ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Прогноз конечных свойств новых материалов (композиционных матералов)

итоговая аттестационная работа по курсу Data Science

Пономарева Мария Александровна

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

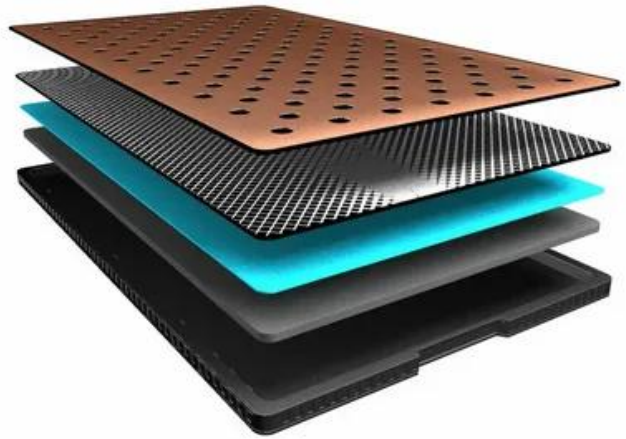
Постановка задачи

Цель работы заключается в разработке моделей для прогнозирования конечных свойств композитов:

- модуля упругости при растяжении

- прочности при растяжении

- модели для рекомендации соотношения «матрица-наполнитель»



Созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов и цифровыми двойниками новых композитов.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Загрузка данных

В качестве исходных данных были представлены два набора данных в формате XLSX.
Загружаем данные Датасеты:

```
#Загружаем первый dataset
df_xbp=pd.read_excel(r"C:\Users\Mariia\Documents\Аналитик данных БАУМАНСКИЙ\АТТЕСТАЦИОННОЕ ЗАДАНИЕ\Датасет для ВКР_композиты\X_bp.xlsx")
df_xbp.head()
```

	Unnamed: 0	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
0	0	1.857143	2030.0	738.736842	30.00	22.267857	100.000000	210.0	70.0	3000.0	220.0
1	1	1.857143	2030.0	738.736842	50.00	23.750000	284.615385	210.0	70.0	3000.0	220.0
2	2	1.857143	2030.0	738.736842	49.90	33.000000	284.615385	210.0	70.0	3000.0	220.0
3	3	1.857143	2030.0	738.736842	129.00	21.250000	300.000000	210.0	70.0	3000.0	220.0
4	4	2.771331	2030.0	753.000000	111.86	22.267857	284.615385	210.0	70.0	3000.0	220.0

```
#Загружаем второй dataset
df_xnup=pd.read_excel(r"C:\Users\Mariia\Documents\Аналитик данных БА
df_xnup.head()
```

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

При помощи команды df.info() выводим данные о Датасетах:

```
#Выведем информацию о первом datasete. Размерность 1023 строк, 11 колонок. Пропусков нет.
df_xbp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1023 entries, 0 to 1022
Data columns (total 11 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   Unnamed: 0                             1023 non-null   int64
 1   Соотношение матрица-наполнитель        1023 non-null   float64
 2   Плотность, кг/м3                       1023 non-null   float64
 3   модуль упругости, ГПа                  1023 non-null   float64
 4   Количество отвердителя, м.%            1023 non-null   float64
 5   Содержание эпоксидных групп,%_2        1023 non-null   float64
 6   Температура вспышки, С_2               1023 non-null   float64
 7   Поверхностная плотность, г/м2          1023 non-null   float64
 8   Модуль упругости при растяжении, ГПа   1023 non-null   float64
 9   Прочность при растяжении, МПа          1023 non-null   float64
 10  Потребление смолы, г/м2                1023 non-null   float64
dtypes: float64(10), int64(1)
memory usage: 88.0 KB
```

```
#Выводим информацию о втором datasete. Пропусков нет. Размерность 1040 строк, 4 колонки. Пропусков нет.
df_xnup.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1040 entries, 0 to 1039
Data columns (total 4 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Unnamed: 0         1040 non-null   int64
 1   Угол нашивки, град 1040 non-null   int64
 2   Шаг нашивки        1040 non-null   float64
 3   Плотность нашивки  1040 non-null   float64
dtypes: float64(2), int64(2)
memory usage: 32.6 KB
```

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Заголовок слайда

Данные Датасеты объединяем по индексу, тип объединения INNER (согласно заданию) при помощи merge

