



**ОБРАЗОВАТЕЛЬНЫЙ
ЦЕНТР** МГТУ им. Н. Э. Баумана

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА по курсу «Data Science»

Прогнозирование конечных свойств новых материалов (композиционных материалов)

Докладчик:

Петров Алексей Сергеевич,
слушатель Образовательного центра МГТУ им. Н. Э. Баумана
под управлением МИЦ «Композиты России»

Цель работы:

Построение модели для прогнозирования ряда конечных свойств композиционных материалов на основе набора входных параметров.

Постановка задачи:

1. Изучение предметной области.
2. Разведочный анализ данных
3. Предобработка данных.
4. Обучение моделей для прогноза **модуля упругости при растяжении и прочности при растяжении.**
5. Создание нейронной сети для прогноза **соотношения матрица-наполнитель.**
6. Оценка точности модели на тренировочном и тестовом наборе данных.
7. Разработка приложения.

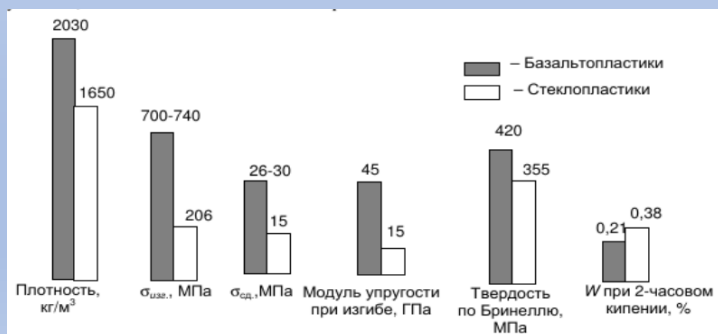
Входные данные

а) результаты испытаний базальтопластика (X_br):

	Соотношение матрица-наполнитель	Плотность, кг/м ³	модуль упругости, ГПа	Количество отвердителя, м. %	Содержание эпоксидных групп, %_2	Температура вспышки, С_2	Поверхностная плотность, г/м ²	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м ²
0.0	1.857143	2030.000000	738.736842	30.000000	22.267857	100.000000	210.000000	70.000000	3000.000000	220.000000
1.0	1.857143	2030.000000	738.736842	50.000000	23.750000	284.615385	210.000000	70.000000	3000.000000	220.000000
2.0	1.857143	2030.000000	738.736842	49.900000	33.000000	284.615385	210.000000	70.000000	3000.000000	220.000000
3.0	1.857143	2030.000000	738.736842	129.000000	21.250000	300.000000	210.000000	70.000000	3000.000000	220.000000
4.0	2.771331	2030.000000	753.000000	111.860000	22.267857	284.615385	210.000000	70.000000	3000.000000	220.000000
...
1018.0	2.271346	1952.087902	912.855545	86.992183	20.123249	324.774576	209.198700	73.090961	2387.292495	125.007669
1019.0	3.444022	2050.089171	444.732634	145.981978	19.599769	254.215401	350.660830	72.920827	2360.392784	117.730099
1020.0	3.280604	1972.372865	416.836524	110.533477	23.957502	248.423047	740.142791	74.734344	2662.906040	236.606764
1021.0	3.705351	2066.799773	741.475517	141.397963	19.246945	275.779840	641.468152	74.042708	2071.715856	197.126067
1022.0	3.808020	1890.413468	417.316232	129.183416	27.474763	300.952708	758.747882	74.309704	2856.328932	194.754342

1023 rows x 10 columns

б) матрица углепластика (X_nup):




	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0.0	0.0	4.000000	57.000000
1.0	0.0	4.000000	60.000000
2.0	0.0	4.000000	70.000000
3.0	0.0	5.000000	47.000000
4.0	0.0	5.000000	57.000000
...
1035.0	90.0	8.088111	47.759177
1036.0	90.0	7.619138	66.931932
1037.0	90.0	9.800926	72.858286
1038.0	90.0	10.079859	65.519479
1039.0	90.0	9.021043	66.920143

1040 rows x 3 columns

Метод решения задачи

Обучение с учителем

```
 X = df.drop(['Соотношение матрица-наполнитель'], axis=1)  
y = df[['Соотношение матрица-наполнитель']]  
# Разделяем на тренировочную и тестовую выборку  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=RANDOM_STATE)
```

I способ.

Решение задачи регрессии с использованием методов библиотеки

Scikit-learn:

- линейная регрессия (linear regression),
- случайный лес (random forest),
- К-ближайших соседей (k-nearest neighbors),
- градиентный бустинг (gradient boosting),
- многослойный перцептрон (MLP).

II способ.

Создание нейронных сетей с использованием высокоуровневого API для построения и тренировки моделей глубокого обучения **Keras**, которое входит во фреймворк **TensorFlow**.

Разведочный анализ данных

```
# количество уникальных значений по каждому п  
df.nunique()
```

Соотношение матрица-наполнитель	1014
Плотность, кг/м3	1013
модуль упругости, ГПа	1020
Количество отвердителя, м.%	1005
Содержание эпоксидных групп, %_2	1004
Температура вспышки, С_2	1003
Поверхностная плотность, г/м2	1004
Модуль упругости при растяжении, ГПа	1004
Прочность при растяжении, МПа	1004
Потребление смолы, г/м2	1003
Угол нашивки, град	2
Шаг нашивки	989
Плотность нашивки	988

dtype: int64

```
# количество пропусков  
df.isnull().sum()
```

Соотношение матрица-наполнитель	0
Плотность, кг/м3	0
модуль упругости, ГПа	0
Количество отвердителя, м.%	0
Содержание эпоксидных групп, %_2	0
Температура вспышки, С_2	0
Поверхностная плотность, г/м2	0
Модуль упругости при растяжении, ГПа	0
Прочность при растяжении, МПа	0
Потребление смолы, г/м2	0
Угол нашивки, град	0
Шаг нашивки	0
Плотность нашивки	0

dtype: int64



```
# количество дубликатов - их нет  
df.duplicated().sum()
```

0



Разведочный анализ данных

```
# описательная статистика
df.describe().round(2).T
```

