ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА по курсу «Data Science Pro»

Тема: Прогнозирование конечных свойств новых материалов (композиционных материалов).

Токарева Людмила Евгеньевна

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Постановка задачи

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).

4 Сделать commit приложения на github.com.

Композиционные материалы — это искусственно созданные материалы, состоящие из нескольких других с четкой границей между ними. Композиты обладают теми свойствами, которые не наблюдаются у компонентов по отдельности. При этом композиты являются монолитным материалом, т.е. компоненты материала неотделимы друг от друга без разрушения конструкции в целом. Яркий пример композита — железобетон.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Входные данные

Первый анализ данных

В качестве входных данных нам был дан zip архив включающий в себя два Excel файла с данными:

- X_bp.xlsx (1023 строки)
- X_nup.xlsx (1040 строк)

Методом визуального наблюдения устанавливаем, что данные в файлах являются структурированными. Каждое свойство хранится в отдельной колонке (10 и 3 колонок данных соответственно), значения представляет собой числа с десятичным разделителем. Первая колонка – индекс (целое число). Первая строка содержит наименование свойств, указанных в

	A	B	C	D
		Угол нашивки, г	Шаг нашивки	Плотность нашивки
0	0	0	4	57
1	1	0	4	60
2	2	0	4	70
3	3	0	5	47
4	4	0	5	57
5	5	0	5	60
6	6	0	5	70
7	7	0	7	47
8	8	0	7	57

	A	B	C	D	E	F	G	H	I	J	K	L
		Соотношение ма	Плотность, кг/м3	модуль упругос	Количество отве	Содержание эпс	Температура вс	Поверхностная г	Модуль упругос	Прочность при р	Потребление смолы, г/м2	
0	0	1,857142857	2030	738,7368421	30	22,26785714	100	210	70	3000	220	
1	1	1,857142857	2030	738,7368421	50	23,75	284,6153846	210	70	3000	220	
2	2	1,857142857	2030	738,7368421	49,9	33	284,6153846	210	70	3000	220	
3	3	1,857142857	2030	738,7368421	129	21,25	300	210	70	3000	220	
4	4	2,771331058	2030	753	111,86	22,26785714	284,6153846	210	70	3000	220	
5	5	2,767918089	2000	748	111,86	22,26785714	284,6153846	210	70	3000	220	
6	6	2,569620253	1910	807	111,86	22,26785714	284,6153846	210	70	3000	220	
7	7	2,56147541	1900	535	111,86	22,26785714	284,6153846	380	75	1800	120	

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Объединение таблиц

Согласно условию задачи необходимо объединить две таблицы по индексу методом INNER JOIN и таким образом получить новый датасет в который войдут строки, присутствующие в обоих таблицах.

```
SELECT поля  INTO "full_X_bp "
FROM "X_bp" AS A INNER JOIN "X_nup" AS B
ON A."indx" = B."Indx";
```

```
X_bp_nup_df = pandas.DataFrame.concat([X_bp_df, X_nup_df],
axis=1, join="inner")
```

В результирующей выборке будет 1023 строки и 13 колонок (свойств материалов)

Командой pandas.DataFrame.info() можно получить признаки датасета.

Признаки объединенного датасета

Название	Тип данных	Наличие пустых значений	Уникальных значений
1	2	3	4
Соотношение матрица-наполнитель	float64	нет	1014
Плотность, кг/м3	float64	нет	1013
модуль упругости, ГПа	float64	нет	1020
Количество отвердителя, м.%	float64	нет	1005
Содержание эпоксидных групп,%_2	float64	нет	1004
Температура вспышки, С_2	float64	нет	1003
Поверхностная плотность, г/м2	float64	нет	1004
Модуль упругости при растяжении, ГПа	float64	нет	1004
Прочность при растяжении, МПа	float64	нет	1004
Потребление смолы, г/м2	float64	нет	1003
Угол нашивки, град	float64	нет	2
Шаг нашивки	float64	нет	989
Плотность нашивки	float64	нет	988

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Описательная статистика

Согласно заданию необходимо получить показатели описательной статистики набора данных (среднее, медиану, максимальное, минимальное и стандартное отклонение).

