ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по курсу

«Data Science»

по теме:

Прогнозирование конечных свойств новых материалов (композиционных материалов)

Слушатель: Антонов С.А.

Актуальность темы

- Теоретический анализ полимерных композиционных материалов путём построения моделей на основе методов вычислительной механики и прогнозирование их эффективных характеристик с завершающей оценкой их надёжности является актуальным.
- Созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

Характеристики анализируемого датасета

- В задании представлены два файла с данными в формате Excel-таблицы. Для формирование единого массива данных, произведено сведения обоих файлов в один.
- Объем и характеристики датасета: в сведенном датасете 1023 записи по каждому показателю, пропуски отсутствуют (нет пустых значений),

Загружаем данные из exel файлов

```
#Считываем данные в датасеты
df1 = pd.read_excel('Datasets\X_bp.xlsx')
df2 = pd.read_excel('Datasets\X_nup.xlsx')
```

```
#Посмотрим на первые 5 строк первого датасета
df1.head()
```

	Unnamed: 0	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностна плотность, г/м
0	0.0	1.857143	2030.0	738.736842	30.00	22.267857	100.000000	210.
1	1.0	1.857143	2030.0	738.736842	50.00	23.750000	284.615385	210.
2	2.0	1.857143	2030.0	738.736842	49.90	33.000000	284.615385	210.
3	3.0	1.857143	2030.0	738.736842	129.00	21.250000	300.000000	210.
4	4.0	2.771331	2030.0	753.000000	111.86	22.267857	284.615385	210.

	0	1	2	3	4
Unnamed: 0	0.000000	1.000000	2.000000	3.000000	4.000000
Соотношение матрица-наполнитель	1.857143	1.857143	1.857143	1.857143	2.771331
Плотность, кг/м3	2030.000000	2030.000000	2030.000000	2030.000000	2030.000000
модуль упругости, ГПа	738.736842	738.736842	738.736842	738.736842	753.000000
Количество отвердителя, м.%	30.000000	50.000000	49.900000	129.000000	111.860000
Содержание эпоксидных групп,%_2	22.267857	23.750000	33.000000	21.250000	22.267857
Температура вспышки, С_2	100.000000	284.615385	284.615385	300.000000	284.615385
Поверхностная плотность, г/м2	210.000000	210.000000	210.000000	210.000000	210.000000
Модуль упругости при растяжении, ГПа	70.000000	70.000000	70.000000	70.000000	70.000000
Прочность при растяжении, МПа	3000.000000	3000.000000	3000.000000	3000.000000	3000.000000
Потребление смолы, г/м2	220.000000	220.000000	220.000000	220.000000	220.000000
Угол нашивки, град	0.000000	0.000000	0.000000	0.000000	0.000000
Шаг нашивки	4.000000	4.000000	4.000000	5.000000	5.000000
Плотность нашивки	57.000000	60.000000	70.000000	47.000000	57.000000

- *В таблице представлены основные характеристики параметров датасета: количество элементов, средние значения параметров, минимальные и максимальные значения, а также медианные значения*

Используемые библиотеки и модули

Импорт внешних библиотек и модулей

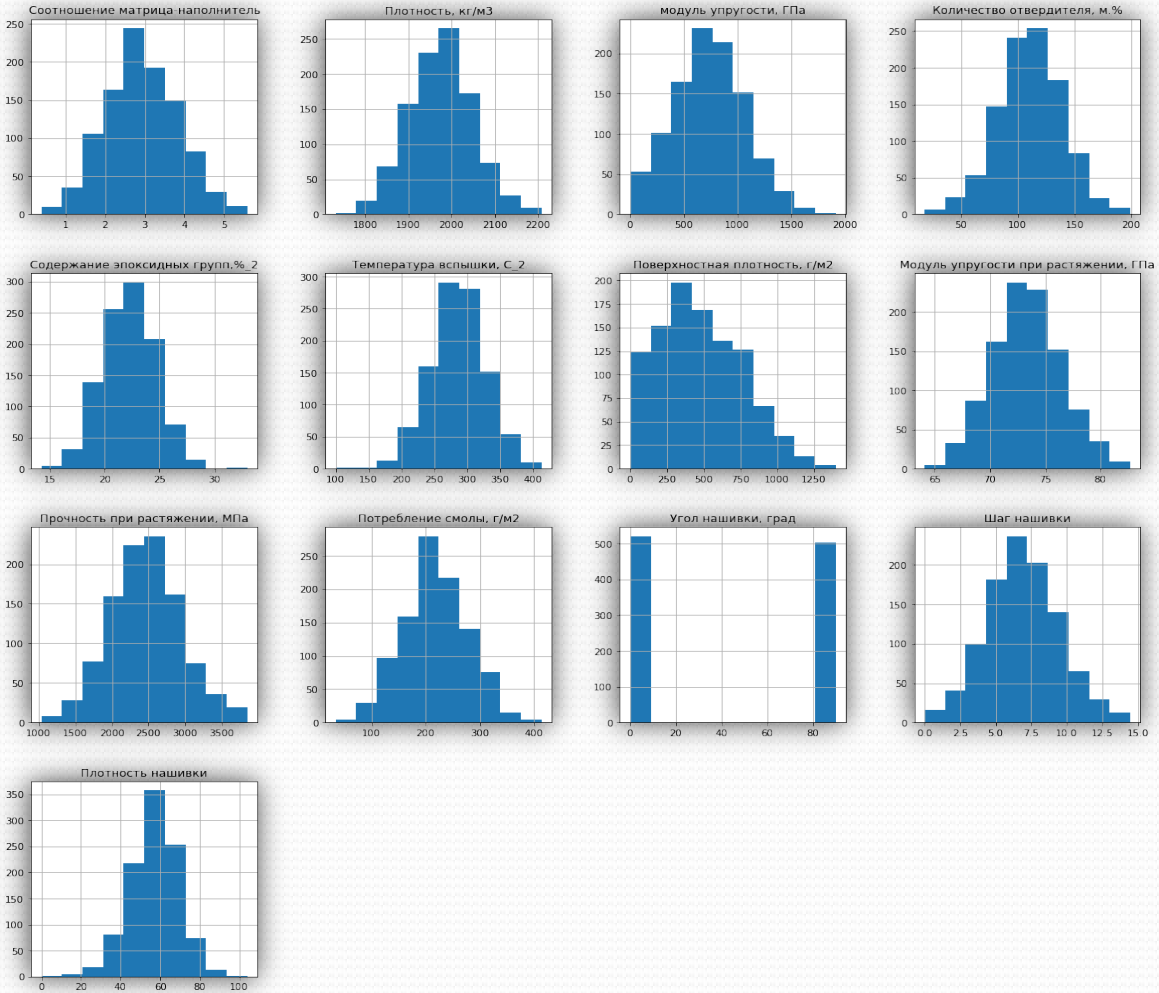
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn
import tensorflow as tf
import seaborn as sns

