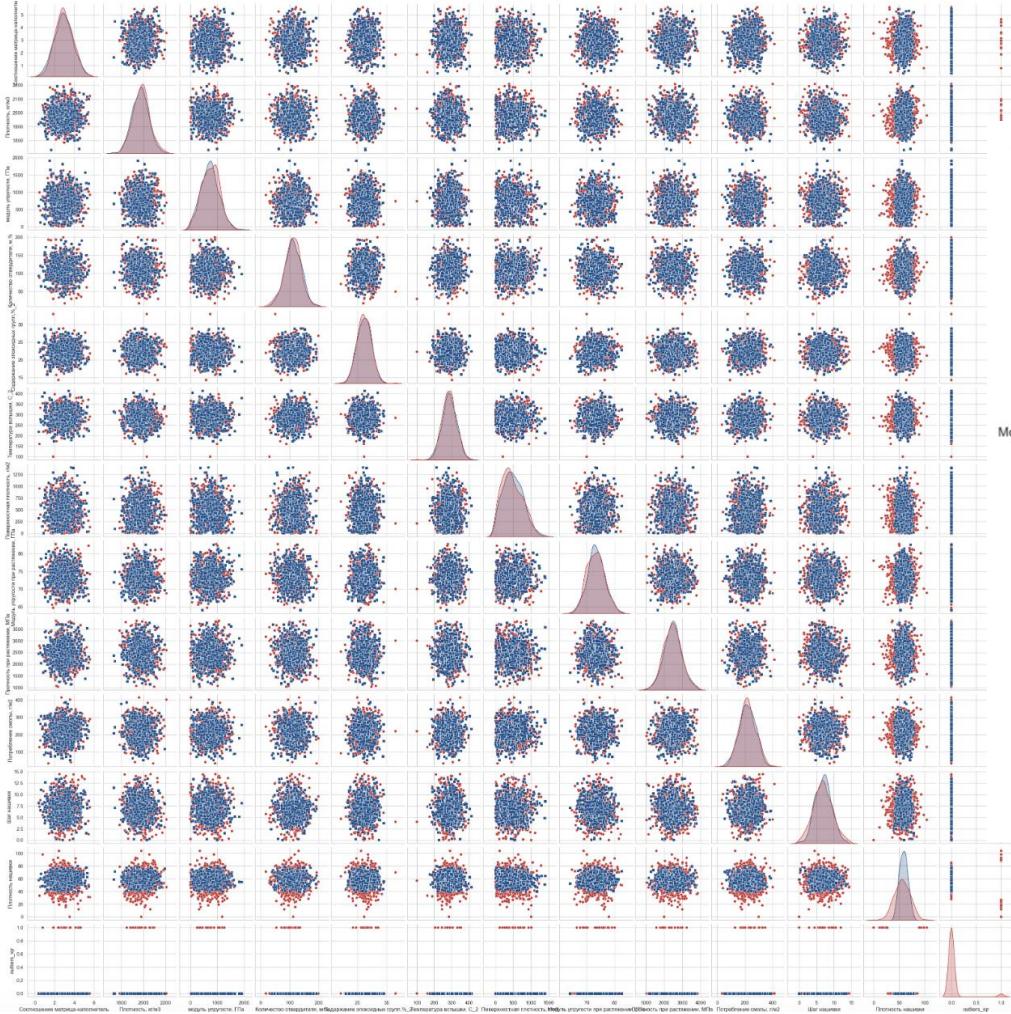
Критерий Колмогорова–Смирнова (Kolmogorov): statistic=0.9970467056181723, pvalue= 0.0

Нулевая гипотеза может быть отвергнута

Критерий К–квадрата Д'Агостино (normaltest): statistic=[38.51062683], pvalue= [4.34033886e–09]

Нулевая гипотеза может быть отвергнута



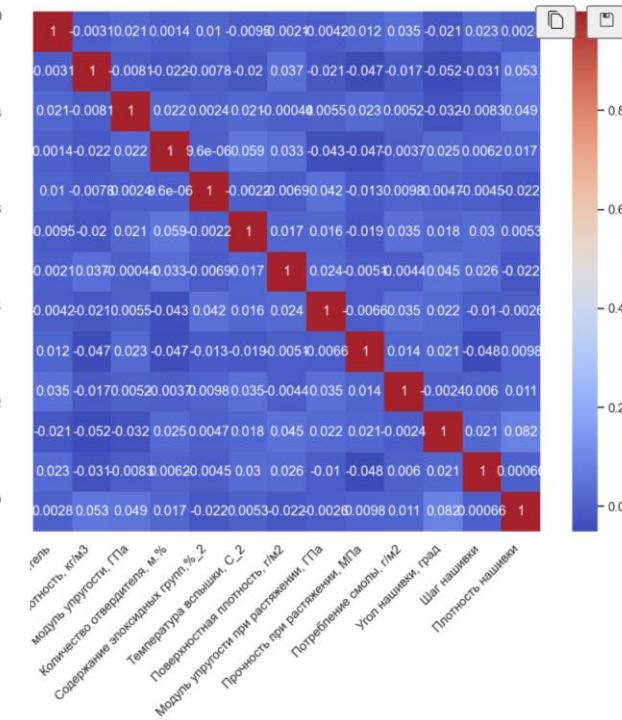


Correlation analysis

Pearson's coef.

