ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

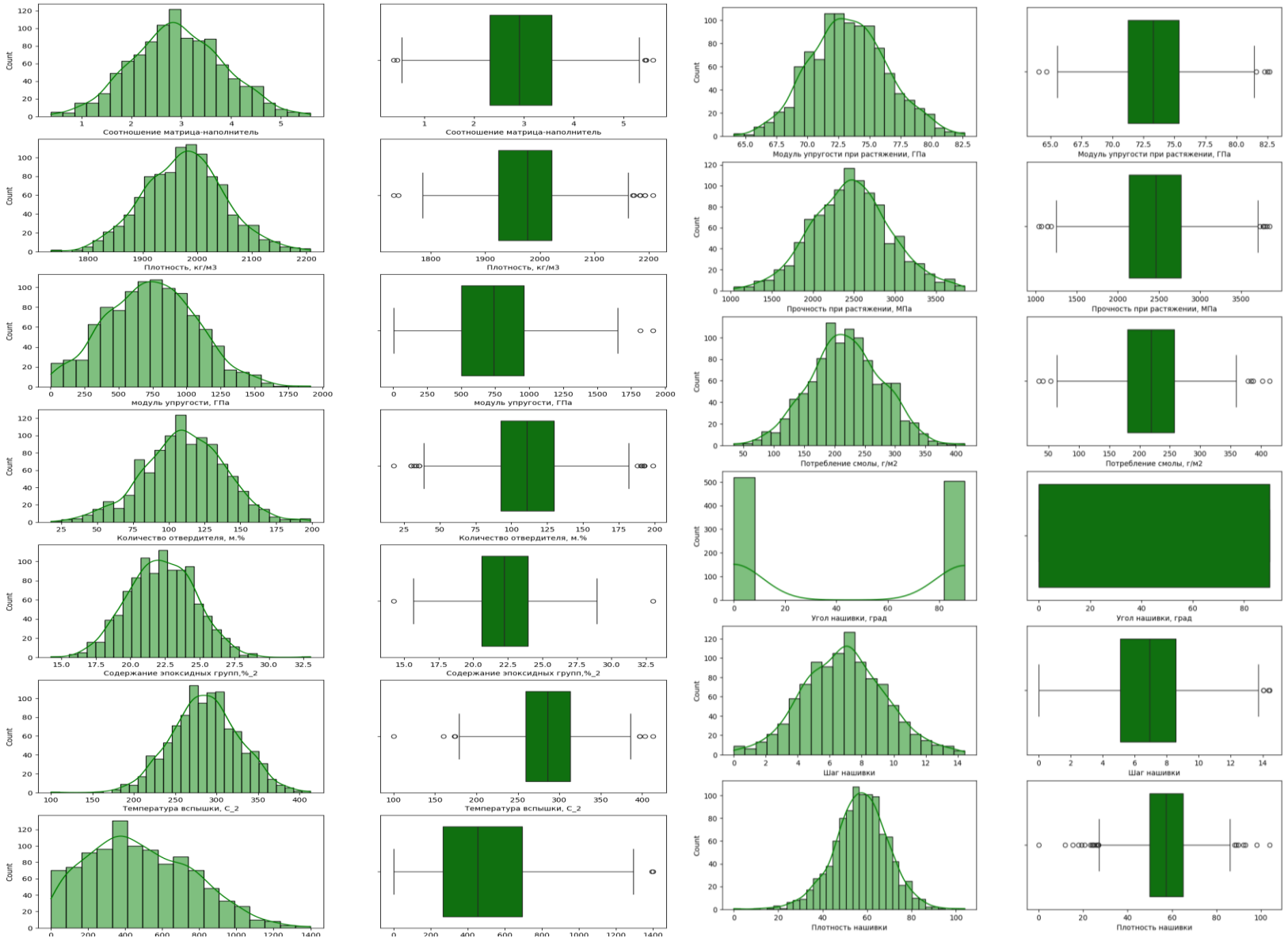
Выпускная квалификационная работа по курсу «Data Science»

Тема: прогнозирование конечных свойств композиционных материалов

Слушатель: Карпов Филипп Алексеевич

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных

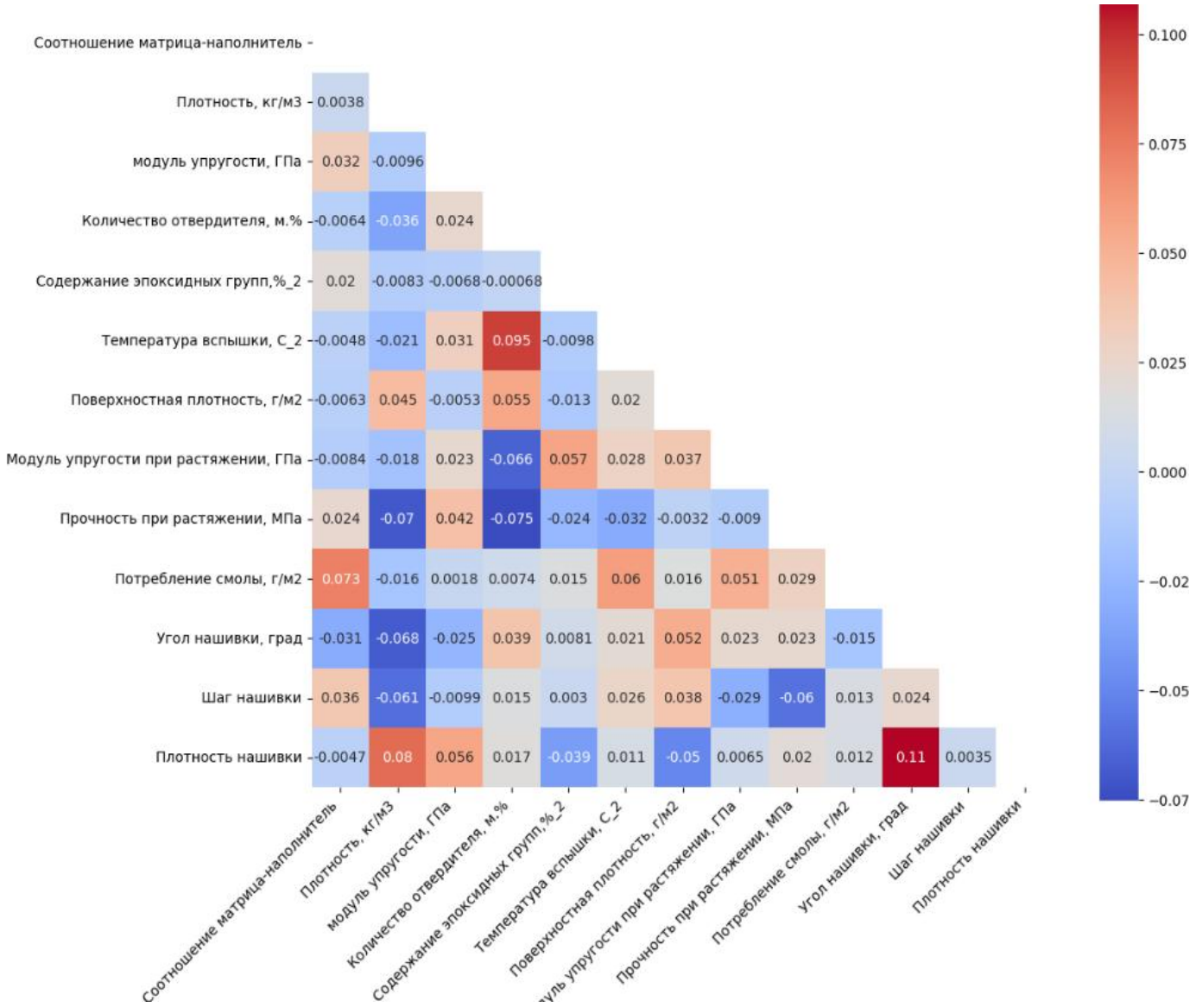


Результаты

1. Всего 1023 объекта, 13 признаков, 3 из которых целевые
2. Все характеристики являются числовыми, пропусков в данных нет
3. Каких-либо зависимостей между признаками не выявлено, попарные коэффициенты корреляции близки к 0
4. График попарного рассеяния точек не выявил зависимостей, видны выбросы