from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder, MinMaxScaler, StandardScaler
from tensorflow import keras
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from pandas import read_excel, DataFrame, Series

from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from keras.wrappers.scikit_learn import KerasClassifier
from keras.models import Sequential
from keras.layers import Activation, Dropout
from numpy.random import seed
```

Этапы обработки данных

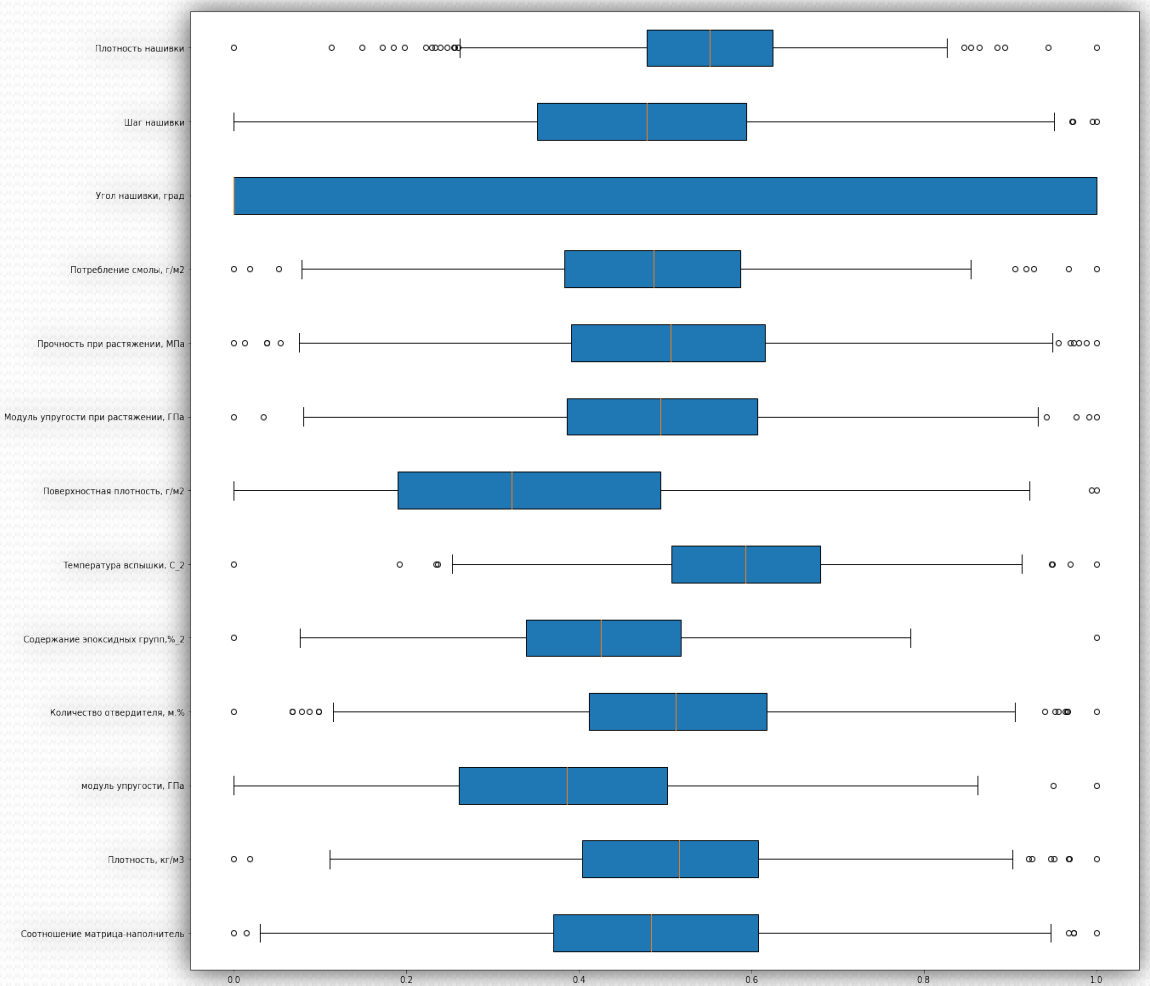
- На первом этапе были проанализированы все признаки для определения их максимальных, минимальных, средних и медианных значений, а также была проведена визуализация данных.
- После этого было проведено исключение выбросов данных, то есть точек данных, которые лежали вдали от обычного распределения данных. Диаграмма ящиков с усами является отличным способом визуализации таких значений.
- На заключительном этапе была проведена нормализация данных.
- После нормализации данных был также проведен анализ взаимосвязи переменных друг с другом. Были построены графики попарного рассеяния переменных, а также была определена корреляция между переменными
