ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
по курсу
«Data Science»

Тема: Прогнозирование конечных свойств новых материалов (композиционных материалов).

Слушатель: Самойличенко А.А.

Цели и задачи

1) Изучить теоретические основы и методы решения поставленной задачи.
2) Провести разведочный анализ предложенных данных. Необходимо нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек. Необходимо также для каждой колонке получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков.
3) Провести предобработку данных (удаление шумов, нормализация и т.д.).
4) Обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель.
5) Разработать приложение с графическим интерфейсом или интерфейсом командной строки, которое будет выдавать прогноз, полученный в задании 4 или 5.
6) Оценить точность модели на тренировочном и тестовом датасете.
7) Создать репозиторий в GitHub / GitLab и разместить там код исследования.

Разведочный анализ данных

Объединённый датасет по индексу и типу объединения INNER после удаления неинформативного столбца 'Unnamed: 0‘ состоит из 13 столбцов и 1023 строк.

•Дубликаты и пропущенные значения в датасете отсутствуют.
•Признак “Угол нашивки, град” представлен двумя значениями.

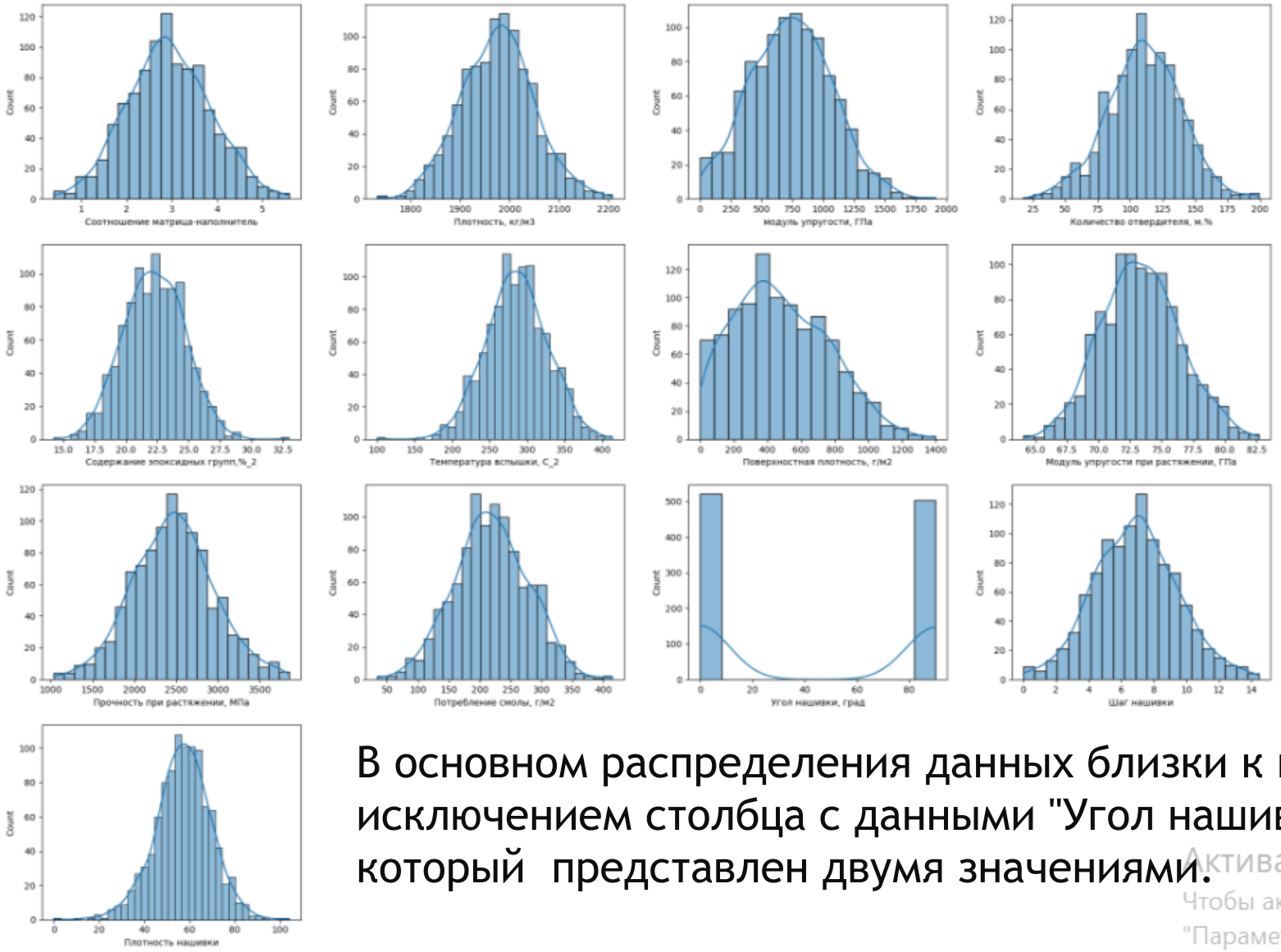
Целевые переменные:

- Модуль упругости при растяжении, Гпа
- Прочность при растяжении, МПа
- Соотношение матрица-наполнитель