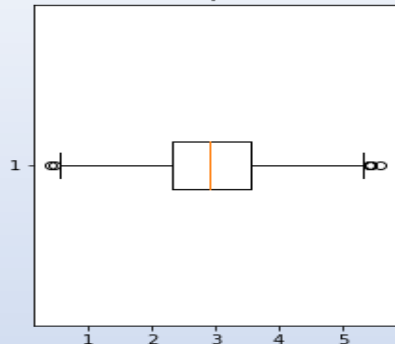
	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	2.93	0.91	0.39	2.32	2.91	3.55	5.59
Плотность, кг/м3	1023.0	1975.73	73.73	1731.76	1924.16	1977.62	2021.37	2207.77
модуль упругости, ГПа	1023.0	739.92	330.23	2.44	500.05	739.66	961.81	1911.54
Количество отвердителя, м.%	1023.0	110.57	28.30	17.74	92.44	110.56	129.73	198.95
Содержание эпоксидных групп,%_2	1023.0	22.24	2.41	14.25	20.61	22.23	23.96	33.00
Температура вспышки, С_2	1023.0	285.88	40.94	100.00	259.07	285.90	313.00	413.27
Поверхностная плотность, г/м2	1023.0	482.73	281.31	0.60	266.82	451.86	693.23	1399.54
Модуль упругости при растяжении, ГПа	1023.0	73.33	3.12	64.05	71.25	73.27	75.36	82.68
Прочность при растяжении, МПа	1023.0	2466.92	485.63	1036.86	2135.85	2459.52	2767.19	3848.44
Потребление смолы, г/м2	1023.0	218.42	59.74	33.80	179.63	219.20	257.48	414.59
Угол нашивки, град	1023.0	44.25	45.02	0.00	0.00	0.00	90.00	90.00
Шаг нашивки	1023.0	6.90	2.56	0.00	5.08	6.92	8.59	14.44
Плотность нашивки	1023.0	57.15	12.35	0.00	49.80	57.34	64.94	103.99

Гистограммы распределения для каждой переменной

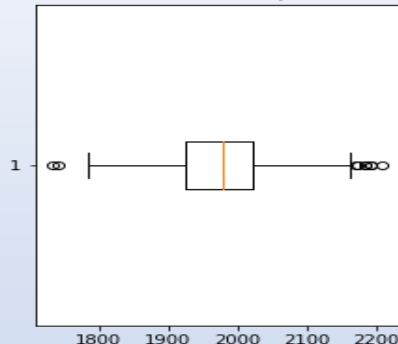


Диаграммы размаха (ящики с усами)

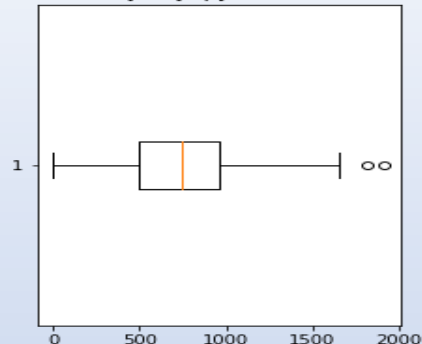
Соотношение матрица-наполнитель



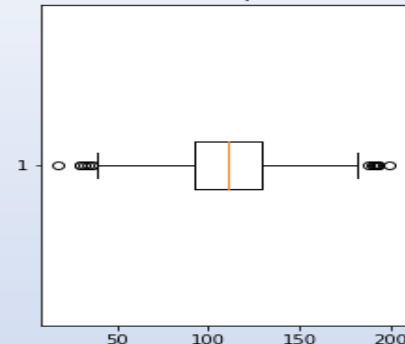
Плотность, кг/м3



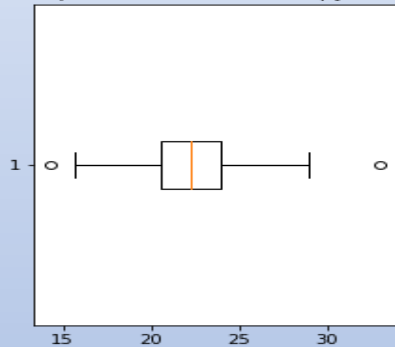
модуль упругости, ГПа



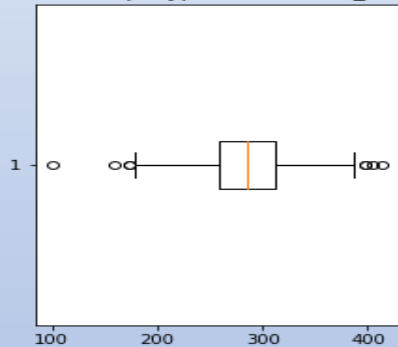
Количество отвердителя, м.%



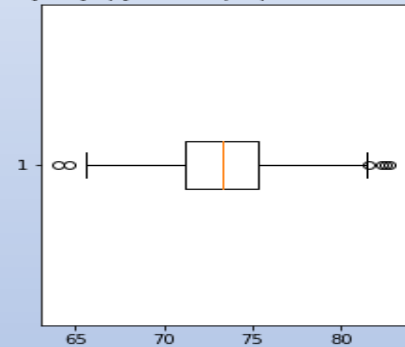
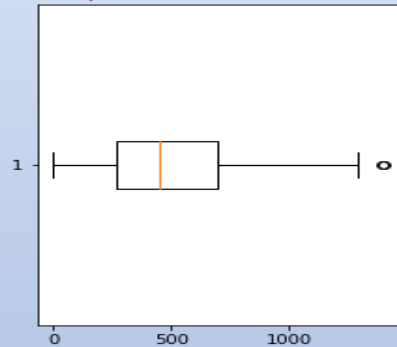
Содержание эпоксидных групп, %_2



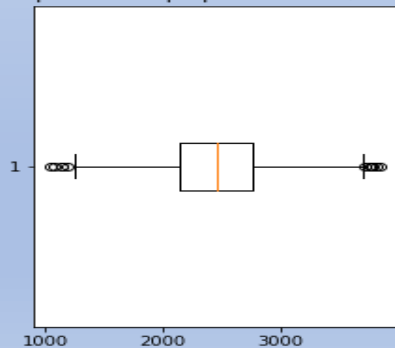
Температура вспышки, С_2



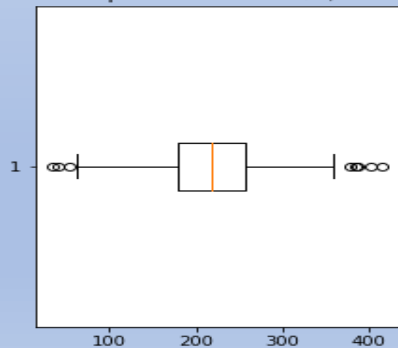
Поверхностная плотность, г/м2 Модуль упругости при растяжении, ГПа