- По результатам предобработки данных можно сделать следующий вывод. Между параметрами модели не наблюдается корреляций и очевидных связей. Число выбросов оказалось незначительным.
- Для рекомендации соотношения «матрица-наполнитель» была разработана простая модель глубокого обучения с помощью Keras.

Этапы обработки данных

Гистограммы до обработки



Диаграммы размаха до удаления выбросов



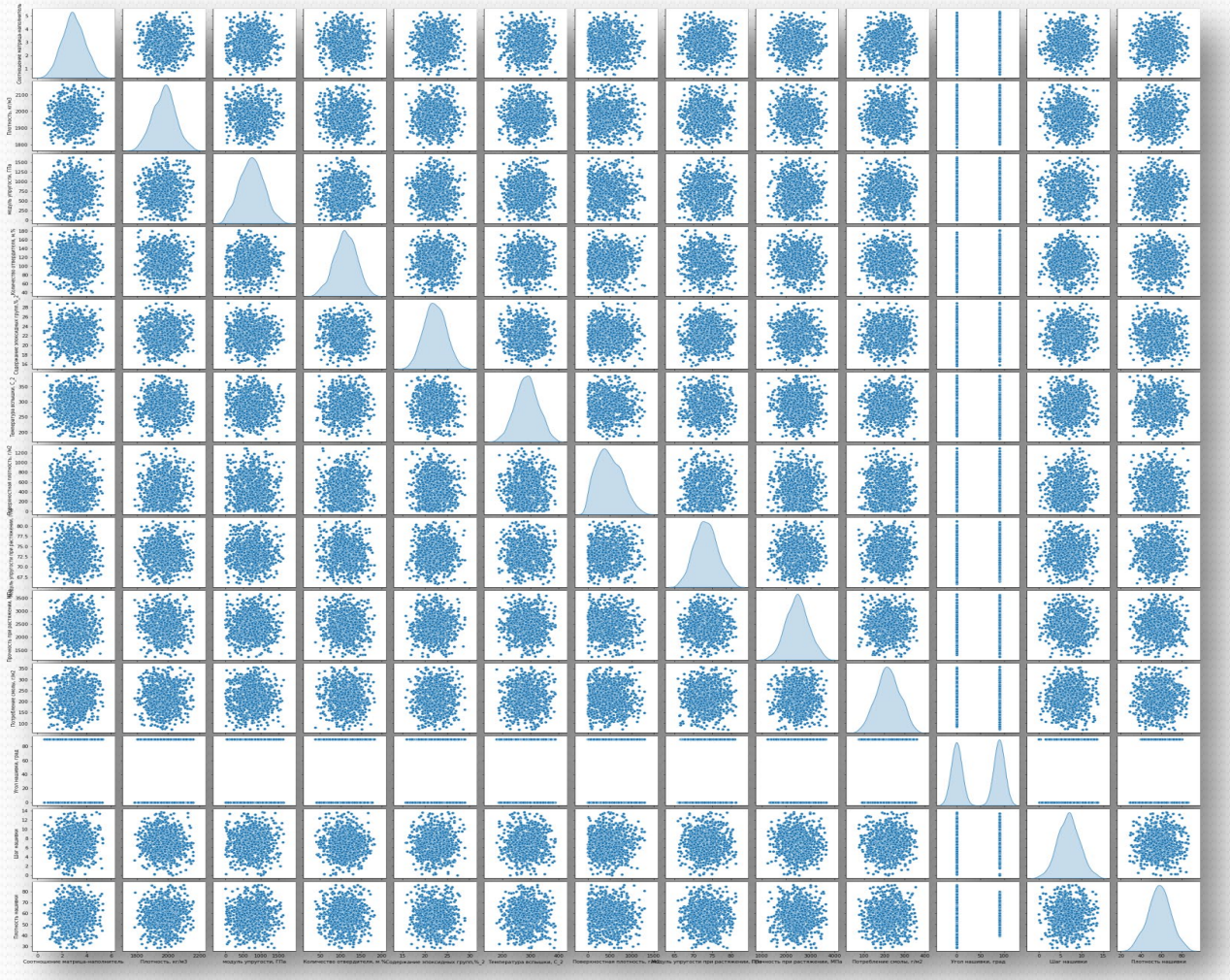
Этапы обработки данных

Описательная статистика датасета после очистки выбросов

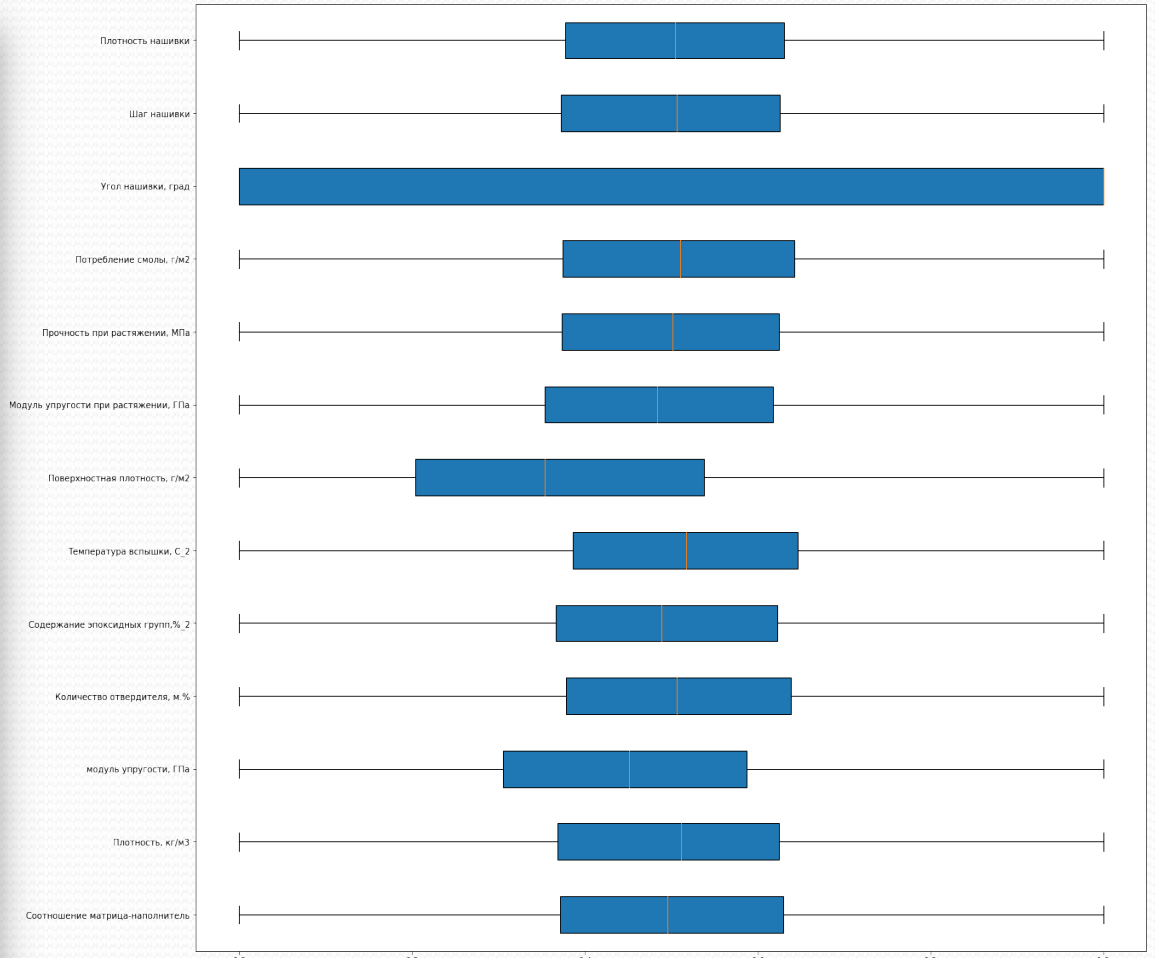
	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град
count	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000	837.000000
mean	0.000868	0.587569	0.216978	0.032887	0.006601	0.085021	0.142062	0.021821	0.742763	0.064855	0.013891
std	0.000275	0.056993	0.092124	0.008474	0.000926	0.014082	0.081065	0.002203	0.056655	0.017766	0.013515
min	0.000163	0.444650	0.000709	0.011339	0.004113	0.049402	0.001902	0.016105	0.590461	0.021630	0.000000
25%	0.000679	0.548948	0.151021	0.027292	0.005925	0.075135	0.078825	0.020292	0.706068	0.052063	0.000000
50%	0.000857	0.585227	0.219229	0.032910	0.006589	0.083934	0.138593	0.021720	0.747345	0.064468	0.022461