Описательная статистика

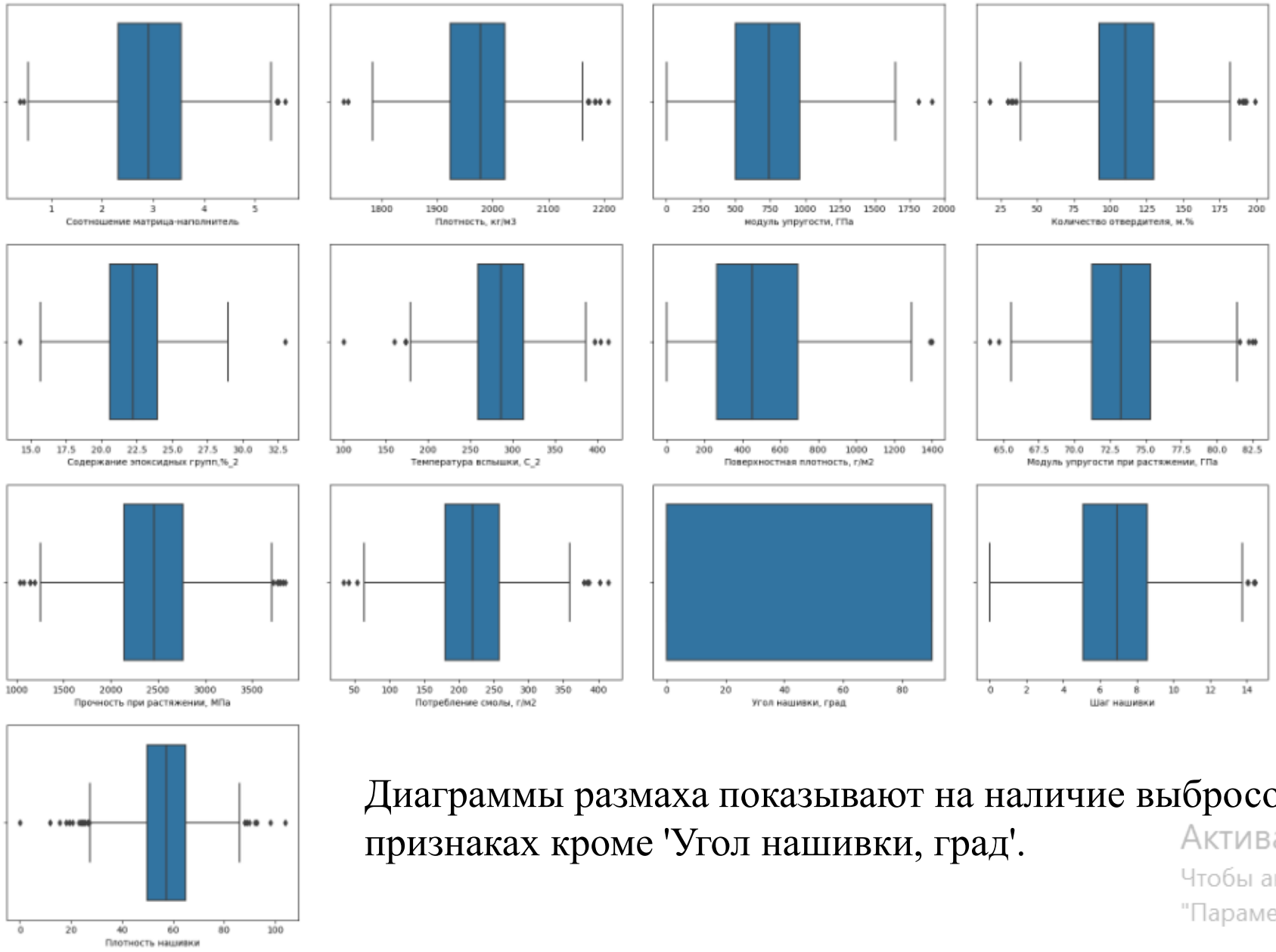
	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660	5.591742
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375	2207.773481
модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	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

Гистограммы распределения для каждой переменной



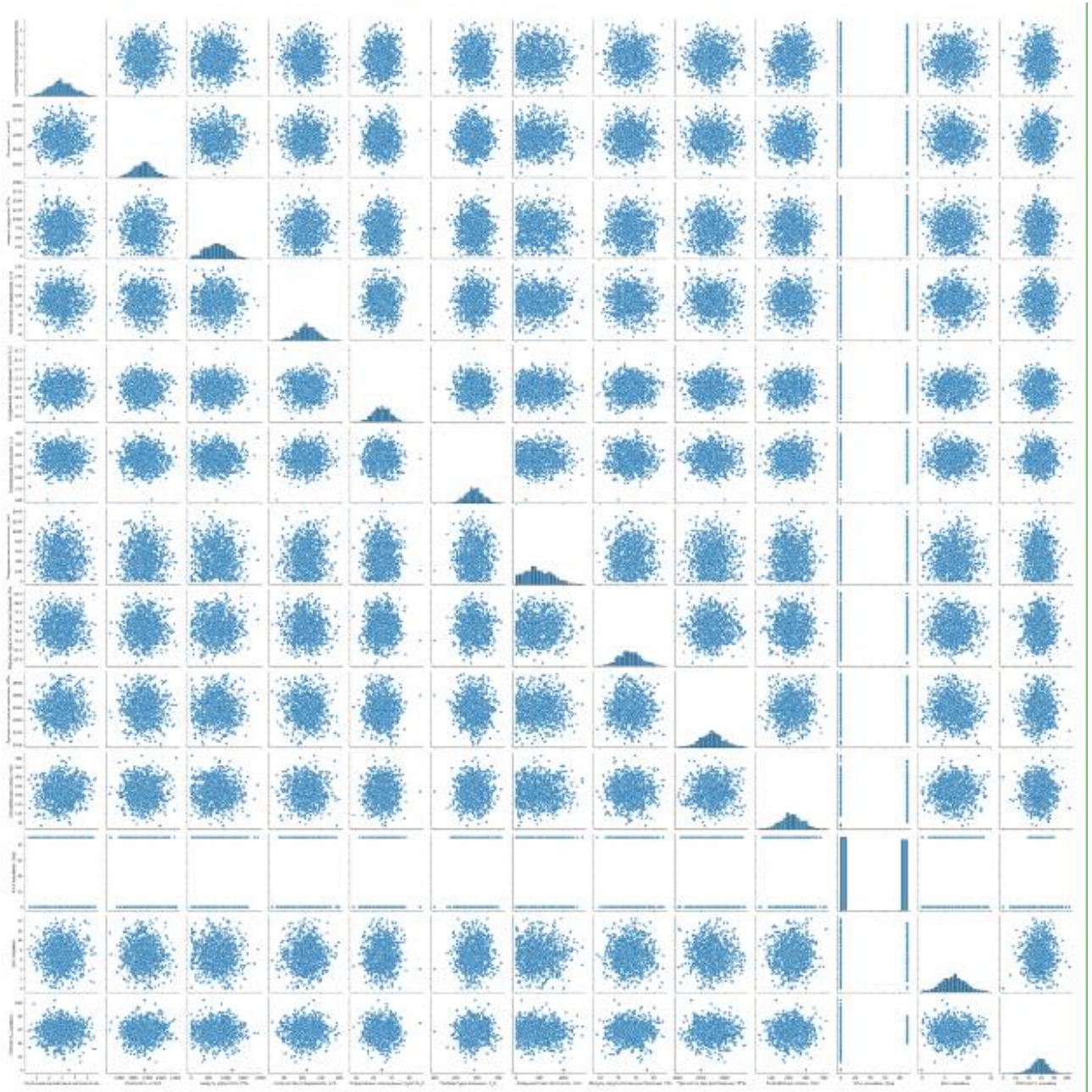
В основном распределения данных близки к нормальному, за исключением столбца с данными "Угол нашивки, град«, который представлен двумя значениями.