Выводим данные о Датасете, полученном в результате объединения:

```
#Выведем информацию о нем
df_all.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
```

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Описательная статистика

```
#Выведем данные описательной статистики при помощи функции describe
df_all.describe()
```

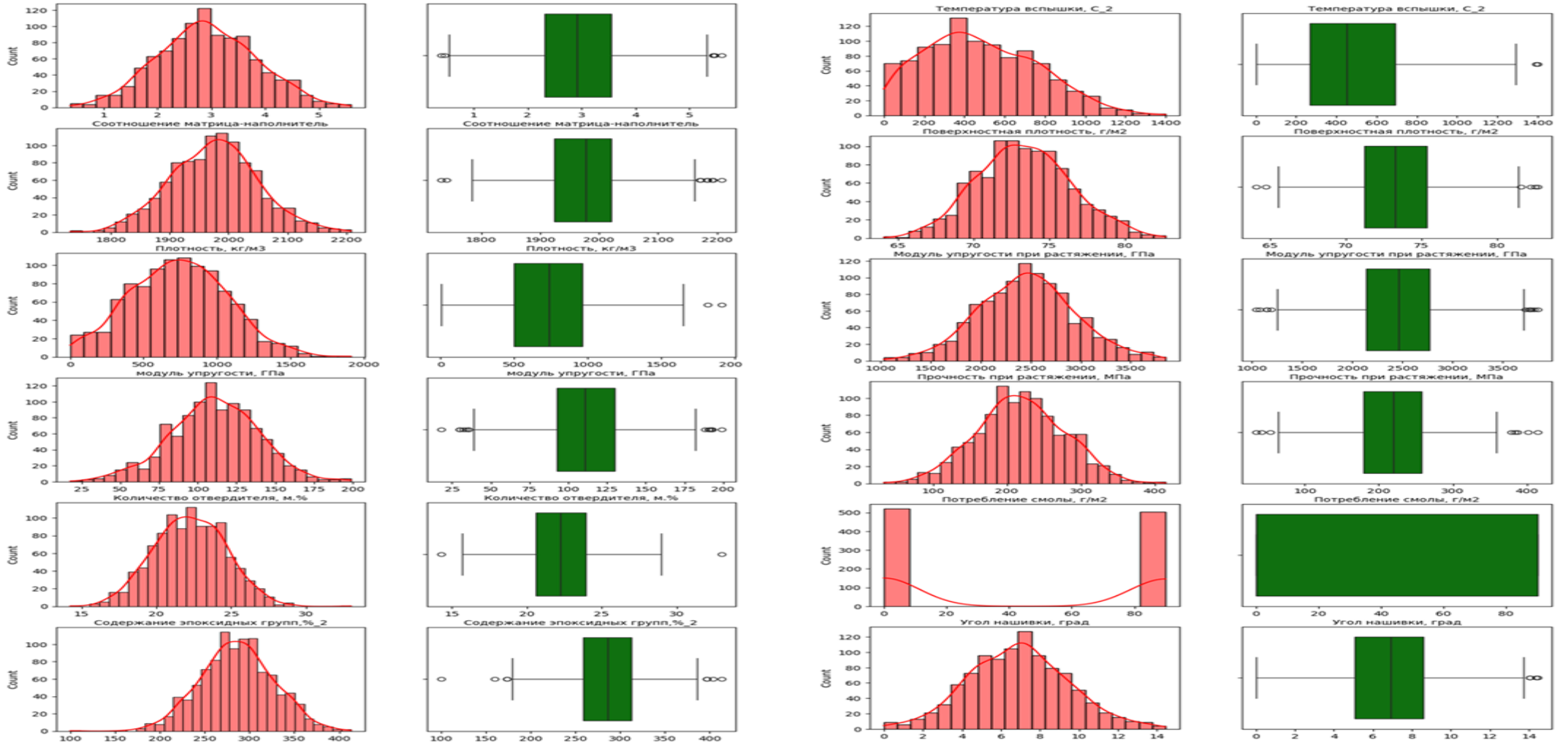
	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
count	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000
mean	2.930366	1975.734888	739.923233	110.570769	22.244390	285.882151	482.731833	73.328571	2466.922843	218.423144	44.252199	6.899222	57.153929
std	0.913222	73.729231	330.231581	28.295911	2.406301	40.943260	281.314690	3.118983	485.628006	59.735931	45.015793	2.563467	12.350969
min	0.389403	1731.764635	2.436909	17.740275	14.254985	100.000000	0.603740	64.054061	1036.856605	33.803026	0.000000	0.000000	0.000000
25%	2.317887	1924.155467	500.047452	92.443497	20.608034	259.066528	266.816645	71.245018	2135.850448	179.627520	0.000000	5.080033	49.799212
50%	2.906878	1977.621657	739.664328	110.564840	22.230744	285.896812	451.864365	73.268805	2459.524526	219.198882	0.000000	6.916144	57.341920
75%	3.552660	2021.374375	961.812526	129.730366	23.961934	313.002106	693.225017	75.356612	2767.193119	257.481724	90.000000	8.586293	64.944961
max	5.591742	2207.773481	1911.536477	198.953207	33.000000	413.273418	1399.542362	82.682051	3848.436732	414.590628	90.000000	14.440522	103.988901