75%	0.001052	0.626059	0.280808	0.038817	0.007208	0.094452	0.199600	0.023319	0.782554	0.076808	0.026792
max	0.001593	0.743130	0.476145	0.055088	0.009122	0.123083	0.368343	0.027834	0.877580	0.114133	0.034285

Этапы обработки данных

Гистограммы рассеяния после очистки выбросов

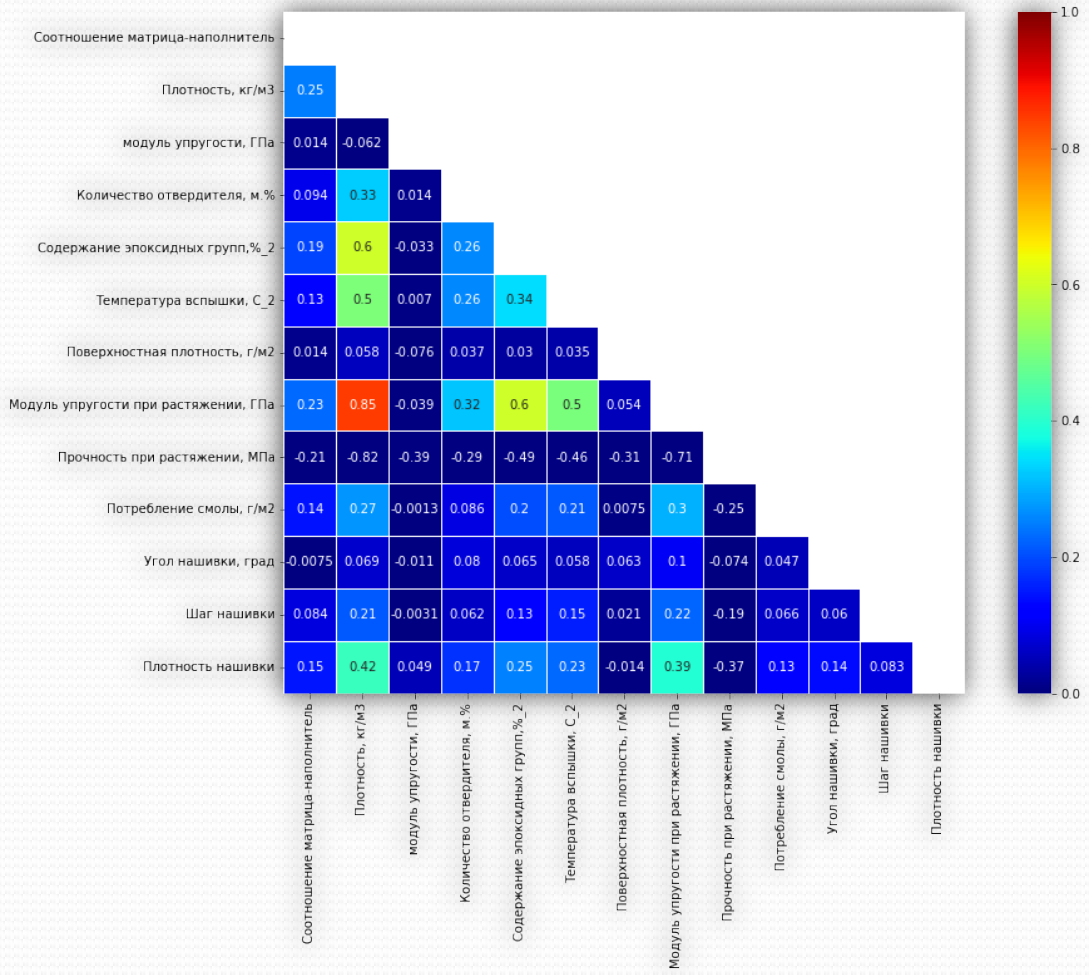


Диаграммы размаха после очистки выбросов

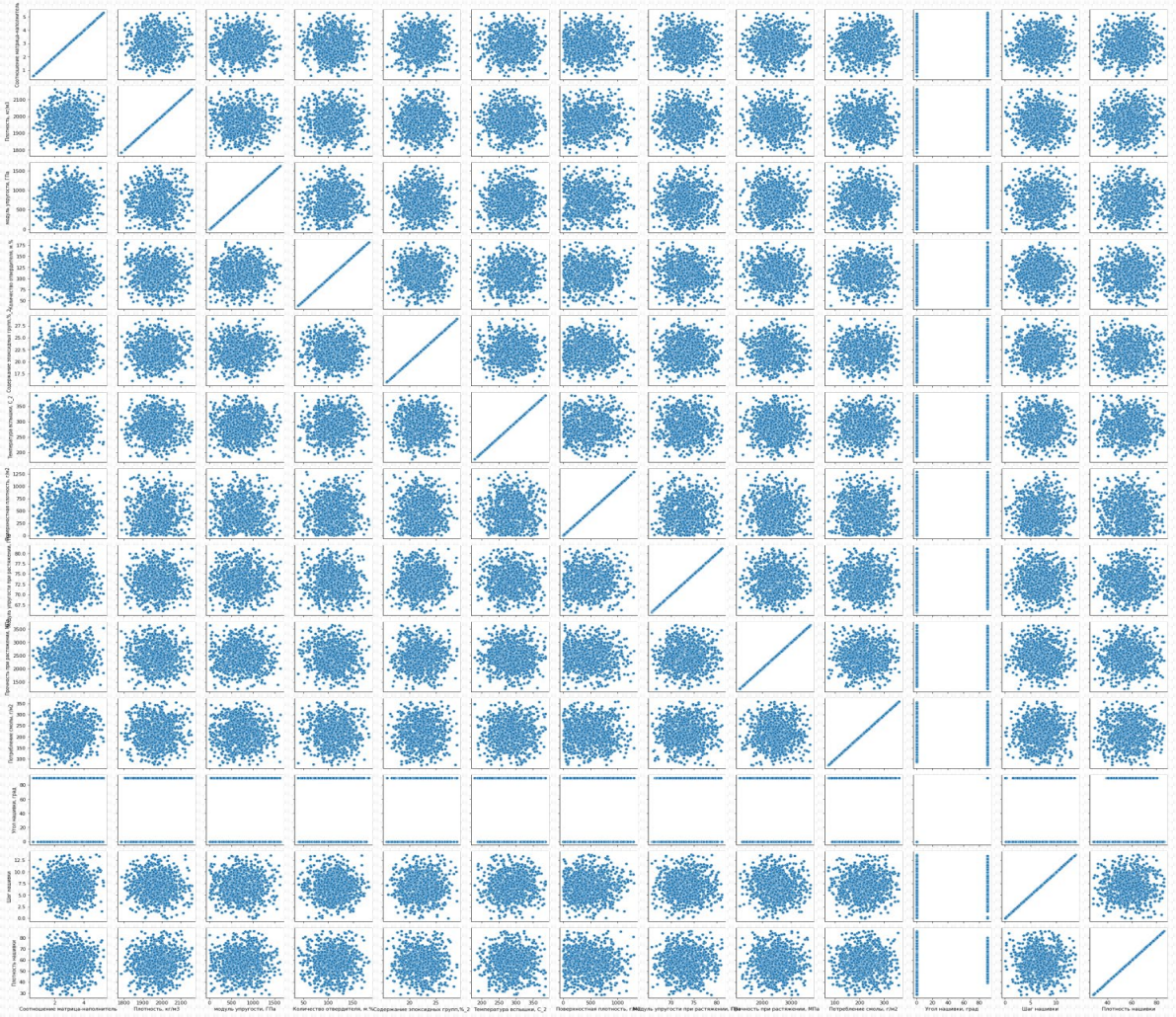


Этапы обработки данных

Матрица корреляции датасета после очистки выбросов



Матрица попарной зависимости датасета



Этапы обработки данных

Анализ датасета на пропуски

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 936 entries, 1 to 1022
Data columns (total 13 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Соотношение матрица-наполнитель   936 non-null    float64
 1   Плотность, кг/м3                  936 non-null    float64
 2   модуль упругости, ГП
 3   Количество отвердите
 4   Содержание эпоксидны
 5   Температура вспышки,
 6   Поверхностная плотно
 7   Модуль упругости при
 8   Прочность при растяж
 9   Потребление смолы, г
 10  Угол нашивки, град
 11  Шаг нашивки
 12  Плотность нашивки
dtypes: float64(13)
memory usage: 102.4 KB
```