Диаграммы размаха для каждой переменной



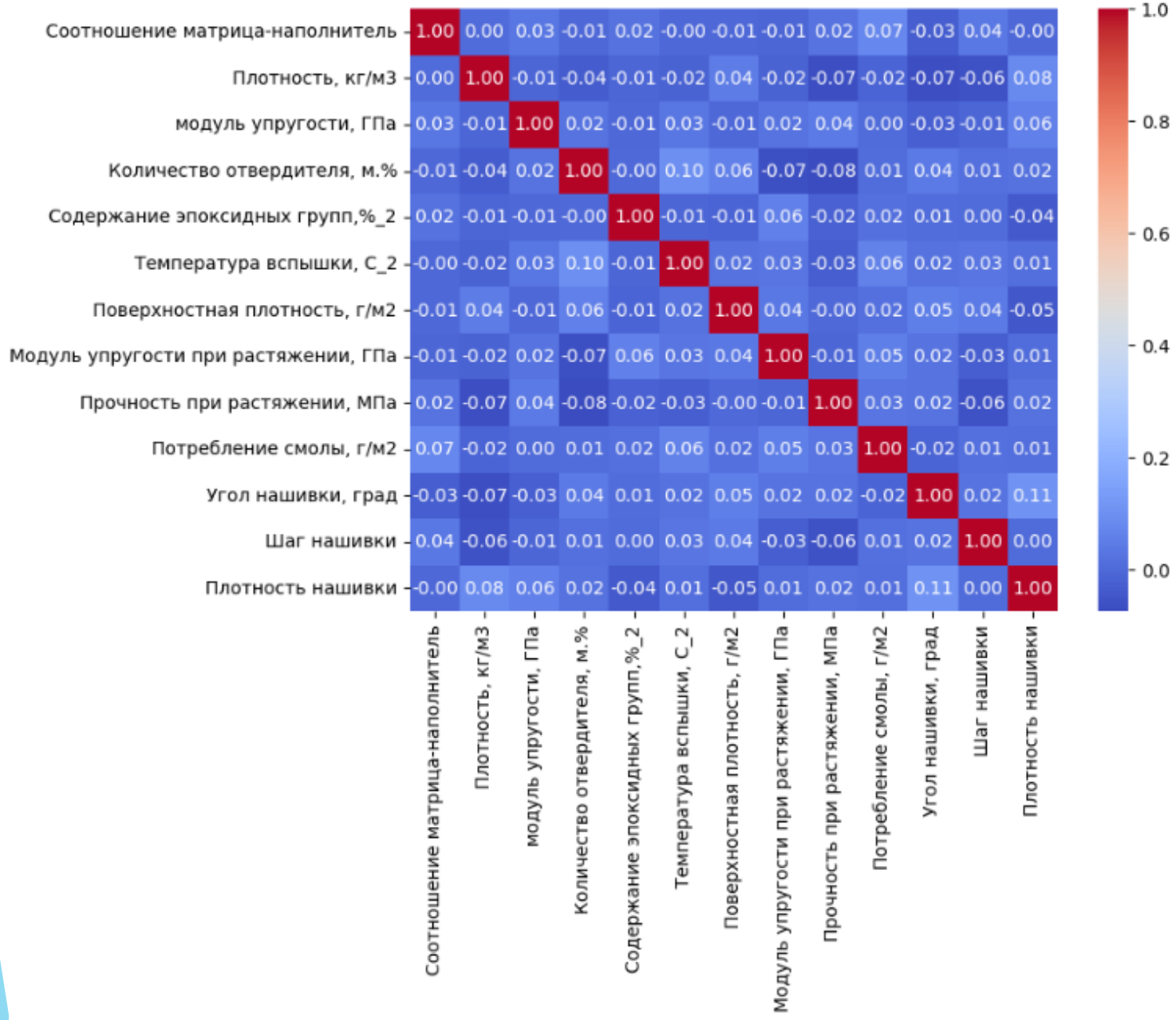
Диаграммы размаха показывают на наличие выбросов во всех признаках кроме 'Угол нашивки, град'.