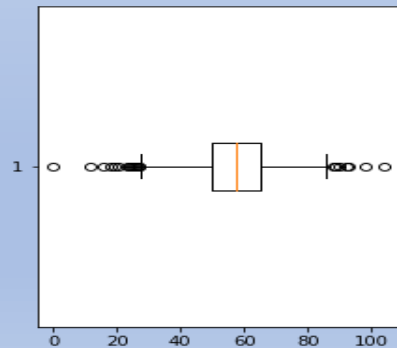
Прочность при растяжении, МПа



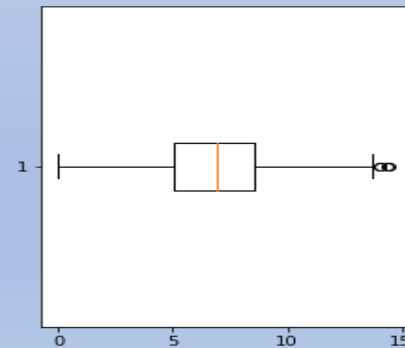
Потребление смолы, г/м2



Плотность нашивки

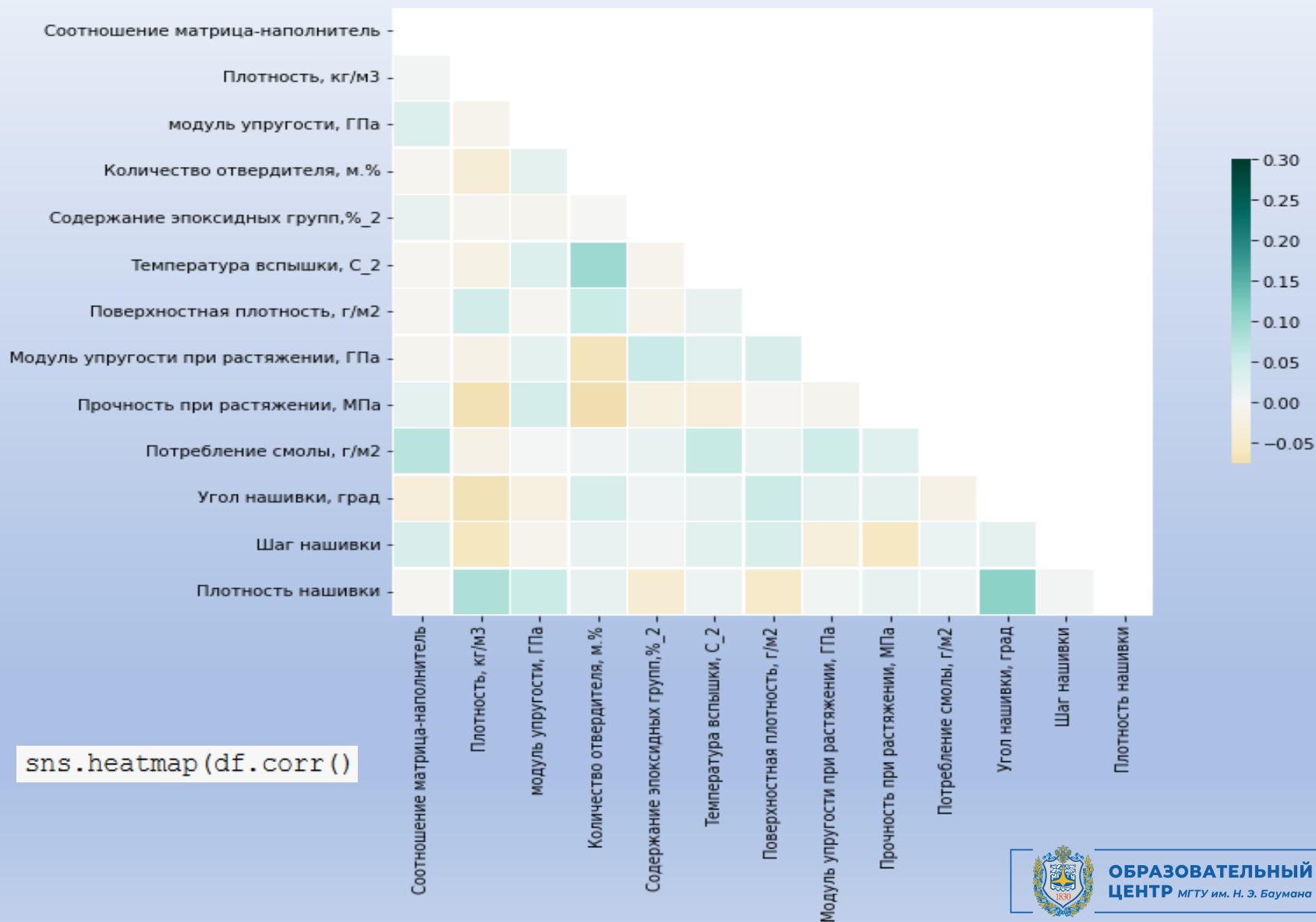


Шаг нашивки



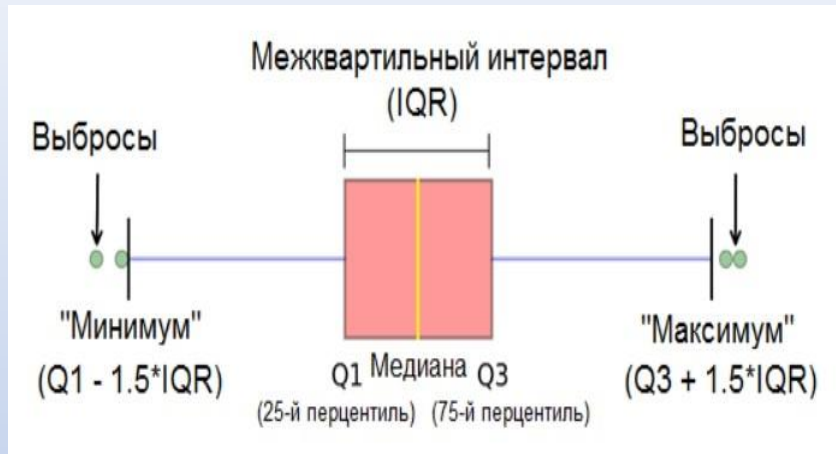
Разведочный анализ данных

Тепловая карта до нормализации



Препроцессинг

Удаление выбросов



Соотношение матрица-наполнитель	6
Плотность, кг/м3	9
модуль упругости, ГПа	2
Количество отвердителя, м.%	14
Содержание эпоксидных групп, %_2	2
Температура вспышки, С_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