count - количество значений
mean - среднее значение
std - стандартное отклонение
min - минимальное значение
25%, 50%, 75% - первый квартиль, медиана, третий квартиль
max - максимальное значение

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных

Строим графики: гистограммы и «ящики с усами»



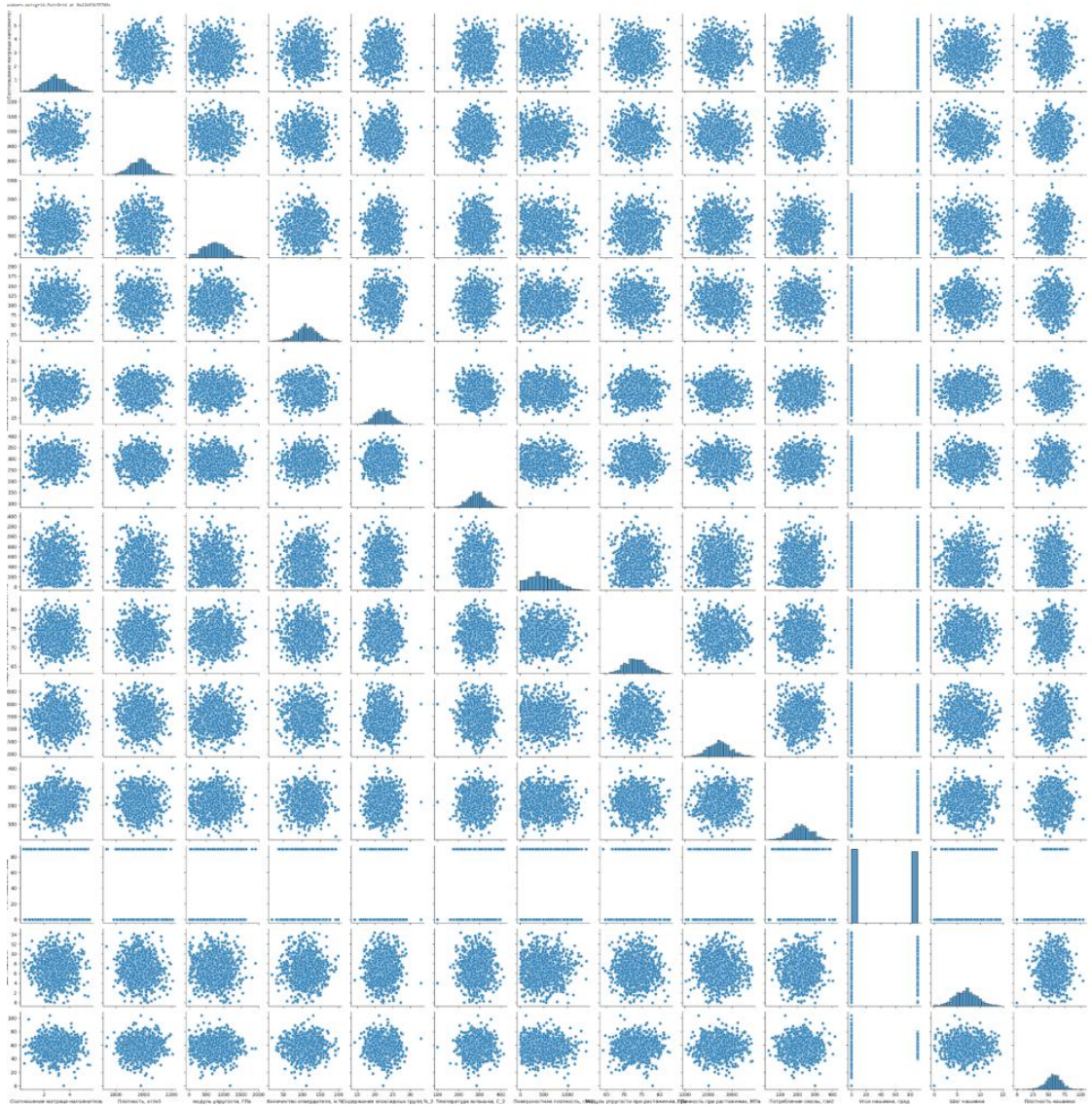
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных

Строим графики рассеяния

Видим выборсы

Применяем методы для нахождения выбросов:
- методом 3-х сигм — 24 выброса;
- методом межквартильных расстояний — 93 выброса



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных

```
# Построим матрицу корреляции
corr = df_all.corr()
fig, ax = plt.subplots(figsize=(10, 7))
sns.heatmap(corr, vmin=-1, vmax=1, annot=True, fmt='.2f', cmap='BrBG')
plt.show()
```

Строим матрицу корреляции.

Корреляционные связи между переменными не наблюдаются

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Предобработка данных

Исключим выбросы:

```
#Данные распределены нормально. Поэтому можем удалить выбросы, обнаруженные методом трех сигм
outliers = pd.DataFrame(index=df_all.index)
for column in df_all:
    zscore = (df_all[column] - df_all[column].mean()) / df_all[column].std()
    outliers[column] = (zscore.abs() > 3)
df_all = df_all[outliers.sum(axis=1)==0]
df_all.shape
```

```
(1000, 13)
```

Используем нормализацию - приведение значений данных в диапазон от 0 до 1 при помощи MinMaxScaler

```
##Нормализация данных
min_max_scaler = MinMaxScaler()
ds_norm=min_max_scaler.fit_transform(np.array(X))
ds_norm[:1]
```

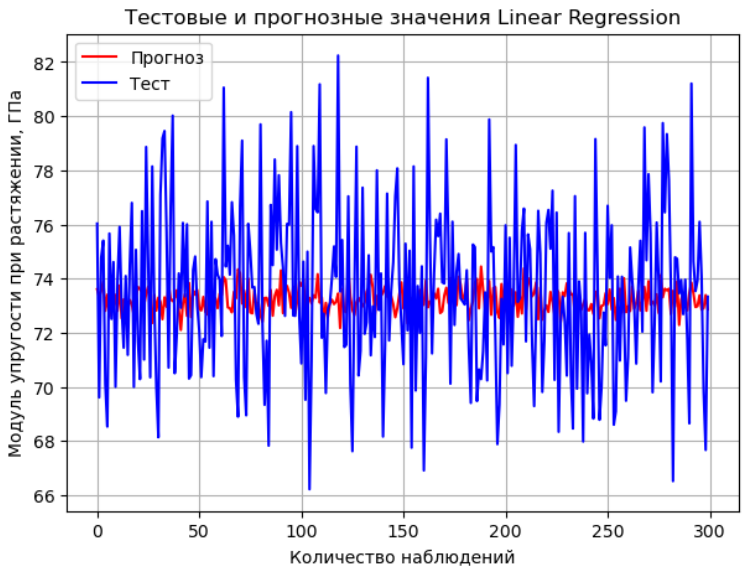
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