Количество выбросов по каждому из столбцов

```
Соотношение матрица-наполнитель      6
Плотность, кг/м3                     9
модуль упругости, ГПа                2
Количество отвердителя, м.%         14
Содержание эпоксидных групп,%_2      2
                                     8
/м2                                  2
кении, ГПа                           6
Па                                  11
                                     8
                                     0
                                     4
                                    21
```

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град
0	0.274768	0.651097	0.452951	0.079153	0.607435	0.509164	0.162230	0.272962	0.727777	0.514688	0.0
1	0.274768	0.651097	0.452951	0.630983	0.418887	0.583596	0.162230	0.272962	0.727777	0.514688	0.0
2	0.466552	0.651097	0.461725	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.0
3	0.465836	0.571539	0.458649	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.0
4	0.424236	0.332865	0.494944	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.0
...	...	...	...	...	...	...	...	...	...	...	...
917	0.361662	0.444480	0.560064	0.337550	0.333908	0.703458	0.161609	0.473553	0.472912	0.183151	1.0
918	0.607674	0.704373	0.272088	0.749605	0.294428	0.362087	0.271207	0.462512	0.461722	0.157752	1.0
919	0.573391	0.498274	0.254927	0.501991	0.623085	0.334063	0.572959	0.580201	0.587558	0.572648	1.0
920	0.662497	0.748688	0.454635	0.717585	0.267818	0.466417	0.496511	0.535317	0.341643	0.434855	1.0
921	0.684036	0.280923	0.255222	0.632264	0.888354	0.588206	0.587373	0.552644	0.668015	0.426577	1.0

922 rows × 13 columns

Нормализация данных с помощью метода MinMaxScaler

Этапы разработки и обучение моделей

- Порядок разработки модели для каждого параметра и для каждого выбранного метода можно разделить на следующие этапы:

- 1) Разделение нормализованных данных на обучающую и тестовую выборки (в соотношении 70 на 30%, согласно поставленной задаче)

- 2) Задание сетки гиперпараметров, по которым будет происходить оптимизация модели. В качестве параметра оценки выбран коэффициент детерминации (R^2)

- 3) Оптимизация подбора гиперпараметров модели с помощью выбора по сетке и перекрестной проверки.

- 4) Подстановка оптимальных гиперпараметров в модель и обучение модели на тренировочных данных.

