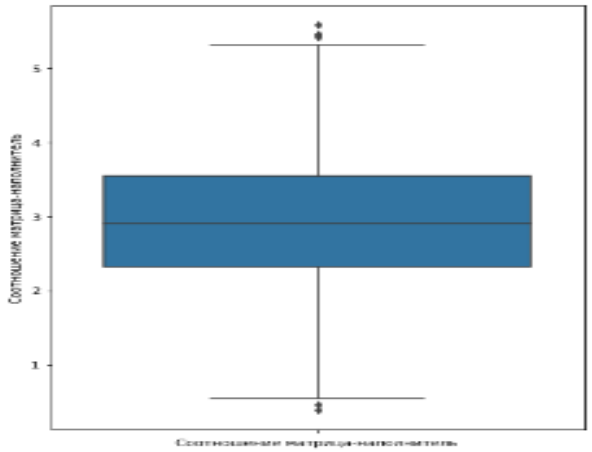
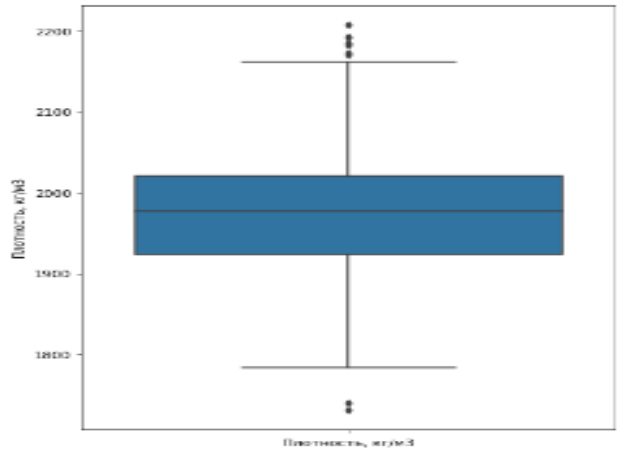
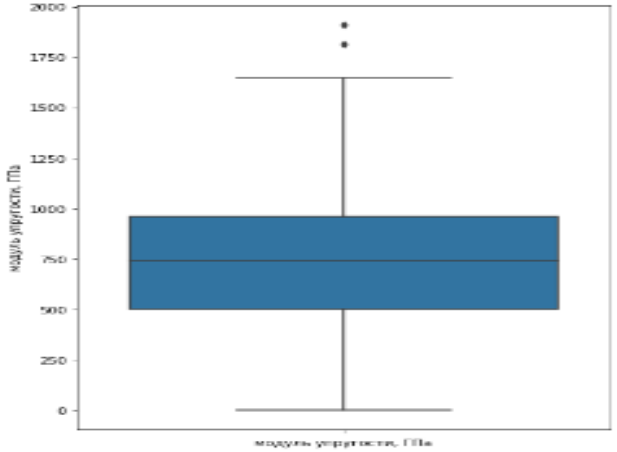
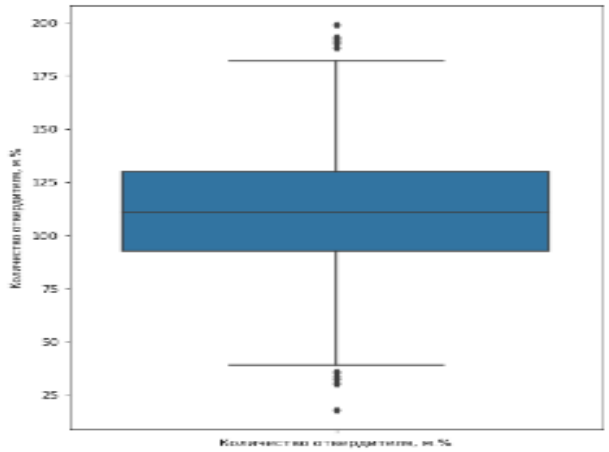
	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660	5.591742
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375	2207.773481
модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	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
ть нашивки	1023.0	57.153929	12.350969	0.000000	49.799212	57.341920	64.944961	103.988901

