



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Выпускная квалификационная работа по курсу "Data Science"

Тема: Прогнозирование конечных свойств
новых материалов (композиционных материалов)

Выполнил: слушатель Пахомов Игорь Александрович



- **Актуальность темы:** Композитных материалы используется практически во всех областях производства. И созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.
- **Предметом проектирования является** построение при помощи методов машинного обучения моделей прогнозирования характеристик «модуль упругости при растяжении» и «прочность при растяжении», рекомендации «соотношение матрица-наполнитель».
- **Цель работы:** разработать модели для прогноза модуля упругости при растяжении, прочности при растяжении и соотношения «матрица-наполнитель».

Задачи проектирования

1. Обучить алгоритм машинного обучения, который будет определять значения:
Модуль упругости при растяжении, ГПа;
Прочность при растяжении, МПа.
2. Написать нейронную сеть, которая будет рекомендовать:
Соотношение матрица-наполнитель.
3. Написать приложение, которое будет выдавать прогноз, полученный в задании 1 или 2 (один или два прогноза, на выбор учащегося)
4. Создать профиль на github.com



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Характеристики анализируемого датасета

Индекс	Признак 1	Признак 2
1	Угол нахлеста	Плотность нахлеста
2	0	4
3	1	0
4	2	0
5	3	0
6	4	0
7	5	0
8	6	0
9	7	0
10	8	0
11	9	0
12	10	0
13	11	0
14	12	0
15	13	0
16	14	0
17	15	0
18	16	0
19	17	0
20	18	0
21	19	0
22	20	90
23	21	90
24	22	90
25	23	90
26	24	90
27	25	90
28	26	90
29	27	90
30	28	90
31	29	90

Индекс	Признак 1	Признак 2	Признак 3	Признак 4	Признак 5	Признак 6	Признак 7	Признак 8	Признак 9
31	29	2.282825314	2008.357592	393.9673255	149.3728324	21.66175068	330.498641	535.3714591	72.24452408
32	30	1.978140173	1973.629097	991.7240946	149.3721279	19.75057789	232.0581913	485.4537781	75.66570056
33	31	1.771436393	1872.49156	801.0338825	79.79454787	22.29630372	340.7368984	864.9291837	70.94759156
34	32	3.277086987	2010.047012	339.5504228	67.49899306	24.28060902	254.9408837	117.5352342	67.47870693
35	33	2.984362226	1912.315437	1183.091845	133.5406007	23.26379657	314.9961255	377.3890094	75.29045222
36	34	2.916149621	1879.969846	1003.270178	109.2395305	25.68275948	294.0485366	408.3542393	71.70085562
37	35	3.247617211	1813.2346	757.874479	81.37987084	23.42246524	279.0801575	575.0628571	69.34113288
38	36	2.423875673	1908.940601	530.2286864	58.26241428	24.07354923	325.138688	456.9080467	74.24435417
39	37	5.08959309	1977.339047	1572.096042	132.3430598	25.39700098	286.5564309	690.3648357	72.34163973
40	38	2.444176986	2085.495837	931.3106361	110.5648399	23.48713976	270.2867651	278.2300203	71.47906047
41	39	2.687696920	2078.854676	1542.168458	132.1474033	22.6501092	357.9728962	787.299217	76.47178847
42	40	3.034395483	1968.401388	455.8710188	61.42129652	23.49072291	316.4145721	637.3768927	75.09037174
43	41	2.465204971	1936.099137	1056.554965	71.29405822	24.5233807	271.9756783	129.0771629	66.42079436
44	42	2.654388649	1996.159145	525.0577741	77.50680254	18.12610706	223.4086654	28.60810234	69.48973340
45	43	1.193629582	1965.929227	899.603701	102.9690686	19.56571624	225.9102229	871.0889548	73.45469452
46	44	2.914333275	2049.373404	382.2633685	81.35204737	16.39159473	233.2960627	561.9921308	69.81451516
47	45	4.315665702	1913.379677	822.9187355	143.5769371	24.27550801	274.9887944	260.6593411	76.95732867
48	46	2.338424280	1963.35156	1155.160504	150.0158290	18.29942138	315.9041781	644.3631421	71.30398977
49	47	1.289167246	1984.511373	1405.786822	130.9427979	21.8292399	288.9520988	161.0077186	74.68091326
50	48	2.134446144	1986.349053	809.2938035	95.08902184	19.36492572	205.499761	196.3576427	76.34020725
51	49	4.147966432	1991.789789	1250.198275	116.8564622	21.57326496	320.7401725	755.006552	70.35546403
52	50	3.057830147	1987.259859	403.3622302	112.719826	19.39051107	336.2453837	362.8319835	73.01648889
53	51	4.162128684	1940.739237	347.9917504	111.9697729	18.80028336	304.1904803	229.9283369	72.76947322
54	52	1.846923155	1956.087297	622.722758	115.7097202	21.04331755	236.8036716	393.3954368	64.6964004
55	53	1.301025044	2057.939193	939.8590088	136.3239746	23.32927188	292.6257156	523.8472486	69.38286607
56	54	3.621934812	2065.51708	553.6209361	120.6958775	24.43189366	308.9785619	234.1616744	69.80219063
57	55	3.960351763	2078.08519	578.4727101	101.9863726	23.52157007	263.9367842	200.9708963	75.92942824
58	56	4.816898968	1907.240881	854.9118785	107.6339657	26.26099277	294.7507581	485.5104287	74.95513851
59	57	4.296378327	1985.438274	481.4670088	86.25804778	21.38104303	240.0931941	61.31785493	71.59802745
60	58	4.487380381	2045.919984	797.0640962	40.30480949	22.23034719	239.5788757	646.0742217	73.80798877
61	59	3.118176223	1892.7578	968.1791713	162.4854467	23.96754282	260.1888972	173.8128117	71.18356105

Исходные данные о свойствах композиционных материалов и способах их компоновки получены структурным подразделением МГТУ им. Н.Э. Баумана – Центр НТИ «Цифровое материаловедение: новые материалы и вещества» в рамках решения производственных задач.

