



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Прогнозирование конечных свойств новых материалов (композиционных материалов)

Семина Юлия Васильевна



Постановка задачи

- 1 Разведочный анализ набора данных
- 2 Изучение методов для решения поставленной задачи
- 3 Применение методов в разработке и обучении моделей
- 4 Разработка пользовательского приложения
- 5 Выгрузка в удаленный репозиторий

Исходный набор данных:

X_bp.xlsx

– данные о
параметрах
базальтопластика
(1024x11)

X_nup.xlsx

– данные о
компоновке композитов
(1041x4)



Объединение по типу INNER

```
In [ ]: # объединение датасетов
df = X_bp.join(X_nup, how='inner')
df.shape
```

```
Out[ ]: (1023, 13)
```

Информация о типах данных

```
df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 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, C_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
```

Описательная статистика данных

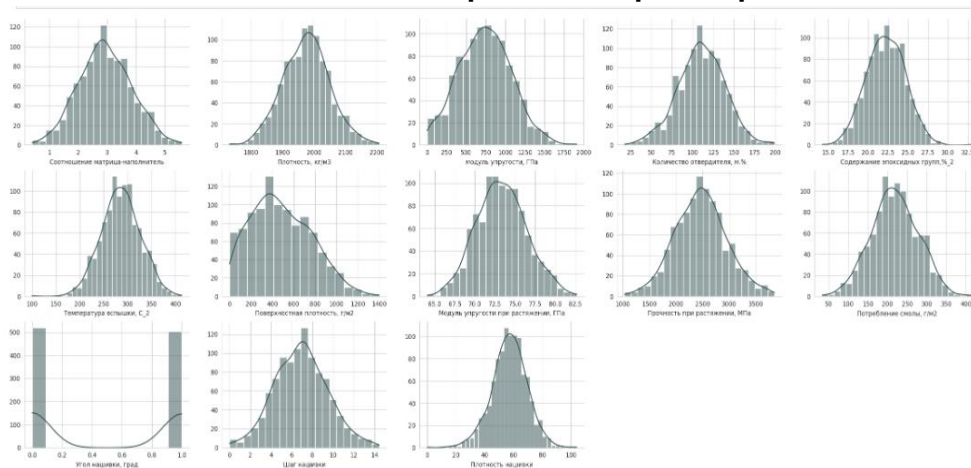
	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0000	2.9304	0.9132	0.3894	2.3179	2.9069	3.5527	5.5917
Плотность, кг/м3	1023.0000	1975.7349	73.7292	1731.7646	1924.1555	1977.6217	2021.3744	2207.7735
модуль упругости, ГПа	1023.0000	739.9232	330.2316	2.4369	500.0475	739.6643	961.8125	1911.5365
Количество отвердителя, м.%	1023.0000	110.5708	28.2959	17.7403	92.4435	110.5648	129.7304	198.9532
Содержание эпоксидных групп,%_2	1023.0000	22.2444	2.4063	14.2550	20.6080	22.2307	23.9619	33.0000
Температура вспышки, C_2	1023.0000	285.8822	40.9433	100.0000	259.0665	285.8968	313.0021	413.2734
Поверхностная плотность, г/м2	1023.0000	482.7318	281.3147	0.6037	266.8166	451.8644	693.2250	1399.5424
Модуль упругости при растяжении, ГПа	1023.0000	73.3286	3.1190	64.0541	71.2450	73.2688	75.3566	82.6821
Прочность при растяжении, МПа	1023.0000	2466.9228	485.6280	1036.8566	2135.8504	2459.5245	2767.1931	3848.4367
Потребление смолы, г/м2	1023.0000	218.4231	59.7359	33.8030	179.6275	219.1989	257.4817	414.5906
Угол нашивки, град	1023.0000	0.4917	0.5002	0.0000	0.0000	0.0000	1.0000	1.0000
Шаг нашивки	1023.0000	6.8992	2.5635	0.0000	5.0800	6.9161	8.5863	14.4405
Плотность нашивки	1023.0000	57.1539	12.3510	0.0000	49.7992	57.3419	64.9450	103.9889

Пропусков: 0, Дубликатов 0.