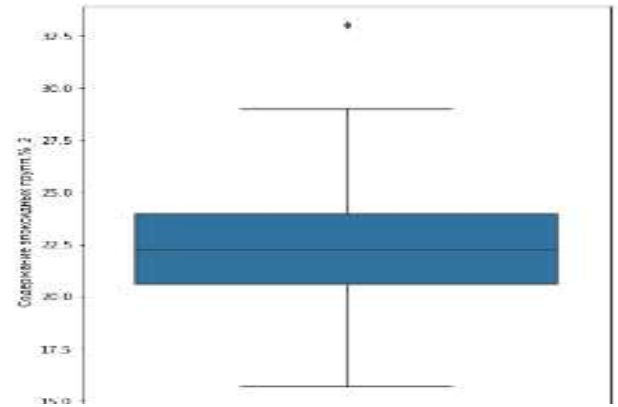
oll output; double click to hide

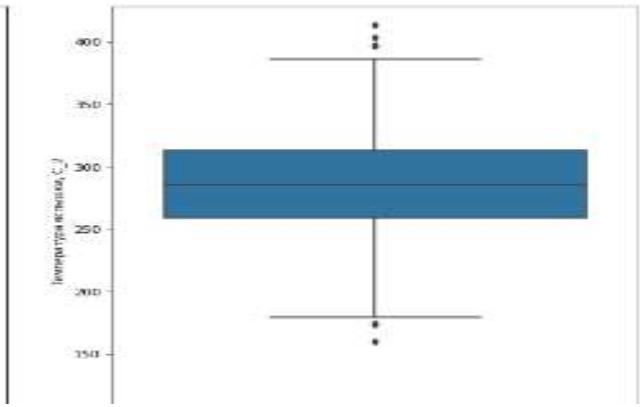

Соотношение матрица-наполнитель
Соотношение матрица-наполнитель

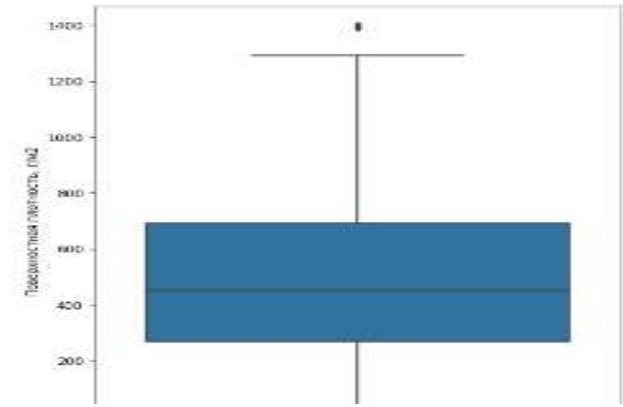

Плотность, кг/м3
Плотность, кг/м3

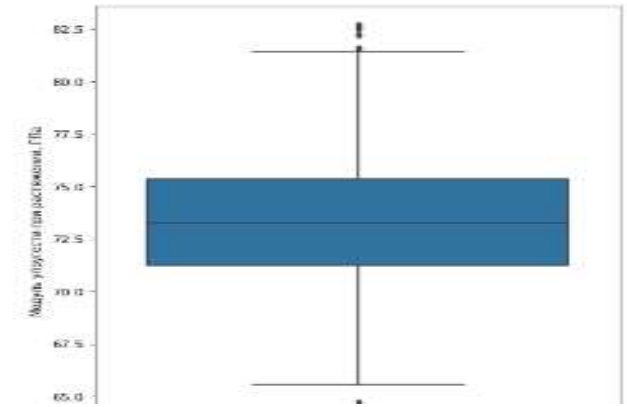

модуль упругости, ГПа
модуль упругости, ГПа

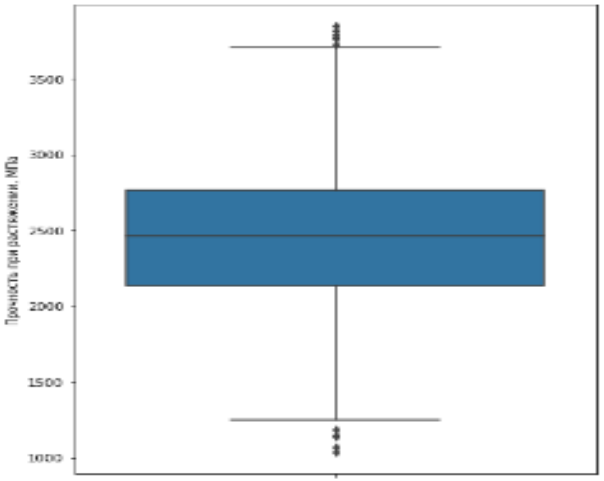

Количество отвердителя, м %
Количество отвердителя, м %

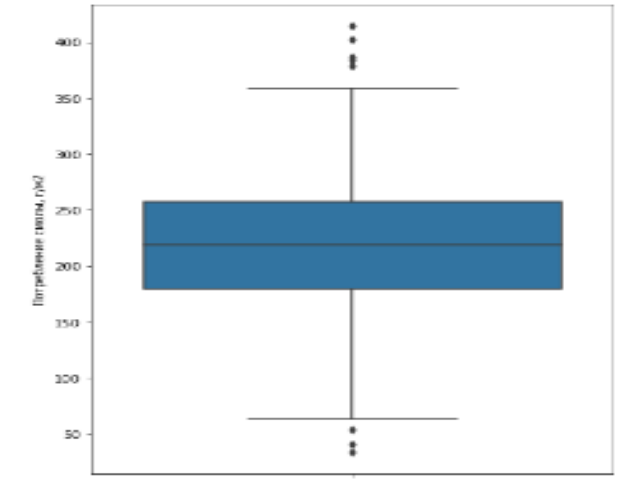

Содержание эпоксидных групп,% 2

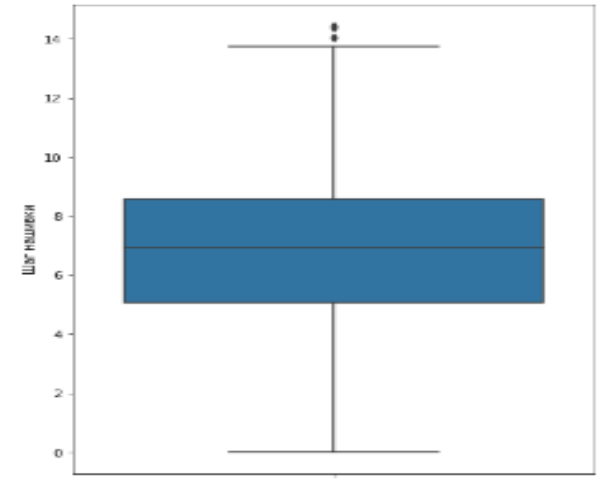

Температура вспышки, С

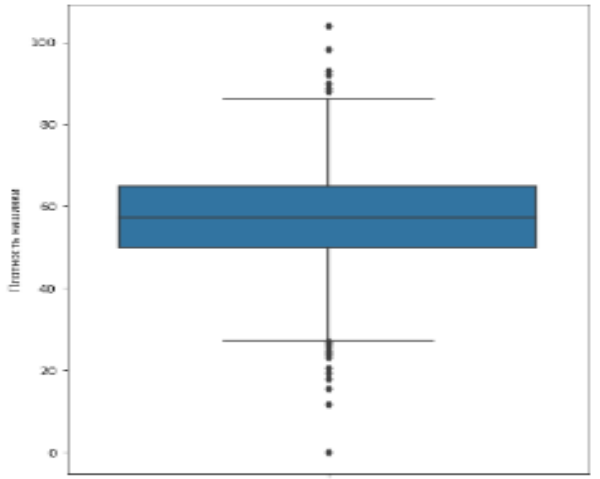

Поверхностная плотность, г/м2

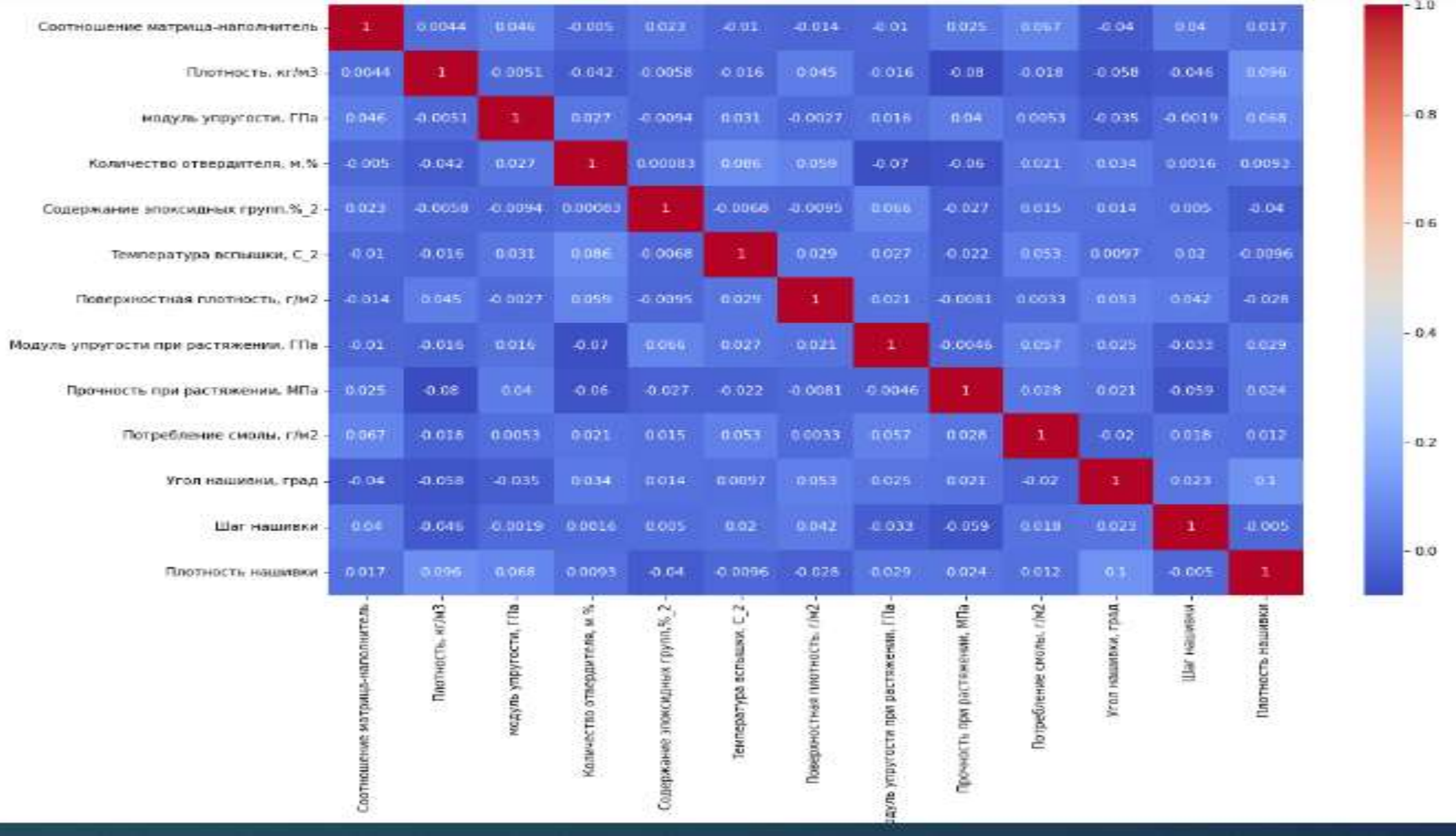

Модуль упругости при растяжении, ГПа