Датасет состоит из двух файлов: X_br.xlsx (признаки базальтопластика) и X_nur.xlsx (признаки углепластика)

- Файл X_br.xlsx -содержит 1023 строки, индекс и 10 признаков.
- Файл X_nur.xlsx -содержит 1040 строк индекс и 3 признака.



Формированный исходного датафрейма

Разведочный анализ данных в рамках данной задачи проведен над датафреймом Pandas, полученным путем импорта (и объединения по типу INNER по полю индекса) таблиц исходных данных. Для объединения датасетов используем методом DataFrame.merge(), с параметром "inner"

информацию о имеющемся датасете

1	df.info()
<pre><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, 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 float64 11 Шаг нашивки 1023 non-null float64 12 Плотность нашивки 1023 non-null float64 dtypes: float64(13) memory usage: 111.9 KB</pre>	

данные объединенного датасета

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



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



Предобработка данных

1. Анализ датасета на пропуски, дубликаты и удаление пропусков, с помощью методов `info()`, `duplicated()` и `describe()`
2. Удаление выбросов из датасета, замена данных, за пределами второго и третьего квантиля на пустые, затем удаление строк, содержащие пустые значения
3. Нормализация данных с помощью метода `MinMaxScaler` и `Normalizer` из библиотеки `sklearn`

```
<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, ГПа                 936 non-null   float64
3   Количество отвердителя, м.%           936 non-null   float64
4   Содержание эпоксидных групп,%_2      936 non-null   float64
5   Температура вспышки, C_2             936 non-null   float64
6   Поверхностная плотность, г/м2        936 non-null   float64
7   Модуль упругости при растяжении, ГПа 936 non-null   float64
8   Прочность при растяжении, МПа        936 non-null   float64
9   Потребление смолы, г/м2              936 non-null   float64
10  Угол нашивки, град                   936 non-null   float64
11  Шаг нашивки                          936 non-null   float64
12  Плотность нашивки                    936 non-null   float64
dtypes: float64(13)
memory usage: 102.4 KB
```

Соотношение матрица-наполнитель	6
Плотность, кг/м3	9
модуль упругости, ГПа	2
Количество отвердителя, м.%	14
Содержание эпоксидных групп,%_2	2
Температура вспышки, C_2	8
Поверхностная плотность, г/м2	2
Модуль упругости при растяжении, ГПа	6
Прочность при растяжении, МПа	11
Потребление смолы, г/м2	8
Угол нашивки, град	0
Шаг нашивки	4
Плотность нашивки	21
dtype: int64	

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, C_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	0.274768	0.651097	0.452951	0.079153	0.607435	0.509164	0.162230	0.272962	0.727777	0.514688	0.0	0.289334	0.546433
1	0.274768	0.651097	0.452951	0.630983	0.418887	0.583596	0.162230	0.272962	0.727777	0.514688	0.0	0.362355	0.319758
2	0.486552	0.651097	0.481725	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.0	0.362355	0.494123
3	0.465836	0.571539	0.450649	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.0	0.362355	0.546433
4	0.424236	0.332865	0.494944	0.511257	0.495653	0.509164	0.162230	0.272962	0.727777	0.514688	0.0	0.362355	0.720799
...
917	0.361662	0.444490	0.580064	0.337550	0.333908	0.703458	0.161609	0.473553	0.472912	0.183151	1.0	0.680014	0.320103
918	0.607674	0.704373	0.272080	0.749605	0.294428	0.362087	0.271207	0.462512	0.461722	0.157752	1.0	0.768759	0.437468
919	0.573391	0.498274	0.254927	0.501991	0.623085	0.334063	0.572659	0.580201	0.587558	0.572648	1.0	0.301102	0.679468
920	0.662497	0.748688	0.454635	0.717585	0.267818	0.466417	0.496511	0.535317	0.341643	0.434855	1.0	0.458245	0.516112
921	0.684036	0.280923	0.255222	0.632264	0.888354	0.588206	0.587373	0.552844	0.688015	0.428577	1.0	0.441137	0.850430