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных

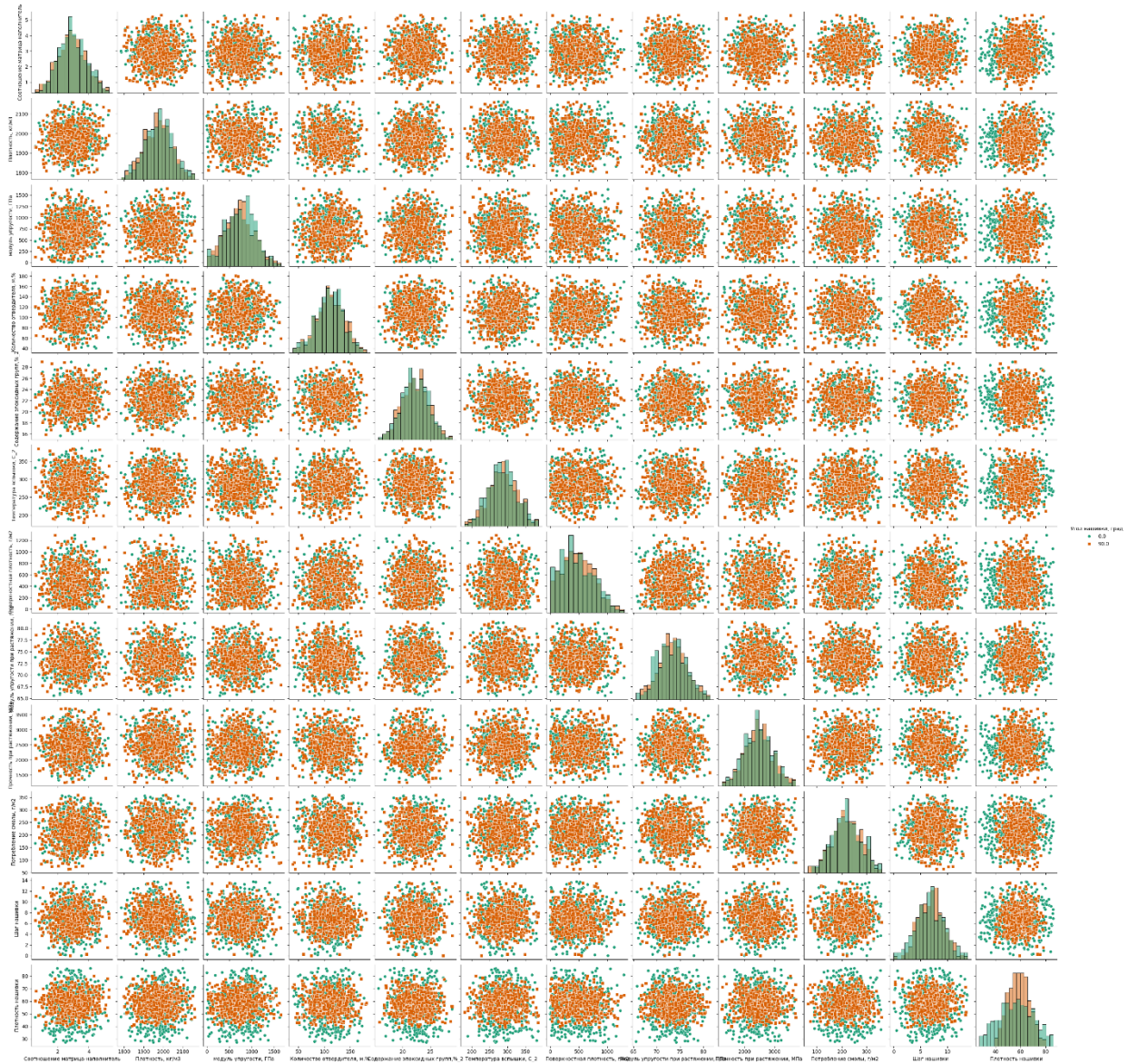


Результаты

1. Всего 1023 объекта, 13 признаков, 3 из которых целевые
2. Все характеристики являются числовыми, пропусков в данных нет
3. Каких-либо зависимостей между признаками не выявлено, попарные коэффициенты корреляции близки к 0
4. График попарного рассеяния точек не выявил зависимостей, видны выбросы

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных



Результаты

1. Всего 1023 объекта, 13 признаков, 3 из которых целевые
2. Все характеристики являются числовыми, пропусков в данных нет
3. Каких-либо зависимостей между признаками не выявлено, попарные коэффициенты корреляции близки к 0
4. График попарного рассеяния точек не выявил зависимостей, видны выбросы

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

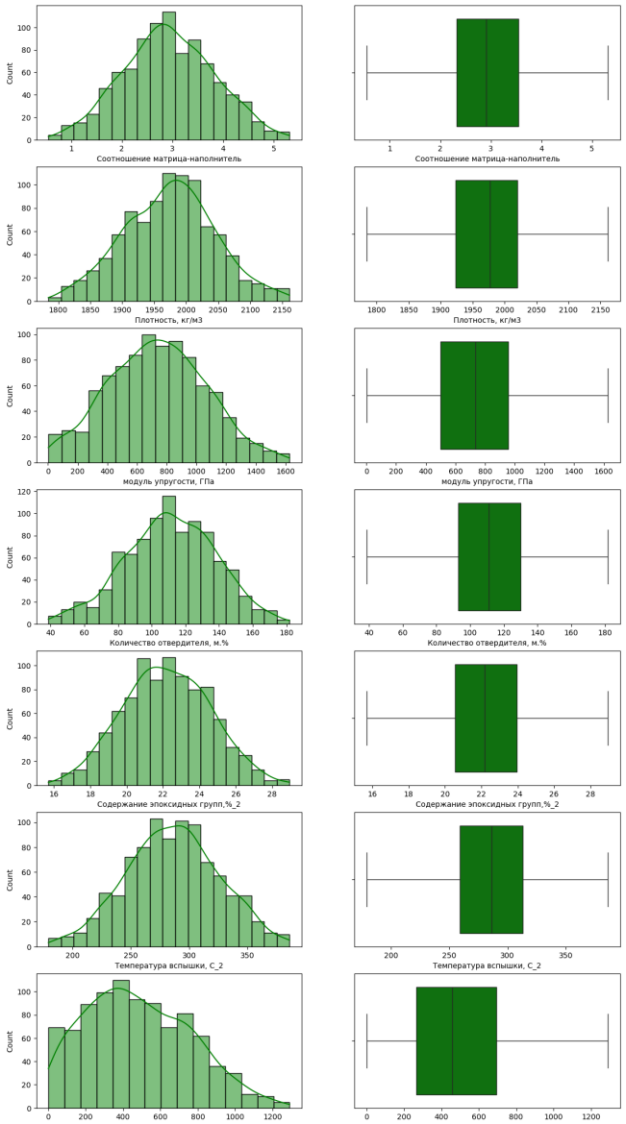
Предобработка данных

Проведено:

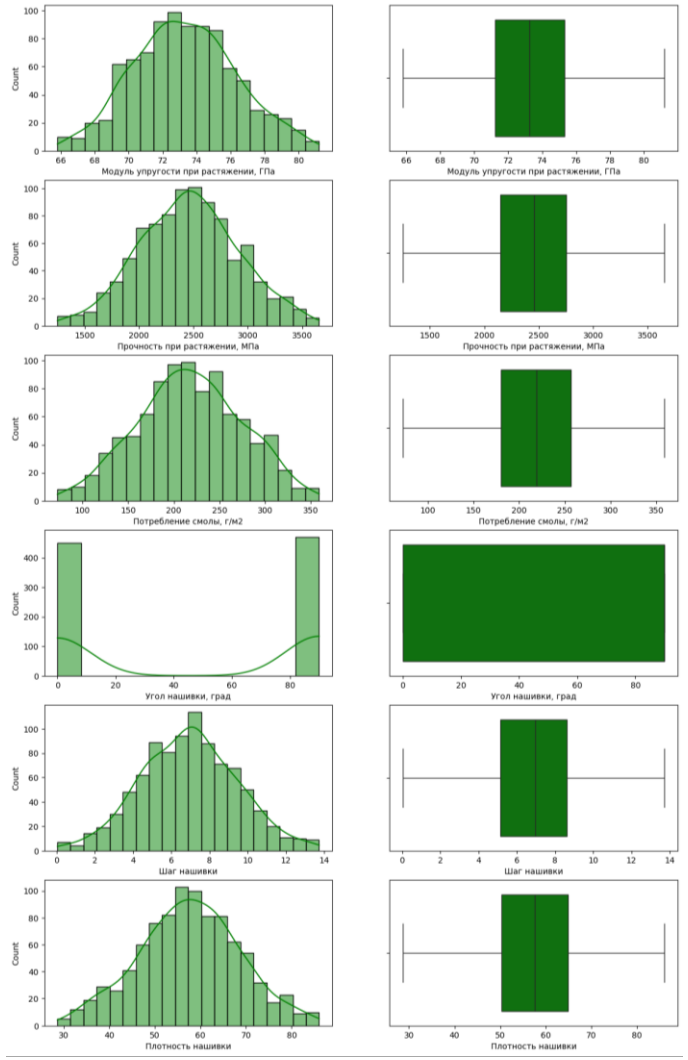
Удаление выбросов методом межквартильного интервала, т.к. для некоторых признаков наблюдаем ассиметричное распределение