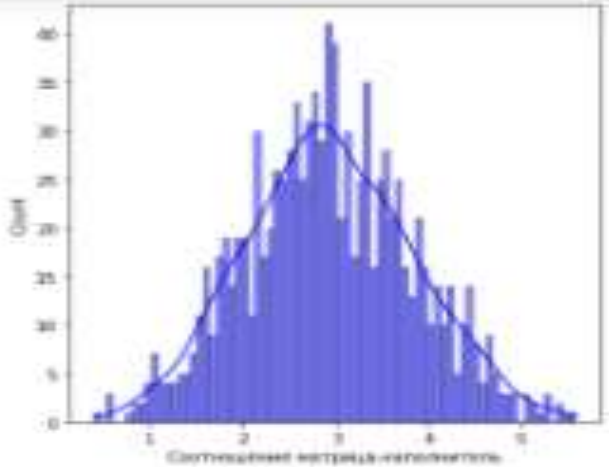

Прочность при растяжении, МПа
Прочность при растяжении, МПа

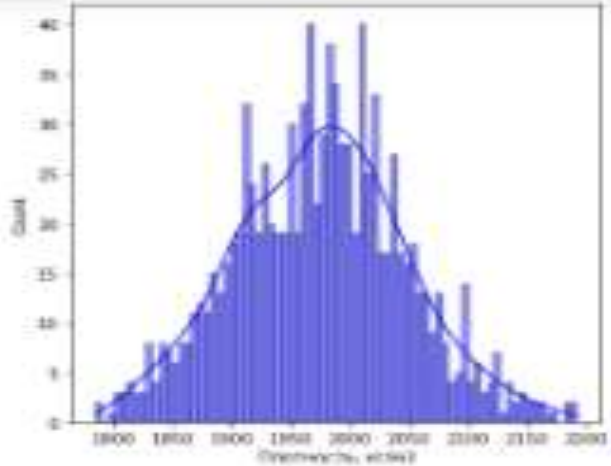

Потребление смолы, г/м2
Потребление смолы, г/м2

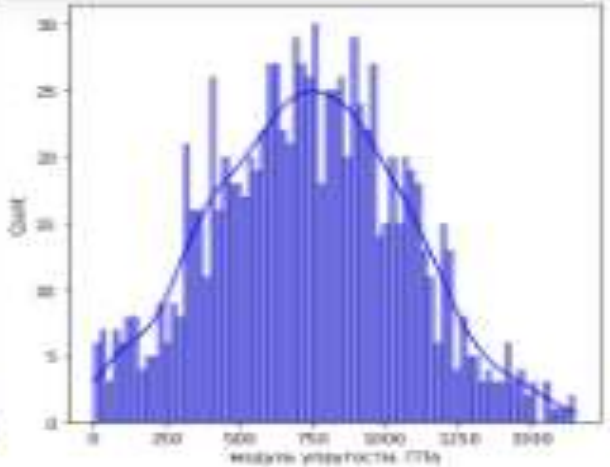

Шаг нашивки
Шаг нашивки

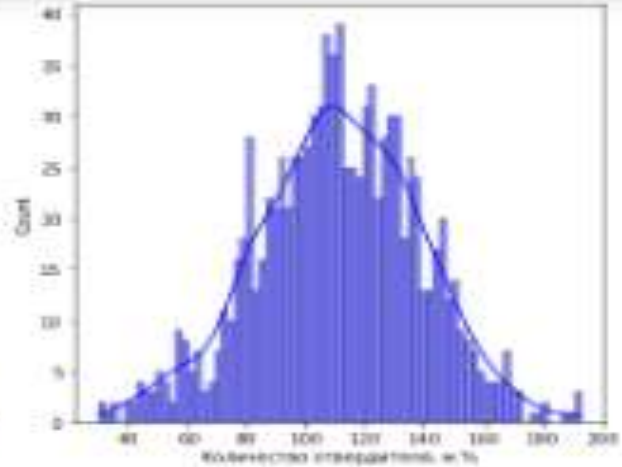

Плотность нашивки
Плотность нашивки

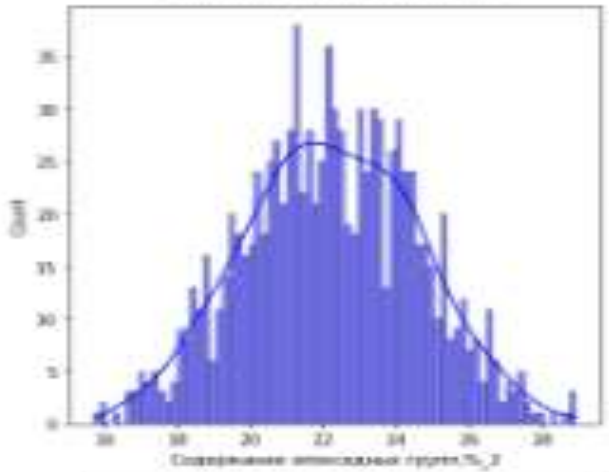

Соотношение матрица-наполнитель
Плотность, кг/м3
модуль упругости, ГПа
Количество отвердителя, м.%
Содержание эпоксидных групп,%_2
Температура вспышки, С_2
Поверхностная плотность, г/м2
Модуль упругости при растяжении, ГПа
Прочность при растяжении, МПа
Потребление смолы, г/м2
Угол нашивки, град
Шаг нашивки
Плотность нашивки
1.0
0.8
0.6
0.4
0.2
0.0