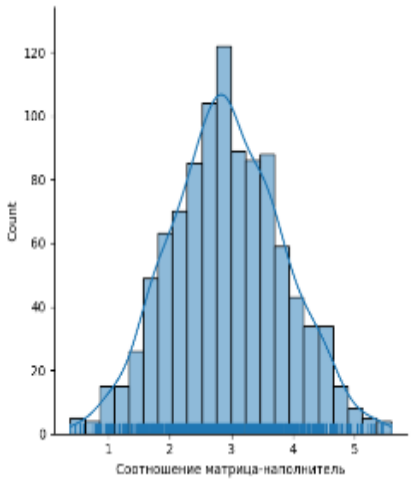
Статистические функции PostgreSQL:
AVG(поле), percentile_cont(0.5), MAX(поле), MIN(поле),stddev(поле)

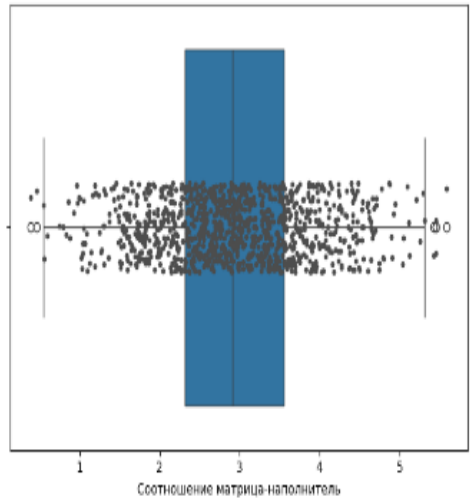
Статистические функции pandas.DataFrame: describe(), mean(), median(), max(), min(), std()

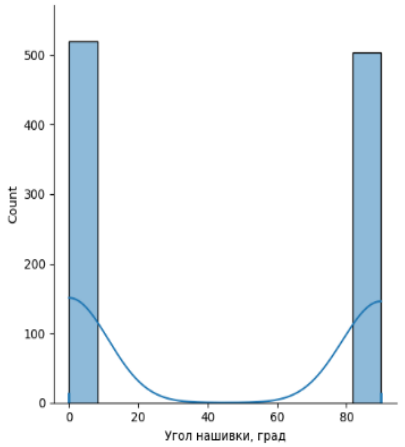
Свойство	среднее	медиана	макс.	минимум	станд. откл.
Соотношение матрица-наполнитель	2.9303657734325483	2.90687765033521	5.591741598697 54	0.389402605178414	0.9132222362148388
Плотность, кг/м3	1975.7348881101545	1977.62165679058	2207.77348061119	1731.764635096	73.72923055065388
модуль упругости, ГПа	739.9232327560721	739.6643276977 92	1911.53647700054	2.43690875350 75	330.23158056102693
Количество отвердителя, м.%	110.57076864736254	110.564839894065	198.953207190451	17.7402745562519	28.2959112887888 15
Содержание эпоксидных групп,%_2	22.24438954776773	22.2307437560244	33.0	14.2549854977161	2.4063012915294304
Температура вспышки, С_2	285.882151351621 87	285.896812331237	413.273418243566	100.0	40.943259952923306
Поверхностная плотность, г/м2	482.731833038418 53	451.86436518306	1399.54236233989	0.603739925153945	281.3146902366 61
Модуль упругости при растяжении, ГПа	73.32857125009068	73.2688045943481	82.682051035271	64.0540605597917	3.1189828894693 03
Прочность при растяжении, МПа	2466.922842697902	2459.52452600309	3848.43673187618	1036.85660535	485.628006278535 96
Потребление смолы, г/м2	218.42314367654285	219.198882195134	414.590628361534	33.8030255329625	59.735930873323504
Угол нашивки, град	44.252199413489734	0.0	90.0	0.0	45.015793407611 42
Шаг нашивки	6.89922207767501 75	6.91614385594 91	14.44052187539 69	0.0	2.5634670728338 82
Плотность нашивки	57.153929432857645	57.3419198469929	103.988901301494	0.0	12.350968798651323