Попарные графики рассеяния



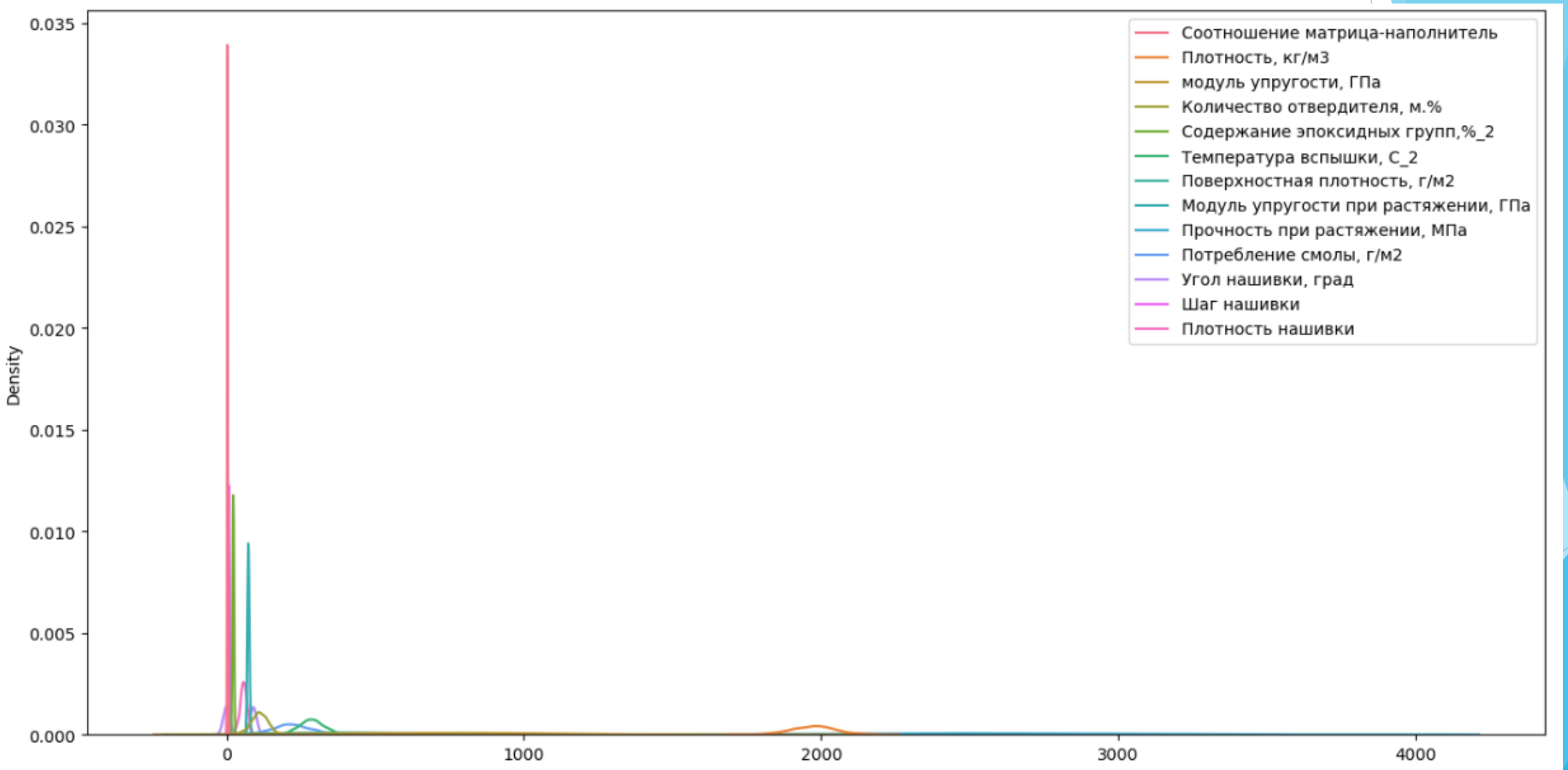
Попарные графики рассеяния указывают на наличие выбросов и отсутствие зависимости между двумя наборами данных.

Тепловая карта



Тепловая карта показывает что корреляции между признаками практически не наблюдается.

График оценки плотности

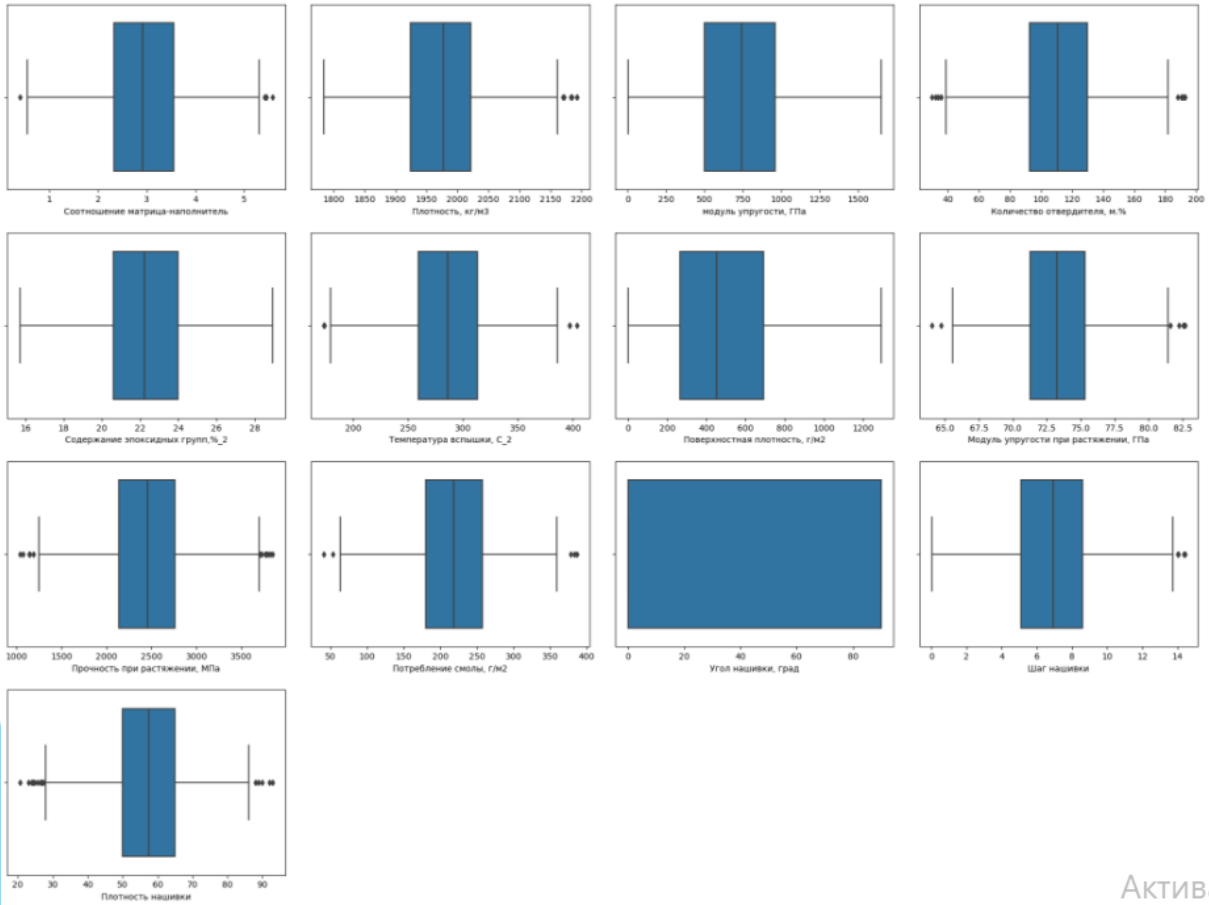


Оценка плотности ядра показывает что значения переменных находятся в разных диапазонах поэтому требуется нормализация данных.