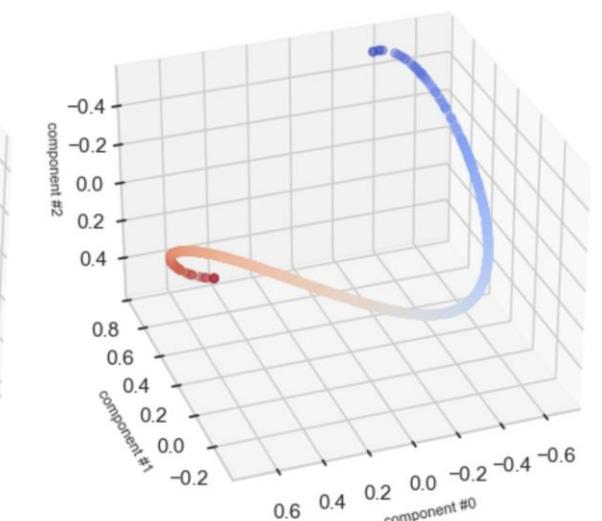
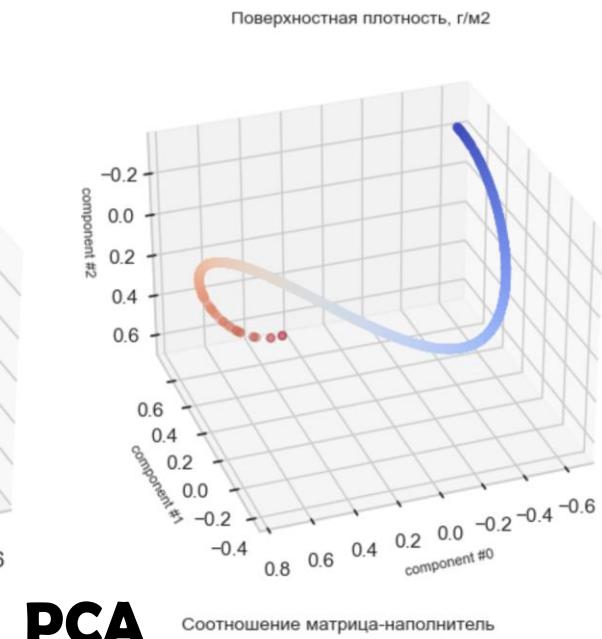
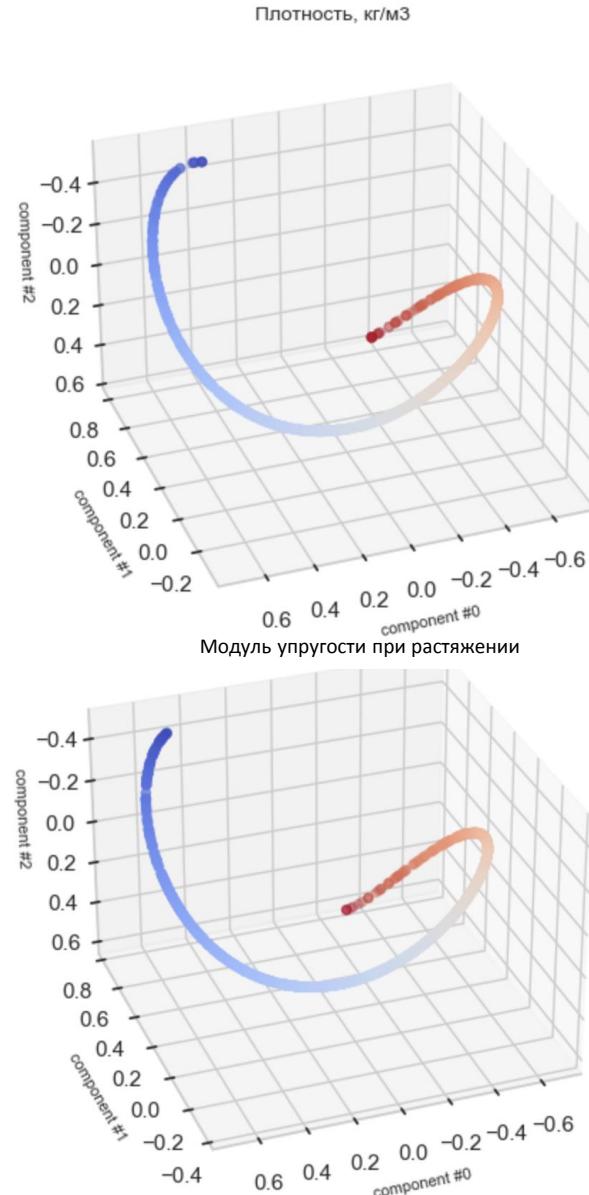
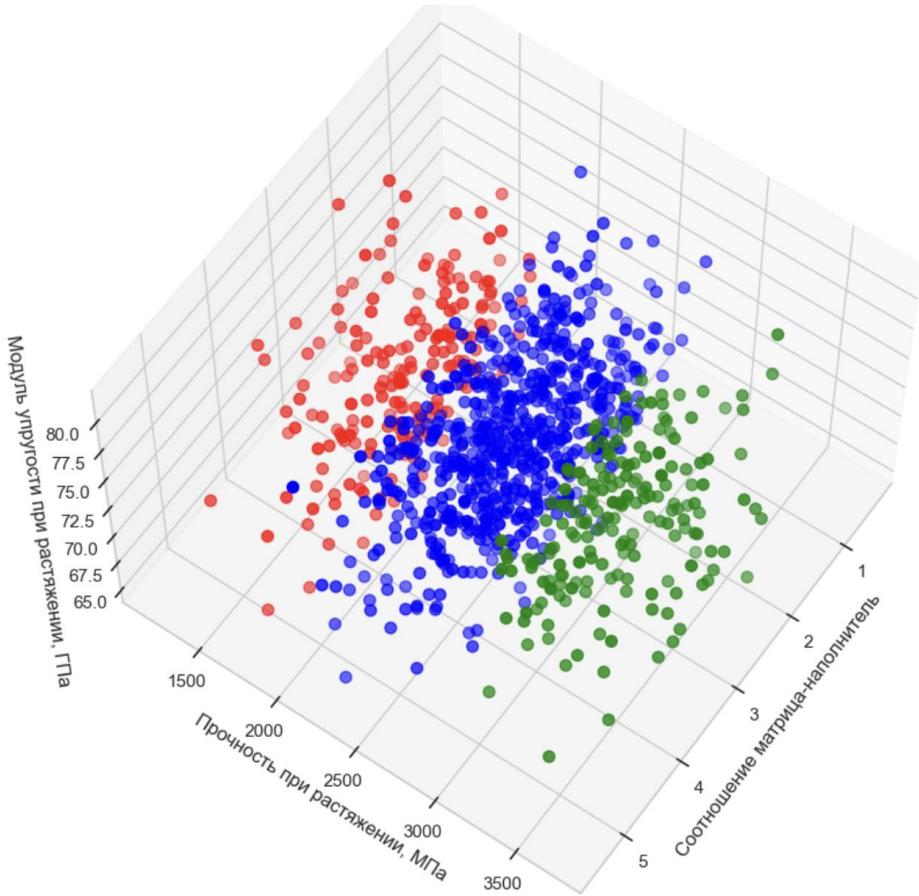
Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп, %_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
1	0.00380.032-0.00640.02-0.00480.00630.00840.0240.073-0.0310.036-0.0047											
Плотность, кг/м3	1	-0.00960.0360.00830.0210.045-0.018-0.07-0.016-0.068-0.0610.08										
модуль упругости, ГПа	0.032-0.0096	1	0.024-0.00680.031-0.00530.0230.0420.0018-0.02500.0990.056									
Количество отвердителя, м.%	0.00640.0360.024	1	-0.000680.0950.055-0.066-0.0750.00740.0390.0150.017									
Содержание эпоксидных групп, %_2	0.02-0.00830.00680.0068	1	-0.00980.0130.057-0.0240.0150.00810.003-0.039									
Температура вспышки, С_2	0.00480.0210.0310.095-0.098	1	0.020.028-0.0320.060.0210.0260.011									
Поверхностная плотность, г/м2	0.00630.045-0.00530.055-0.0130.02	1	0.037-0.00320.0160.0520.038-0.05									
Модуль упругости при растяжении, ГПа	0.00840.0180.023-0.0660.0570.0280.037	1	-0.0090.0510.023-0.0290.0065									
Прочность при растяжении, МПа	0.024-0.070.042-0.075-0.0240.0320.00320.009	1	0.0290.023-0.060.02									
Потребление смолы, г/м2	0.073-0.0160.0180.00740.0150.06	1	0.0160.0510.029-0.015									
Угол нашивки, град	-0.031-0.068-0.0250.0390.00810.0210.052	1	0.0230.023-0.0150.0240.11									
Шаг нашивки	0.036-0.0610.00990.0150.0030.026	1	0.038-0.029-0.060.0130.0241									
Плотность нашивки	0.00470.080.0560.017-0.0390.011	1	-0.050.00650.020.0120.110.0035									

Kendall's coef.