Этапы разработки и обучение моделей

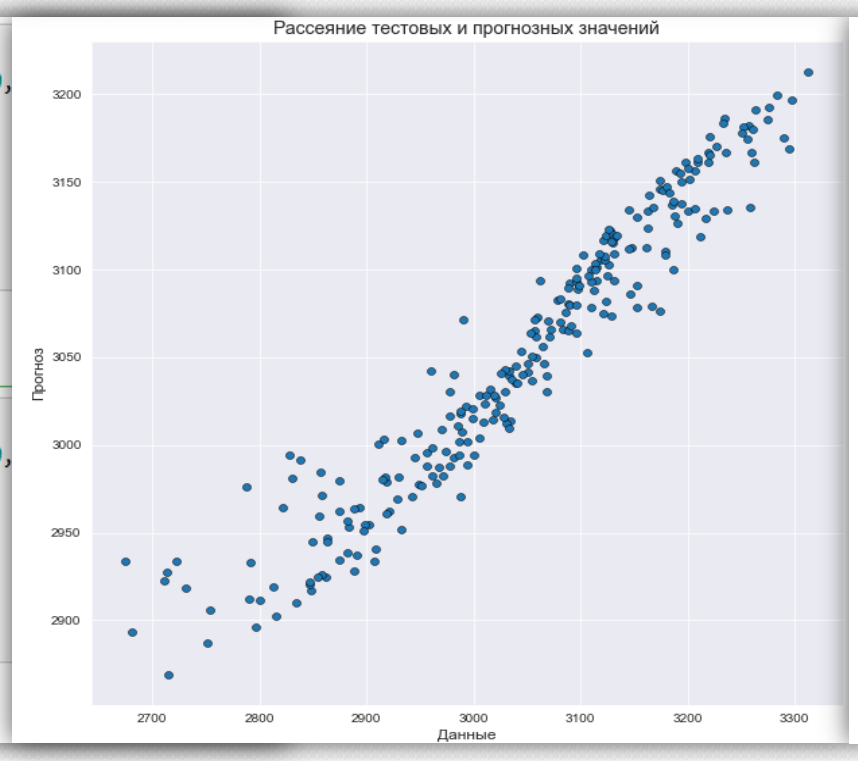
Тестирование моделей

```
#Подставляем оптимальные гиперпараметры в модель
model_base_1 = KNeighborsRegressor(algorithm='brute', leaf_size=10, n_neighbors=100,
#Обучаем модель
model_base_1.fit(Xtrain1_1,Ytrain1_1)
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate(model_base_1, Xtrain1_1,Ytrain1_1)
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate(model_base_1, Xtest1_1,Ytest1_1)
```

```
Средняя абсолютная ошибка: 0.0000
Средняя абсолютная ошибка: 0.1546
```

```
#Подставляем оптимальные гиперпараметры в модель
model_base_2 = KNeighborsRegressor(algorithm='brute', leaf_size=10, n_neighbors=100,
#Обучаем модель
model_base_2.fit(Xtrain2_1,Ytrain2_1)
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate(model_base_2, Xtrain2_1,Ytrain2_1)
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate(model_base_2, Xtest2_1,Ytest2_1)
```

```
Средняя абсолютная ошибка: 0.0000
Средняя абсолютная ошибка: 0.0202
```



	Данные	Прогноз
0	2731.187634	2918.466881
1	3005.029020	3003.927511
2	3095.598833	3100.544471
3	2910.996260	3000.398009
4	3118.143272	3105.850278
...	...	...
247	2985.209632	3010.800784
248	3039.836511	3045.180054
249	2874.154431	2962.153945
250	3067.883163	3030.528017
251	2987.594837	3019.108070

252 rows × 2 columns

Результаты модели k ближайших соседейдля параметра «Прочность при растяжении»

Этапы разработки и обучение моделей

Тестирование моделей

```
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate_2(model_base_11, Xtrain1_2, Ytrain
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate_2(model_base_11, Xtest1_2, Ytest1_
```

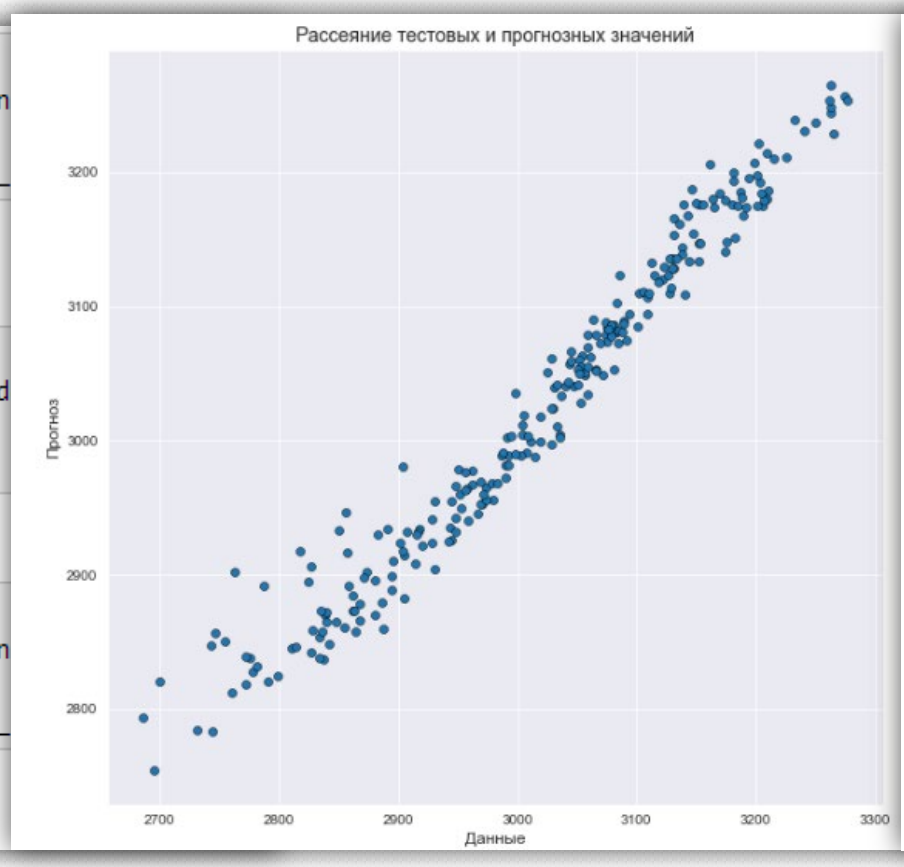
```
Средняя абсолютная ошибка: 0.1288
Средняя абсолютная ошибка: 0.1655
```

```
#Подставляем оптимальные гиперпараметры в модель
model_base_22 = GradientBoostingRegressor(loss='lad', max_d
#Обучаем модель
model_base_22.fit(Xtrain2_2,np.ravel(Ytrain2_2))
```

```
GradientBoostingRegressor(loss='lad', max_depth=2)
```

```
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate_2(model_base_22, Xtrain2_2, Ytrain
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate_2(model_base_22, Xtest2_2, Ytest2_
```

```
Средняя абсолютная ошибка: 0.0050
Средняя абсолютная ошибка: 0.0085
```



	Данные	Прогноз
0	3075.513003	3084.535920
1	3019.465343	2999.180338
2	3058.533799	3034.767054
3	3129.901407	3136.223928
4	3083.324200	3102.851986
...	...	...
247	2685.747892	2793.348055
248	3028.420736	3024.451303
249	3152.256939	3133.135592
250	3032.628666	3041.220577
251	3051.973752	3049.619002

252 rows × 2 columns

Результаты модели повышения градиента для параметра «Прочность при растяжении»