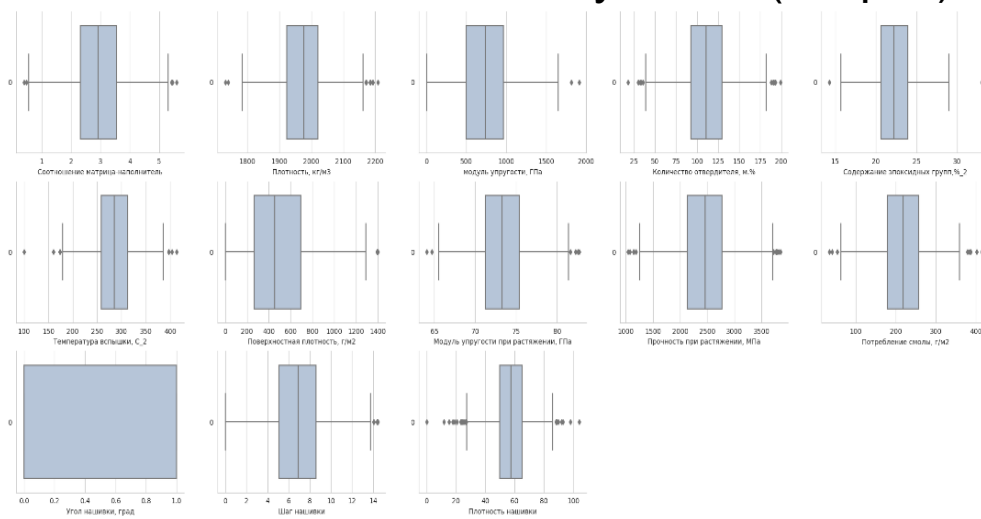


Графическое распределение данных

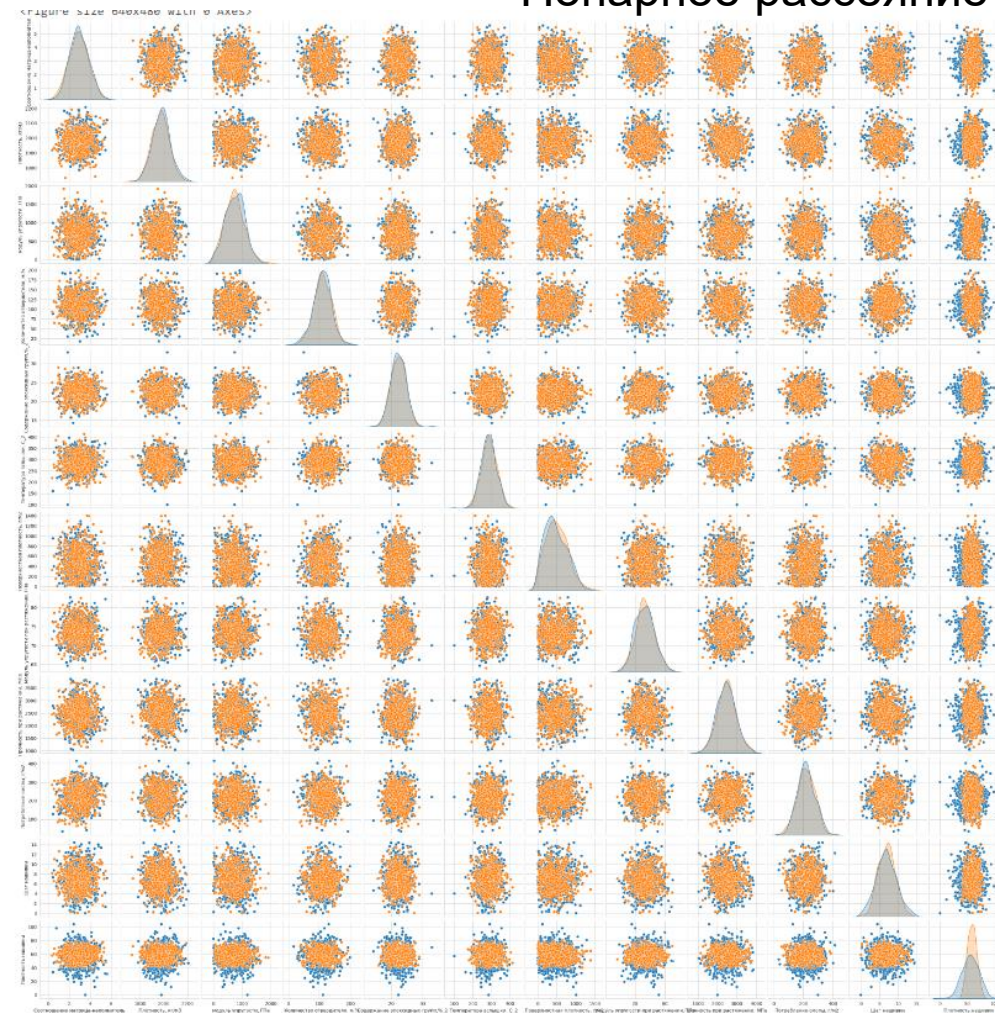
Гистограммы распределения



«Ящики с усами» (boxplot)