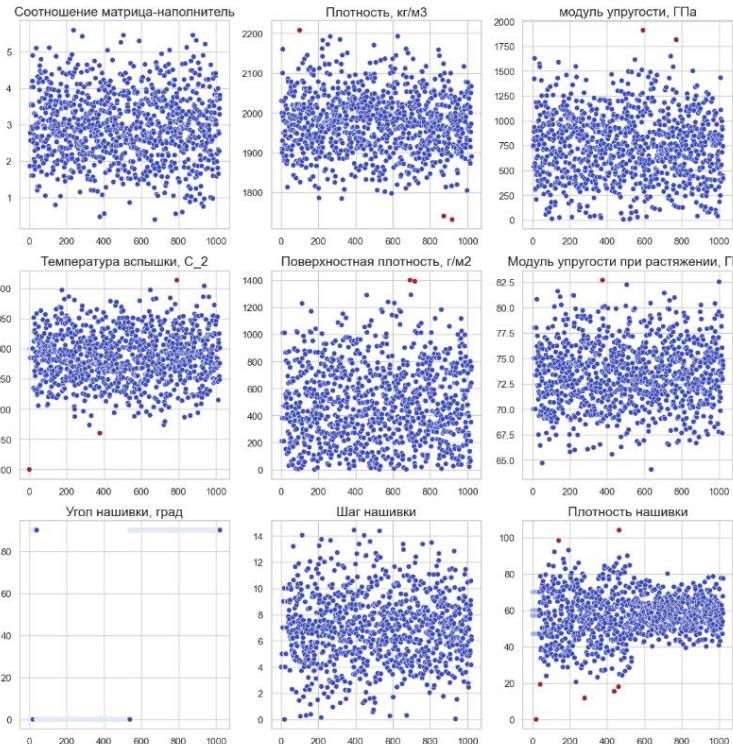
Exploratory data analysis: PCA



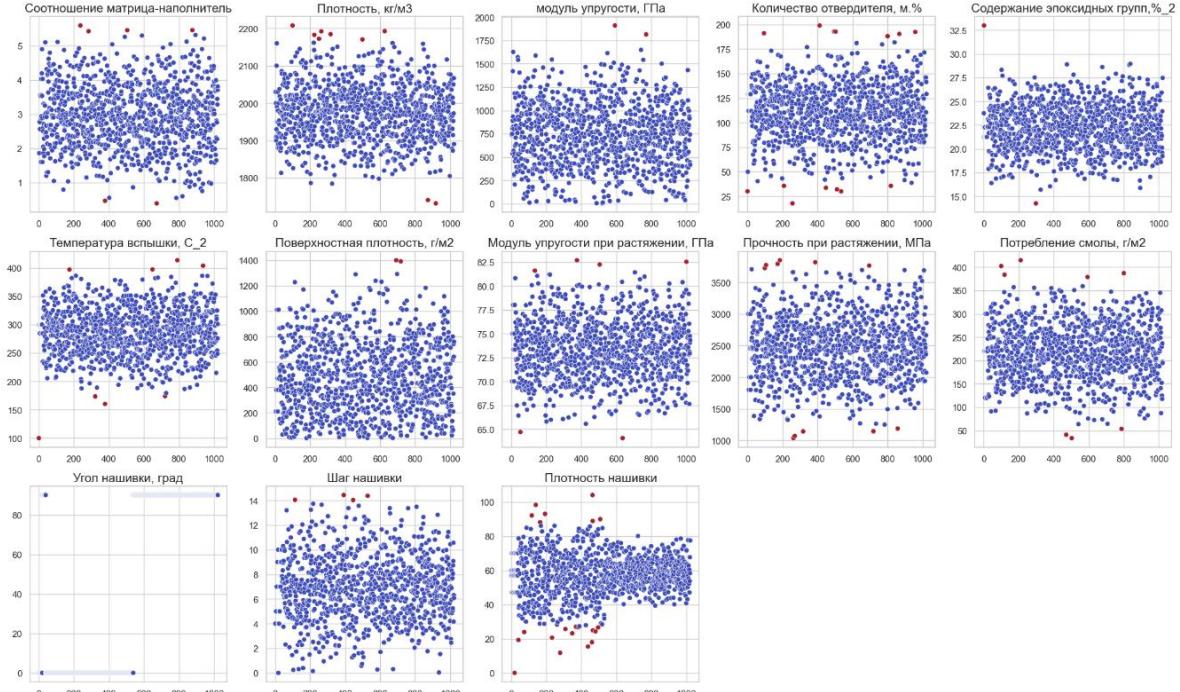


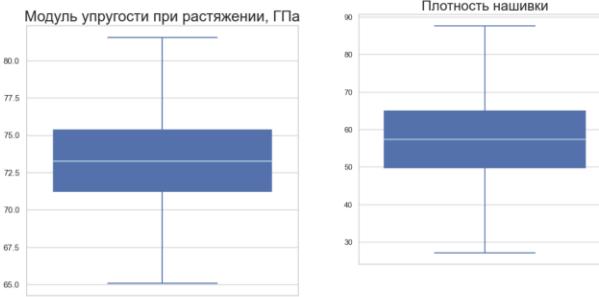
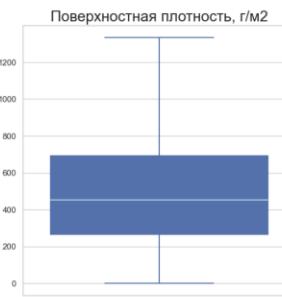
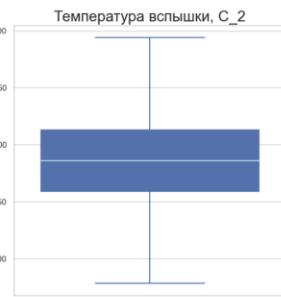
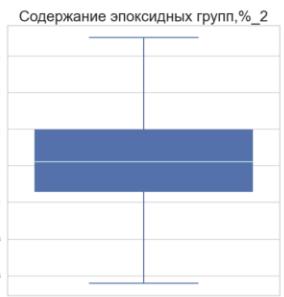
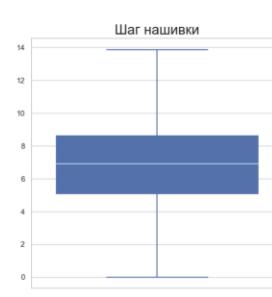
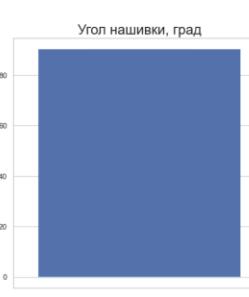
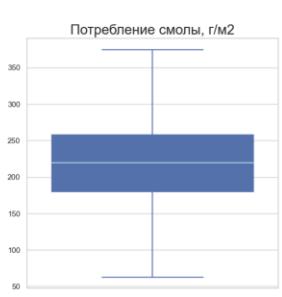
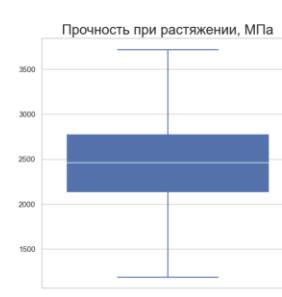
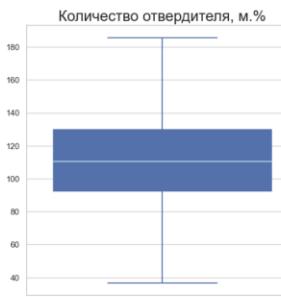
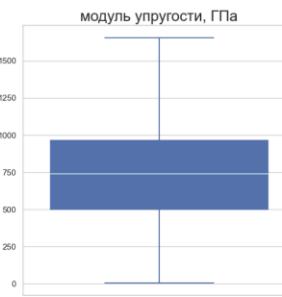
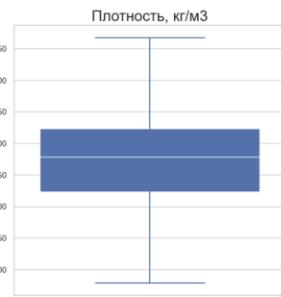
Outliers detection