Этапы разработки и обучение моделей

Тестирование моделей

```
best_estimator = model11.best_estimator_
#Выводим гиперпараметры для оптимальной модели
print(best_estimator)
#выводим точность оптимального трейнера
print(model11.best_score_)
```

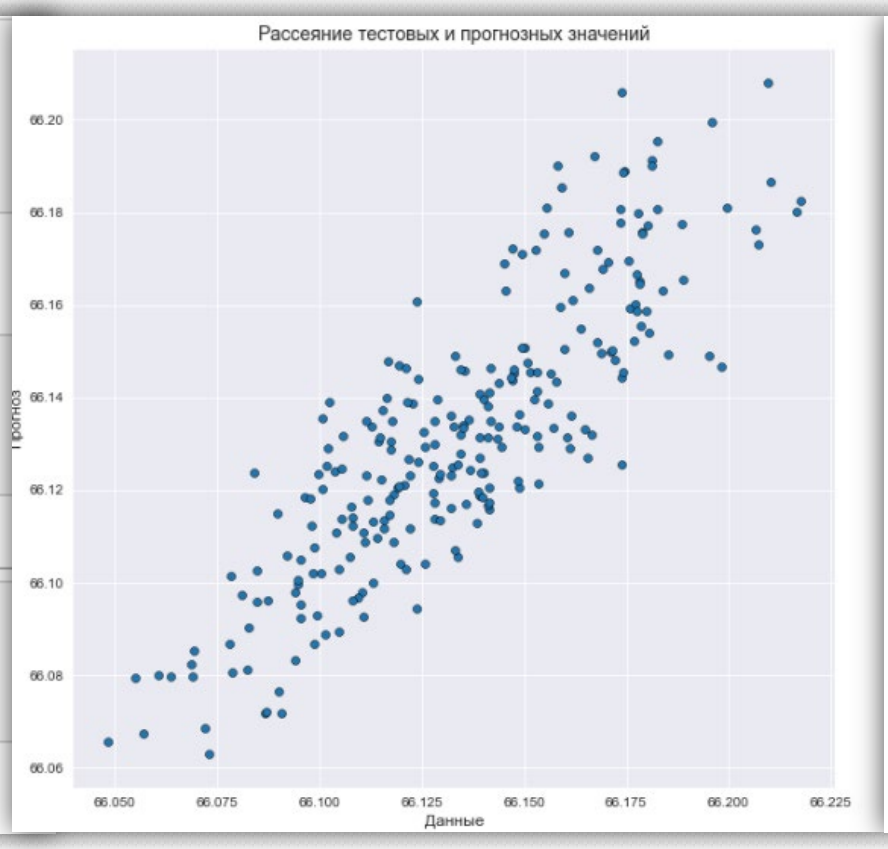
```
LinearRegression()
-0.026111116626460085
```

```
#Подставляем оптимальные гиперпараметры в модель
model_base11 = LinearRegression()
#Обучаем модель
model_base11.fit(Xtrain11,Ytrain11)
```

```
LinearRegression()
```

```
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate(model_base11, Xtrain11,Ytrain11)
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate(model_base11, Xtest11, Ytest11)
```

```
Средняя абсолютная ошибка: 0.1608
Средняя абсолютная ошибка: 0.1509
```



	Данные	Прогноз
0	66.104819	66.103029
1	66.160714	66.175785
2	66.159139	66.185353
3	66.127873	66.125390
4	66.177853	66.179916
...	...	...
247	66.210186	66.186612
248	66.111366	66.123212
249	66.129569	66.123408
250	66.135667	66.117103
251	66.139894	66.118540

252 rows × 2 columns

Результаты модели LinearRegression для параметра «Модуль упругости при растяжении»

Этапы разработки и обучение моделей

Тестирование моделей

```
best_estimator = model111.best_estimator_
#Выводим гиперпараметры для оптимальной модели
print(best_estimator)
#выводим точность оптимального трейнера
print(model111.best_score_)
```

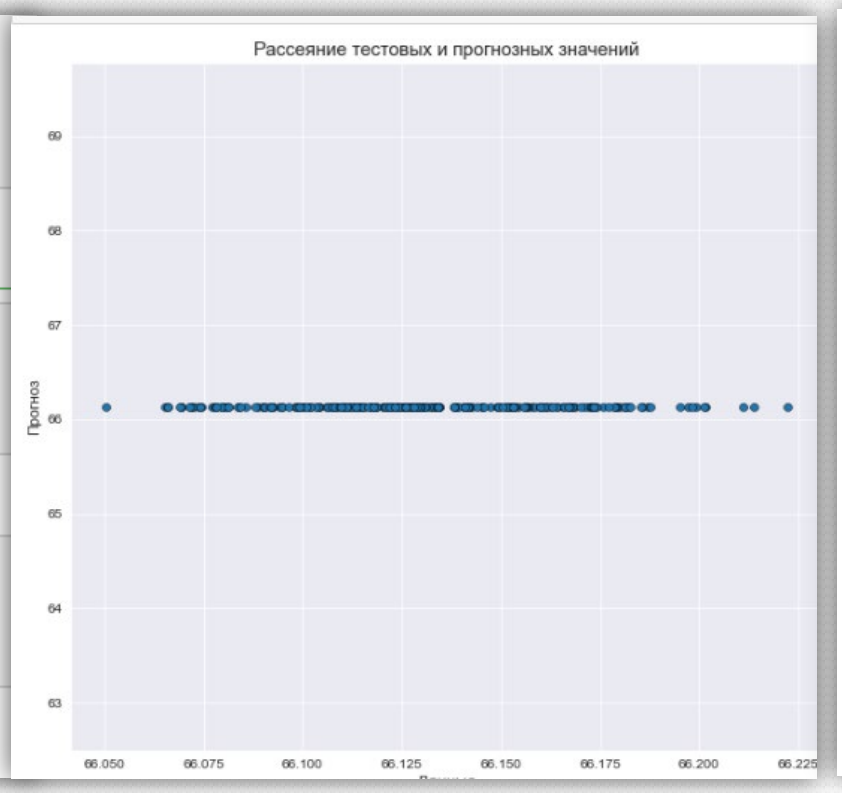
```
SVR(kernel='linear')
-0.0304111012748157
```

```
#Подставляем оптимальные гиперпараметры в модель
model_base111 = SVR(kernel='linear')
#Обучаем модель
model_base111.fit(Xtrain111,np.ravel(Ytrain111))
```

```
SVR(kernel='linear')
```

```
#Оцениваем точность на тренировочном наборе
base_accuracy = evaluate_2(model_base111, Xtrain111, Ytrain111)
#Оцениваем точность на тестовом наборе
base_accuracy = evaluate_2(model_base111,Xtest111, Ytest111)
```

```
Средняя абсолютная ошибка: 0.1581
Средняя абсолютная ошибка: 0.1585
```