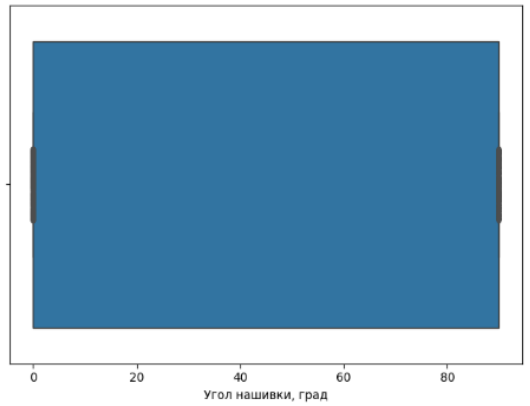
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Плотность распределения и ящик с «усами»









Библиотека Seaborn и ее функции displot, rugplot, boxplot, stripplot позволяют построить плотность распределения и ящик с «усами».

Из гистограмм распределения переменных и диаграмм «ящик с усами» видно, что все признаки, кроме «Угол нашивки», имеют нормальное распределение (график в виде «колокола»), принимают неотрицательные значения и непрерывны. Также визуально видны «выбросы» – аномальные значения, которые отрицательно будут сказываться на выявлении общей закономерности и процессу обучения и предсказания.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Выявление выбросов (аномалий)

Есть следующие методы выявления выбросов для признаков с нормальным распределением:

1. Метод стандартного отклонения (3-х сигм), где верхняя (Limmax) и нижняя (Limmin) граница выбросов определяются по формуле:

   Limmax = mean + Ns∗S

   Limmin = mean - Ns∗S ,

   где mean - среднее значение;

   S - стандартное отклонение;

   Ns = 3 - заданное число стандартных отклонений (3-х сигм);

2. Метод межквартильных расстояний, где верхняя (Limmax) и нижняя (Limmin) граница выбросов определяются по формуле:

   Limmax=Q3+Ni∗IQR

   Limmin=Q1−Ni∗IQR ,

   где Q3 - третий квартиль (значение 75%);

   Q1 - первый квартиль (значение 25%);

   IQR — интерквартильное расстояние (или интерквартильный размах), определяемое по формуле IQR= Q3− Q1;

   Ni = 1.5 — заданное число интерквартильного размаха.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Фильтрарция данных методом трех сигм

Реализовать формулы отсечения выбросов можно как средствами PostgreSQL конструкцией языка запросов SQL, так и Pandas.DataFrame.

```
SELECT  * FROM public."full_X_bp" WHERE mat_nap > ((SELECT AVG(mat_nap) FROM public."full_X_bp")
- 3.0 * (SELECT stddev(mat_nap) FROM public."full_X_bp" )) AND mat_nap < ((SELECT AVG(mat_nap)
FROM public."full_X_bp") + 3.0 * (SELECT stddev(mat_nap) FROM public."full_X_bp" ))….
```

Из 1023 строк остается 1002 строки удален 21 выброс

```
for i in filtered_df.columns:
  filtered_df = filtered_df[(filtered_df[i] > filtered_df[i].mean() - 3 *filtered_df[i].std())
    &
  (filtered_df[i] < filtered_df[i].mean() + 3 *filtered_df[i].std())]
```

Из 1023 строк остается 1000 строк. Удалено 23 выброса

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Задача регрессии

Предсказание значений вещественной, непрерывной переменной — это задача регрессии.

В настоящее время разработано много математических методов регрессионного анализа и многие из них реализованы в программные алгоритмы, например, для Python такие алгоритмы собраны в библиотеку scikit-learn.

- Математические методы регрессивного анализа и их релизация в библиотеке scikit-learn, например:

Математический метод регрессивного анализа	Функция библиотеки scikit-learn	Гиперпараметры со значениями по умалчанию
Линейная регрессия лассо	Lasso	{'alpha': 1.0, 'copy_X': True, 'fit_intercept': True, 'max_iter': 1000, 'positive': False, 'precompute': False, 'random_state': None, 'selection': 'cyclic', 'tol': 0.0001, 'warm_start': False}

Использование:

- объявить объект reg = Lasso()
- обучить histoty = reg.Fit(X_train, Y_train)
- предсказать predict = reg.predict(x_train

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Нормализация данных и разделение на 3 датасета

Нормализация данных это процесс предпроцессинга, приводящий данные к одному масштабу от значения от 0 до 1

- Имеются средства sklearn.preprocessing.StandardScaler и sklearn.preprocessing.MinMaxScaler, но мы будем испольховать ручную нормалицацию по формуле (x – min()) / (max() – min()) с сохранением значений дельты и множителя для каждого поля.