Удаление выбросов

Удаляем выбросы в датасете с помощью метода трёх сигм.

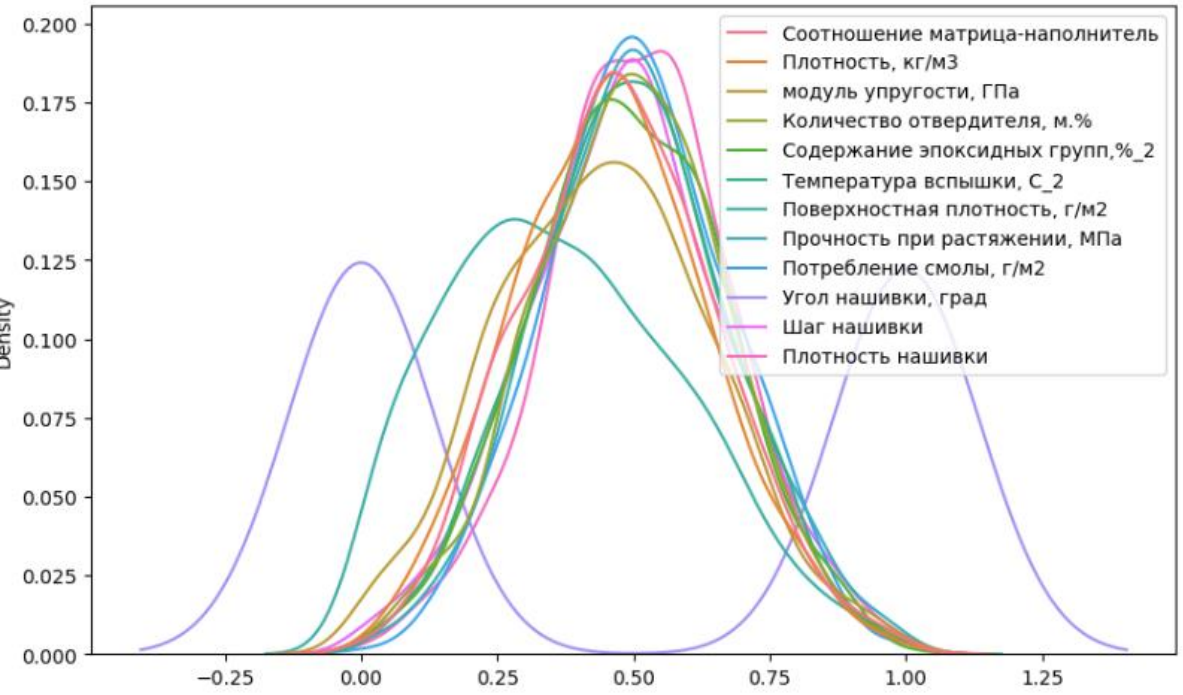
После удаления выбросов размерность датасета составляет 1000 строк и 13 признаков .



Диаграммы размаха после удаления выбросов

Нормализация данных

График оценки плотности после нормализации



Описательная статистика после нормализации

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	700.0	0.475080	0.178852	0.0	0.354198	0.470486	0.594725	1.0
Плотность, кг/м3	700.0	0.460015	0.179172	0.0	0.331535	0.456148	0.577349	1.0
модуль упругости, ГПа	700.0	0.446995	0.198017	0.0	0.300911	0.447128	0.578954	1.0
Количество отвердителя, м.%	700.0	0.492974	0.176204	0.0	0.371904	0.493143	0.612012	1.0
Содержание эпоксидных групп,%_2	700.0	0.490718	0.176745	0.0	0.371371	0.488621	0.619951	1.0
Температура вспышки, С_2	700.0	0.489708	0.176965	0.0	0.368951	0.489479	0.608334	1.0
Поверхностная плотность, г/м2	700.0	0.373061	0.216199	0.0	0.206143	0.358842	0.532634	1.0
Прочность при растяжении, МПа	700.0	0.503359	0.177701	0.0	0.383379	0.498630	0.616059	1.0
Потребление смолы, г/м2	700.0	0.508277	0.168013	0.0	0.393195	0.507419	0.619798	1.0
Угол нашивки, град	700.0	0.495714	0.500339	0.0	0.000000	0.000000	1.000000	1.0
Шаг нашивки	700.0	0.488922	0.185128	0.0	0.364410	0.490795	0.610799	1.0
Плотность нашивки	700.0	0.508781	0.166270	0.0	0.401573	0.507366	0.615928	1.0

Нормализация данных сделана с помощью метода MinMaxScaler

Создание модели машинного обучения для предсказания - модуля упругости при растяжении и прочности при растяжении.

Для прогнозирования модуля упругости при растяжении и прочности при растяжении были использованы следующие методы машинного обучения:

- Лассо-регрессия (Lasso);
- Линейная регрессия (LinearRegression);
- Гребневая регрессия (Ridge);
- Регрессионное дерево решений (DecisionTreeRegressor);
- Градиентный бустинг регрессии (GradientBoostingRegressor);
- Случайный лес регрессии (RandomForestRegressor);
- Эластичная сеть регрессии (ElasticNetCV);
- Метод опорных векторов для регрессии (SVR);
- Байесовская линейная регрессия (BayesianRidge);
- Ядерная регрессия (KernelRidge).

Лучшие гиперпараметры для модели были подобранны методом GridSearchCV из библиотеки sklearn

Анализ результатов работы моделей машинного обучения для предсказания - модуля упругости при растяжении

	R2	RMSE	MAE
Lasso	-0.021946	3.104460	2.512111
LinearRegression	-0.042239	3.135130	2.542403
Ridge	-0.036315	3.126208	2.533229
DecisionTreeRegressor	-0.065842	3.170432	2.551608
GradientBoostingRegressor	-0.124855	3.257019	2.596898
RandomForestRegressor	-0.058812	3.159959	2.527859
ElasticNetCV	-0.021981	3.104512	2.510919
SVR	-0.062883	3.166028	2.556054
BayesianRidge	-0.021522	3.103815	2.510063
KernelRidge	-0.031442	3.118849	2.524650

Метрики для оценки качества работы модели:

- R2 (коэффициент детерминации)
- RMSE (среднеквадратичная ошибка)
- MAE (средняя абсолютная ошибка)

```
# BayesianRidge
br_model = BayesianRidge(alpha_1 = 2.0, alpha_2 = 0.01, lambda_1 = 0.01, lambda_2 = 1)
br_model.fit(X_el_train_norm, y_elastic_train)
y_pred_br = br_model.predict(X_el_test_norm)
```

Все модели показали неудовлетворительные результаты, значения коэффициента детерминации находятся около нуля. Лучшие показатели на тестовой выборке с подобранными гиперпараметрами показывает модель BayesianRidge (Байесовская линейная регрессия).