922 rows x 13 columns

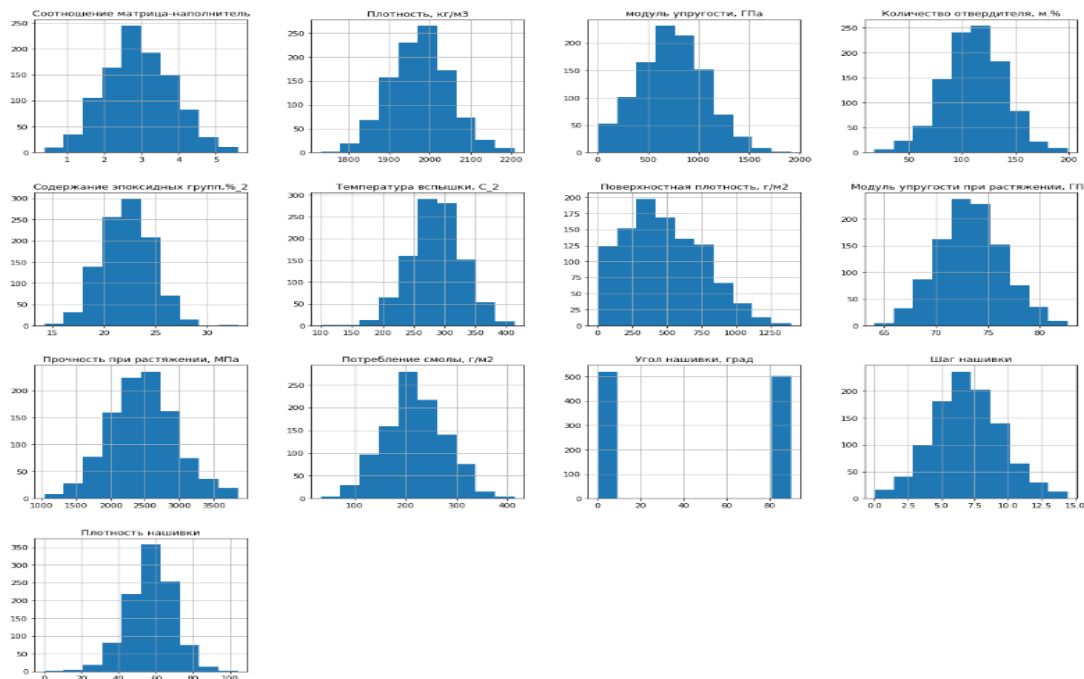
Анализ датасета на пропуски

Количество выбросов
по каждому из столбцов

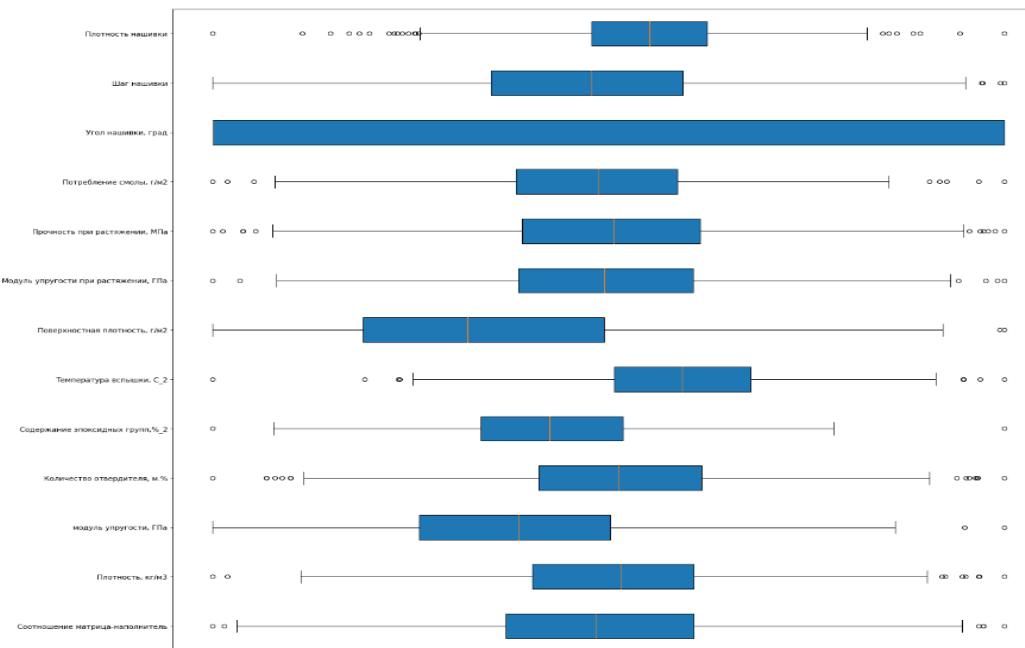
Нормализация данных с помощью метода `MinMaxScaler`



Гистограммы до обработки и диаграммы размаха до удаления выбросов



Гистограммы распределения по каждой переменной для оценки повторяющихся значений в многомерном пространстве



Для каждой переменной построены диаграммы размаха для определения наличия выбросов в данных.

Шкалы приведем к величинам в диапазоне $[0,1]$, чтобы "ящики с усами" были одного масштаба