	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

Описательная статистика до обработки



Гистограммы распределения и диаграммы "ящика с усами" после удаления выбросов



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Предобработка данных

Проведены:

1. Нормализация с помощью MinMaxScaler библиотеки sklearn, привела все признаки к диапазону от 0 до 1

	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

Описательная статистика до обработки

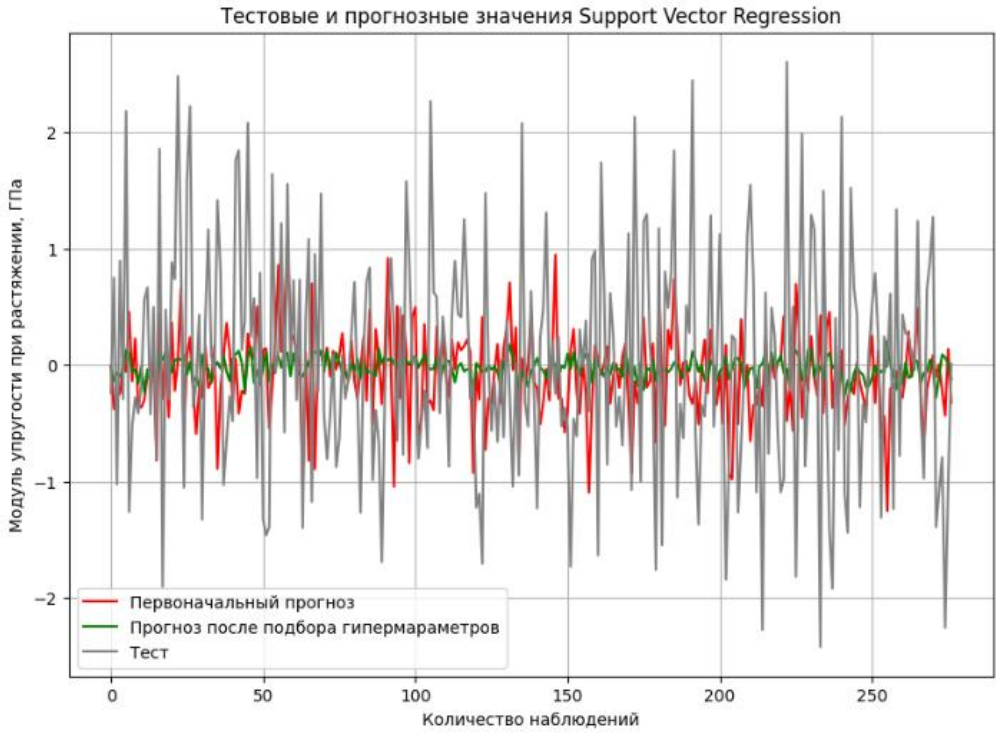
	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	922.0	0.499412	0.187858	0.0	0.371909	0.495189	0.629774	1.0
Плотность, кг/м3	922.0	0.502904	0.188395	0.0	0.368184	0.511396	0.624719	1.0
модуль упругости, ГПа	922.0	0.451341	0.201534	0.0	0.305188	0.451377	0.587193	1.0
Количество отвердителя, м.%	922.0	0.506200	0.186876	0.0	0.378514	0.506382	0.638735	1.0
Содержание эпоксидных групп,%_2	922.0	0.490578	0.180548	0.0	0.366571	0.488852	0.623046	1.0
Температура вспышки, С_2	922.0	0.516739	0.190721	0.0	0.386228	0.516931	0.646553	1.0
Поверхностная плотность, г/м2	922.0	0.373295	0.217269	0.0	0.204335	0.354161	0.538397	1.0
Модуль упругости при растяжении, ГПа	922.0	0.487343	0.196366	0.0	0.353512	0.483718	0.617568	1.0
Прочность при растяжении, МПа	922.0	0.503776	0.188668	0.0	0.373447	0.501481	0.624299	1.0
Потребление смолы, г/м2	922.0	0.507876	0.199418	0.0	0.374647	0.510143	0.642511	1.0
Угол нашивки, град	922.0	0.510846	0.500154	0.0	0.000000	1.000000	1.000000	1.0
Шаг нашивки	922.0	0.503426	0.183587	0.0	0.372844	0.506414	0.626112	1.0
Плотность нашивки	922.0	0.503938	0.193933	0.0	0.376869	0.504310	0.630842	1.0

Описательная статистика после обработки

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Метод опорных векторов

Модуль упругости при растяжении

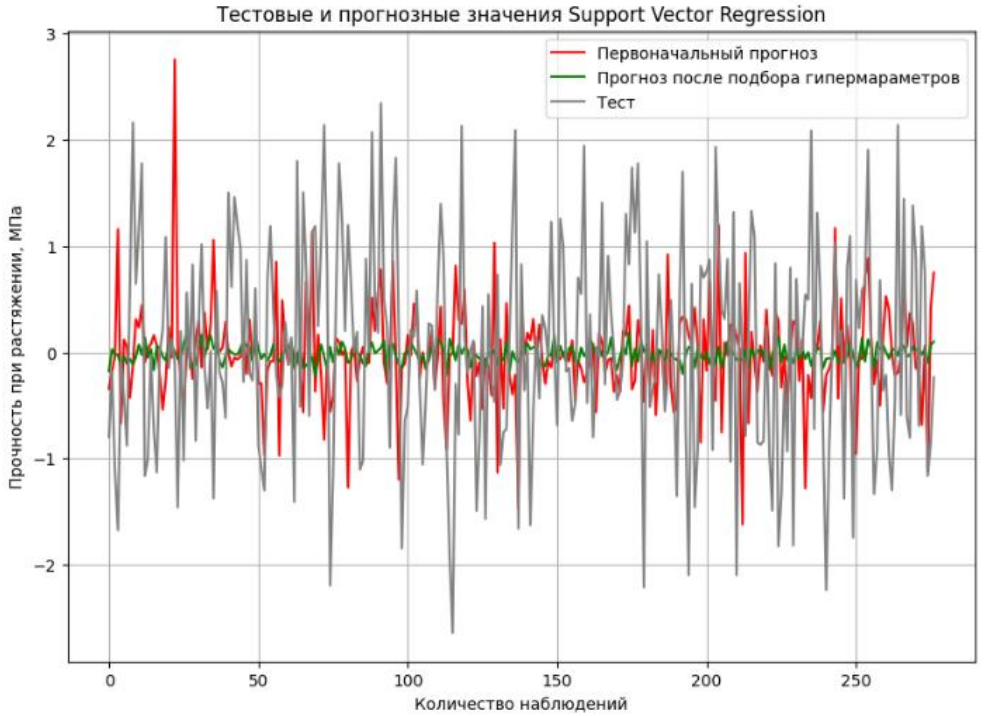



```
svr_upr_hyp = SVR(C=0.1, epsilon=1)
svr_upr_hyp.fit(X_std_train_upr, np.ravel(y_std_train_upr)) #обучаем модель
y_pred_svr_upr_hyp = svr_upr_hyp.predict(X_std_test_upr)
print('Результаты обучения методом опорных векторов для величины "Модуль упругости при растяжении, ГПа":')
print_metrics(y_std_test_upr, y_pred_svr_upr_hyp)
```

✓ 0.0s

```
Результаты обучения методом опорных векторов для величины "Модуль упругости при растяжении, ГПа":
Среднеквадратическая ошибка MSE: 0.969
Средняя абсолютная ошибка MAE: 0.792
Корень из среднеквадратической ошибки RMSE: 0.985
Коэффициент детерминации R2: -0.009
```