Попарное рассеяние





Поиск выбросов. Очистка данных

Методом трех сигм найдено выбросов: 24

Методом межквартильных расстояний найдено выбросов: 93

```
In [ ]: # Удалить выбросы методом 3-х сигм
outliers = pd.DataFrame(index=df.index)
for column in df:
    zscore = (df[column] - df[column].mean()) / df[column].std()
    outliers[column] = (zscore.abs() > 3)
df = df[outliers.sum(axis=1)==0]
df.shape
```

Out[]: (1000, 13)

При разведочном анализе выявлено:

- пропусков 0, дубликатов 0;
- 1000x13 после удаления выбросов;
- корреляция между признаками близка к 0.

```
In [ ]: # получим количество уникальных значений в каждом столбце
df.nunique()
```

```
Out[ ]: Соотношение матрица-наполнитель      994
Плотность, кг/м3                             993
модуль упругости, ГПа                         999
Количество отвердителя, м.%                  983
Содержание эпоксидных групп,%_2              983
Температура вспышки, С_2                     982
Поверхностная плотность, г/м2                984
Модуль упругости при растяжении, ГПа         984
Прочность при растяжении, МПа                984
Потребление смолы, г/м2                      983
Угол нашивки, град                           2
Шаг нашивки                                  968
Плотность нашивки                            967
dtype: int64
```



Препроцессинг данных

1 Выделение признаков и целевых переменных X, y

2 Разделение данных на тренировочную и тестовую выборки 70/30%

```
1 # Разделяем на тренировочную и тестовую выборку
2 x1_train_raw, x1_test_raw, y1_train, y1_test = train_test_split(x1, y1, test_size=0.3, random_state=RANDOM_STATE)
3 print('x1_train_raw:', x1_train_raw.shape, 'y1_train:', y1_train.shape)
4 print('x1_test_raw:', x1_test_raw.shape, 'y1_test:', y1_test.shape)
```

```
x1_train_raw: (700, 11) y1_train: (700, 1)
x1_test_raw: (300, 11) y1_test: (300, 1)
```

3 Препроцессинг данных – стандартизация

```
1 # препроцессинг входных переменных
2 scaler = StandardScaler()
3 scaled2 = ColumnTransformer(transformers=[('scaler', scaler, x2_columns)])
4 x2_train = scaled2.fit_transform(x2_train_raw)
5 x2_test = scaled2.transform(x2_test_raw)
```

```
[ ] 1 # массивы y_train и y_test принимают исходные значения
2 y2_train = y2_train['Прочность при растяжении, МПа'].values
3 y2_test = y2_test['Прочность при растяжении, МПа'].values
```



Разработка и обучение модели

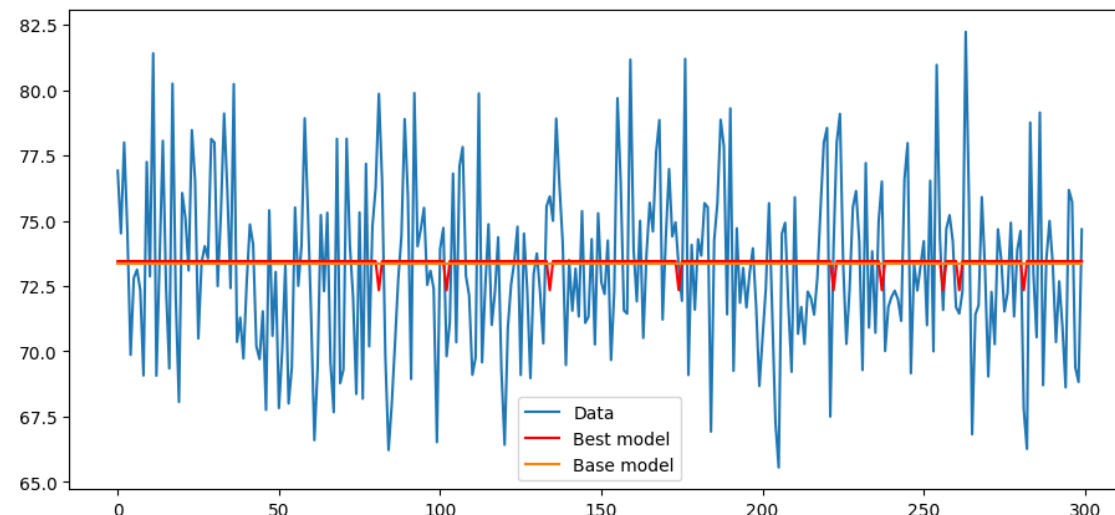
Модель предсказания Модуля упругости при растяжении, ГПа