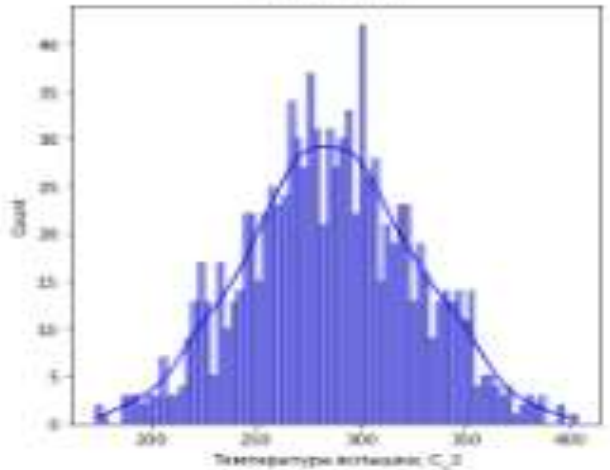

Count
Соотношение матрица-наполнитель

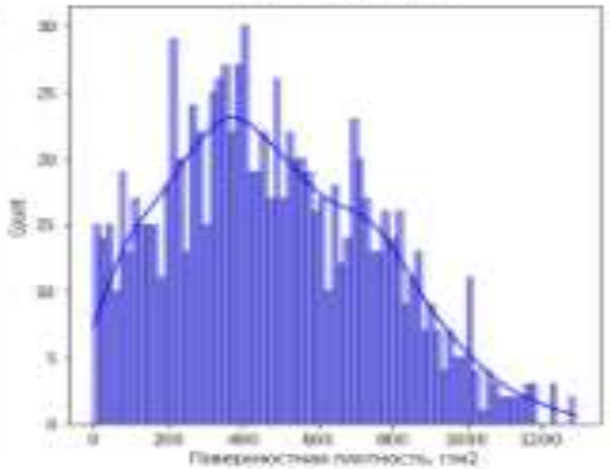

Count

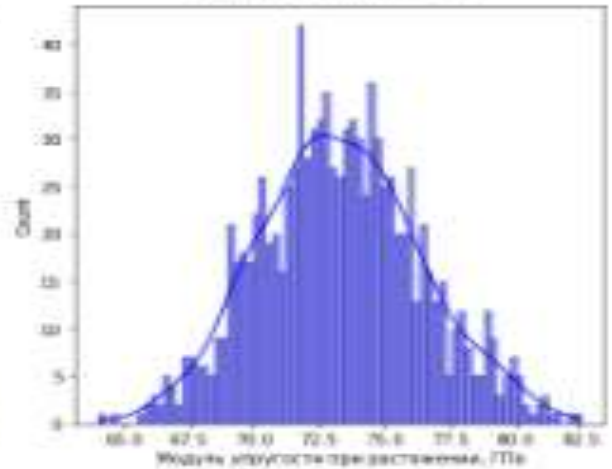

Count
Модуль упругости, ГПа

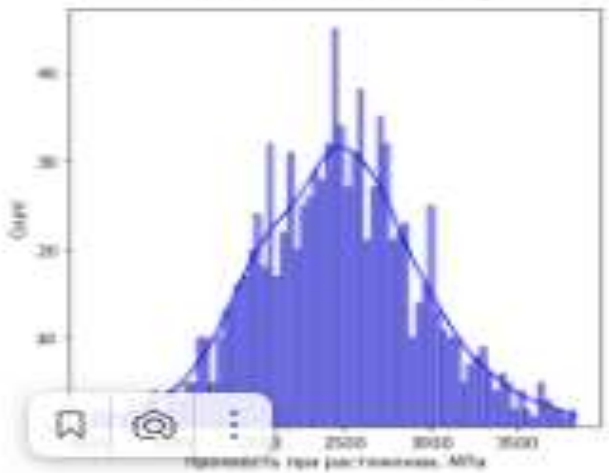

Count
Количество отвердителя, м.%

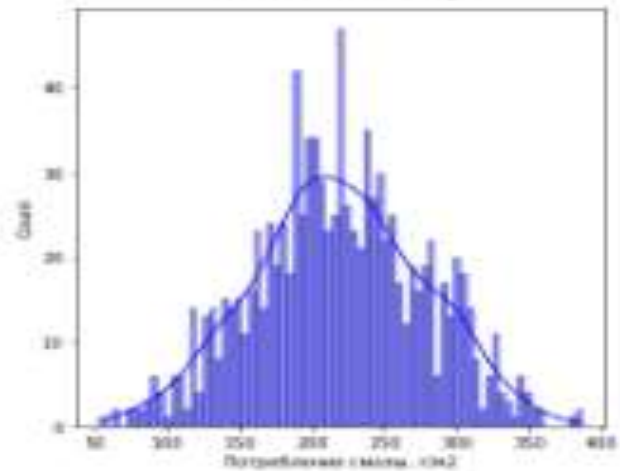

Count
Содержание эпоксидных групп,%_2

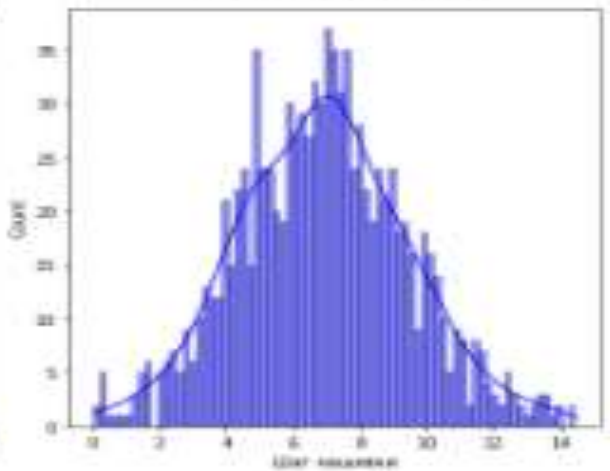

Count
Температура вспышки, С_2

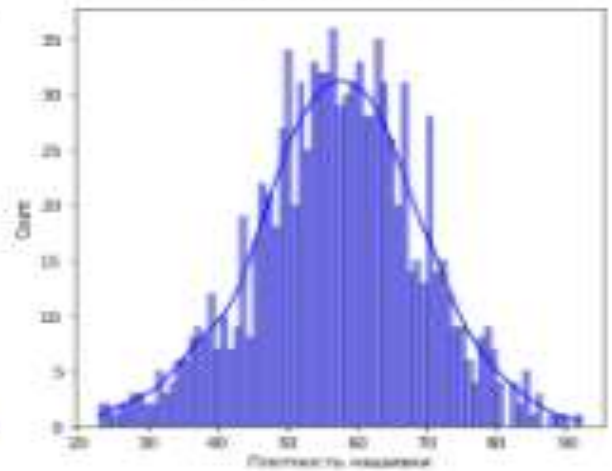

Count
Поверхностная плотность, г/м2

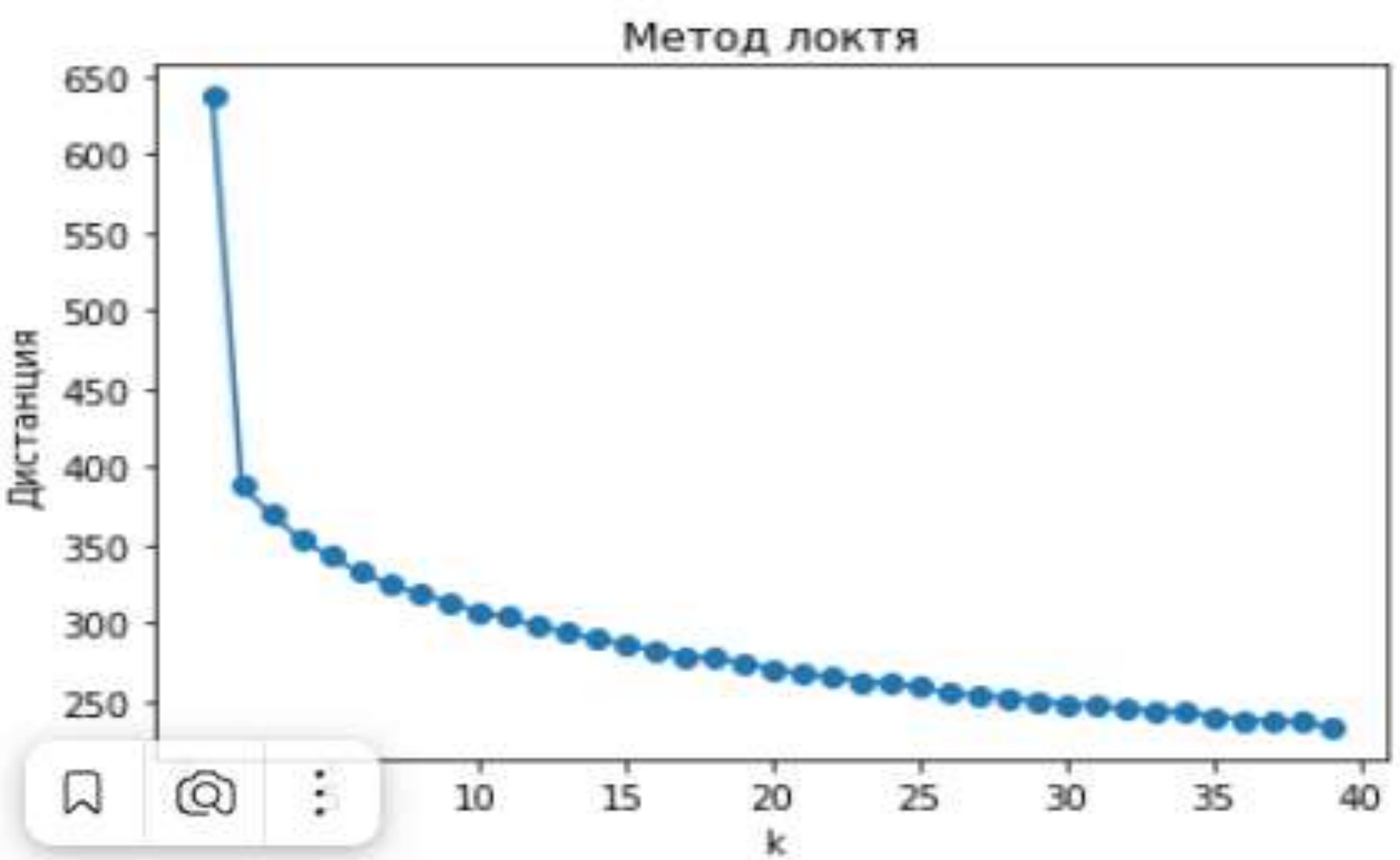

Count
Модуль упругости при растяжении, ГПа


Count
Прочность при растяжении, МПа

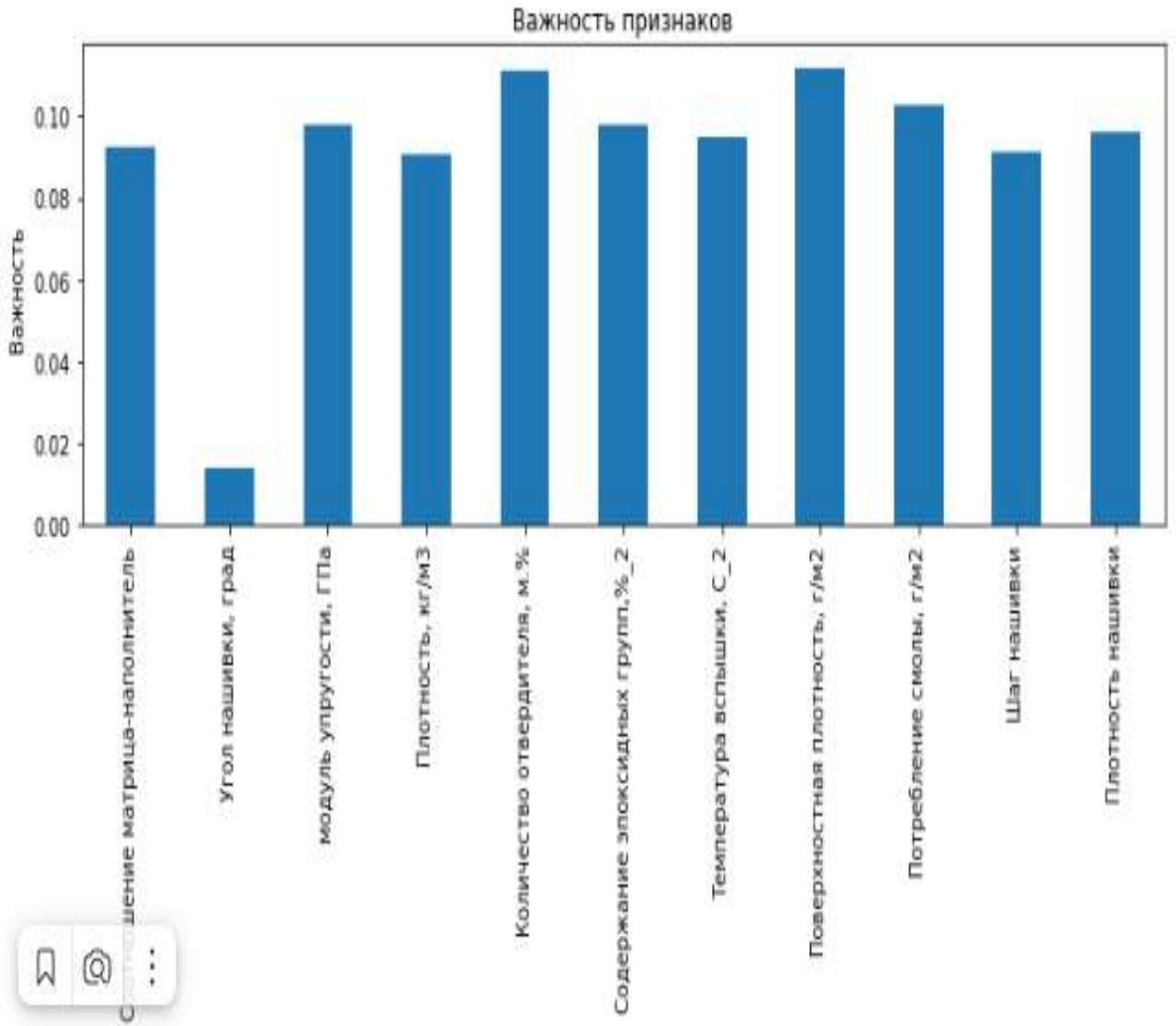

Count
Потребление смолы, г/м2

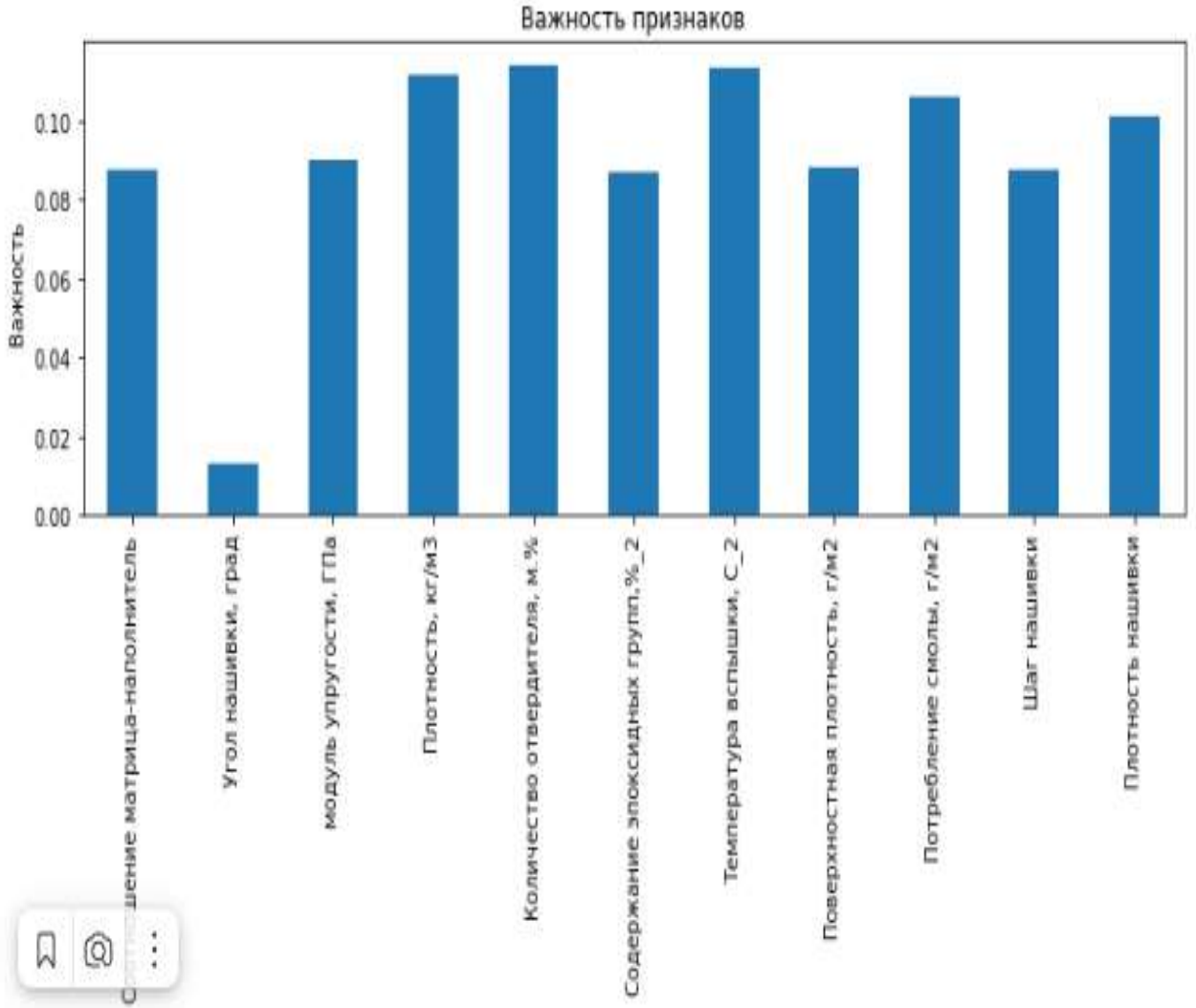

Count
Шаг нашивки


Count
Плотность нашивки


Метод локтя
Дистанция
650
600
550
500
450
400
350
300
250
10
15
20
25
30
35
40
k

feature	Поверхностная плотность, г/м2	Количество отвердителя, м.%	Потребление смолы, г/м2	модуль упругости, ГПа	Содержание эпоксидных групп,%_2	Плотность нашивки	Температура вспышки, С_2	Соотношение матрица-наполнитель	Шаг нашивки	Плотность, кг/м3	Угол нашивки, град