Анализ результатов работы моделей машинного обучения для предсказания - прочности при растяжении

	R2	RMSE	MAE
Lasso	-0.019856	469.004096	365.853564
LinearRegression	-0.041048	473.851843	370.003339
Ridge	-0.034649	472.393187	368.940852
DecisionTreeRegressor	-0.025416	470.280839	369.984701
GradientBoostingRegressor	-0.032582	471.921119	371.712346
RandomForestRegressor	-0.037358	473.011207	372.443349
ElasticNetCV	-0.000928	464.631542	363.886617
SVR	-0.049553	475.783433	368.535000
BayesianRidge	-0.000928	464.631544	363.886619
KernelRidge	-0.010346	466.812301	364.936087

Метрики для оценки качества работы модели:

- R2 (коэффициент детерминации)
- RMSE (среднеквадратичная ошибка)
- MAE (средняя абсолютная ошибка)

```
# ElasticNetCV
en_model2 = ElasticNetCV(l1_ratio = 0.01)
en_model2.fit(X_strength_train_norm, y_strength_train)
y_pred_en2 = en_model2.predict(X_strength_test_norm)
```

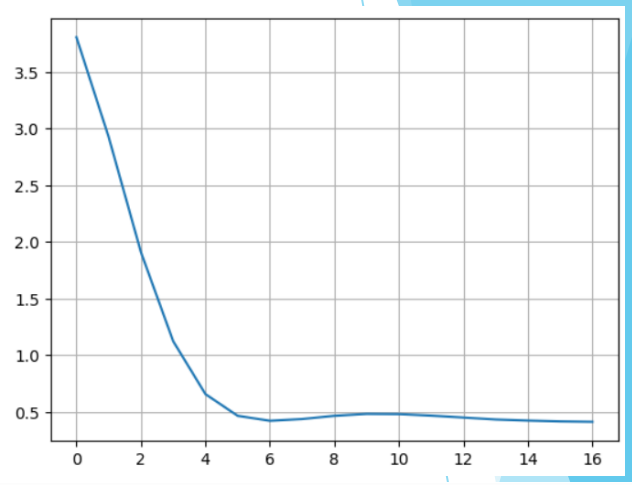
```
# BayesianRidge
br_model2 = BayesianRidge(alpha_1 = 0.01, alpha_2 = 2.0, lambda_1 = 1, lambda_2 = 0.01)
br_model2.fit(X_strength_train_norm, y_strength_train)
y_pred_br2 = br_model2.predict(X_strength_test_norm)
```

Все модели показали неудовлетворительные результаты, значения коэффициента детерминации находятся около нуля. Лучшие показатели на тестовой выборке с подобранными гиперпараметрами показывает модель BayesianRidge и ElasticNetCV.

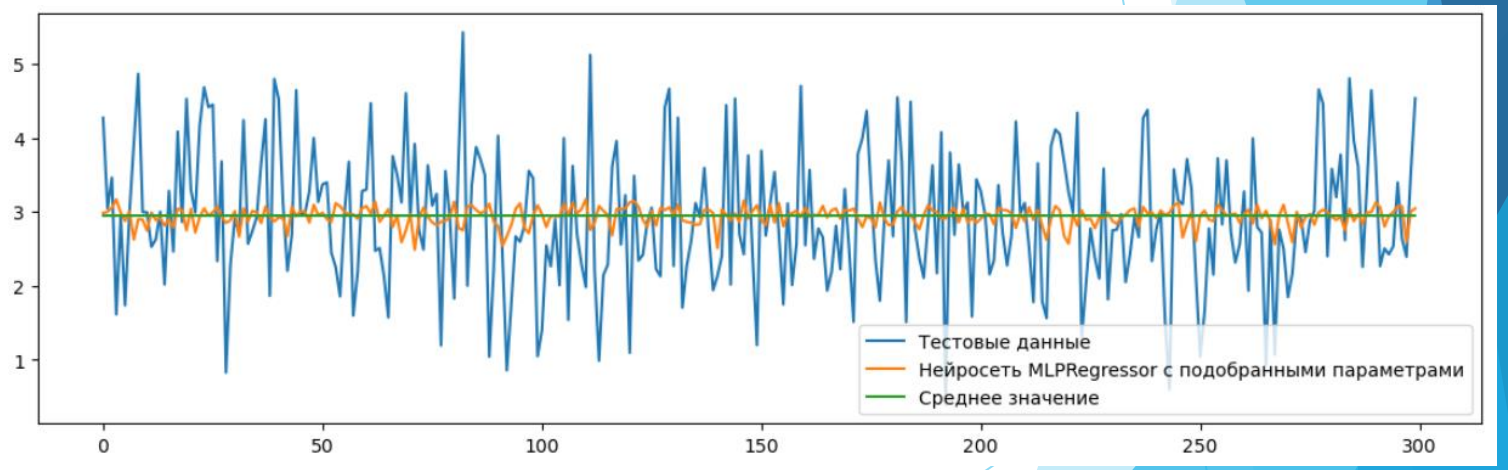
Нейронная сеть MLPRegressor для прогнозирования соотношения матрица-наполнитель.

```
# Воспользуемся GridSearchCV из библиотеки sklearn для подбора лучших параметров для MLPRegressor.
param_list = {"hidden_layer_sizes": [(12, 12, 12), (8, 12), (24, 24), (24, 24, 24), (8, 8), (16, 16)],
              "activation": ["identity", "logistic", "tanh", "relu"],
              "solver": ["lbfgs", "sgd", "adam"],
              "alpha": [0.00005, 0.0001, 0.0005, 0.005, 0.05],
              }
model_matrix = MLPRegressor()

grid_search_matrix2 = GridSearchCV(model_matrix, param_list, cv=5, verbose=1, n_jobs=-1)
grid_search_matrix2.fit(X_matrix_train_norm.values, y_matrix_train)
best_params = grid_search_matrix2.best_params_
best_score = grid_search_matrix2.best_score_
print(f'Best params: {best_params}')
print(f'Best score: {best_score}')
```