Разделим нормализованные данные на три датасета, где целевой параметр поставим в последней колонке. Это позволить выделять входные параметры конструкцией X = values[:,:-1] # все кроме последнего и выходные параметры конструкцией Y = values[:,-1] # только последний.

Модуль упругости

Модуль прочности
Модуль прочности

Материал - наполнитель

Json


```
Соотношение матрица-наполнитель:
    mnozitel:                    5.202338993519127
    delta:                       0.389402605178414
Плотность, кг/м3:
    mnozitel:                    408.2565382314201
    delta:                       1784.48224524858
модуль упругости, ГПа:
    mnozitel:                    1646.9787971223625
    delta:                       2.4369087535075
Количество отвердителя, м.%:
    mnozitel:                    162.8955521795974
    delta:                       29.9561496534826
Содержание эпоксидных групп,%_2:
    mnozitel:                    13.259200571021099
    delta:                       15.6958938036288
Температура вспышки, С_2:
    mnozitel:                    230.16794095235497
    delta:                       173.484919924459
Поверхностная плотность, г/м2:
    mnozitel:                    1290.736374710296
    delta:                       0.603739925153945
Модуль упругости при растяжении, ГПа:
    mnozitel:                    18.627990475479308
    delta:                       64.0540605597917
Прочность при растяжении, МПа:
    mnozitel:                    2811.58012652618
    delta:                       1036.85660535
Потребление смолы, г/м2:
    mnozitel:                    345.8551534796663
    delta:                       41.0482779512307
Угол нашивки, град:
    mnozitel:                    90
    delta:                       0
Шаг нашивки:
    mnozitel:                    14.402882938698157
    delta:                       0.0376389366987437
Плотность нашивки:
    mnozitel:                    72.3918586171032
    delta:                       20.5716333306441
```

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Подбор гиперпараметров

Гиперпараметры

Модуль упругости

Модуль прочности

Материал - наполнитель

С помощью GridSearchCV провожу оценку параметров с использование поиска по сетке с перекрестной проверкой с количеством блоков 10. Перекрестная проверка уже встроена в GridSearchCV, а количество блоков указывается в параметрах функции GridSearchCV - cv = 10, а значения гиперпараметров, которые необходимо перебирать, указывается в массиве. Такую операцию провожу для каждой модели и каждого целевого параметра.

Данные	Метрика	Лучшие параметры
moduprrast.csv	max_error	{'alpha': 0.01}
moduprrast.csv	neg_mean_absolute_error	{'alpha': 0.0}
moduprrast.csv	neg_root_mean_squared_error	{'alpha': 0.0}
moduprrast.csv	r2	{'alpha': 0.0}
modprochrast.csv	max_error	{'alpha': 0.01}
modprochrast.csv	neg_mean_absolute_error	{'alpha': 0.01}
modprochrast.csv	neg_root_mean_squared_error	{'alpha': 0.01}
modprochrast.csv	r2	{'alpha': 0.01}
mat_nap.csv	max_error	{'alpha': 0.01}
mat_nap.csv	neg_mean_absolute_error	{'alpha': 0.01}
mat_nap.csv	neg_root_mean_squared_error	{'alpha': 0.01}
mat_nap.csv	r2	{'alpha': 0.01}

Рекомендованные параметры

LinearRegression
Lasso
Ridge
svm.SVR
neighbors.KneighborsRegressor
tree.DecisionTreeRegressor
ensemble.RandomForestRegressor
ensemble.GradientBoostingRegressor

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Испытание моделей

Настройки по умолчанию (default) и оптимальные (optimize)