importance	0.1119	0.110906	0.102944	0.097964	0.097704	0.095844	0.09457	0.092356	0.090992	0.090846	0.013973





```python
#Рассмотрим применение Случайного Леса для разных наборов данных
def model_rf(x_train, x_test, y_train, y_test):
    model = RandomForestRegressor(random_state=42,
                                  # опции, относящиеся к отдельным деревьям такие же, как в tree.DecisionTreeRegressor
                                  # число деревьев в лесу
                                  n_estimators=50,
                                  # Функция для измерения качества разделения.
                                  criterion='squared_error',
                                  #максимальная глубина дерева
                                  max_depth=4
                                      )
```

	train_r2	test_r2	mse	mae	mape
Полный датасет	0.135939	-0.015642	0.028650	0.134170	3.783847e+12
0-й кластер	0.239273	0.020304	0.028387	0.131582	7.879145e+12
1-й кластер	0.221050	0.064647	0.023527	0.122880	3.494982e-01
Полный датасет 6 столбцов	0.121267	0.001264	0.028174	0.132902	3.718998e+12
0-й кластер 6 столбцов	0.222402	-0.009753	0.029255	0.134236	7.869201e+12
1-й кластер 6 столбцов	0.213179	0.060093	0.023636	0.123519	3.462081e-01

```python
def lasso_reg(x_train, x_test, y_train, y_test):
    # строим модель, обучаем ее, проверяем прогнозные значения, смотрим метрики
    lasso = Lasso(alpha=1.0, #Константа, которая контролируя силу регулярицции
                  max_iter=1000, #Максимальное количество итераций.
                  tol=0.0001, #Допуск для оптимизации
                  random_state=42,#Начальное значение генератора псевдослучайных чисел
                  selection='cyclic')
    lasso.fit(x_train, y_train)
    predict = lasso.predict(x_train)
```

	train_r2	test_r2	mse	mae	mape
Полный датасет	2.220446e-16	-0.009483	0.028476	0.132860	3.810883e+12
0-й кластер	-5.551115e-16	-0.003623	0.029093	0.134513	8.189343e+12
1-й кластер	3.885781e-16	-0.003551	0.025251	0.125529	3.714942e-01
Полный датасет 6 столбцов	2.220446e-16	-0.009483	0.028476	0.132860	3.810883e+12
0-й кластер 6 столбцов	-5.551115e-16	-0.003623	0.029093	0.134513	8.189343e+12
1-й кластер 6 столбцов	3.885781e-16	-0.003551	0.025251	0.125529	3.714942e-01