Общее количество выбросов: 25



Общее количество выбросов: 93







Data normalization

Следующим этапом для приведения данных к единой шкале будет нормализация.

Учитывая, что все данные – положительные числа, первоначально используем для нормализации `MinMaxScaler()`.

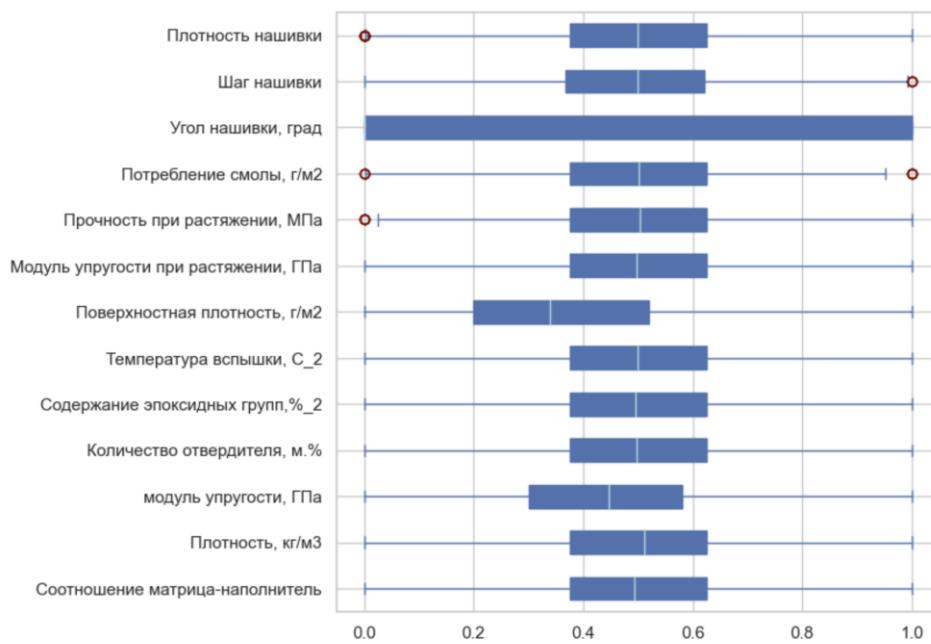
```
min_max_scaler = preprocessing.MinMaxScaler()  
df_min_max_scaled = min_max_scaler.fit_transform(df)  
df_min_max_scaled = pd.DataFrame(df_min_max_scaled, columns=df.columns)  
# проверим получившуюся после нормализации таблицу  
df_min_max_scaled.head(3)
```

то есть сделаем нормализацию по рядам в массиве

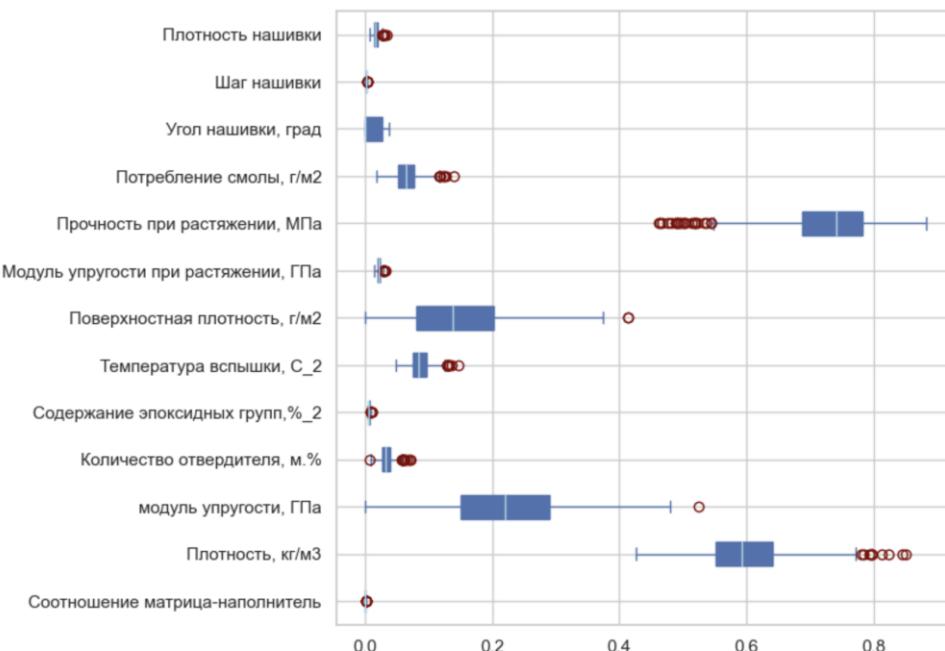
в качестве нормы для расчета используем норму (тип нормализации) по умолчанию – L2