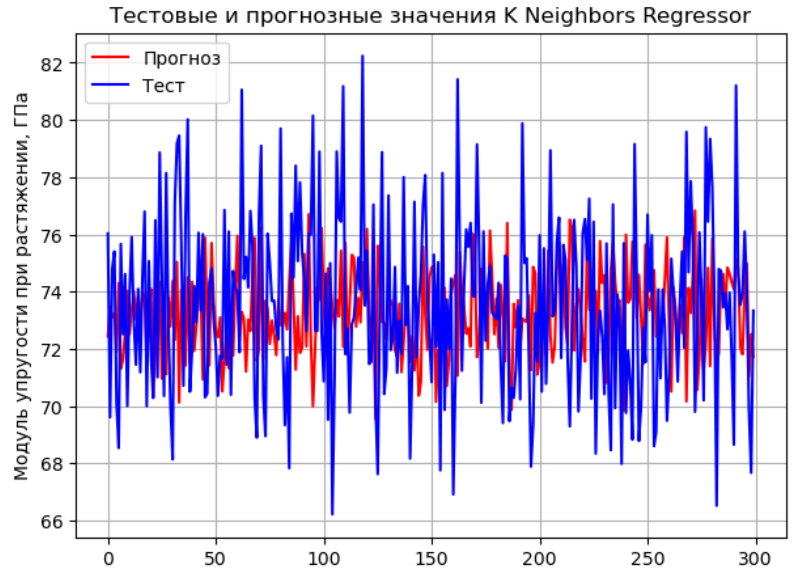
Методы регресионного анализа

- Линейная регресиия
- Метод опорных вектторов
- Метод k-ближайших соседей
- Деревья решений
- Градиентный бустинг
- Случайный лес

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Модуль упругости при растяжении





ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Модуль упругости при растяжении

После подбора лучших гиперпараметров методом GridSearchCV получаем.
MAE - средняя абсолютная ошибка

mae_df1

	Регрессор	MAE
0	RandomForest	2.578553
1	Linear Regression	2.510989
2	KNeighbors	2.803555
3	Support Vector	3.124316
4	GradientBoosting	2.597720
5	DecisionTree	3.551074

Linear Regression Results Train:
Ttain score: 0.02
Linear Regression Results:
lr_MAE: 396
lr_MAPE: 0.18
lr_MSE: 240591.32
lr_RMSE: 490.50
Test score: 0.02

K Neighbors Regressor Results Train:
Train score: 0.24
K Neighbors Regressor Results:
KNN_MAE: 3
KNN_MAPE: 0.04
KNN_MSE: 12.45
KNN_RMSE: 3.53
Test score: -0.30

```
Gradient Boosting Regressor Results Train:
Train score: 0.49
Gradient Boosting Regressor Results:
GBR_MAE:  3
GBR_MAPE: 0.04
GBR_MSE: 10.34
GBR_RMSE: 3.22
Test score: -0.08
```

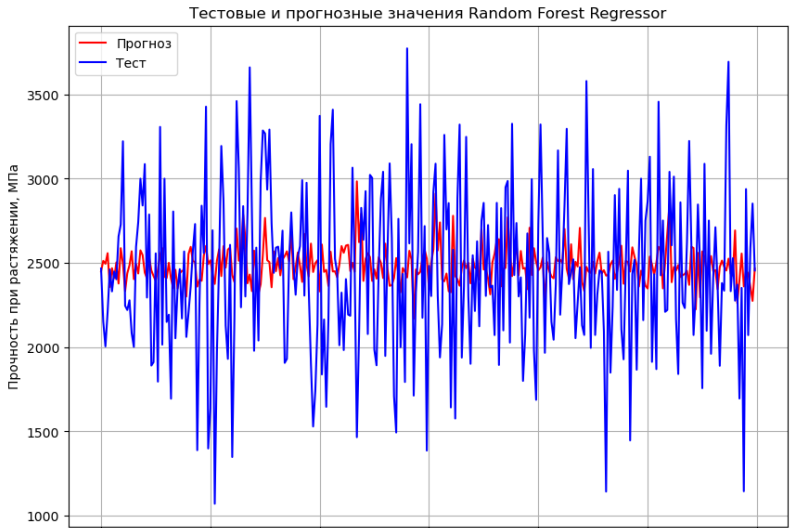
```
Support Vector Regression Results Train:
Train score: 0.87
Support Vector Regression Results:
SVR_MAE: 3
SVR_MAPE: 0.04
SVR_MSE: 15.88
SVR_RMSE: 3.98
Test score: -0.66
```