Метрики модели в результате подбора гиперпараметров кросс-валидации

DecisionTreeRegressor (criterion='absolute_error', max_depth=1, max_features=4, random_state=38, splitter='random')	-0.0245	-3.0199	-2.4248	-0.0331	-7.7355
---	---------	---------	---------	---------	---------

Точность на тренировочных и тестовых данных

Данные	R2	RMSE	MAE	MAPE	max_error
Модуль упругости, train	0.006730	-3.023163	-2.410568	-0.032922	-9.400634
Модуль упругости, test	-0.008675	-3.300236	-2.665111	-0.036518	-8.782905



Decision Tree



Разработка и обучение модели

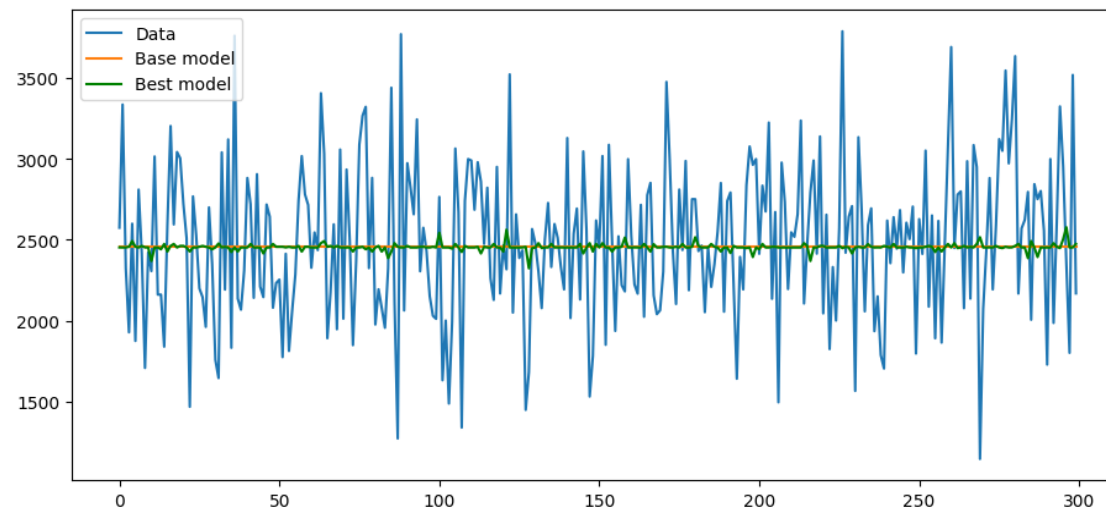
Модель предсказания Прочности при растяжении, МПа

Метрики модели в результате подбора гиперпараметров кросс-валидации

GradientBoostingRegressor (max_depth=6, max_features=1, n_estimators=1, random_state=3128)	-0.018646	484.82637	379.50171	0.168228	1269.36172
GradientBoostingRegressor (loss='absolute_error', max_depth=5, max_features=1, n_estimators=1, random_state=3128)	-0.021328	485.48386	380.46830	0.168037	1268.91142

Точность на тренировочных и тестовых данных

Данные	R2	RMSE	MAE	MAPE	max_error
Прочность при растяжении, train	0.033631	477.8012	374.10805	0.165670	1415.5576
Прочность при растяжении, test	0.00914	483.0871	389.41809	0.16625	1372.9174