Прочность при растяжении




```
svr_pr = SVR(kernel = 'poly', C = 1, epsilon = 1.0)
svr_pr.fit(X_std_train_pr, np.ravel(y_std_train_pr)) #обучаем модель
y_pred_svr_pr = svr_pr.predict(X_std_test_pr)
print('Результаты обучения методом опорных векторов "Прочность при растяжении, МПа":')
print_metrics(y_std_test_pr, y_pred_svr_pr)
```

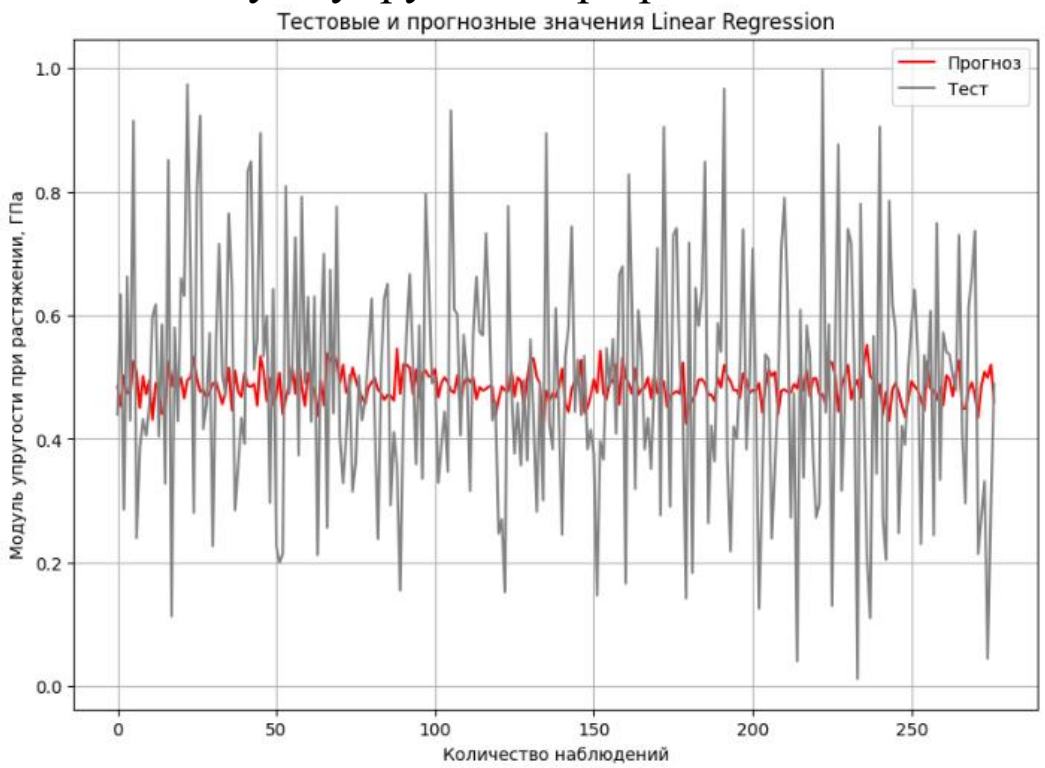
✓ 0.0s

```
Результаты обучения методом опорных векторов "Прочность при растяжении, МПа":
Среднеквадратическая ошибка MSE: 1.13
Средняя абсолютная ошибка MAE: 0.848
Корень из среднеквадратической ошибки RMSE: 1.063
Коэффициент детерминации R2: -0.153
```

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Линейная регрессия

Модуль упругости при растяжении

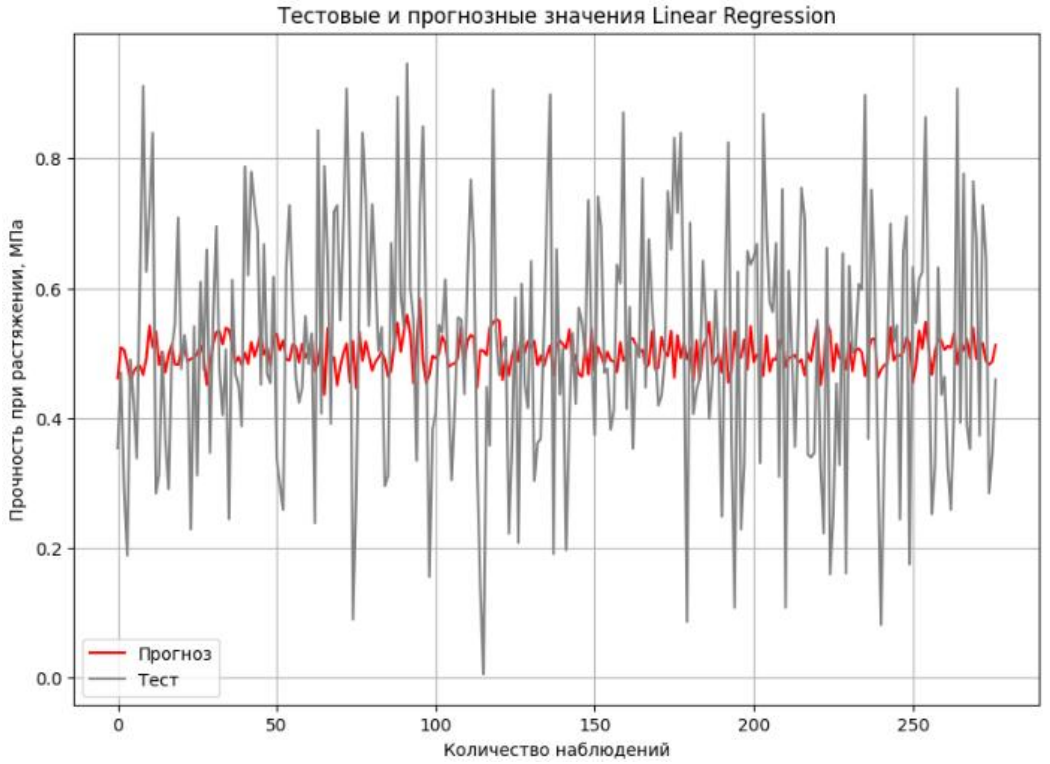



```
lr_upr = LinearRegression()
lr_upr.fit(X_norm_train_upr, y_norm_train_upr) #обучаем модель
y_pred_lr_upr = lr_upr.predict(X_norm_test_upr)
print('Результаты обучения методом линейной регрессии для величины "Модуль упругости при растяжении, ГПа":')
print_metrics(y_norm_test_upr, y_pred_lr_upr)
```

✓ 0.0s

```
Результаты обучения методом линейной регрессии для величины "Модуль упругости при растяжении, ГПа":
Среднеквадратическая ошибка MSE: 0.038
Средняя абсолютная ошибка MAE: 0.156
Корень из среднеквадратической ошибки RMSE: 0.194
Коэффициент детерминации R2: -0.016
Точность модели (%) 68.328
```

Прочность при растяжении




```
lr_pr = LinearRegression()
lr_pr.fit(X_norm_train_pr, y_norm_train_pr) #обучаем модель
y_pred_lr_pr = lr_pr.predict(X_norm_test_pr)
print('Результаты обучения методом линейной регрессии "Прочность при растяжении, МПа":')
print_metrics(y_norm_test_pr, y_pred_lr_pr)
```