Описательная статистика датасета после очистки выбросов

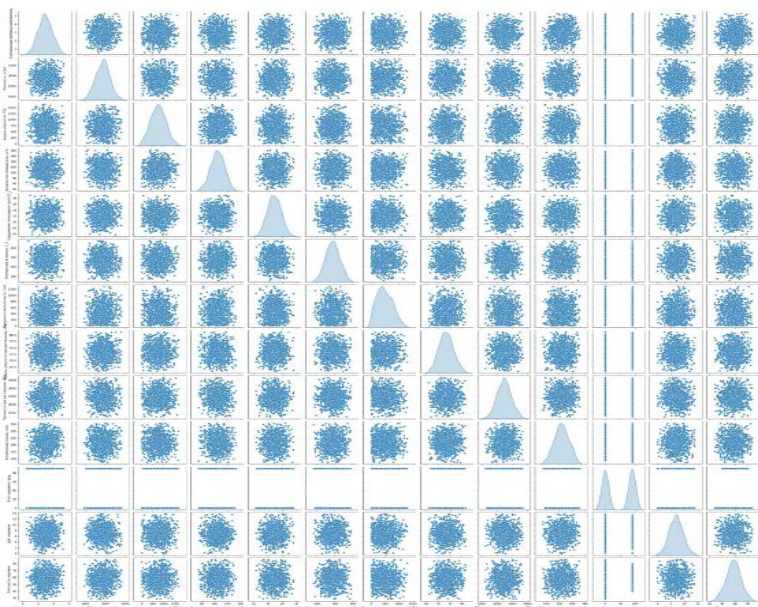
	Соотношение матрица- наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/ м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град
count	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	0.094462	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

Для каждой переменной построены диаграммы размаха для определения наличия выбросов в данных.

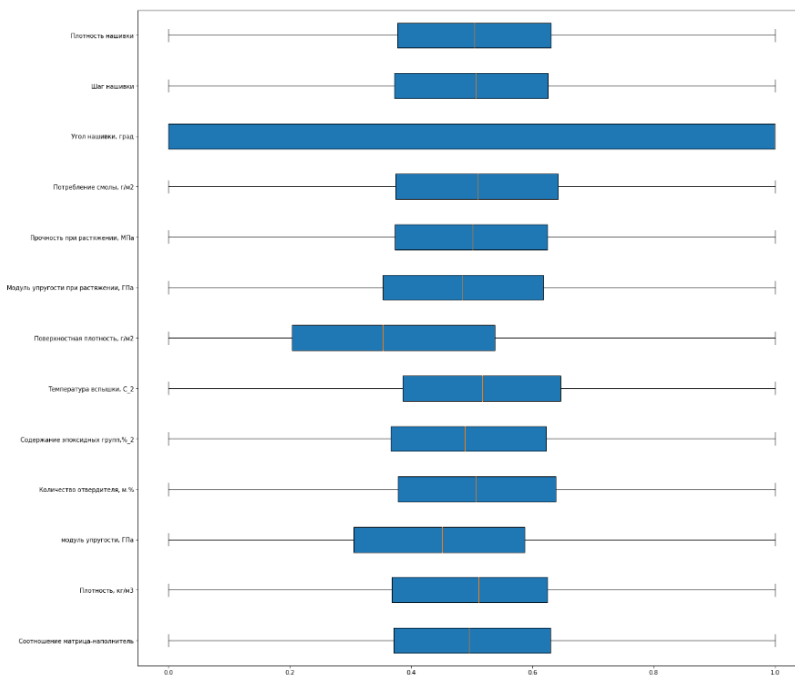
Шкалы приведем к величинам в диапазоне [0,1], чтобы "ящики с усами" были одного масштаба



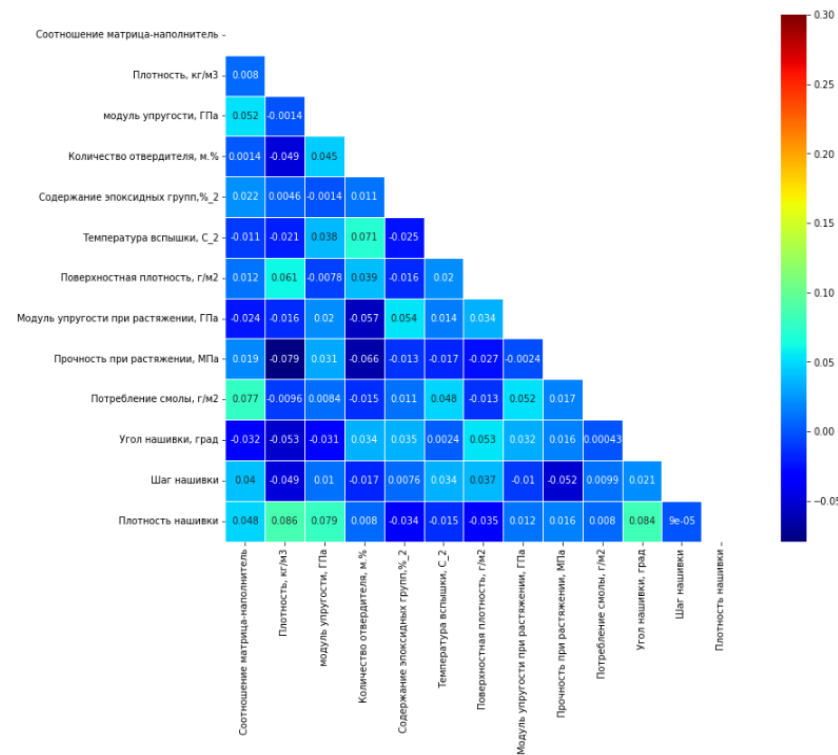
Гистограммы рассеяния и диаграммы размаха и Матрица корреляции датасета после очистки выбросов