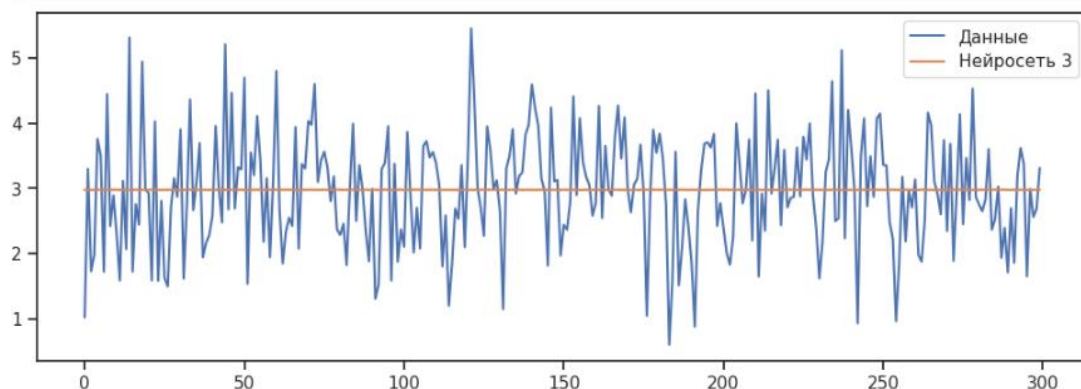
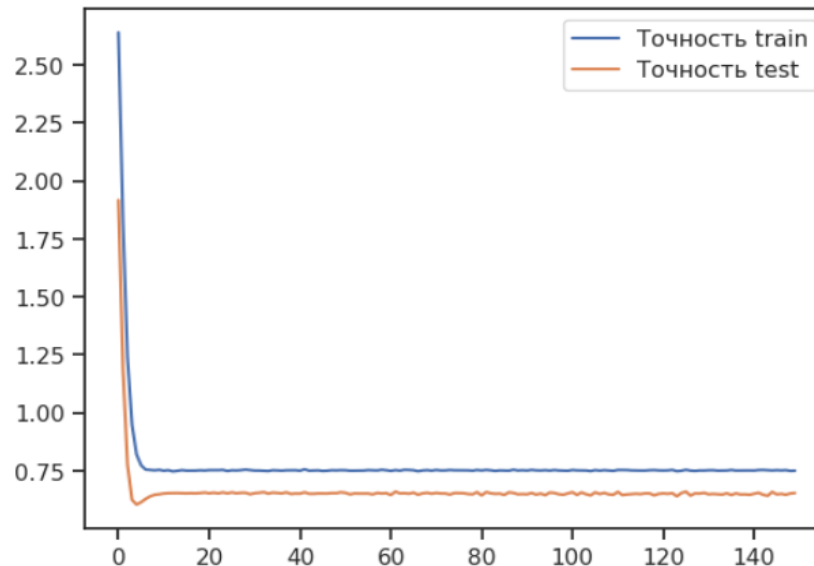
Gradient Boosting



Нейронная сеть для рекомендаций Соотношение матрица-наполнитель

```
modelMF_3 = tf.keras.Sequential([
    keras.layers.Dense(32, input_dim=12, activation = 'sigmoid'),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(64, activation = 'sigmoid'),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(32, activation = 'sigmoid'),
    keras.layers.Dense(16, activation = 'sigmoid'),
    keras.layers.Dense(1, activation = 'linear')])
modelMF_3.compile(optimizer='adam', loss='mse', metrics=['mae'])
modelMF_3.summary()
history = modelMF_3.fit(X_train, y_train,
                        epochs=150,
                        validation_split=0.1,
                        verbose=2)
```

Epoch 150/150
20/20 - 0s - loss: 0.8730 - mae: 0.7498 - val_loss: 0.6475 - val_mae: 0.6535 - 75ms/epoch - 4ms/step



	R2	RMSE	MAE	MAPE	max_error
Соотношение матрица-наполнитель, тренировочный	-0.002767	0.923670	0.741881	0.330315	2.613384
Соотношение матрица-наполнитель, тестовый	-0.000423	0.874308	0.695222	0.296718	2.479797



Разработка приложения

Интеграция модели нейросети из TensorFlow. Python. Flask. Jinja

← 127.0.0.1:5000 Рекомендации для соотношения матрица-наполнитель

Рекомендации для соотношения матрица-наполнитель

Плотность, кг/м3 (1700...2300)

Модуль упругости, ГПа (2...2000)

Количество отвердителя, м.% (17...200)

Содержание эпоксидных групп, %_2 (14...34)

Температура вспышки, С_2 (100...414)

Поверхностная плотность, г/м2 (0.6...1400)

Модуль упругости при растяжении, ГПа (64...83)

Прочность при растяжении, МПа (1036...3849)

Потребление смолы, г/м2 (33...414)

Угол нашивки, град (0 или 90)

Шаг нашивки (0...15)

Плотность нашивки (0...104)

Входные переменные:

	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	1880.0	622.0	111.86	22.267857	284.615385	470.0	73.333333	2455.555556	220.0	90.0	4.0	60.0

Результат модели:

Соотношение матрица-наполнитель
[2.9509132]



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Спасибо за внимание!

github.com/maaliskuussa



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана



do.bmstu.ru