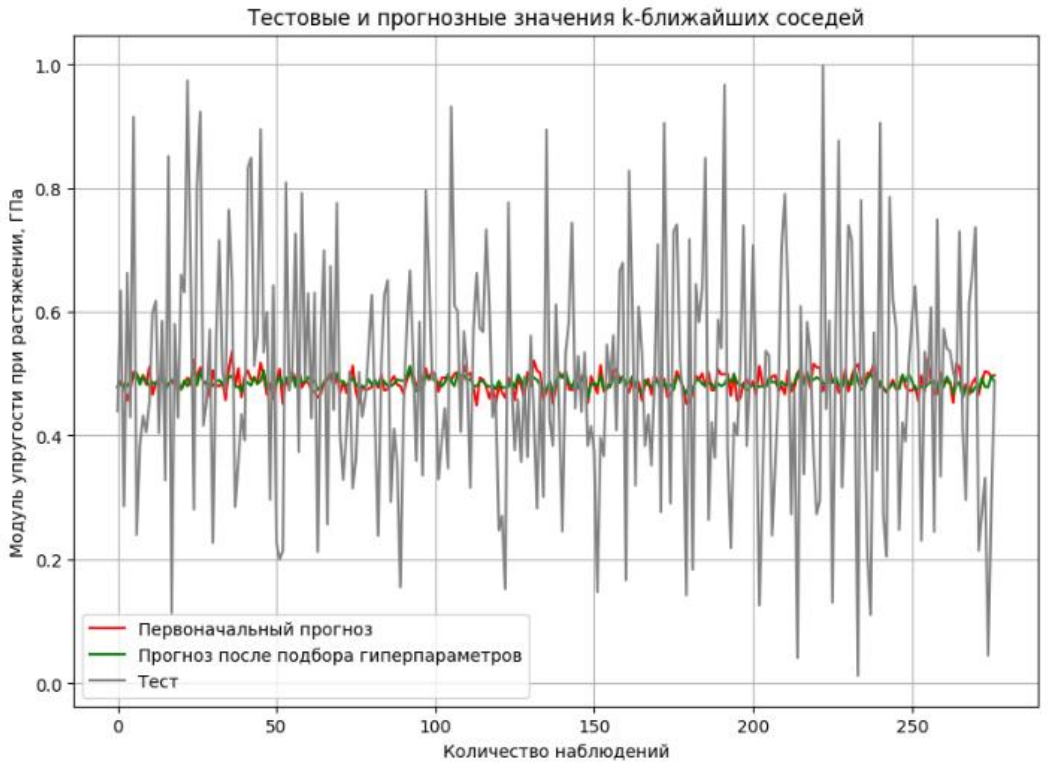
✓ 0.0s

```
Результаты обучения методом линейной регрессии "Прочность при растяжении, МПа":
Среднеквадратическая ошибка MSE: 0.035
Средняя абсолютная ошибка MAE: 0.15
Корень из среднеквадратической ошибки RMSE: 0.186
Коэффициент детерминации R2: 0.009
Точность модели (%) 70.841
```

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

k-ближайших соседей

Модуль упругости при растяжении

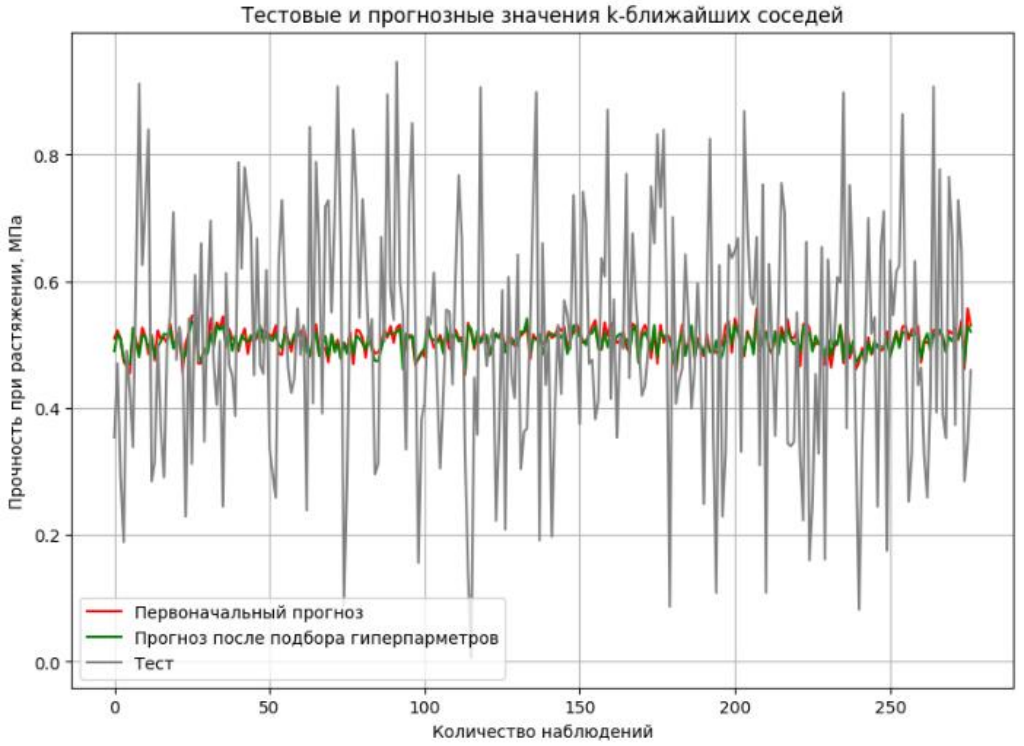



```
knr_upr = KNeighborsRegressor(n_neighbors=100)
knr_upr.fit(X_norm_train_upr, y_norm_train_upr) #обучаем модель
y_pred_knr_upr = knr_upr.predict(X_norm_test_upr)
print('Результаты обучения методом k-ближайших соседей для величины "Модуль упругости при растяжении, ГПа":')
print_metrics(y_norm_test_upr, y_pred_knr_upr)
```

✓ 0.0s

```
Результаты обучения методом k-ближайших соседей для величины "Модуль упругости при растяжении, ГПа":
Среднеквадратическая ошибка MSE: 0.037
Средняя абсолютная ошибка MAE: 0.155
Корень из среднеквадратической ошибки RMSE: 0.193
Коэффициент детерминации R2: -0.005
Точность модели (%) 68.69
```

Прочность при растяжении




```
knr_pr = KNeighborsRegressor(n_neighbors=100)
knr_pr.fit(X_norm_train_pr, y_norm_train_pr) #обучаем модель
y_pred_knr_pr = knr_pr.predict(X_norm_test_pr)
print('Результаты обучения методом k-ближайших соседей "Прочность при растяжении, МПа":')
print_metrics(y_norm_test_pr, y_pred_knr_pr)
```

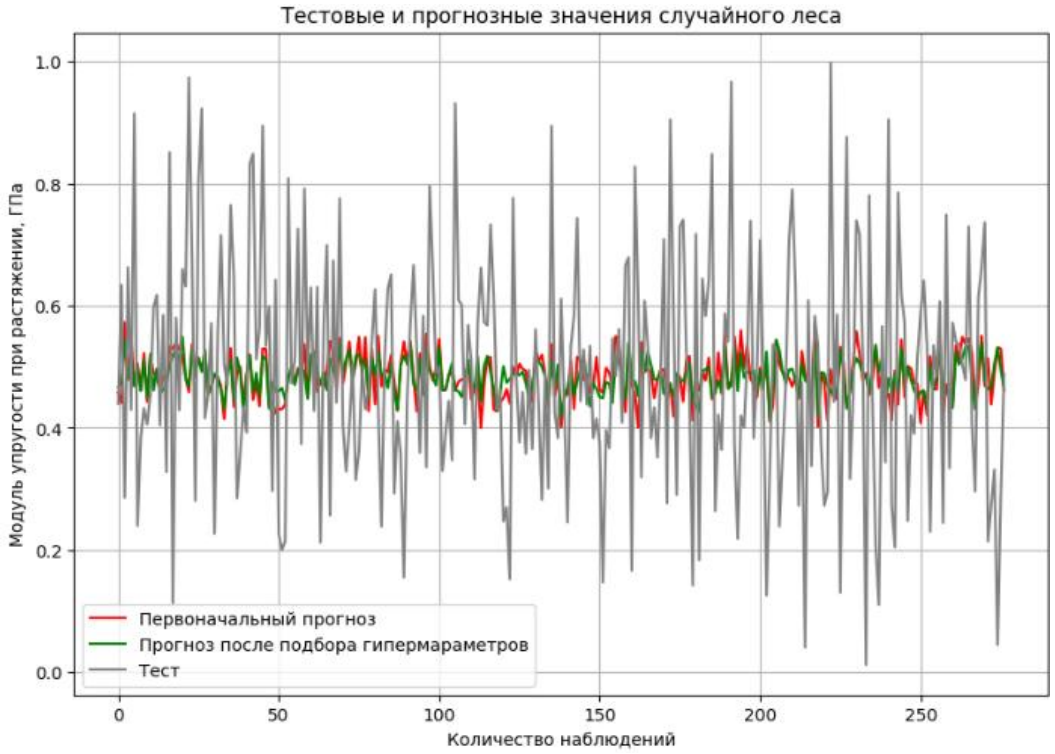
✓ 0.0s

```
Результаты обучения методом k-ближайших соседей "Прочность при растяжении, МПа":
Среднеквадратическая ошибка MSE: 0.035
Средняя абсолютная ошибка MAE: 0.152
Корень из среднеквадратической ошибки RMSE: 0.188
Коэффициент детерминации R2: -0.012
Точность модели (%) 70.472
```