Нормализация

```
# масштабирование значений в диапазон [0, 1]
scaler = MinMaxScaler()
df_norm=pd.DataFrame(scaler.fit_transform(df), columns=df.columns, index=df.index)
```

```
df_norm.head(5)
```

	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
1.0	0.274768	0.651097	0.447061	0.079153	0.607435	0.509164	0.162230	0.280303	0.712590	0.529221	0.0	0.289334
3.0	0.274768	0.651097	0.447061	0.630983	0.418887	0.583596	0.162230	0.280303	0.712590	0.529221	0.0	0.362355
4.0	0.466552	0.651097	0.455721	0.511257	0.495653	0.509164	0.162230	0.280303	0.712590	0.529221	0.0	0.362355
5.0	0.465836	0.571539	0.452685	0.511257	0.495653	0.509164	0.162230	0.280303	0.712590	0.529221	0.0	0.362355
6.0	0.424236	0.332865	0.488508	0.511257	0.495653	0.509164	0.162230	0.280303	0.712590	0.529221	0.0	0.362355
...

Препроцессинг

```
# Описательная статистика для нормализованного датасета
df_norm.describe().round(5).T
```

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	936.0	0.49893	0.18749	0.0	0.37227	0.49454	0.62920	1.0
Плотность, кг/м3	936.0	0.50270	0.18778	0.0	0.36852	0.51123	0.62500	1.0
модуль упругости, ГПа	936.0	0.44676	0.19958	0.0	0.30124	0.44706	0.58045	1.0
Количество отвердителя, м.%	936.0	0.50466	0.18886	0.0	0.37619	0.50604	0.63798	1.0
Содержание эпоксидных групп,%_2	936.0	0.49122	0.18062	0.0	0.36772	0.48938	0.62341	1.0
Температура вспышки, С_2	936.0	0.51606	0.19062	0.0	0.38613	0.51598	0.64645	1.0
Поверхностная плотность, г/м2	936.0	0.37373	0.21708	0.0	0.20562	0.35416	0.53868	1.0
Модуль упругости при растяжении, ГПа	936.0	0.48865	0.19147	0.0	0.35902	0.48575	0.61508	1.0
Прочность при растяжении, МПа	936.0	0.49571	0.18891	0.0	0.36515	0.49182	0.61287	1.0
Потребление смолы, г/м2	936.0	0.52114	0.19578	0.0	0.39207	0.52377	0.65245	1.0
Угол нашивки, град	936.0	0.51175	0.50013	0.0	0.00000	1.00000	1.00000	1.0
Шаг нашивки	936.0	0.50223	0.18326	0.0	0.37221	0.50426	0.62460	1.0
Плотность нашивки	936.0	0.51378	0.19134	0.0	0.39048	0.51603	0.63884	1.0



Препроцессинг

Гистограммы после нормализации



Обучение моделей



```
# обучение с параметрами по умолчанию
RandomForestRegressor().fit(X_train_upr, np.ravel(y_train_upr))
# запуск предсказания для модели на основе тестового набора данных X_test
y_pred_upr = RandomForestRegressor().predict(X_test_upr)
```

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}|$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

Where,

\hat{y} – predicted value of y
 \bar{y} – mean value of y

```
'MSE': metrics.mean_squared_error(y_test_upr, y_pred_upr),
'MAE': metrics.mean_absolute_error(y_test_upr, y_pred_upr),
'R^2': metrics.r2_score(y_test_upr, y_pred_upr)
```

- оценка результатов отработки каждой модели проводилась по метрикам, рассчитанным с использованием библиотеки Scikit-Learn



Обучение моделей

Результаты для параметра Модуль упругости при растяжении

Модуль упругости при растяжении			
Model	MSE	MAE	R^2
RandomForestRegressor_best	0.03472	0.15124	-0.00263
KNeighborsRegressor_best	0.03472	0.15123	-0.00280
LinearRegression_default	0.03481	0.15221	-0.00522
LinearRegression_best	0.03481	0.15221	-0.00522
DummyRegression_mean	0.03483	0.15120	-0.00593
MLPRegressor_best	0.03483	0.15121	-0.00599
GradientBooster_best	0.03510	0.15179	-0.01367
RandomForestRegressor_default	0.03558	0.15344	-0.02753
MLPRegressor_default	0.03643	0.15627	-0.05194
GradientBooster_default	0.03780	0.15633	-0.09154
KNeighborsRegressor_default	0.04169	0.16648	-0.20410