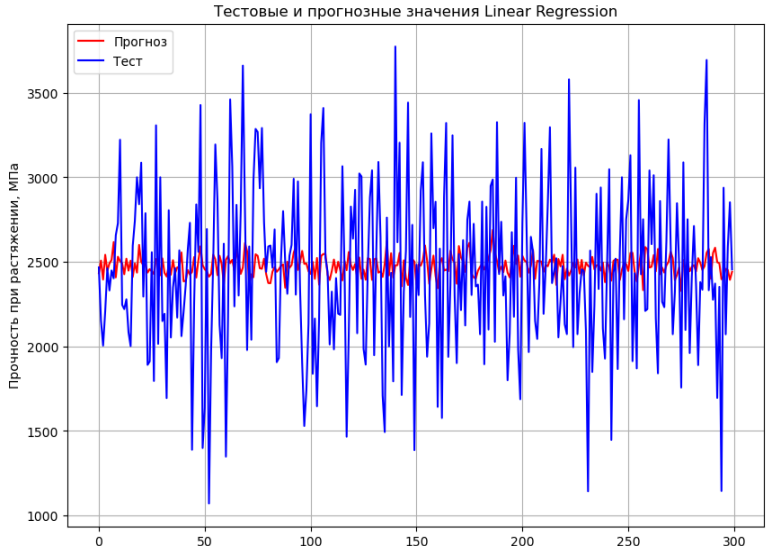
```
Decision Tree Regressor Results Train:
Train score: 1.00
Decision Tree Regressor Results:
DTR_MAE:  4
DTR_MSE: 20.24
DTR_RMSE: 4.50
DTR_MAPE: 0.05
Test score: -1.12
```

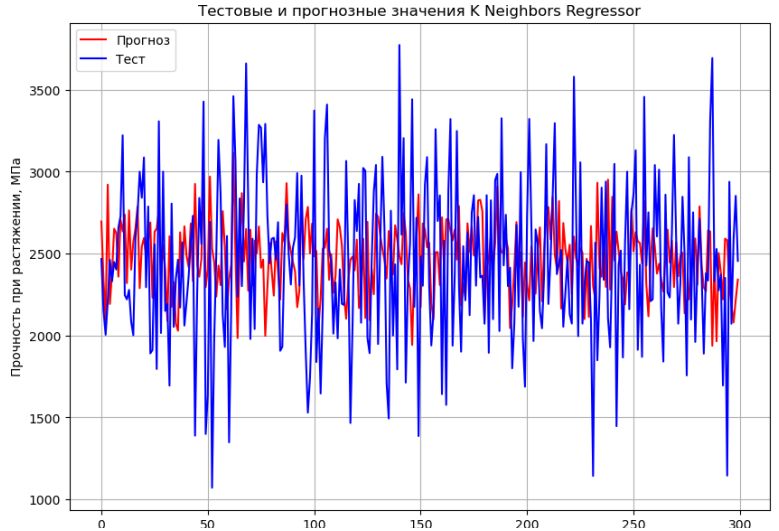
```
Random Forest Regressor Results Train:
Train score: 0.40
Random Forest Regressor Results:
RF_MAE:  3
RF_MAPE: 0.03
RF_MSE: 10.19
RF_RMSE: 3.19
Test score: -0.07
```

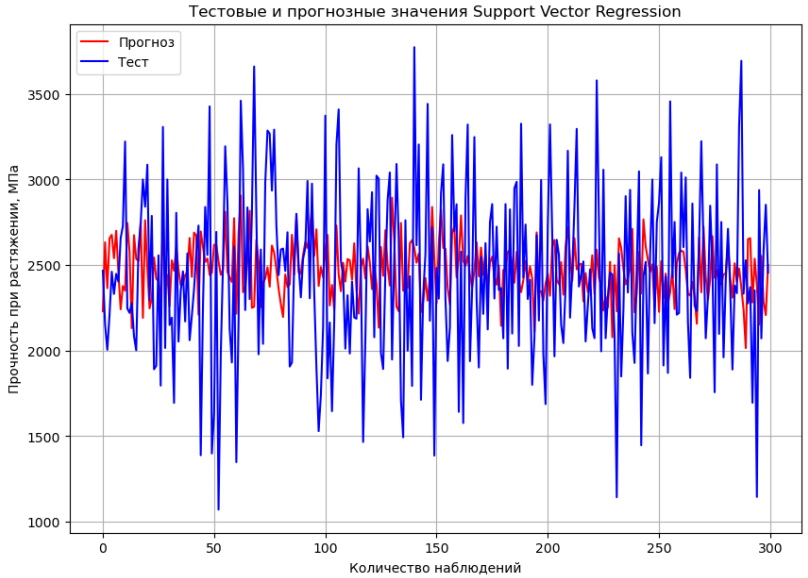
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