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Случайный лес

Модуль упругости при растяжении

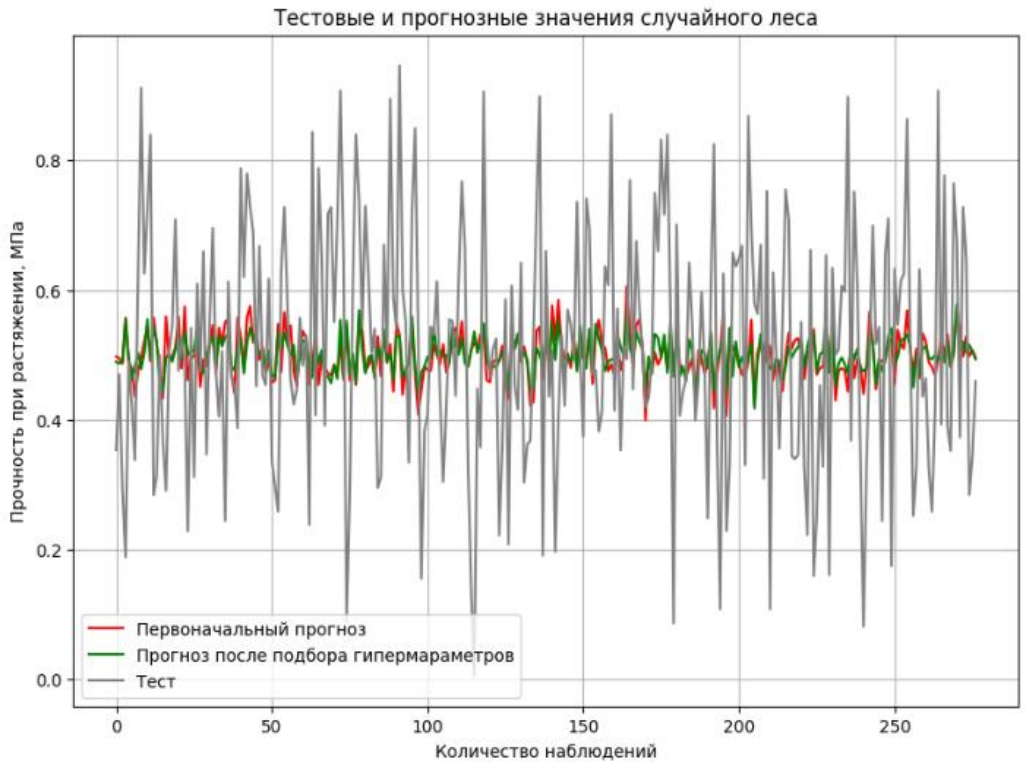



```
rf_upr = RandomForestRegressor(min_samples_split=10, max_features='sqrt', random_state=3)
rf_upr.fit(X_norm_train_upr, y_norm_train_upr) #обучаем модель
y_pred_rf_upr = rf_upr.predict(X_norm_test_upr)
print('Результаты обучения методом случайного леса для величины "Модуль упругости при растяжении, ГПа":')
print_metrics(y_norm_test_upr, y_pred_rf_upr)
```

✓ 0.3s

```
Результаты обучения методом случайного леса для величины "Модуль упругости при растяжении, ГПа":
Среднеквадратическая ошибка MSE: 0.038
Средняя абсолютная ошибка MAE: 0.156
Корень из среднеквадратической ошибки RMSE: 0.195
Коэффициент детерминации R2: -0.027
Точность модели (%) 68.331
```

Прочность при растяжении




```
rf_pr = RandomForestRegressor(min_samples_split=10, max_features='sqrt', random_state=3)
rf_pr.fit(X_norm_train_pr, y_norm_train_pr) #обучаем модель
y_pred_rf_pr = rf_pr.predict(X_norm_test_pr)
print('Результаты обучения случайного леса "Прочность при растяжении, МПа":')
print_metrics(y_norm_test_pr, y_pred_rf_pr)
```

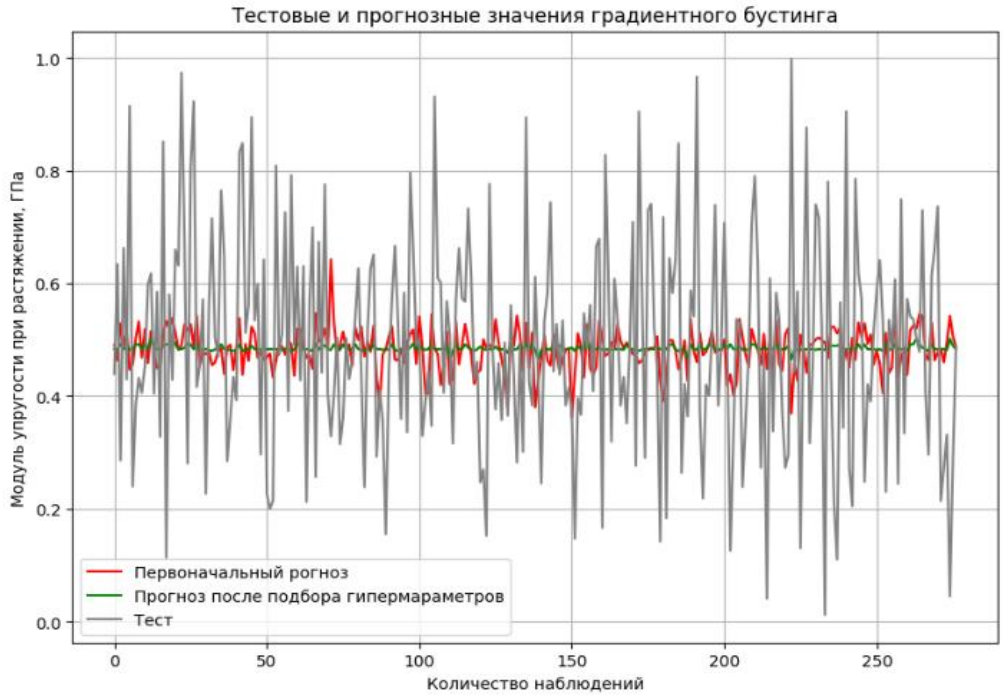
✓ 0.2s

```
Результаты обучения случайного леса "Прочность при растяжении, МПа":
Среднеквадратическая ошибка MSE: 0.035
Средняя абсолютная ошибка MAE: 0.151
Корень из среднеквадратической ошибки RMSE: 0.186
Коэффициент детерминации R2: 0.003
Точность модели (%) 70.693
```

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Градиентный бустинг

Модуль упругости при растяжении

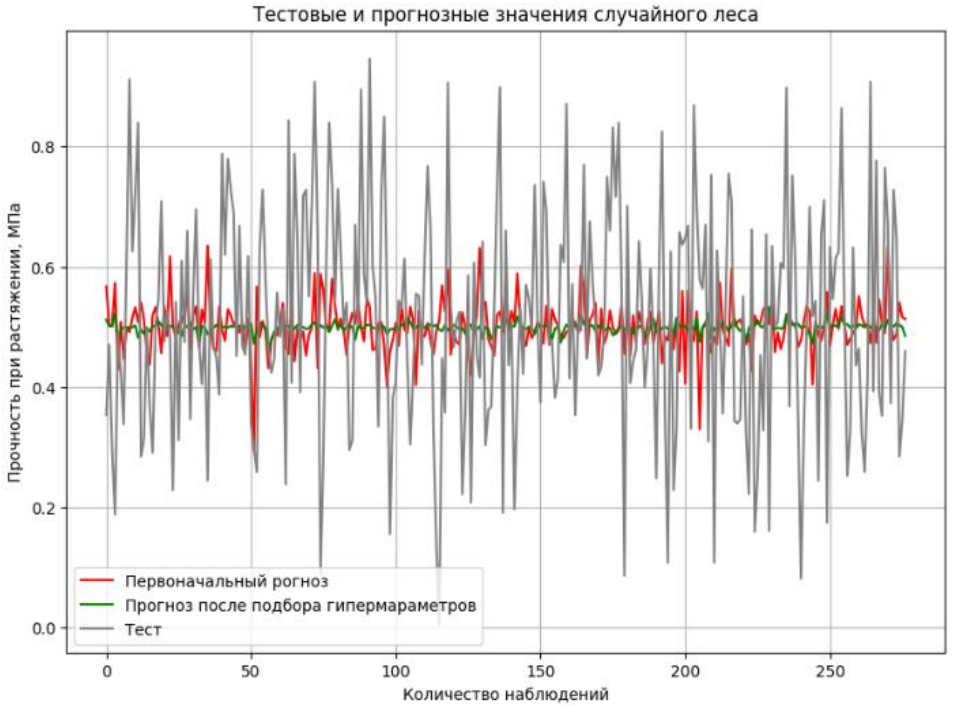



```
gbr_upr = GradientBoostingRegressor(n_estimators=40, learning_rate=0.1, random_state=3)
gbr_upr.fit(X_norm_train_upr, y_norm_train_upr) #обучаем модель
y_pred_gbr_upr = gbr_upr.predict(X_norm_test_upr)
print('Результаты обучения методом градиентного бустинга для величины "Модуль упругости при растяжении, ГПа":')
print_metrics(y_norm_test_upr, y_pred_gbr_upr)
```