Результаты для параметра Прочность при растяжении

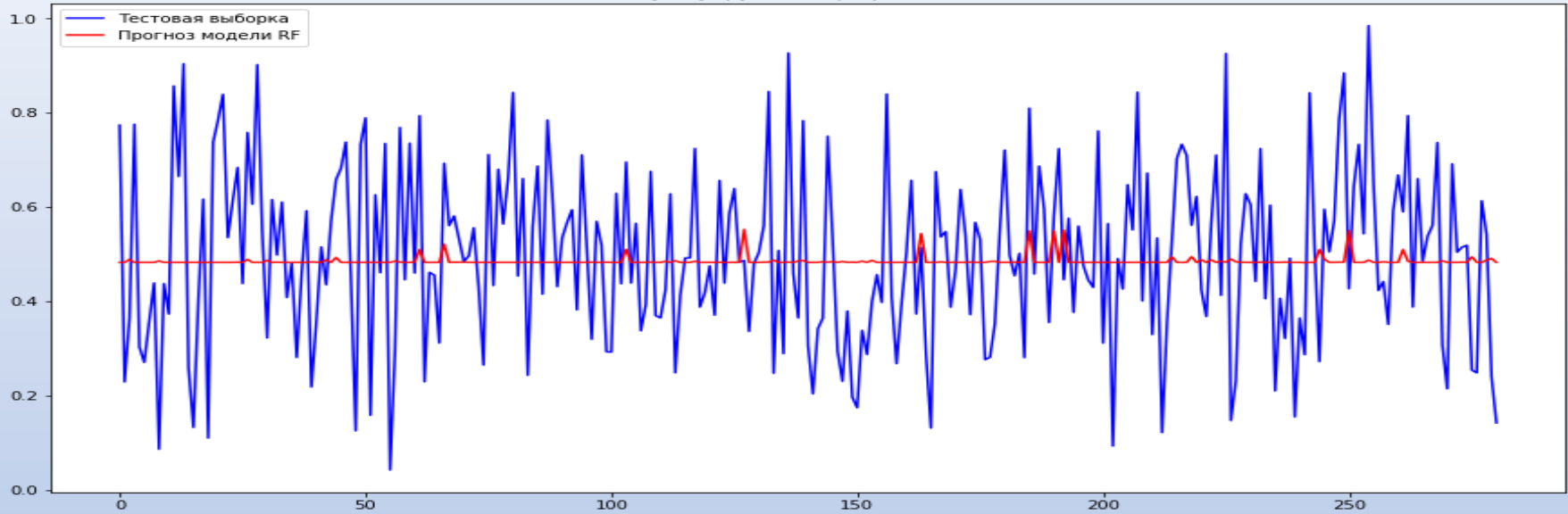
Прочность при растяжении			
Model	MSE	MAE	R^2
KNeighborsRegressor_best	0.03481	0.14658	-0.02799
DummyRegression_mean	0.03486	0.14675	-0.02967
RandomForestRegressor_best	0.03495	0.14725	-0.03239
LinearRegression_default	0.03545	0.14790	-0.04700
LinearRegression_best	0.03545	0.14790	-0.04700
MLPRegressor_best	0.03562	0.14855	-0.05194
GradientBooster_best	0.03567	0.15002	-0.05363
RandomForestRegressor_default	0.03599	0.15050	-0.06305
GradientBooster_default	0.03789	0.15570	-0.11903
MLPRegressor_default	0.03789	0.15570	-0.11903
KNeighborsRegressor_default	0.04143	0.15725	-0.22358



Обучение моделей

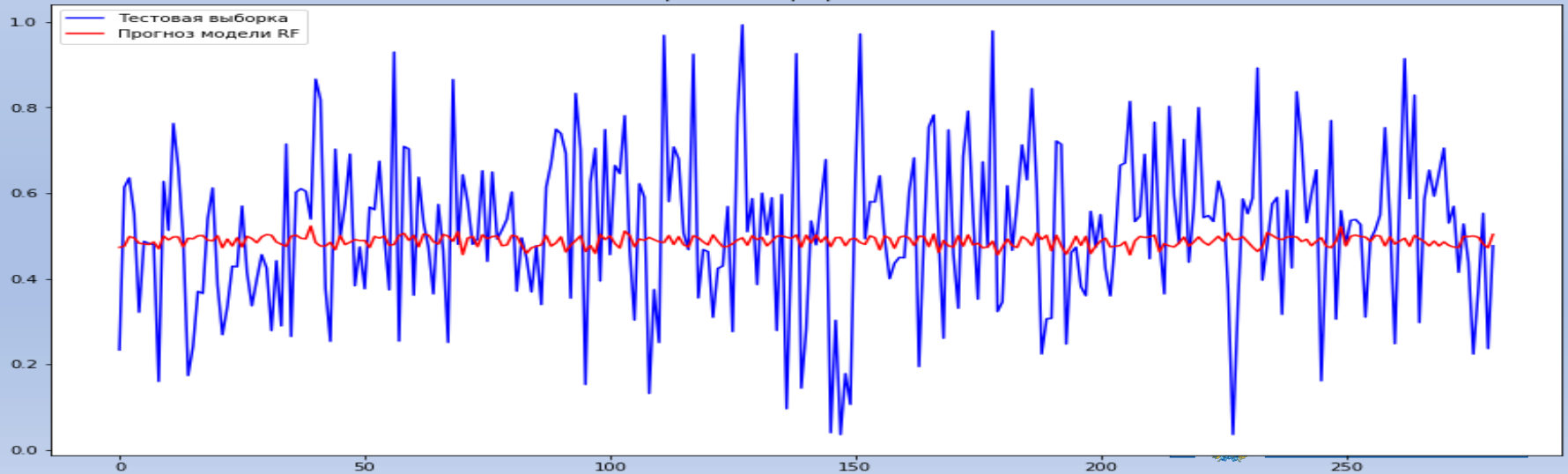
Лучшая модель Случайный лес:

Модуль упругости при растяжении



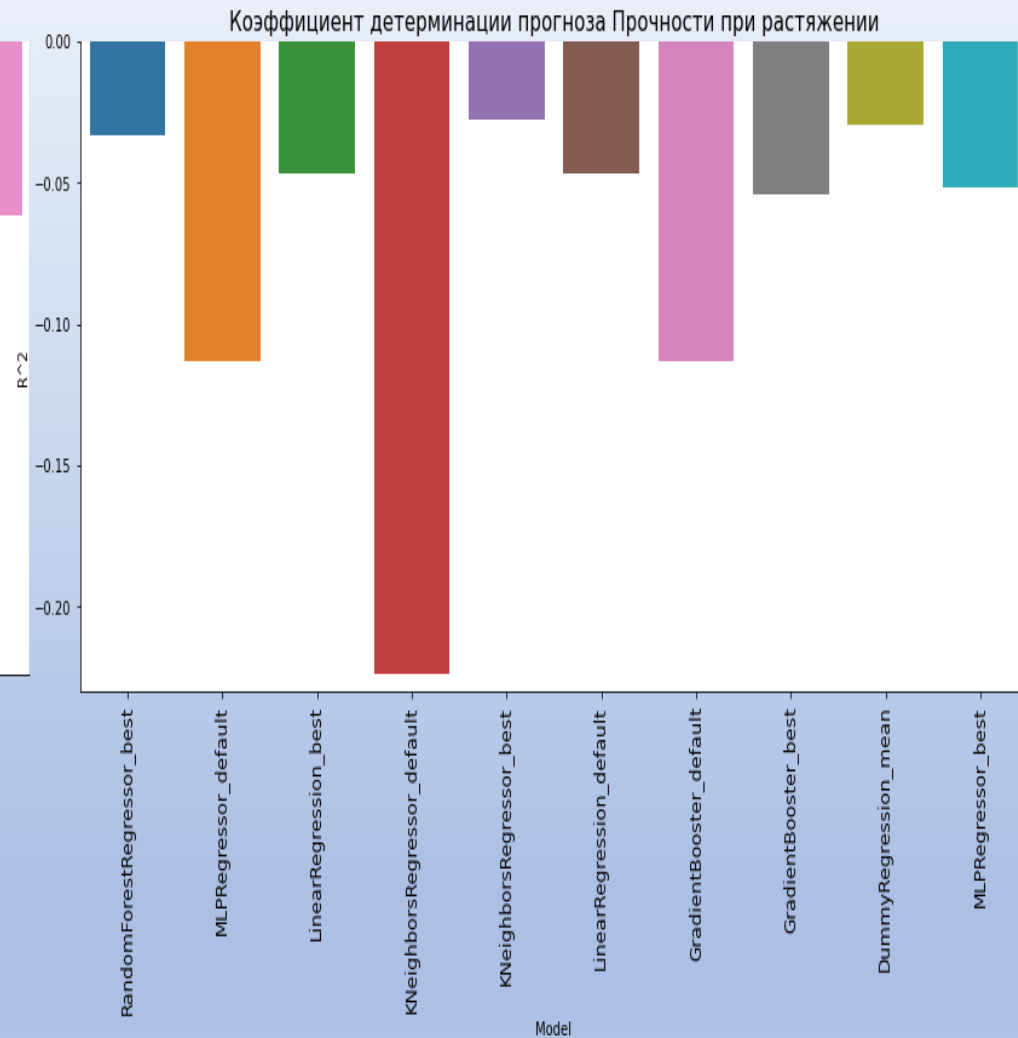
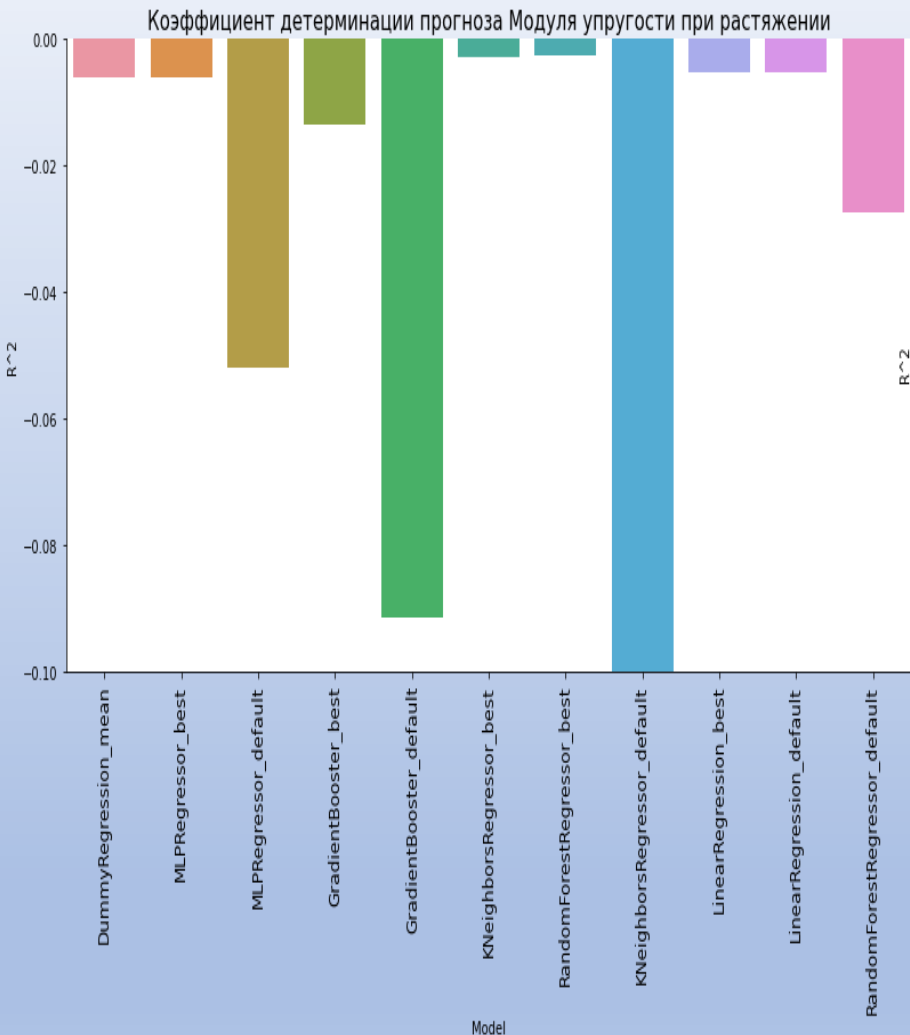
Лучшая модель К-ближайших соседей:

Прочность при растяжении



Обучение моделей

Коэффициенты детерминации R²:



Нейронные сети

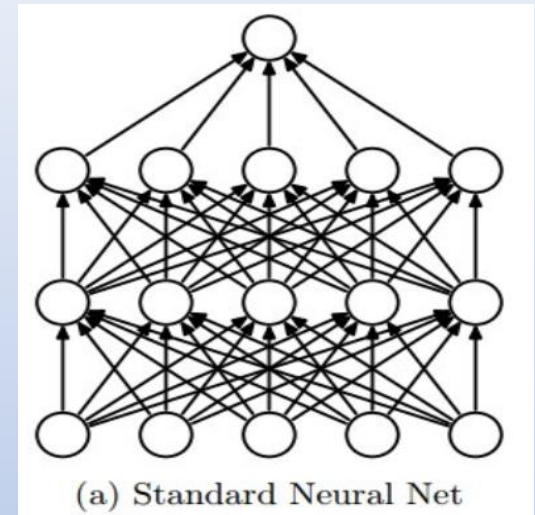
Архитектура сети без dropout

```
# сводная информация по архитектуре сети  
model_1.summary()
```

Model: "sequential_3"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 64)	832
dense_2 (Dense)	(None, 32)	2080
dense_3 (Dense)	(None, 12)	396
out (Dense)	(None, 1)	13