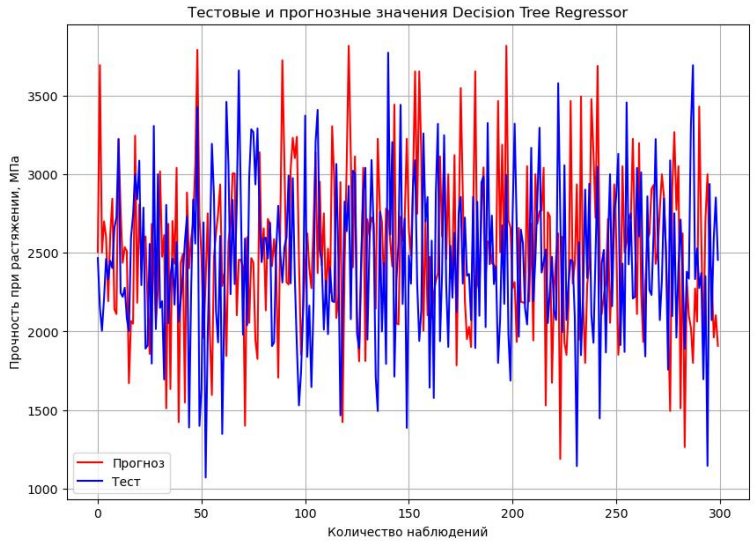
Прочность при растяжении

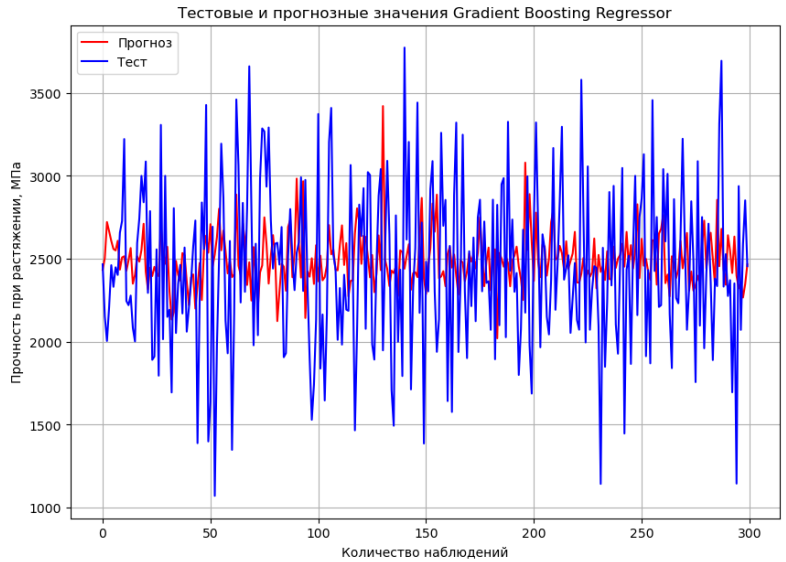












ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Модель для соотношения матрица-наполнитель

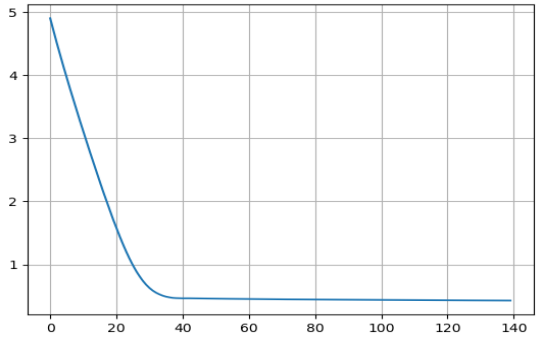


График ошибки при обучении MLPRegressor и визуализация тестовых/прогнозных значений для данной модели

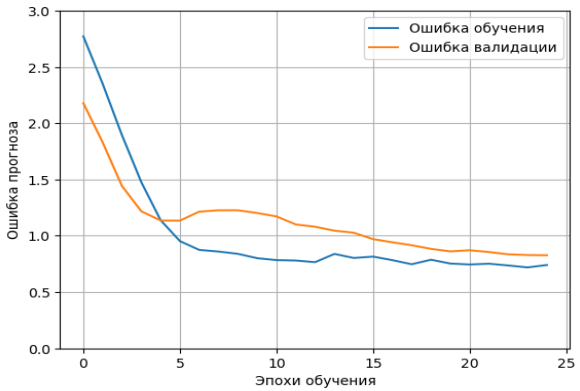


График ошибок при построении НС Sequential и визуализация тест/прогноз для данной модели