	Данные	Прогноз
0	66.123800	66.132373
1	66.151416	66.132373
2	66.167024	66.132373
3	66.161214	66.132373
4	66.173399	66.132373
...	...	...
247	66.126000	66.132373
248	66.150265	66.132373
249	66.081114	66.132373
250	66.201315	66.132373
251	66.123064	66.132373

252 rows × 2 columns

Результаты модели регрессии опорных векторов (SVR) для параметра «Модуль упругости при растяжении»

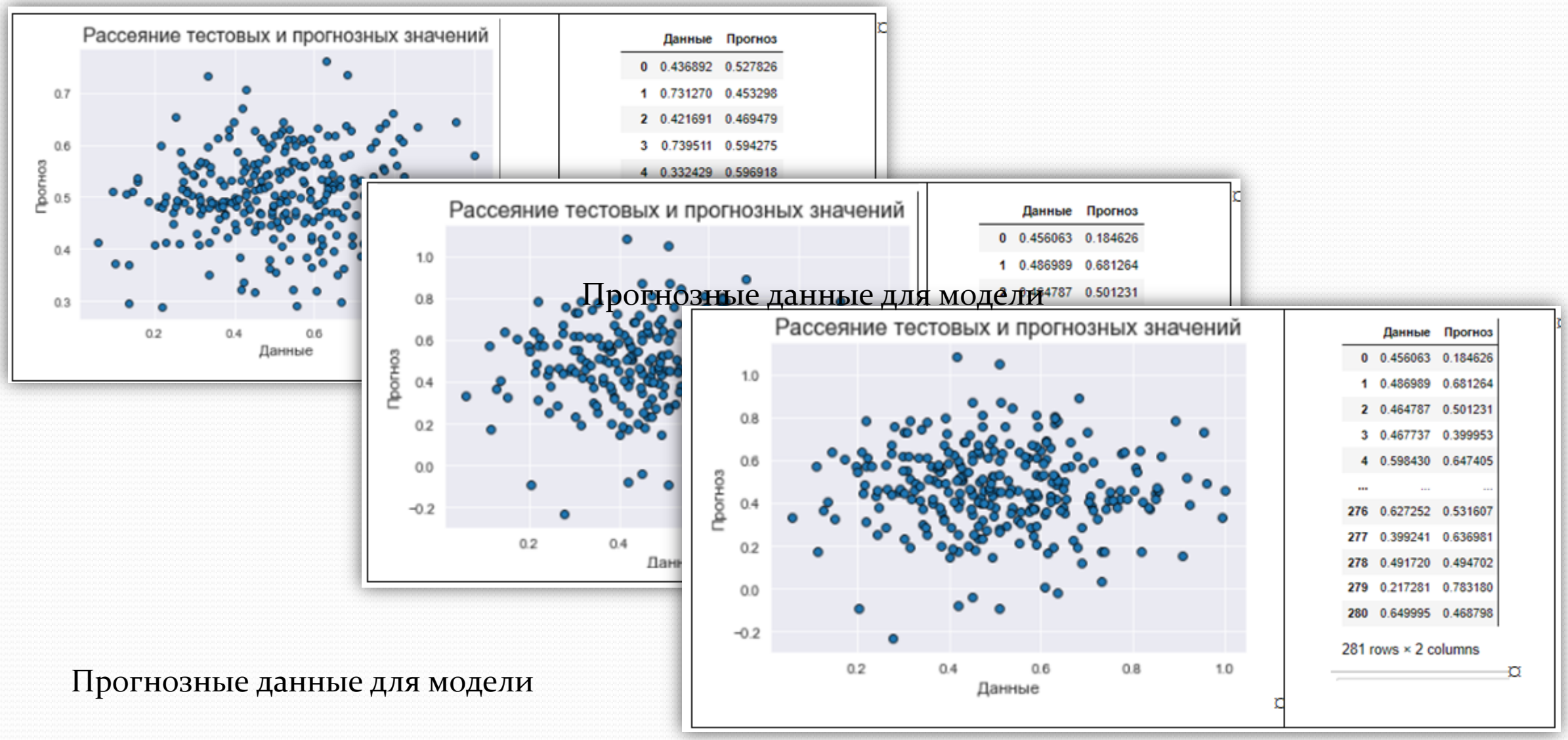
Разработка нейронной сети

Архитектура нейронной сети

```
def build_model1():
    model1=models.Sequential()
    model1.add(layers.Dense(64, activation='tanh', input_shape=(X1trn1.shape[1],)))
    model1.add(layers.Dense(64, activation='tanh'))
    model1.add(layers.Dense(1))
    model1.compile(optimizer='rmsprop', loss='mse', metrics=['mae'])
    return model1
```

- Для рекомендации соотношения «матрица-наполнитель» разработана простая модель глубокого обучения с помощью библиотеки Keras.
  - Для эксперимента были выбраны три функции активации:
  - tanh (арктангенс),
  - relu (выпрямленная линейная единица),
  - sigmoid (сигмоида 1/(1+exp(-x)))

Разработка нейронной сети



Прогнозные данные для модели

Прогнозные данные для модели

Разработка web-приложения

Прогнозное значение параметра «Соотношение матрица-наполнитель»

Результат

Плотность, кг/м3	
модуль упругости, ГПа	
Количество отвердителя, м.%	
Содержание эпоксидных групп,%_2	
Температура вспышки, С_2	
Поверхностная плотность, г/м2	
Потребление смолы, г/м2	
Угол нашивки, град	
Шаг нашивки	
Плотность нашивки	

Отправить

Удалённый репозиторий

- Страница создана на GitHub.
- Адрес страницы https:// https://github.com/Kiser1983/Antonov_VKR_DS
- В репозитории находятся:
  - файлы тетрадок Юпитера,
  - наборы данных, модели,
  - Web-приложение,
  - ВКР в текстовом формате.

Заключение

- Теоретически разработанный метод определения надёжности изделий из композиционных материалов, основанный на использовании статистически достоверных характеристик материалов, полученных физическим и вычислительным экспериментом, позволяет оценивать уровень надежности изделий как в отдельных точках, так и по всему объёму в целом.

Спасибо за внимание!