Гистограммы рассеяния
после очистки выбросов



Диаграммы размаха после
очистки выбросов

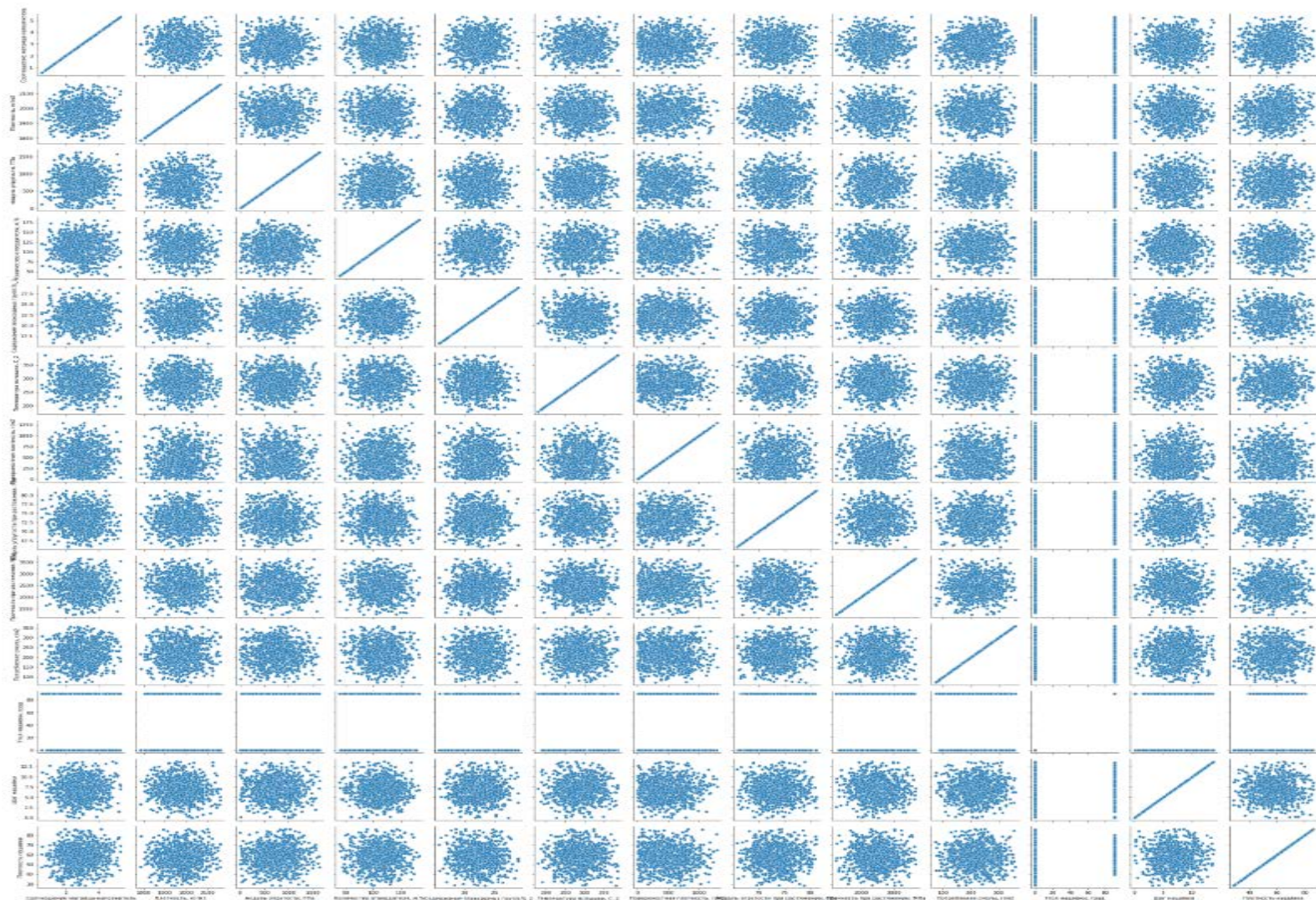


Матрица корреляции датасета
после очистки выбросов



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГУ им. Н.Э. Баумана

Матрица попарной зависимости датасета





Разработка и обучение моделей

В данной работе разработка и обучение моделей машинного обучения осуществляется для двух выходных параметров: «Прочность при растяжении» и «Модуль упругости при растяжении». Для каждого признака построение моделей осуществляется отдельно.

Для признака «Прочность при растяжении» были разработаны и обучены следующие модели:

- модель k ближайших соседей (метод `KNeighborsRegressor()`);
- модель на основе градиентного бустинга (метод `GradientBoostingRegressor()`).

Для признака «Модуль упругости при растяжении» были разработаны и обучены следующие модели:

- модель на основе линейной регрессии (метод `LinearRegression()`);
- модель на основе опорных векторов (метод `SVR()`).

Порядок разработки модели для каждого параметра и для каждого выбранного метода

- 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

Тестирование моделей

Подставляем оптимальные гиперпараметры в модель

```

1 model_base_1 = KNeighborsRegressor(algorithm='brute', leaf_size=10, n_neighbors=100, weights='distance')
2 #Обучаем модель
3 model_base_1.fit(Xtrain1_1,Ytrain1_1)
4 #Оцениваем точность на тренировочном наборе
5 base_accuracy = evaluate(model_base_1, Xtrain1_1,Ytrain1_1)
6 #Оцениваем точность на тестовом наборе
7 base_accuracy = evaluate(model_base_1, Xtest1_1,Ytest1_1)