График функции потерь




```
# Настраиваем нейросеть по параметрам GridSearchCV
model_matrix3_1 = MLPRegressor(
    hidden_layer_sizes = (12, 12, 12),
    activation = 'tanh',
    solver='sgd',
    alpha = 0.005,
    max_iter=500,
    early_stopping = True,
    validation_fraction = 0.3,
    random_state=42,
    verbose=True)
```



Коэффициент детерминации (R2) для MLPRegressor с подобранными параметрами: -0.019444

Среднеквадратичная ошибка для (RMSE) MLPRegressor с подобранными параметрами: 0.932634

Нейронная сеть MLPRegressor из библиотеки sklearn показала неудовлетворительный результат, коэффициент детерминации показывает около нулевые значения.

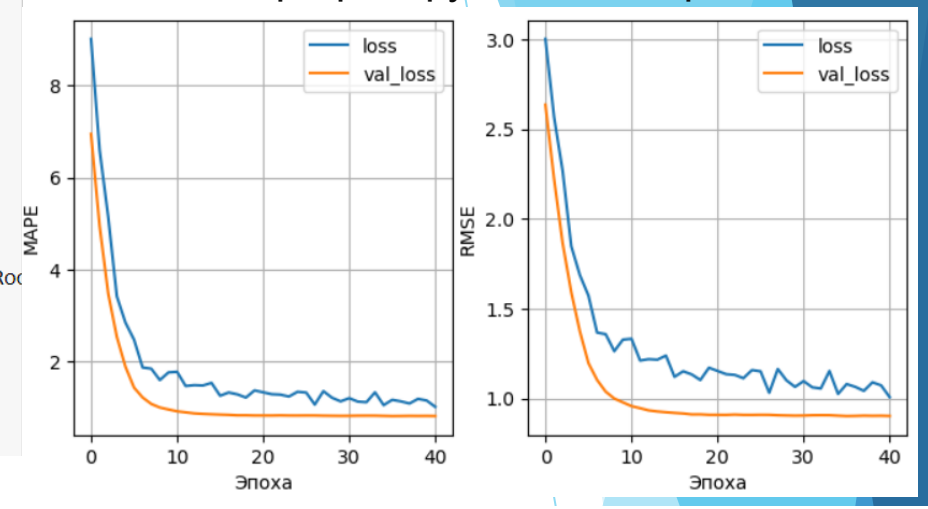
Нейронная сеть для прогнозирования соотношения матрица-наполнитель из библиотеки tensorflow.

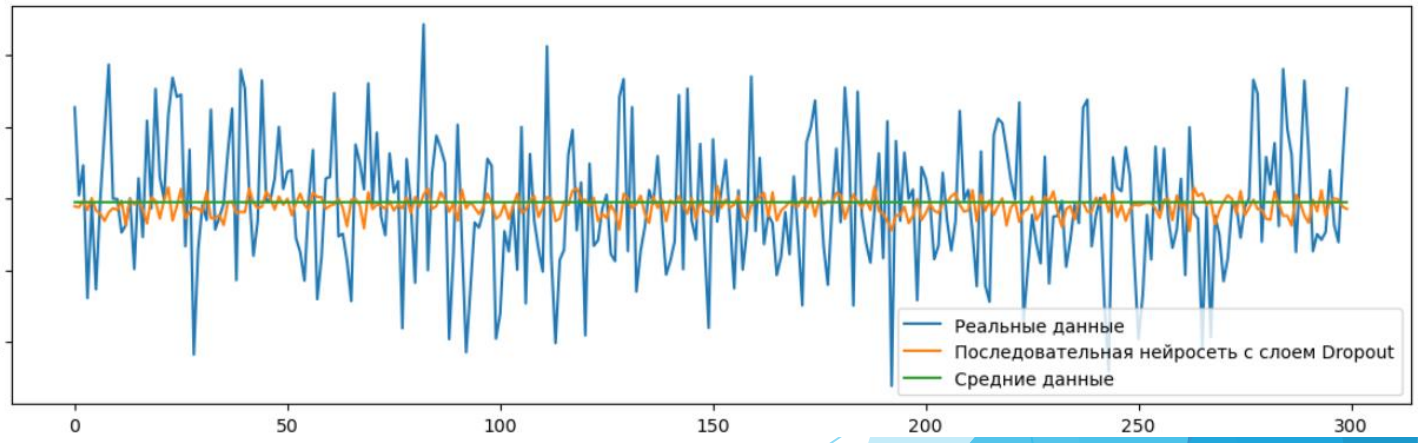
```
# Создадим последовательную модель с слоем Dropout и callback.
model_matrix_tf3 = tf.keras.Sequential([X_matrix_train_n_k, layers.Dense(8, activation='tanh'),
                                                            layers.Dropout(0.2),
                                                            layers.Dense(16, activation='tanh'),
                                                            layers.Dropout(0.3),
                                                            layers.Dense(24, activation='tanh'),
                                                            layers.Dropout(0.5),
                                                            layers.Dense(1)
                                        ])
# Компиляция модели с оптимизатором, функцией потерь и метриками
model_matrix_tf3.compile(optimizer = tf.keras.optimizers.SGD(0.005), loss = 'mean_squared_error', metrics = [tf.keras.metrics.Roo
# Создание callback-функции для остановки обучения, если значение потерь на валидационных данных не улучшится в течение 5 эпох

early_stopping_callback = keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)

# Посмотрим на архитектуру модели
model_matrix_tf3.summary()
```

```
%%time
#Обучение нейросети
history = model_matrix_tf3.fit(
  X_matrix_test,
  y_matrix_train,
  epochs=100,
  validation_split=0.3,
  verbose=1,
  callbacks=[early_stopping_callback])
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 normalization (Normalizatio  (None, 12)               25
 n)

 dense_12 (Dense)            (None, 8)                 104

 dropout_3 (Dropout)         (None, 8)                 0

 dense_13 (Dense)            (None, 16)                144

 dropout_4 (Dropout)         (None, 16)                0

 dense_14 (Dense)            (None, 24)                408

 dropout_5 (Dropout)         (None, 24)                0

 dense_15 (Dense)            (None, 1)                 25

=================================================================
Total params: 706
Trainable params: 681
Non-trainable params: 25
```