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

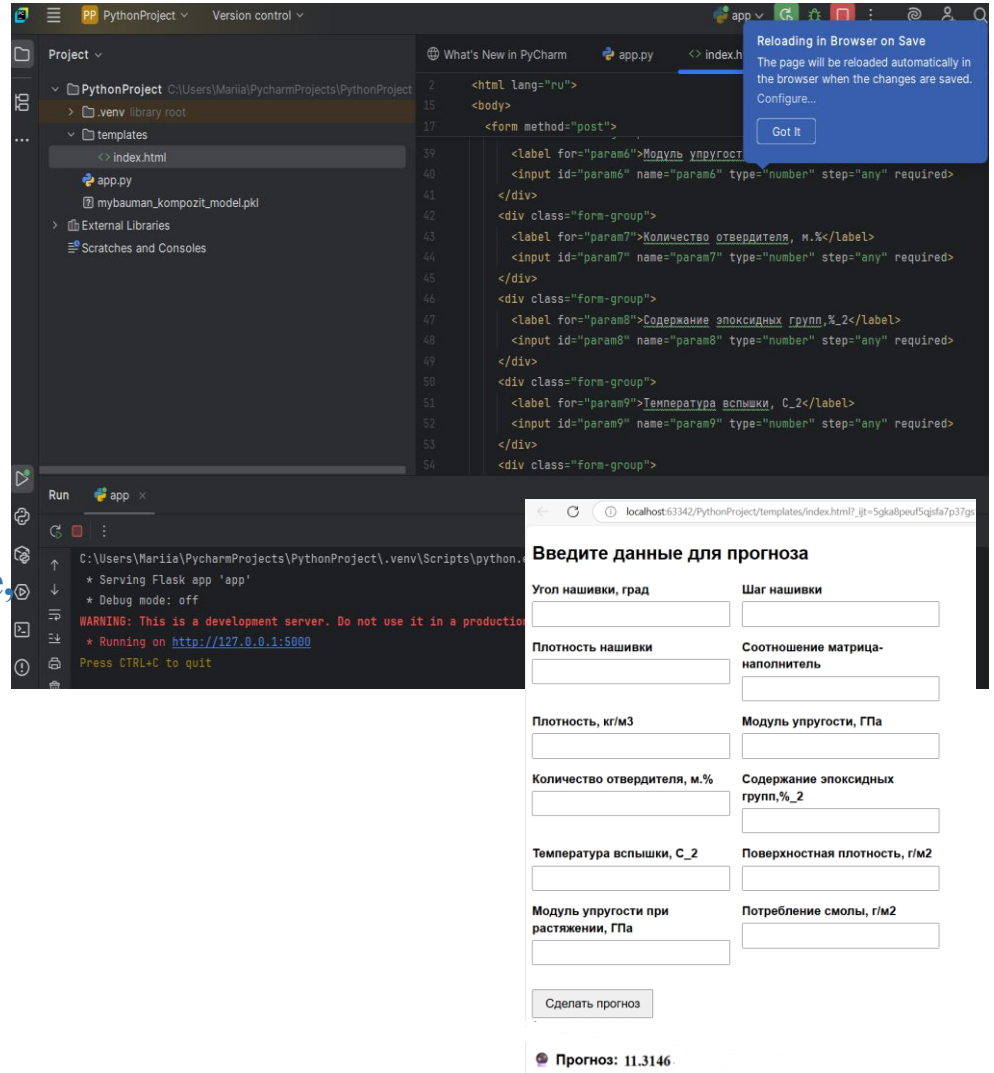
Приложение и выводы

К сожалению, с практической стороны результаты работы оказались неудовлетворительными, и не могут быть применены на практике.

Рекомендации:

- **необходима консультация со специалистом в предметной области;**
- **получить дополнительную информацию по исходным данным;**
- **применить другие модели нейронных сетей;**
- **применить преобразование исходных данных (логарифмирование, возведение в степень и др.);**
- **изучить вопрос разделения данных на отдельные кластеры.**

Приложение для расчета Прочности при растяжении при использовании среды разработки PyCharm



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

do.bmstu.ru