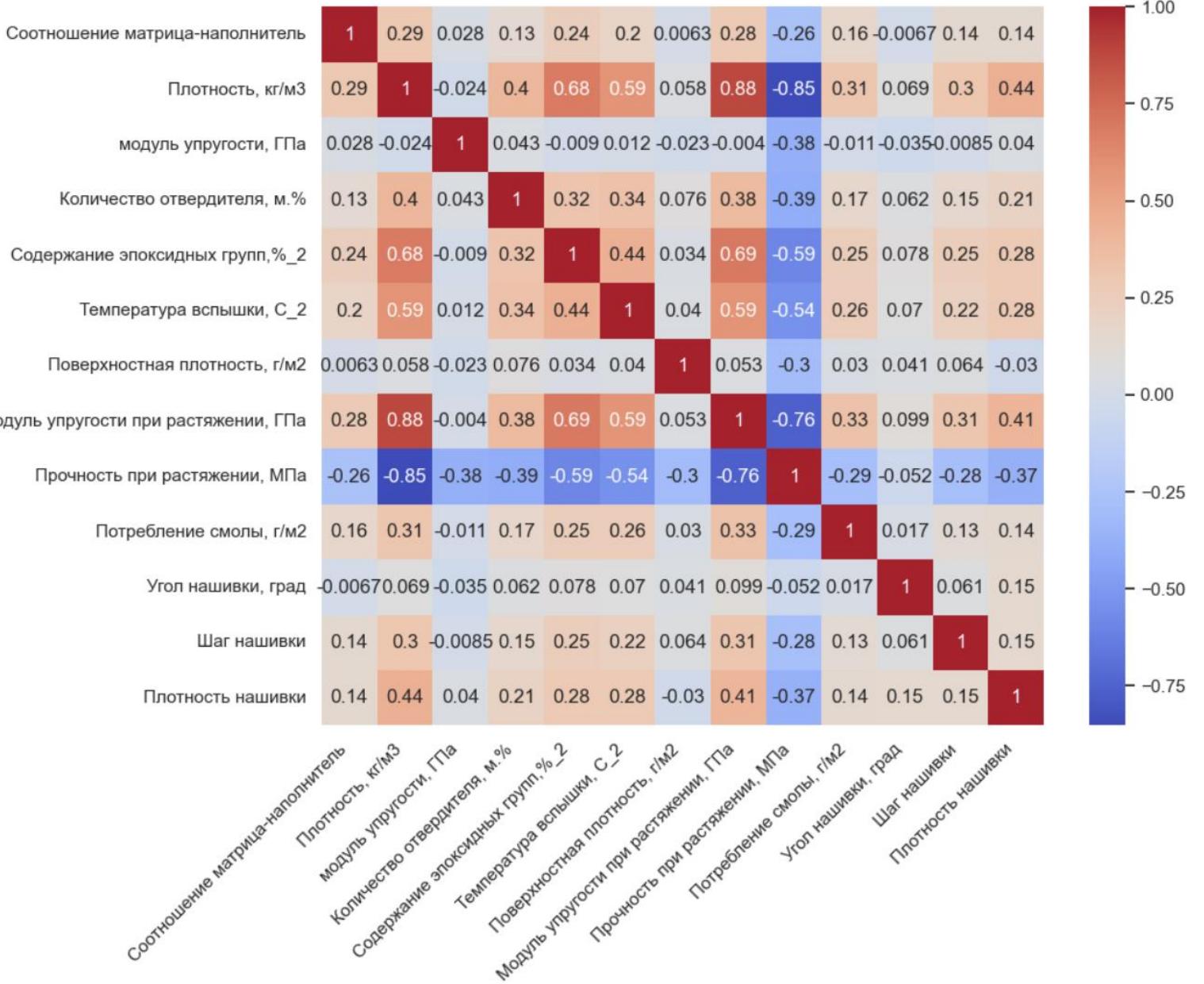
```
df_normalized_sample = preprocessing.normalize(np.array(df))  
df_normalized_sample = pd.DataFrame(df_normalized_sample, columns=df.columns)  
df_normalized_sample.head(3)
```

Диаграммы "ящики с усами"



Диаграммы "ящики с усами"



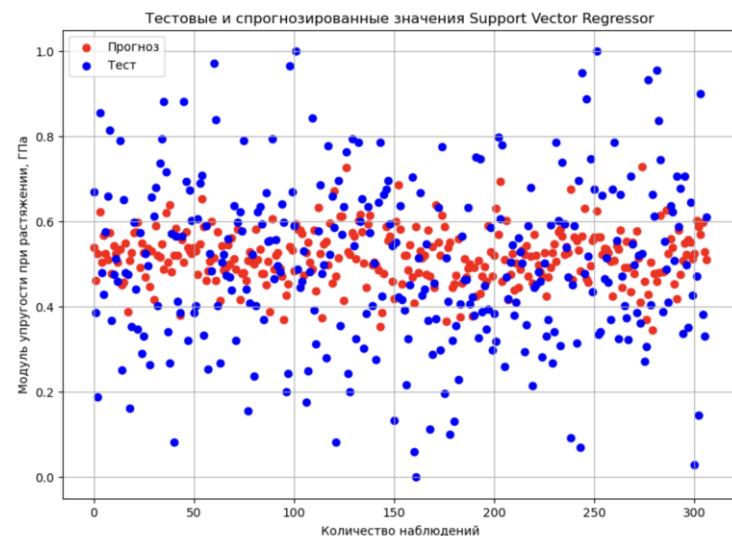




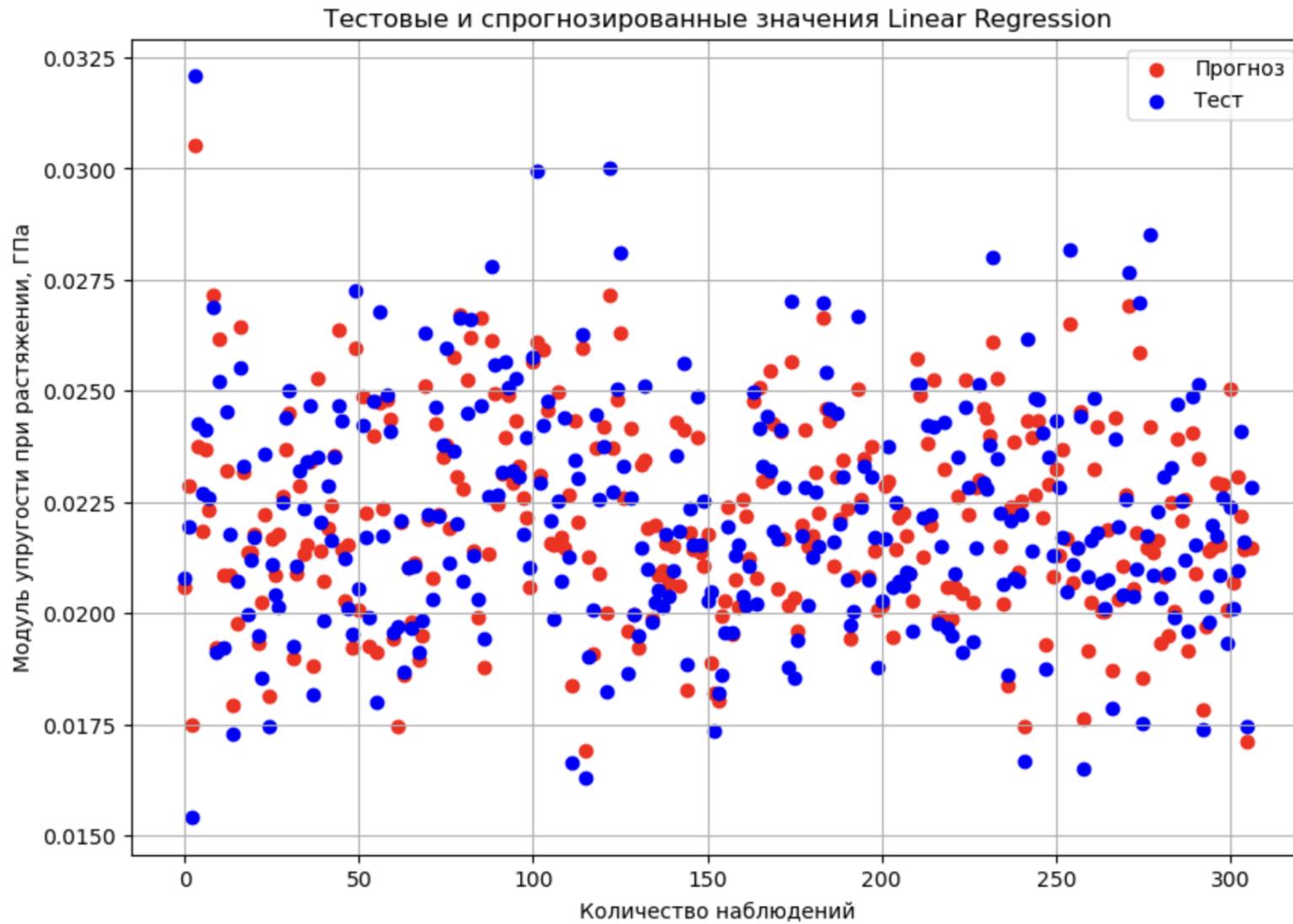
Prediction of tensile modulus

Steps:

- data preparation (for several models);
- test/train (30%; 70%);
- training;
- visualization;
- choosing hyperparameters (GridSearch);



	Perceptron	RMSE	r2 score
0	Linear_Regression	0.190390	0.016809
1	Linear_Regression_sample_normalization	3.126316	0.019888
2	Linear_Regression_categ_angle	0.190390	0.016809
3	Linear_Regression_sample_normalization_categ_a...	3.126312	0.019890
4	Support Vector Regressor	0.198266	-0.066223
5	Support Vector Regressor_sample_normalization	3.155871	0.015510
6	Support Vector Regressor_categ_angle	0.198266	-0.066223
7	Random Forest Regressor	3.172353	0.005200
8	KNeighbors Regressor	0.172178	-0.016857
9	Gradient Boosting Regressor	3.164110	0.010363
10	Support Vector Regressor + GridSearch	0.170952	-0.002427

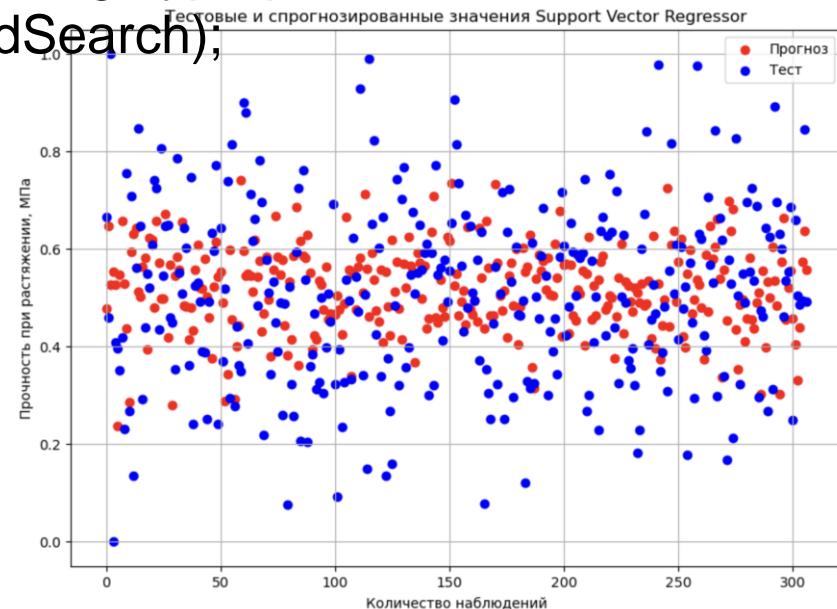




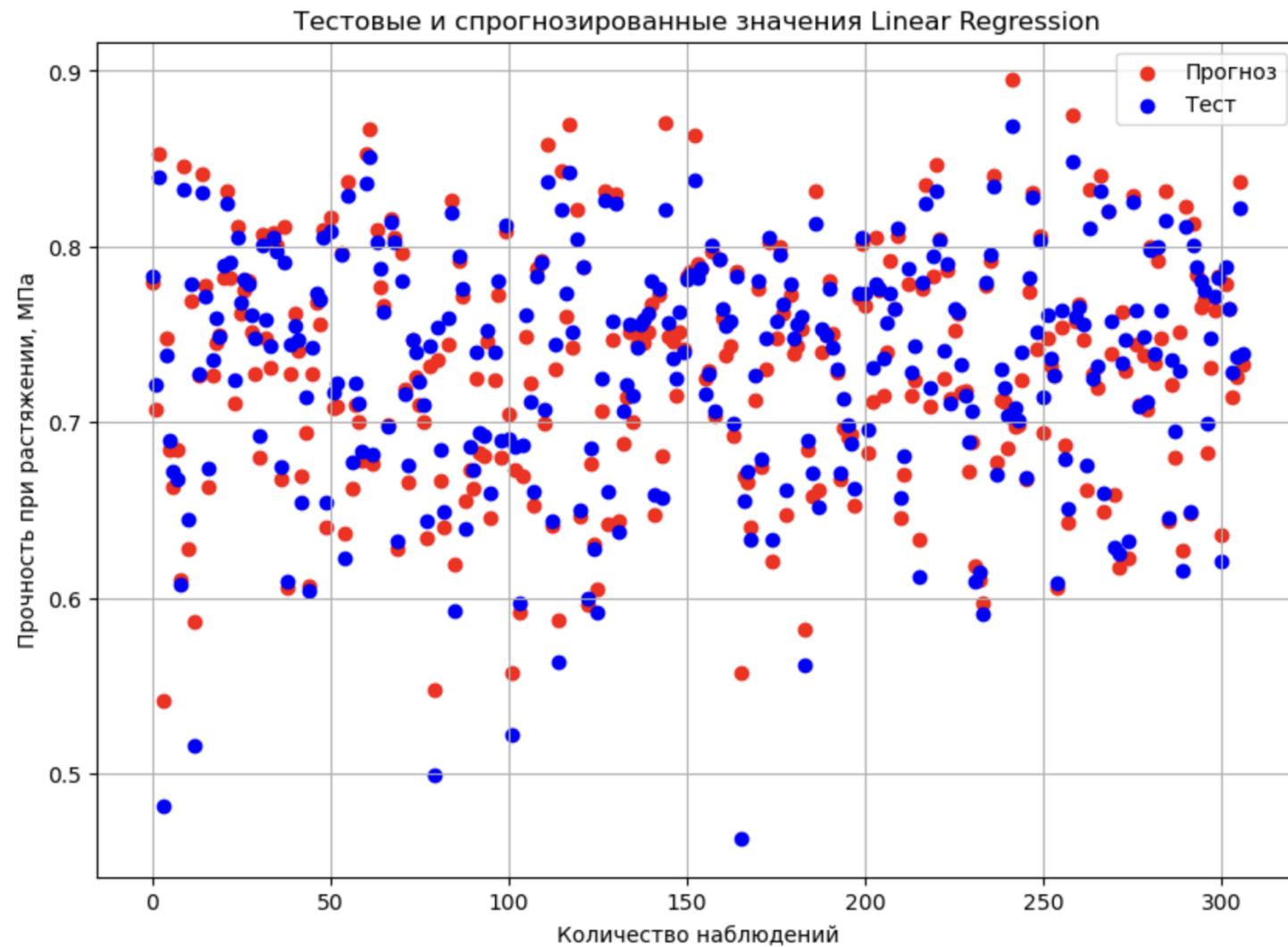
Prediction of tensile strength

Steps:

- data preparation (for several models);
- test/train (30%; 70%);
- training;
- visualization;
- choosing hyperparameters (GridSearch);



	Perceptron	RMSE	r2 score
0	Linear_Regression	0.186273	-0.029695
1	Linear_Regression_sample_normalization	470.122198	-0.028440
2	Support Vector Regressor	0.204037	-0.235450
3	Support Vector Regressor_sample_normalization	463.729506	-0.000661
4	Random Forest Regressor	468.614005	-0.021852
5	KNeighbors Regressor	0.188264	-0.051825
6	Gradient Boosting Regressor	477.076369	-0.059091
7	Support Vector Regressor + GridSearch	0.183695	-0.001387





Development of a neural network to predict the matrix-filler ratio

```
# выводим архитектуру модели  
model.summary()
```

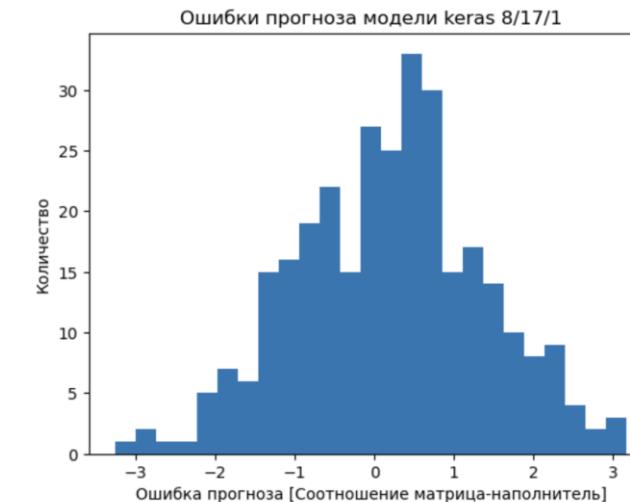
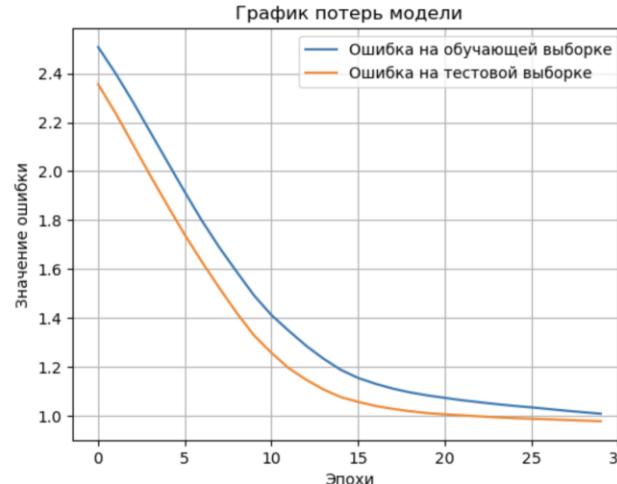
Model: "sequential"

Layer (type)	Output Shape	Param #
<hr/>		
normalization_1 (Normalization)	(None, 12)	25
dense (Dense)	(None, 8)	104
dense_1 (Dense)	(None, 17)	153
dense_2 (Dense)	(None, 1)	18
<hr/>		

Total params: 300

Trainable params: 275

Non-trainable params: 25



CPU times: user 2.25 s, sys: 467 ms, total: 2.72 s
Wall time: 1.55 s

Model Results:

Model_MAE: 1

Model_MAPE: 0.38

Test score: 1.44





Development of a neural network to predict the matrix-filler ratio

Model: "sequential_1"

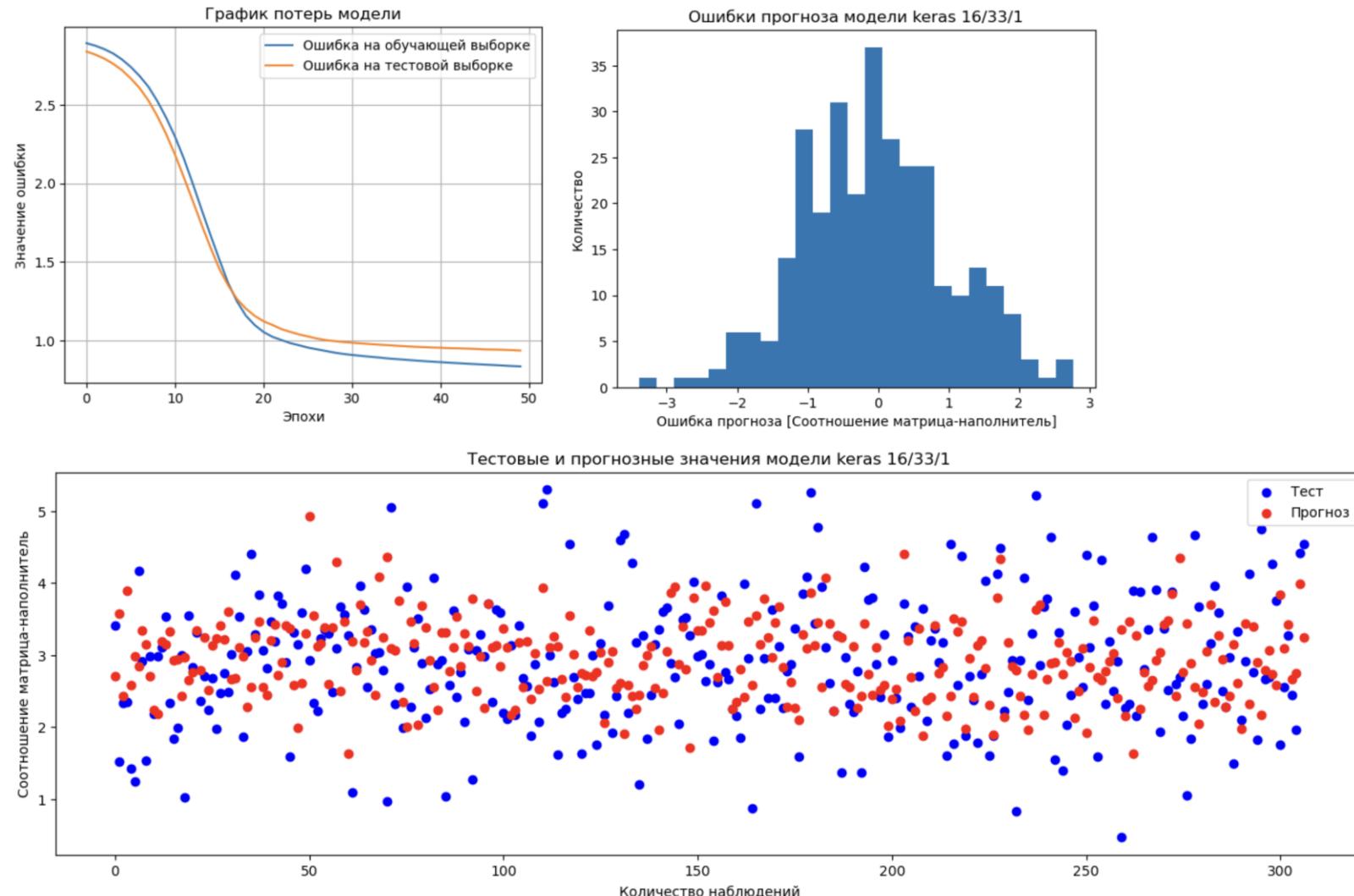
Layer (type)	Output Shape	Param #
<hr/>		
normalization_1 (Normalization)	(None, 12)	25
dense_3 (Dense)	(None, 16)	208
dense_4 (Dense)	(None, 33)	561
dense_5 (Dense)	(None, 1)	34
<hr/>		
Total params:	828	
Trainable params:	803	
Non-trainable params:	25	