График функции потерь



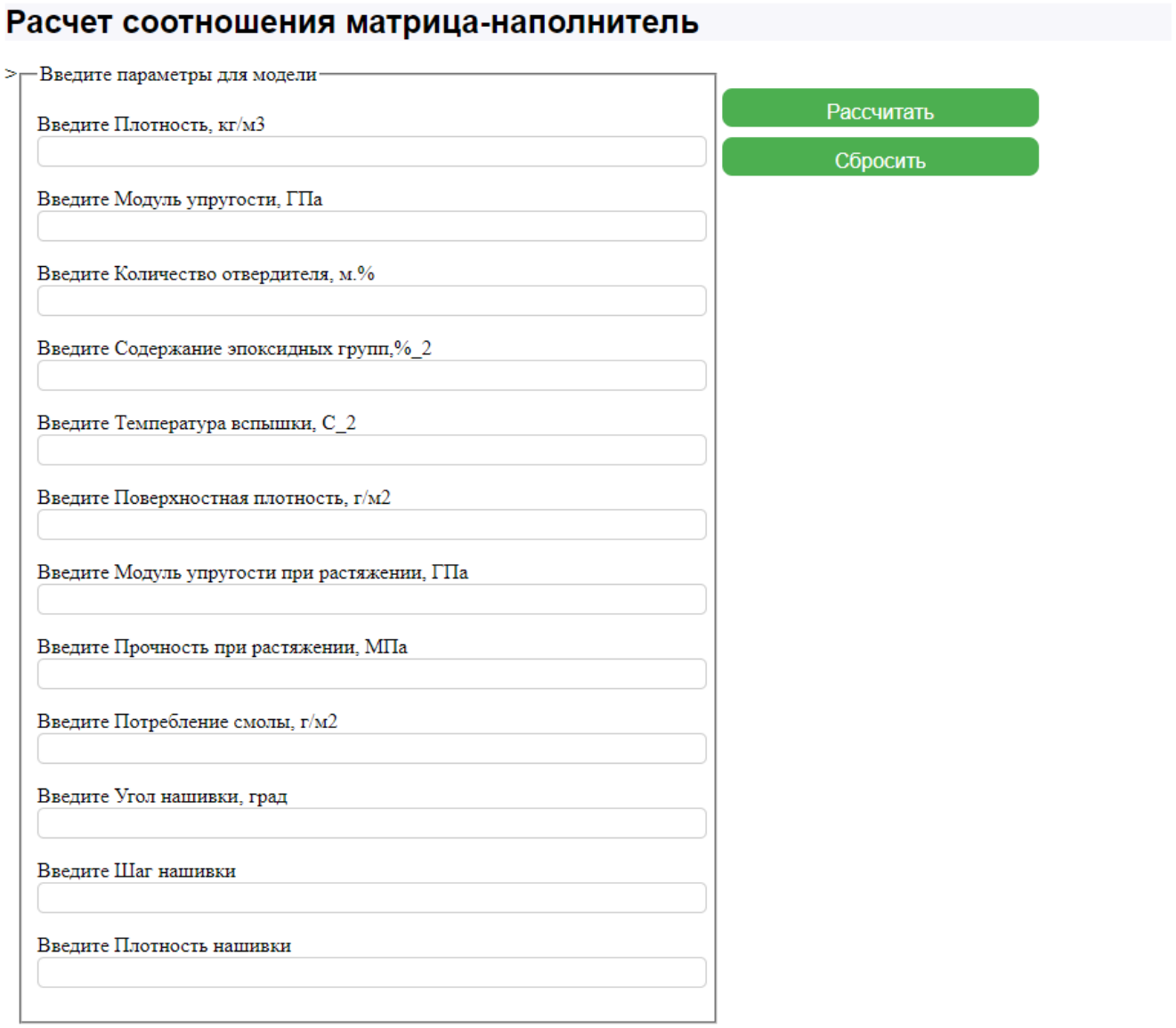


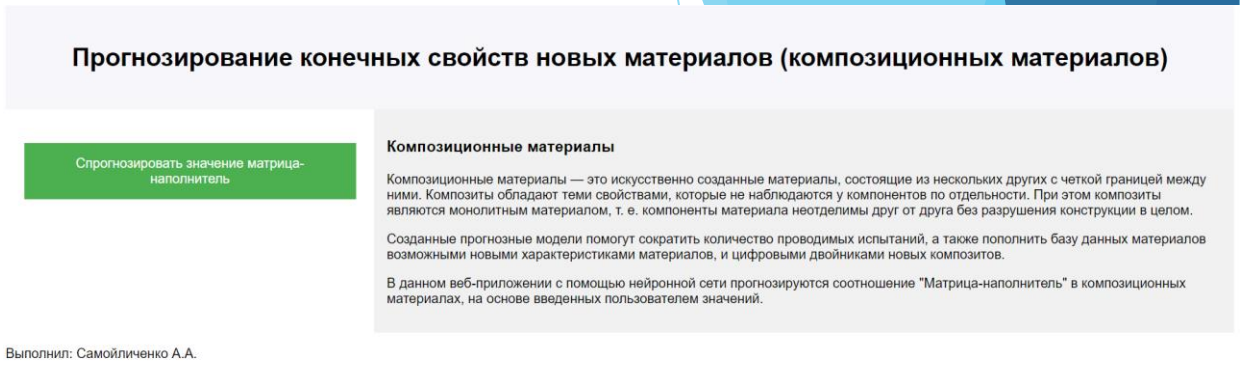
Коэффициент детерминации для последовательной неиросети со слоем Dropout и ранней остановкой: -0.027137

Среднеквадратичная ошибка для последовательной неиросети со слоем Dropout и ранней остановкой: 0.936147

Последовательная нейронная сеть из библиотеки tensorflow показала неудовлетворительный результат, коэффициент детерминации показывает около нулевые значения.

Разработка Web-приложения





Ссылка на GitHub:

https://github.com/andreigh89/vkr

Спасибо за внимание!