```

Средняя абсолютная ошибка: 0.0000
Средняя абсолютная ошибка: 0.1546

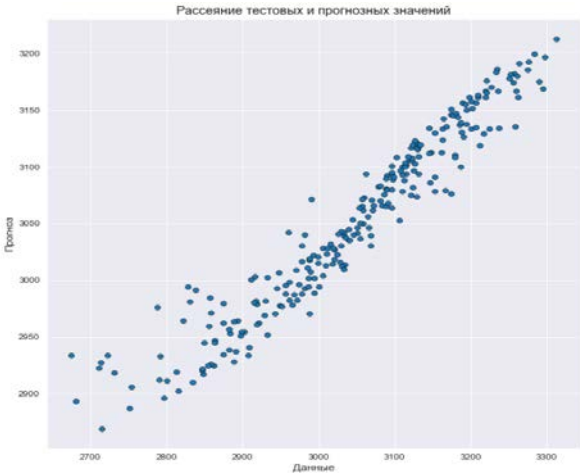
Подставляем оптимальные гиперпараметры в модель

```

1 model_base_2 = KNeighborsRegressor(algorithm='brute', leaf_size=10, n_neighbors=100, weights='distance')
2 #Обучаем модель
3 model_base_2.fit(Xtrain2_1,Ytrain2_1)
4 #Оцениваем точность на тренировочном наборе
5 base_accuracy = evaluate(model_base_2, Xtrain2_1,Ytrain2_1)
6 #Оцениваем точность на тестовом наборе
7 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»

Оцениваем точность на тренировочном наборе

```

1 base_accuracy = evaluate_2(model_base_11, Xtrain1_2, Ytrain1_2)
2 #Оцениваем точность на тестовом наборе
3 base_accuracy = evaluate_2(model_base_11, Xtest1_2, Ytest1_2)

```

Средняя абсолютная ошибка: 0.1288
Средняя абсолютная ошибка: 0.1655

```

1 Подставляем оптимальные гиперпараметры в модель

```

```

1 model_base_22 = GradientBoostingRegressor(loss='lad', max_depth=2)
2 #Обучаем модель
3 model_base_22.fit(Xtrain2_2,np.ravel(Ytrain2_2))

```

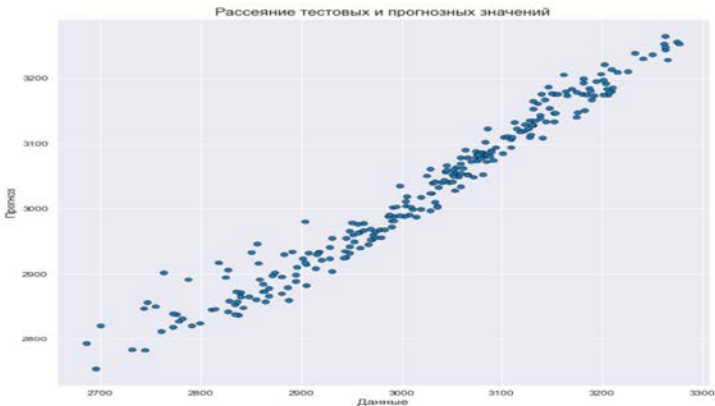
GradientBoostingRegressor(loss='lad', max_depth=2)

```

1 #Оцениваем точность на тренировочном наборе
2 base_accuracy = evaluate_2(model_base_22, Xtrain2_2, Ytrain2_2)
3 #Оцениваем точность на тестовом наборе
4 base_accuracy = evaluate_2(model_base_22, Xtest2_2, Ytest2_2)

```

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»



Тестирование моделей

```
1 best_estimator = model11.best_estimator_  
2 #Выводим гиперпараметры для оптимальной модели  
3 print(best_estimator)  
4 #Выводим точность оптимального трейнера  
5 print(model11.best_score_)
```

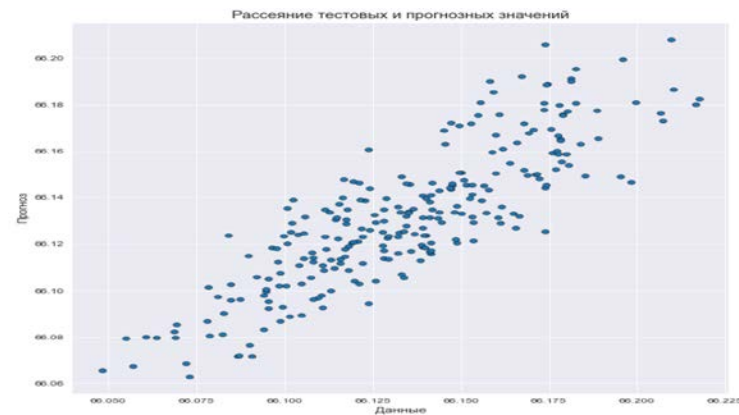
```
LinearRegression()  
-0.026111116626460085
```

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

```
LinearRegression()
```

```
1 #Оцениваем точность на тренировочном наборе  
2 base_accuracy = evaluate(model_base11, Xtrain11,Ytrain11)  
3 #Оцениваем точность на тестовом наборе  
4 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»

```
1 best_estimator = model111.best_estimator_  
2 #Выводим гиперпараметры для оптимальной модели  
3 print(best_estimator)  
4 #Выводим точность оптимального трейнера  
5 print(model111.best_score_)
```