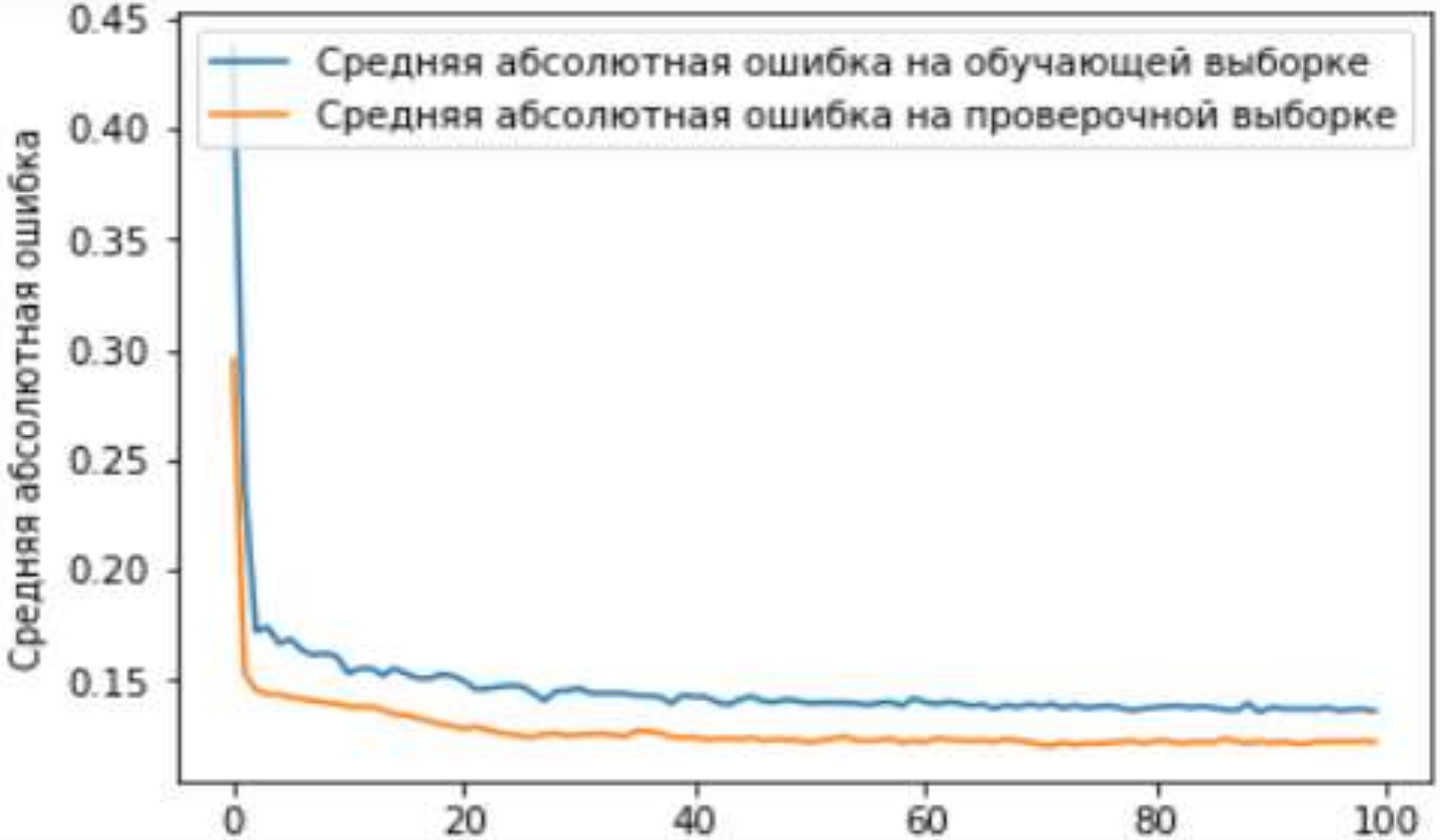
```python
def Neighbors(x_train, x_test, y_train, y_test):
    model = KNeighborsRegressor(n_neighbors=25, #Количество соседей, которые будут использоваться по умолчанию
                                algorithm='auto', #Алгоритм, используемый для вычисления ближайших соседей
                                p=2, #Параметр мощности для метрики Минковского
                                metric='minkowski', #Метрика, используемая для вычисления расстояния.
                                )
```

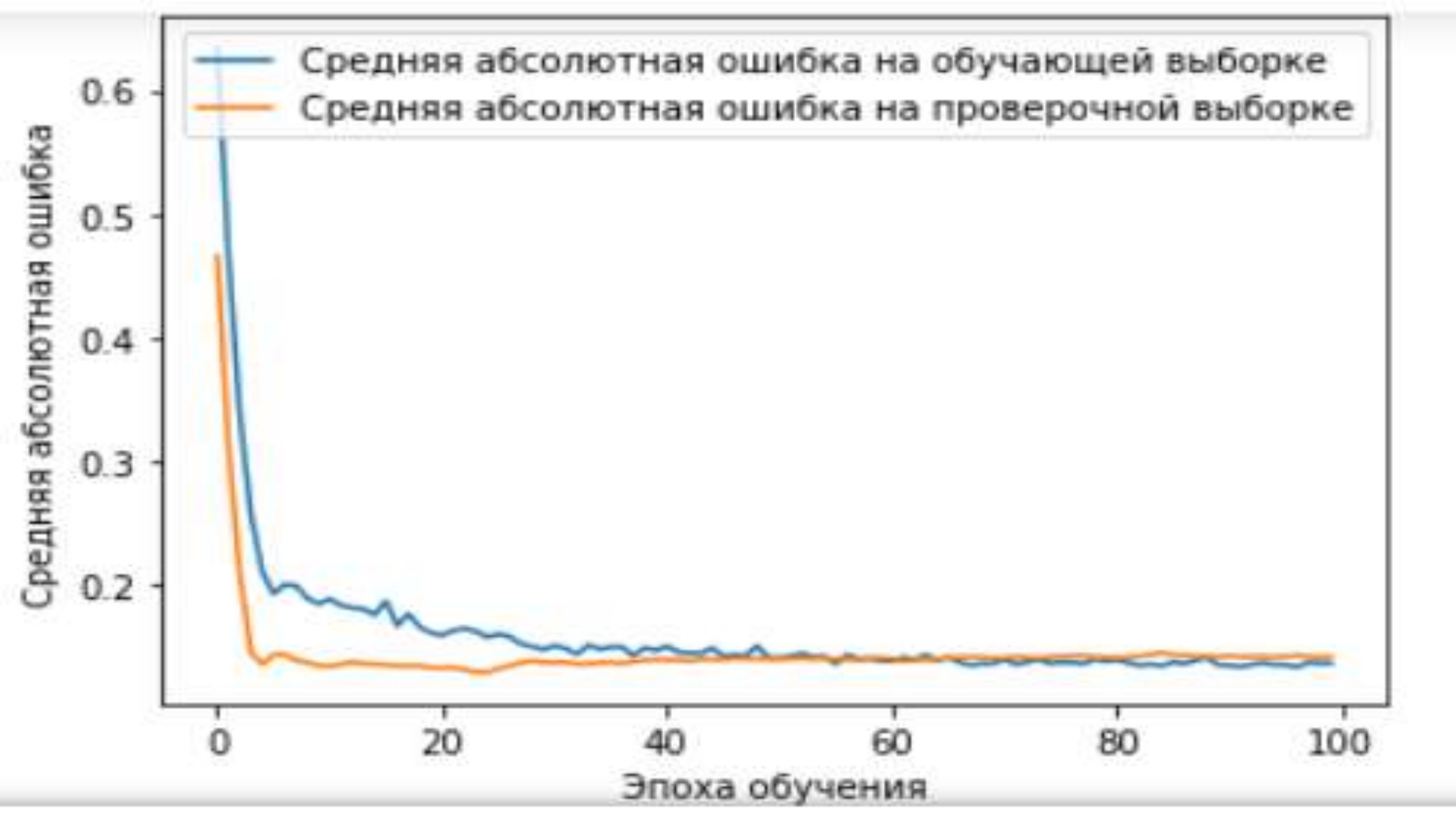
	train_r2	test_r2	mse	mae	mape
Полный датасет	0.068137	-0.036886	0.029237	0.135432	3.534803e+12
0-й кластер	0.054912	-0.016316	0.029444	0.135089	7.204661e+12
1-й кластер	0.040216	-0.027797	0.025863	0.127571	3.773910e-01
Полный датасет 6 столбцов	0.048213	-0.040667	0.029343	0.135491	3.582180e+12
0-й кластер 6 столбцов	0.054864	-0.040367	0.030154	0.137815	7.844449e+12
1-й кластер 6 столбцов	0.065094	-0.025240	0.025799	0.127145	3.725394e-01