✓ 0.1s

```
Результаты обучения методом градиентного бустинга для величины "Модуль упругости при растяжении, ГПа":
Среднеквадратическая ошибка MSE: 0.039
Средняя абсолютная ошибка MAE: 0.156
Корень из среднеквадратической ошибки RMSE: 0.196
Коэффициент детерминации R2: -0.041
Точность модели (%) 68.463
```

Прочность при растяжении




```
gbr_pr = GradientBoostingRegressor(n_estimators=50, learning_rate=0.1, random_state=3)
gbr_pr.fit(X_norm_train_pr, y_norm_train_pr) #обучаем модель
y_pred_gbr_pr = gbr_pr.predict(X_norm_test_pr)
print('Результаты обучения случайного леса "Прочность при растяжении, МПа":')
print_metrics(y_norm_test_pr, y_pred_gbr_pr)
```

✓ 0.1s

```
Результаты обучения случайного леса "Прочность при растяжении, МПа":
Среднеквадратическая ошибка MSE: 0.037
Средняя абсолютная ошибка MAE: 0.156
Корень из среднеквадратической ошибки RMSE: 0.191
Коэффициент детерминации R2: -0.048
Точность модели (%) 69.753
```

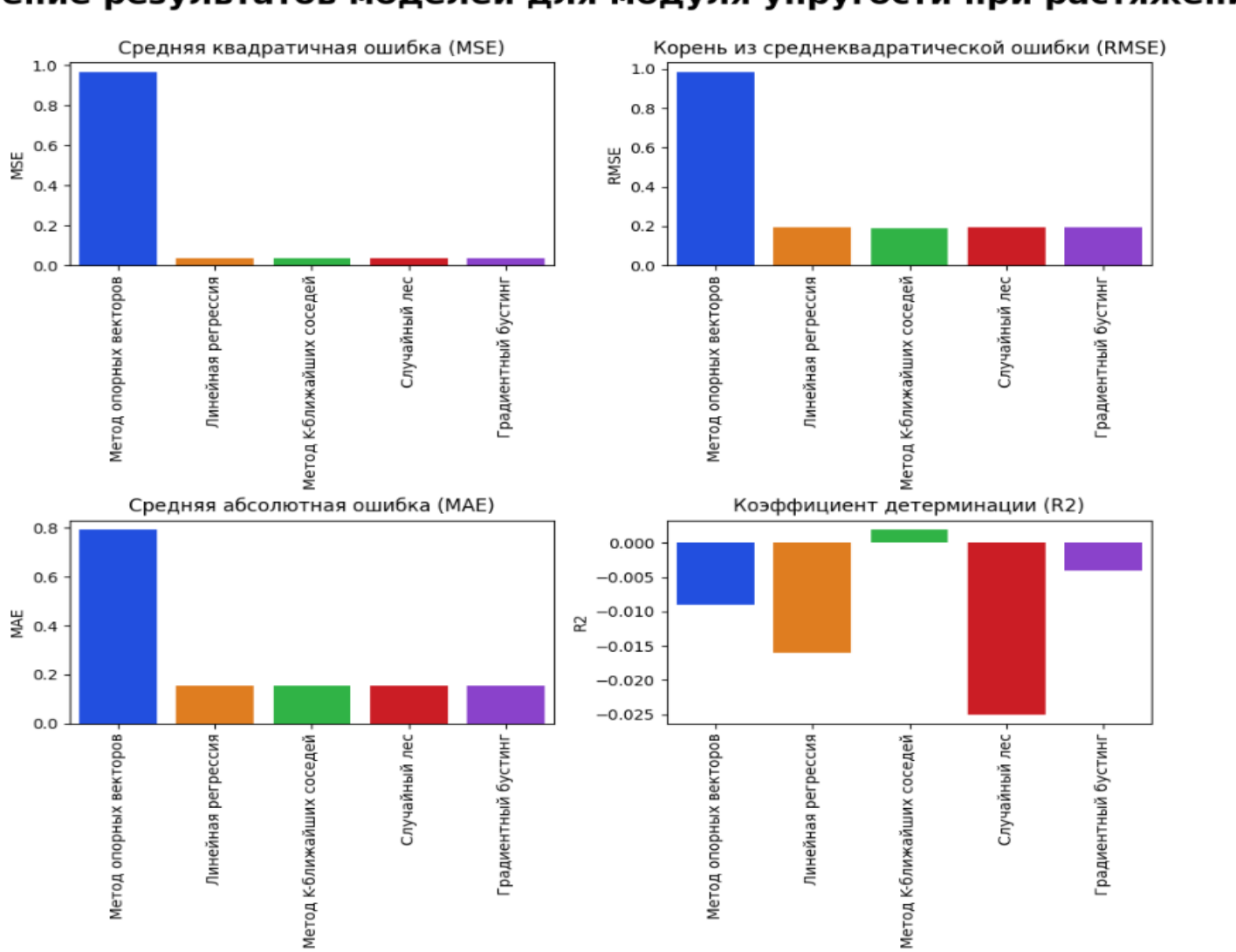
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Построение моделей прогнозирования модуля упругости и прочности при растяжении

Сравнение моделей для прогнозирования

Модуль упругости при растяжении

	Модель	MSE	RMSE	MAE	R2
0	Метод опорных векторов	0.969	0.985	0.792	-0.009
1	Линейная регрессия	0.038	0.194	0.156	-0.016
2	Метод К-ближайших соседей	0.037	0.192	0.154	0.002
3	Случайный лес	0.038	0.195	0.155	-0.025
4	Градиентный бустинг	0.037	0.193	0.154	-0.004



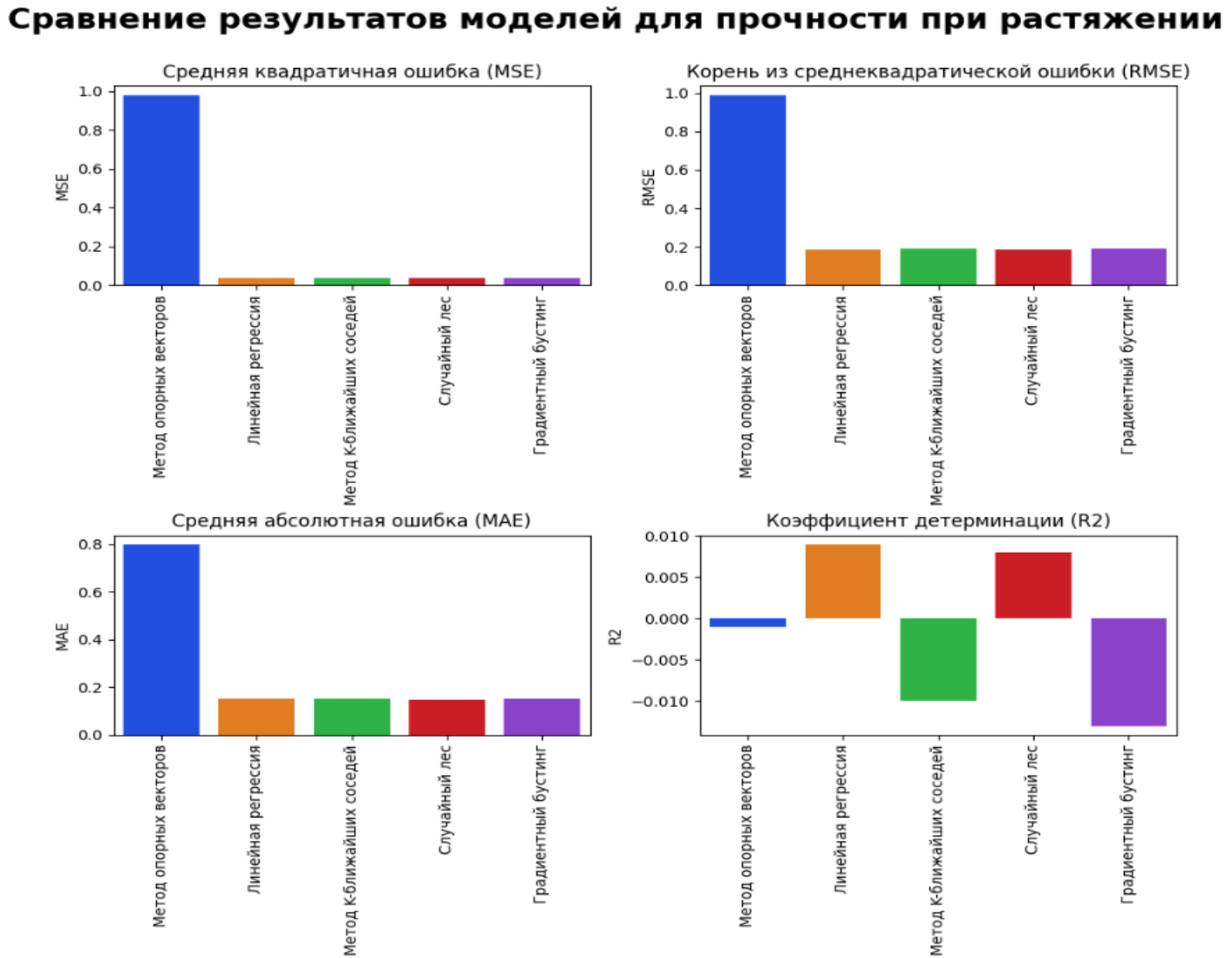
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Построение моделей прогнозирования модуля упругости и прочности при растяжении