=====
Total params: 3,321
Trainable params: 3,321
Non-trainable params: 0

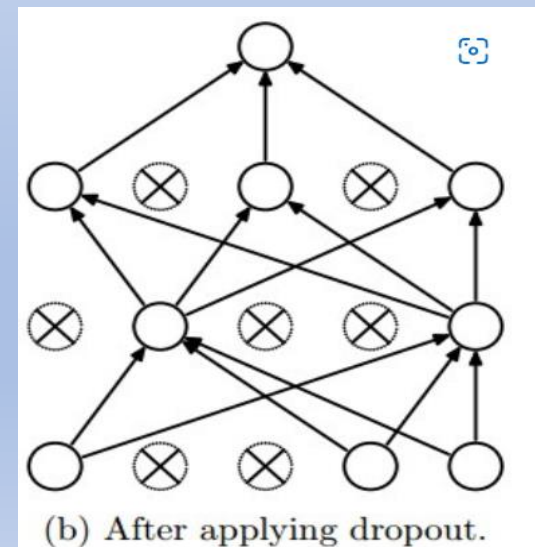


Архитектура сети с dropout

Model: "sequential_5"

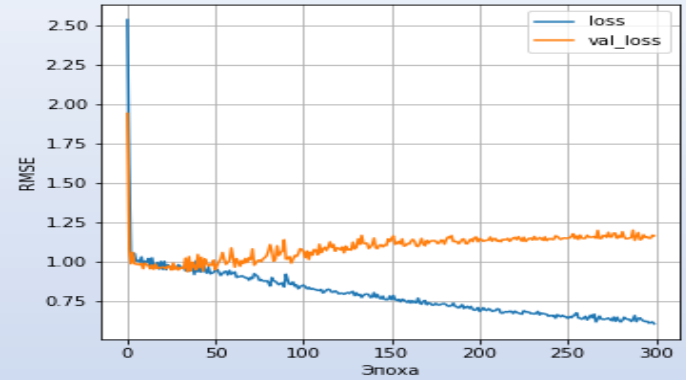
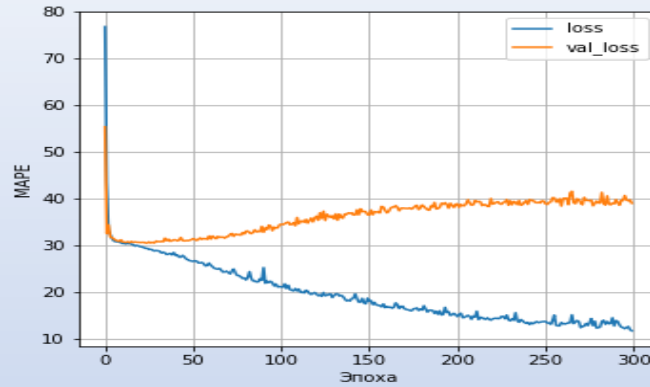
Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 64)	832
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2080
dropout_2 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 12)	396
dropout_3 (Dropout)	(None, 12)	0
out (Dense)	(None, 1)	13

=====
Total params: 3,321
Trainable params: 3,321
Non-trainable params: 0

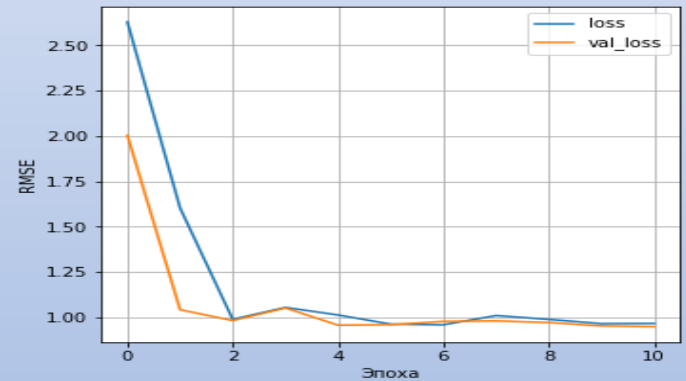
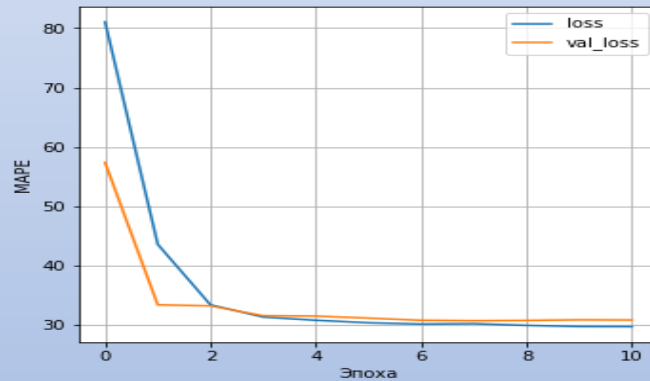


Нейронные сети

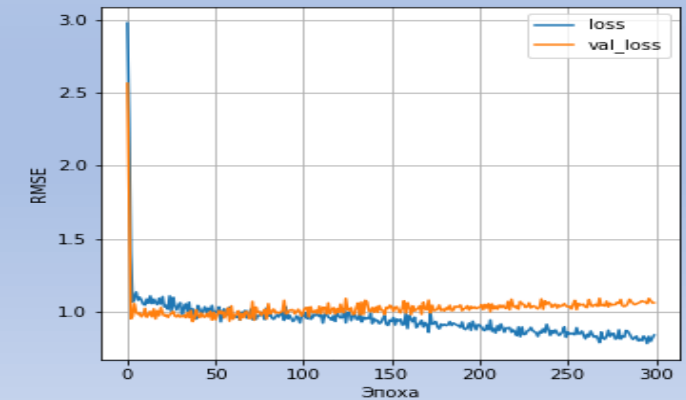
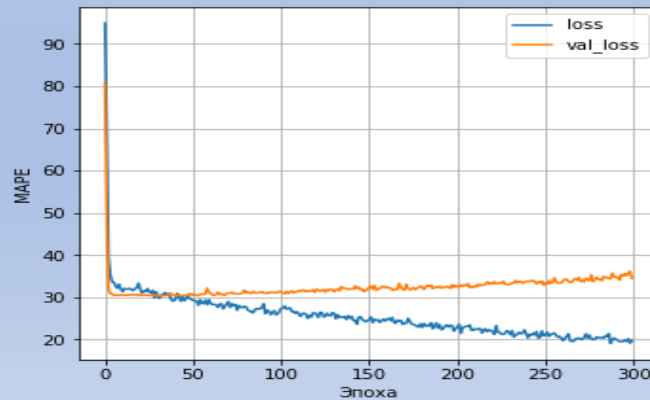
«стандарт»:



с ОСТАНОВКОЙ:



dropout-слой

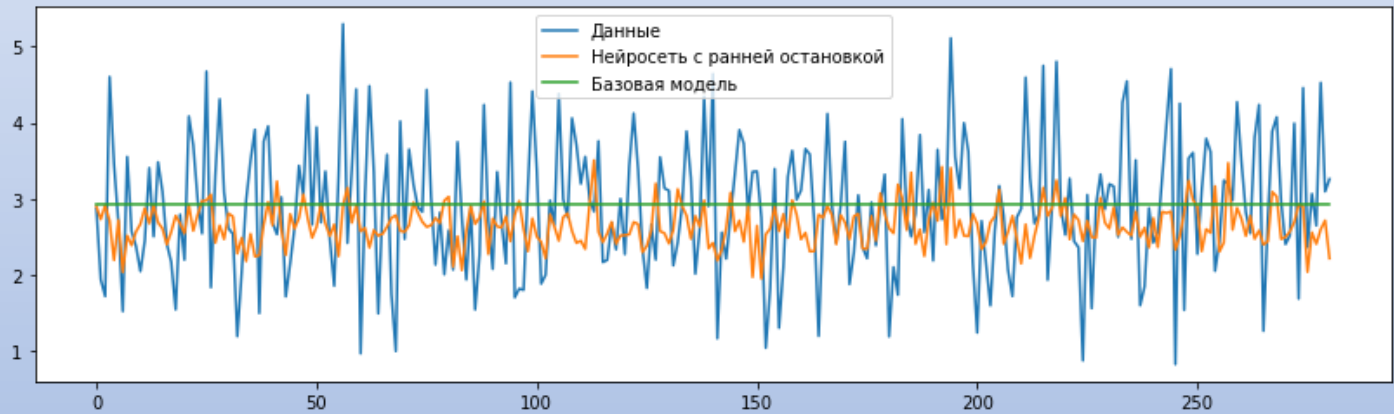


Нейронные сети

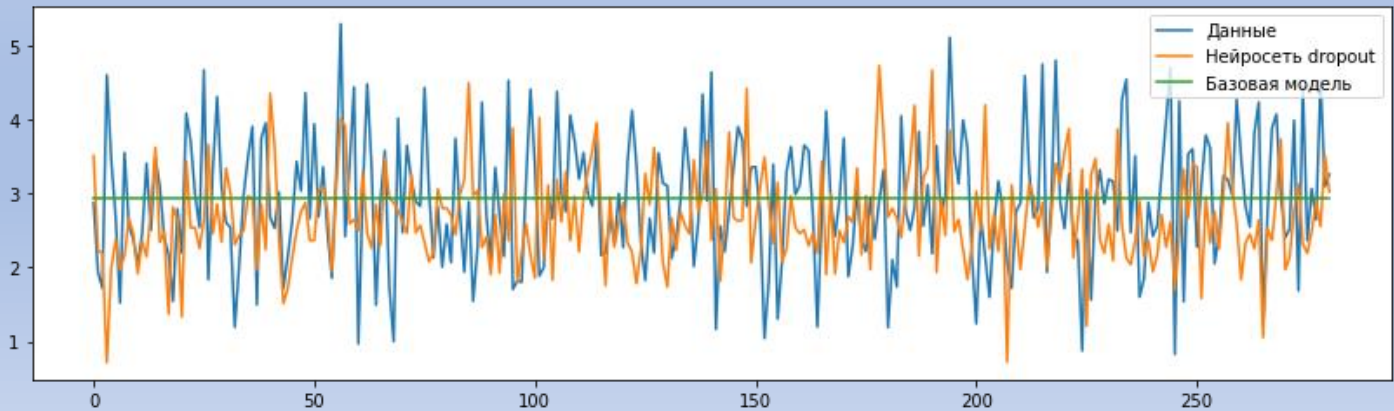
«стандарт»:



с остановкой:



dropout-слой



Нейронные сети

Сравнение метрик смоделированных нейронных сетей с «предсказанием» среднего значения:

	R2	RMSE	MAE	MAPE	max_error
DummyRegressor	-0.001569	-0.873433	-0.694175	-0.298186	-2.359781
Нейросеть переобученная	-0.539060	-1.082722	-0.863116	-0.338348	-3.268139
Нейросеть с ранней остановкой	-0.111431	-0.920090	-0.738757	-0.277341	-2.416448
Нейросеть dropout	-0.652519	-1.121921	-0.883273	-0.345372	-3.127986

Лучшая обобщающая способность и меньшие значения ошибок на тестовом множестве оказались у нейронной сети, обученной с ранней остановкой. Но и она предсказывает гораздо хуже базовой модели (усредненное значение).

Оценка моделей на тренировочной и тестовой выборках

	R2	RMSE	MAE	max_error
Случайный лес				
Модуль упругости при растяжении, тренировочный	0.01238	-0.19223	-0.15483	-0.51702
Модуль упругости при растяжении, тестовый	-0.00263	-0.18633	-0.15124	-0.49743
К-ближайших соседей				
Прочность при растяжении, тренировочный	0.00650	-0.18943	-0.15111	-0.50857
Прочность при растяжении, тестовый	-0.02799	-0.18656	-0.14658	-0.49695
Сеть с ранней остановкой				
Соотношение матрица-наполнитель, тренировочный	-0.18256	-0.98054	-0.79410	-3.03017
Соотношение матрица-наполнитель, тестовый	-0.11143	-0.92009	-0.73876	-2.41645

обучение модели

оценка качества работы модели

train	test
70%	30%

Выводы

- распределение исходных данных близко к нормальному;
- корреляция между парами признаков практически отсутствует;
- примененные модели и нейронные сети не показали высокой эффективности в прогнозировании свойств композитов в этом конкретном исследовании, необходимы дополнительные вводные данные для улучшения моделей;
- лучшие метрики на основании исключительно количественного сравнения показали модели, созданные с использованием алгоритмов Случайный лес, К-ближайших соседей и нейронная сеть с ранней остановкой.

СПАСИБО ЗА ВНИМАНИЕ!