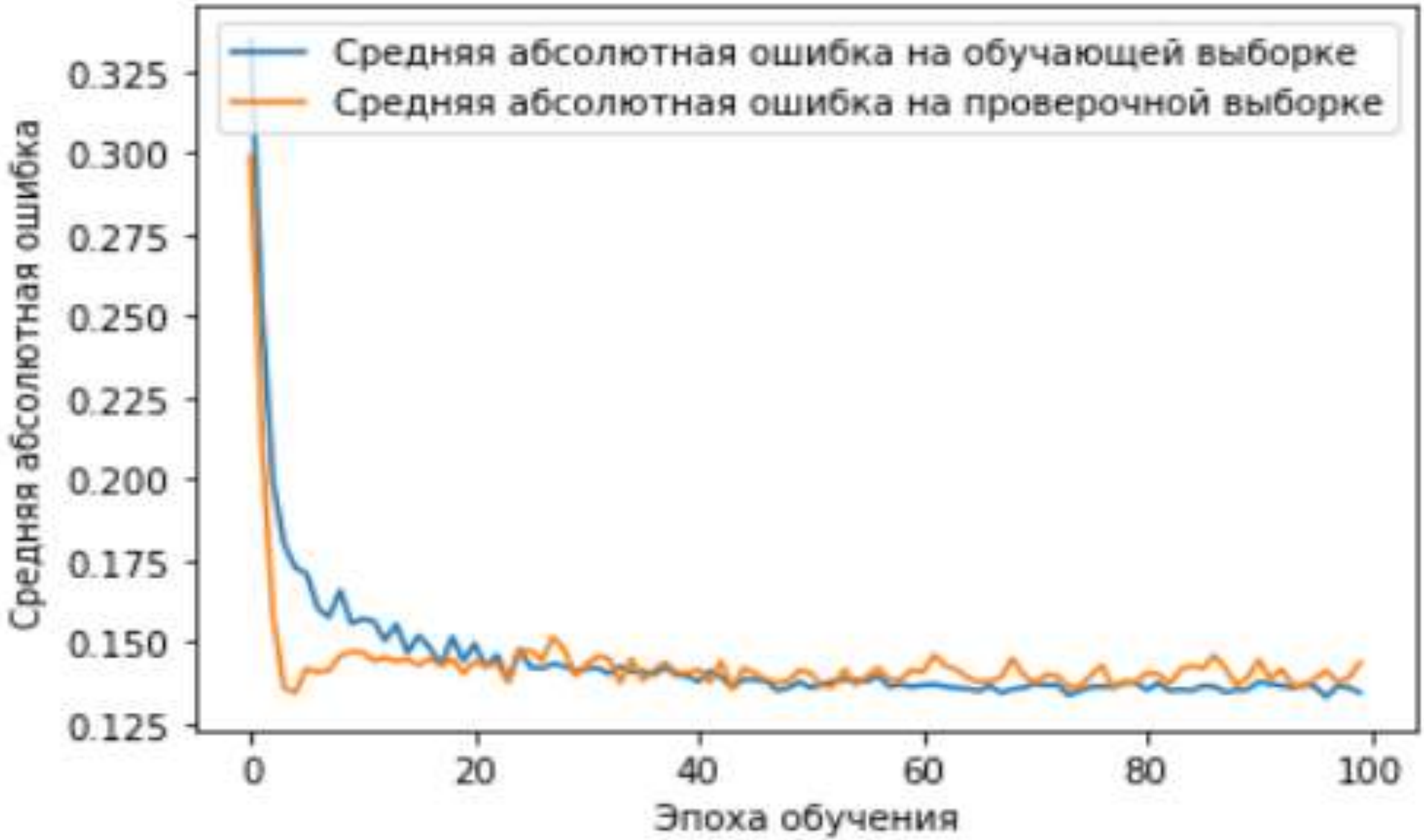
```python
# Многослойный персептрон
def perceptron(x_train, x_test, y_train, y_test):
    model = keras.Sequential()
    model.add(Dense(20, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(10, activation='relu'))
    model.add(Dense(1, activation='linear'))

    model.compile(loss='mean_squared_error', optimizer='adam', metrics=['mae'])
    history = model.fit(x_train, y_train, epochs=100, validation_split = 0.1, verbose = 2)
```

Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 20)	260
dropout_2 (Dropout)	(None, 20)	0
dense_7 (Dense)	(None, 10)	210
dense_8 (Dense)	(None, 1)	11

Total params: 481
Trainable params: 481
Non-trainable params: 0


Средняя абсолютная ошибка на обучающей выборке
Средняя абсолютная ошибка на проверочной выборке
Средняя абсолютная ошибка
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0
20
40
60
80
100


Средняя абсолютная ошибка на обучающей выборке
Средняя абсолютная ошибка на проверочной выборке
Средняя абсолютная ошибка
0.6
0.5
0.4
0.3
0.2
0
20
40
60
80
100
Эпоха обучения


Средняя абсолютная ошибка на обучающей выборке
Средняя абсолютная ошибка на проверочной выборке
Средняя абсолютная ошибка
0.325
0.300
0.275
0.250
0.225
0.200
0.175
0.150
0.125
0
20
40
60
80
100
Эпоха обучения