Model Results:

Model_MAE: 1

Model_MAPE: 0.36

Test score: 1.10





Model: "sequential_3"

Layer (type)	Output Shape	Param #
<hr/>		
normalization_1 (Normalization)	(None, 12)	25
dense_9 (Dense)	(None, 16)	208
dropout_2 (Dropout)	(None, 16)	0
dense_10 (Dense)	(None, 33)	561
dropout_3 (Dropout)	(None, 33)	0
dense_11 (Dense)	(None, 1)	34
<hr/>		
Total params:	828	
Trainable params:	803	
Non-trainable params:	25	

Model Results:

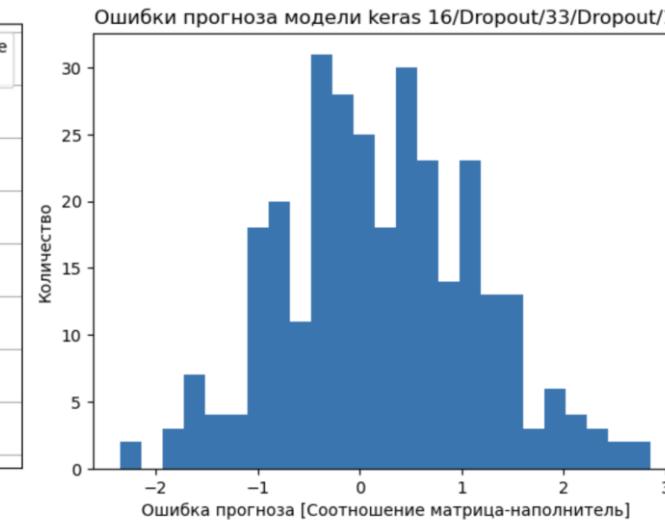
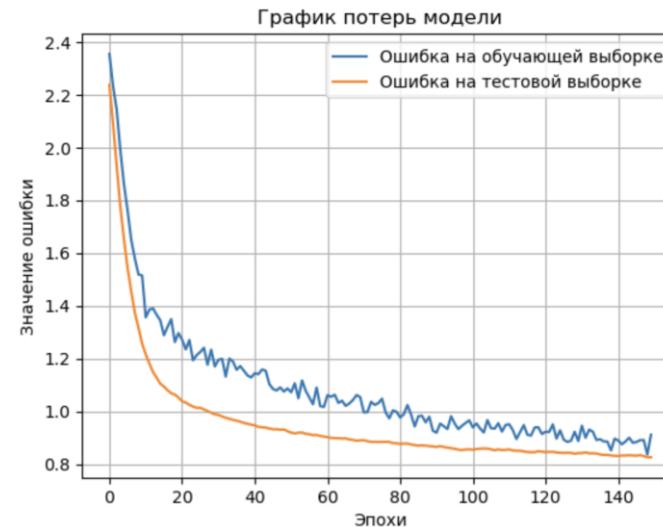
Model_MAE: 1

Model_MAPE: 0.31

Test score: 0.94

MAE [Соотношение матрица-наполнитель]	
model_8/17/1	0.962809
model_16/33/1	0.830088
model_16/Dropout/33/Dropout/1	0.773735
model_16/Dropout/33/Dropout/1_Adam_opt	0.739763

Development of a neural network to predict the matrix-filler ratio





Application development

app.py 1, U X

```
final_project > app.py > ...
1  import flask
2  from flask import render_template
3  import tensorflow as tf
4  from tensorflow import keras
5
6  app = Flask(__name__)
7
8  # загружаем модель и определяем параметры функции – будущие входы для модели (всего 12 параметров)
9
10 def set_params(param1, param2, param3, param4, param5, param6, param7, param8, param9, param10, param11, param12):
11
12     model = keras.models.load_model("/Users/kusita_1/Desktop/final_project/saved_model/composites_model_16/")
13     prediction = model.predict([param1, param2, param3, param4, param5, param6, param7, param8, param9, param10, param11, param12])
14
15     return prediction[0][0]
16
17 @app.route('/', methods=['post', 'get'])
18
19 def app_calculation():
20     param_list = []
21     message = ''
22     if flask.request.method == 'POST':
23
24         # получим данные из форм и добавим их в список, который затем передадим функции set_params
25         for i in range(1,13,1):
26             param = flask.request.form.get(f'param{i}')
27             param_list.append(float(param))
28
29         message = set_params(*param_list)
30
31     # указываем шаблон и прототип сайта для вывода
32     return render_template("/Users/kusita_1/Desktop/final_project/templates/main.html", message=message)
33
34 # запускаем приложение
35 app.run()
```

01//EN"

```
:text/html; charset=utf-8">
□#ffffff;"><span style="font-size: 12px;"><span style="font-family: Times New Roman,Times,serif;">
border="3" cellpadding="5" cellspacing="5" dir="ltr" width="500px">
:&font-family: Times New Roman,Times,serif;"><span style="color: □#ffffff;"><span style="color: □#000000;">
юолнитель&raquo;</span></span></span></th>
```

```
:ipx;"><span style="font-family: Times New Roman,Times,serif;"><span style="color: □#ffff00;">
```

```
:8B2" border="3" cellpadding="5" cellspacing="5" dir="ltr" width="500px">
20   <form action="" method="post">
21   <table align="center" bgcolor="darkgreen" border="3" bordercolor="#1D48B2" cellpadding="5" cellspacing="5" width="500px">
22     <tbody>
23       <tr>
```



GitHub repository

The screenshot shows a GitHub repository page for the user 'Ana7-7' named 'Composite-materials'. The repository contains files such as .gitignore, LICENSE, README.md, compositum_part1.ipynb (which is highlighted), compositum_part2.ipynb, and compositum_part3.ipynb. The main file, 'compositum_part1.ipynb', is displayed on the right. It has a size of 16.1 MB and was uploaded by Ana7-7. A note indicates that code can be 55% faster with GitHub Copilot. Below the file preview, there is a message in Russian about a project for the 'Data Science pro' course.

Files

- .gitignore
- LICENSE
- README.md
- compositum_part1.ipynb
- compositum_part2.ipynb
- compositum_part3.ipynb

Composite-materials / compositum_part1.ipynb

Ana7-7 Add files via upload

16.1 MB Code 55% faster with GitHub Copilot

Проект по курсу "Data Science pro"

слушатель: Анна Погорелова

Цель проекта: прогнозирование конечных свойств композиционных материалов. Задачи



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана



do.bmstu.ru

For further information, please,
connect
anna.pogorelova.md@gmail.co
m