Сравнение моделей для прогнозирования

Прочность при растяжении

	Модель	MSE	RMSE	MAE	R2
0	Метод опорных векторов	0.981	0.99	0.799	-0.001
1	Линейная регрессия	0.035	0.186	0.15	0.009
2	Метод К-ближайших соседей	0.035	0.188	0.152	-0.01
3	Случайный лес	0.035	0.186	0.149	0.008
4	Градиентный бустинг	0.035	0.188	0.152	-0.013



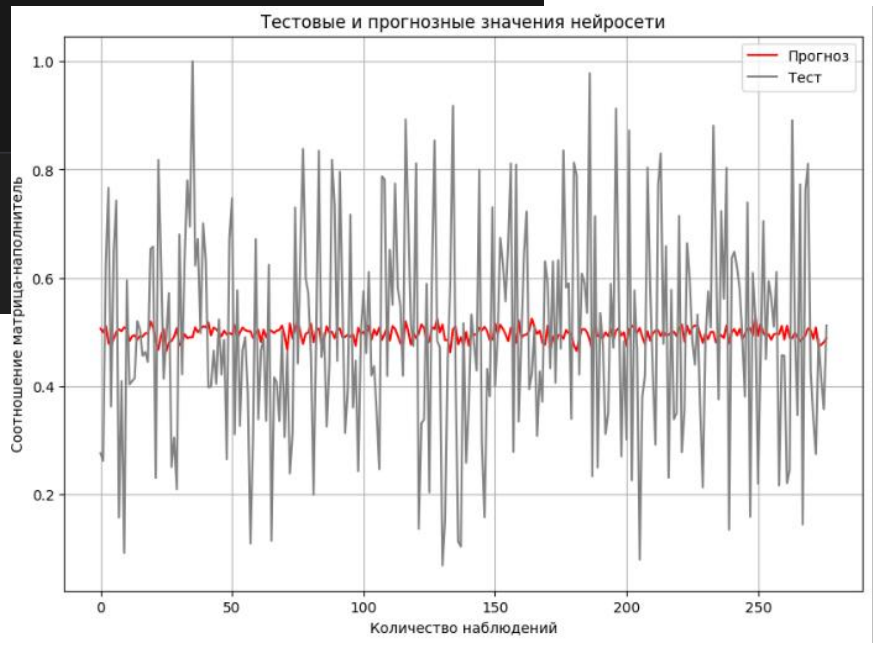
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разработка НС для прогнозирования Соотношения матрица-наполнитель

- 11 входных признаков, 1 - целевой.
- Нейронную сеть строим с помощью класса Sequential библиотеки keras.

```
def base_model():
    model = Sequential()
    model.add(Dense(8, input_dim=12, activation='tanh')) # скрытый полносвязный слой 1 input_dim=12
    model.add(Dense(8, activation='tanh')) # скрытый полносвязный слой 2
    model.add(Dropout(0.5))
    model.add(Dense(1, activation='linear')) # выходной слой

    model.compile(loss='mean_squared_error', optimizer='sgd')
    return model

#Создаем НС и обучаем её
ns = base_model()
history = ns.fit(X_train_ns, y_train_ns,
                 epochs=400,
                 verbose=0, validation_data=(X_test_ns, y_test_ns))

#Предсказываем значения
y_pred_ns = ns.predict(X_test_ns)

#Печатаем метрики и график
print_metrics(y_test_ns, y_pred_ns)
```

```
✓ 49.6s
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step
Среднеквадратическая ошибка MSE: 0.037
Средняя абсолютная ошибка MAE: 0.156
Корень из среднеквадратической ошибки RMSE: 0.193
Коэффициент детерминации R2: -0.005
Точность модели (%) 68.785
```



Параметры НС:

- Входной слой: полносвязный, 8 нейронов, ФА tanh;
- Один скрытый полносвязный слой с 8 нейронами, ФА tanh;
- Один Dropout слой;
- Выходной слой: полносвязный, 1 нейрон, ФА leaner;
- loss-функция: среднеквадратичная ошибка (mean_squared_error).

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Консольное приложение

Приложение прогнозирует соотношение матрица-наполнитель в соответствии с обученной нейронной сетью

```python
# Создадим функцию для ввода данных
def input_variable():
    x1 = float(input('Плотность, кг/м3: '))
    x2 = float(input('Модуль упругости, ГПа: '))
    x3 = float(input('Количество отвердителя, м.%: '))
    x4 = float(input('Содержание эпоксидных групп,%_2: '))
    x5 = float(input('Температура вспышки, С_2: '))
    x6 = float(input('Поверхностная плотность, г/м2: '))
    x7 = float(input('Модуль упругости при растяжении, ГПа: '))
    x8 = float(input('Прочность при растяжении, МПа: '))
    x9 = float(input('Потребление смолы, г/м2: '))
    x10 = float(input('Угол нашивки: '))
    x11 = float(input('Шаг нашивки: '))
    x12 = float(input('Плотность нашивки: '))
    return x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12
```

✓ 0.0s

```python
# Создадим функцию для вызова приложения
def app_model():
    # Загружаем модель и масштабаторы
    nn_model = load_model('../model_dir/my_model.keras')
    scaler_x = load('../model_dir/minmax_scl_x.pkl')
    scaler_y = load('../model_dir/minmax_scl_y.pkl')

    print('Приложение прогнозирует соотношение "матрица-наполнитель"')
    for i in range(110):
        try:
            print('Введите "1" для прогноза, "2" для выхода')
            check = input()

            if check == '1':
                print('Введите данные для прогноза')
                X = input_variable()
                X =  scaler_x.transform(np.array(X).reshape(1,-1))
                prediction = nn_model.predict(X)
                output = scaler_y.inverse_transform(prediction)
                print('Прогнозное значение соотношения "матрица-наполнитель": ')
                print(output[0][0])

            elif check == '2':
                break
            else:
                print('Повторите выбор')

        except Exception as e:
            print(e)
            print('Введены некорректные данные. Пожалуйста, повторите операцию')
app_model()
```

✓ 5m 2.8s

```
Приложение прогнозирует соотношение "матрица-наполнитель"
Введите "1" для прогноза, "2" для выхода
```

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Заключение

Рассмотренные модели машинного обучения: метод опорных векторов, линейная регрессия, метод К-ближайших соседей, случайный лес, градиентный бустинг, нейронная сеть показали неудовлетворительные результаты при прогнозировании целевых переменных даже с учетом подбора гиперпараметров.

Для достижения целей работы необходимо провести более обширное исследование, включающее:

- привлечение большего количества моделей машинного обучения;

- осуществление более тщательный подбора гиперпараметров для каждой модели

- экспертная оценка фактического влиянии исследуемых параметров на целевые переменные, и на основании нее избавиться от малозначительных переменных, или провести дополнительные эксперименты.

ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

do.bmstu.ru