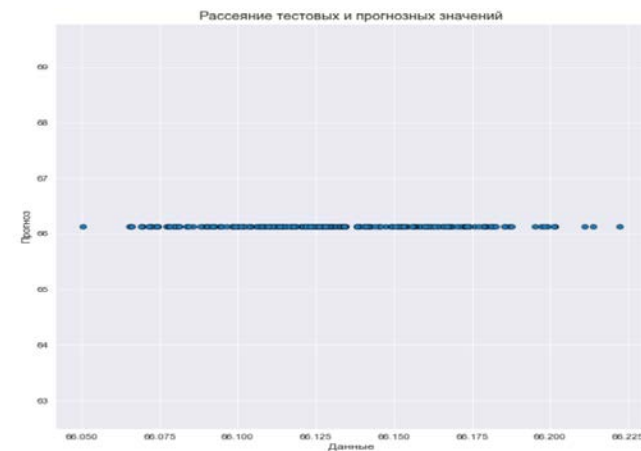
```
SVR(kernel='linear')  
-0.0304111012748157
```

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

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

```
1 #Оцениваем точность на тренировочном наборе  
2 base_accuracy = evaluate_2(model_base111, Xtrain111, Ytrain111)  
3 #Оцениваем точность на тестовом наборе  
4 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



Разработка нейронной сети

Для рекомендации соотношения «матрица-наполнитель» разработана простая модель глубокого обучения с помощью библиотеки Keras.

Модель состоит из трех скрытых уровней. Первый уровень содержит 64 нейрона, что немногим более чем в три раза превышает объем входных данных (10 входных переменных). Последующие скрытые уровни содержат 64 и 1 нейрон. Снижение числа нейронов на каждом уровне сжимает информацию, которую сеть обработала на предыдущих уровнях.

Скрытые уровни нейронной сети трансформируются функциями активации. Эти функции являются важными элементами сетевой инфраструктуры, так как они вносят в систему нелинейность.

Для эксперимента были выбраны три функции активации:

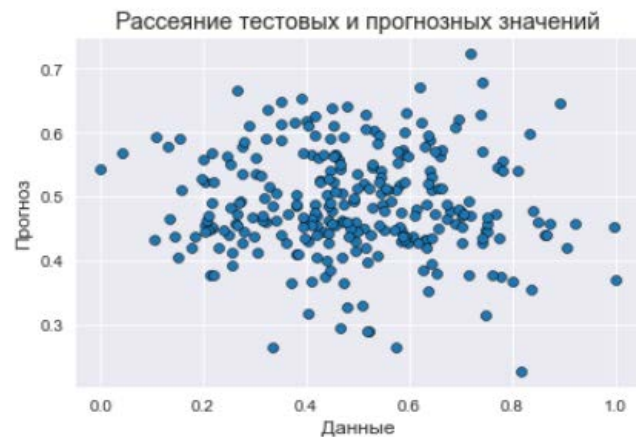
1. tanh (арктангенс),
2. relu (выпрямленная линейная единица),
3. sigmoid (сигмоида $1/(1+\exp(-x))$)

```
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
```

Архитектура нейронной сети

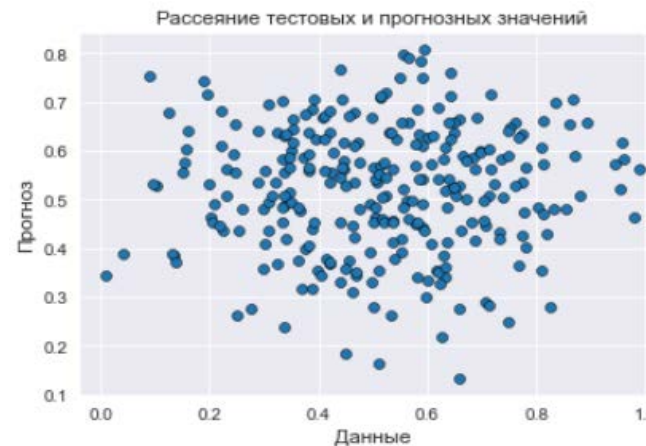


Разработка нейронной сети



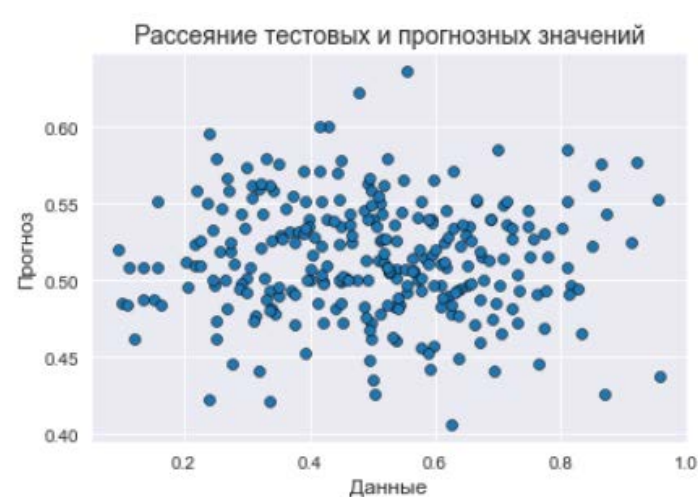
	Данные	Прогноз
0	0.439831	0.501854
1	0.638328	0.539216
2	0.274799	0.427661
3	0.381253	0.410576
4	0.455313	0.564241
...
272	0.648514	0.571335
273	0.419445	0.405875
274	0.341934	0.505846
275	0.526656	0.550951
276	0.590948	0.601100

Прогнозные данные для модели с функцией tanh



	Данные	Прогноз
0	0.387258	0.438928
1	0.588735	0.784112
2	0.504202	0.463572
3	0.528359	0.638538
4	0.505854	0.534838
...
272	0.739697	0.503467
273	0.524701	0.544280
274	0.198417	0.533502
275	0.409386	0.625002
276	0.748254	0.639669

Прогнозные данные для модели с функцией relu



	Данные	Прогноз
0	0.319138	0.441282
1	0.366300	0.531129
2	0.397774	0.540102
3	0.722423	0.481907
4	0.666214	0.550901
...
272	0.589733	0.510593
273	0.494837	0.467828
274	0.223971	0.509313
275	0.298759	0.492468
276	0.555986	0.496744

Прогнозные данные для модели с функцией sigmoid



Разработка нейронной сети, второй вариант

- Сформируем входы и выход для модели.
- Разобьём выборку на обучающую и тестовую.
- Нормализуем данные.
- Создадим функцию для поиска наилучших параметров и слоёв.
- Построим модель, определим параметры, найдем оптимальные параметры посмотрим на результаты;
- Повторим все эти этапы до построения окончательной модели;
- Обучим нейросеть;
- Посмотрим на потери модели;
- Построим график потерь на тренировочной и тестовой выборках.
- Построим график результата работы модели.

```
def create_model(lyrs=[32], act='softmax', opt='SGD', dr=0.1):
```

```
    seed = 7  
    np.random.seed(seed)  
    tf.random.set_seed(seed)
```

```
    model = Sequential()  
    model.add(Dense(lyrs[0], input_dim=x_train.shape[1], activation=act))  
    for i in range(1, len(lyrs)):  
        model.add(Dense(lyrs[i], activation=act))
```

```
    model.add(Dropout(dr))  
    model.add(Dense(3, activation='tanh')) # выходной слой
```

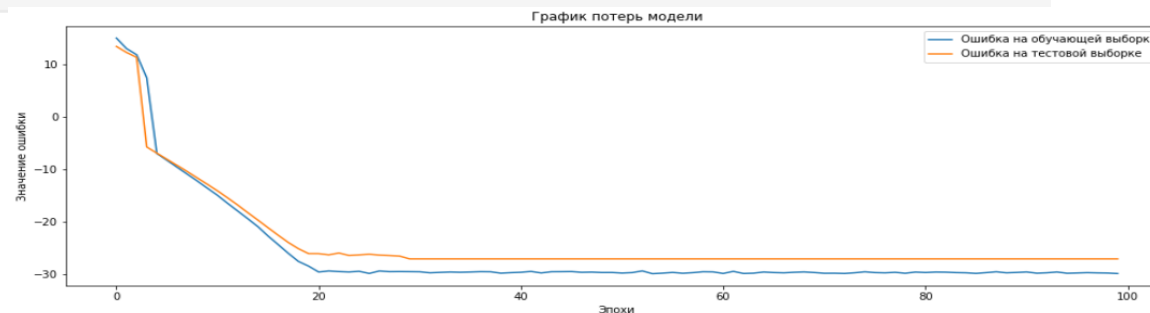
```
    model.compile(loss='binary_crossentropy', optimizer=opt, metrics=['mae', 'accuracy'])
```

```
    return model
```

```
# построение окончательной модели
```

```
model = create_model(lyrs=[128, 64, 16, 3], dr=0.05)
```

```
print(model.summary())
```





Разработка приложения и Создание удаленного репозитория



При разработке приложения нам понадобилась библиотека Flask также ещё использовали библиотеку tensorflow и pickle для внедрения наших моделей в приложение.

Прогнозирование для "соотношения матрица-наполнитель"

Плотность, кг/м3 (1700...2300)	1880.0
Модуль упругости, ГПа (2...2000)	622.0
Количество отвердителя, м. % (17...200)	111.86
Содержание эпоксидных групп, % 2 (14...34)	22.2678571428571
Температура вспыхива, С 2 (100...414)	284.615384615384
Поверхностная плотность, г/м2 (0.6...1400)	470.0
Модуль упругости при растяжении, ГПа (64...83)	73.3333333333333
Прочность при растяжении, МПа (1036...3849)	2455.55555555555
Потребление смолы, г/м2 (33...414)	220.0
Угол нашивки, град (0...90)	90.0
Шаг нашивки (0...15)	4.0
Плотность нашивки (0...104)	60.0

Отправить

Входные переменные:

Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м. %	Содержание эпоксидных групп, % 2	Температура вспыхива, 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 [2.9509137]

Страница создана на GitHub.

Адрес страницы:

https://github.com/ComrGarry/DS_PahomovIA

В репозитории находятся: файлы тетрадок Юпитера, наборы данных, модели, приложение, ВКР в текстовом формате.



ВЫВОД ПО ТЕМЕ

Были сделаны выводы что использованные при разработке моделей подходы не позволили получить сколько-нибудь достоверных прогнозов. Применённые модели регрессии не показали высокой эффективности в прогнозировании свойств композитов. Невозможно определить из свойств материалов соотношение «матрица – наполнитель»

Данный факт не указывает на то, что прогнозирование характеристик композитных материалов на основании предоставленного набора данных невозможно, но может указывать на недостатки базы данных, подходов, использованных при прогнозе, необходимости пересмотра инструментов для прогнозирования.

Вывод: текущим набором алгоритмов задача не решается, возможно, решается трудно или не решается совсем.



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана



do.bmstu.ru