Модуль упругости

Модуль прочности

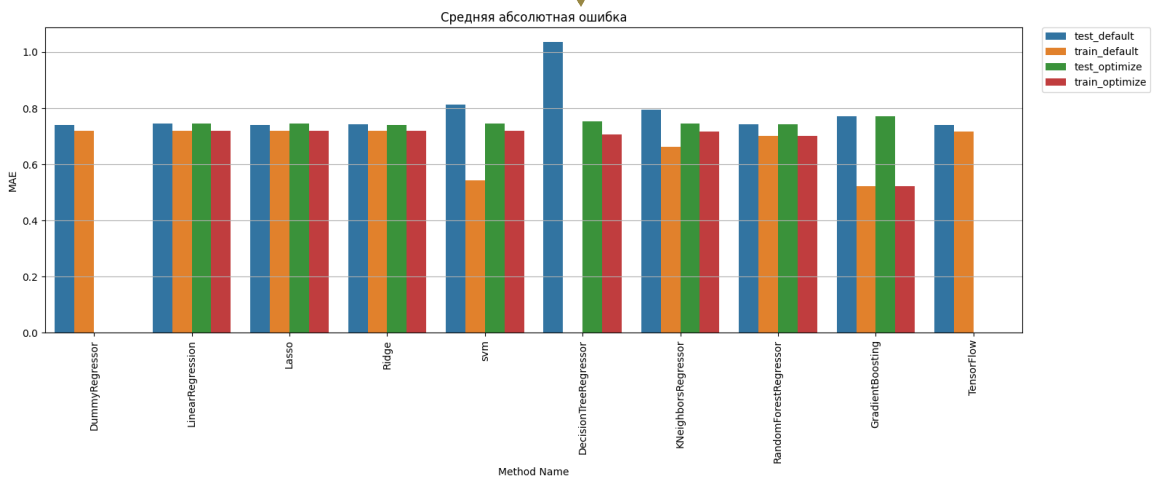
Материал - наполнитель

Train, test

Train, test

Train, test

Скриптами (9_examine_mod_upr.py, 9_examine_mat_nap.py, 9_examine_modprocn.py) в цикле запускаю обучение и тестирование моделей с параметрами по умолчанию, т.е. модель из «коробки», и оптимально настроенными параметрами. Накапливаю оценочные показатели для тестовой и тренировочной выборки, для оптимальных и «дефолтовых» настроек и на основании, которых выстраиваю сравнительные графики и таблицу оценочных метрик В перечень моделей добавлен DummyRegressor, который делает прогнозы, используя простые правила. Этот регрессор полезен в качестве простой базовой линии для сравнения с другими (реальными) регрессорами.

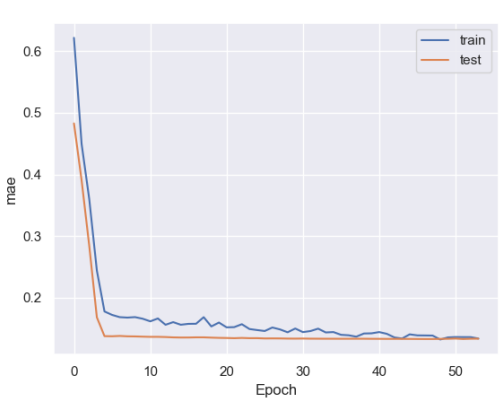


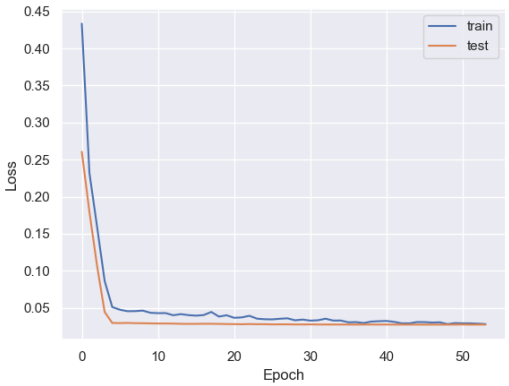
DummyRegressor
LinearRegression
Lasso
Ridge
svm.SVR
neighbors.KneighborsRegressor
tree.DecisionTreeRegressor
ensemble.RandomForestRegressor
ensemble.GradientBoostingRegressor

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

TensorFlow

- В качестве рабочего варианта модели для предсказаний написания приложения выберем tensorflow, которая имеет встроенную функцию сохранение обученной модели, которую можно загрузить из приложения и использовать в работе. Создание модели, обучение, тестирование и отображение хода обучения, а также сохранение обученной модели в файл выполним скриптами 12.1_keras__modupr.py, 12.2_keras_modproch.py, и 12.3_keras__mat_nap.py и соответственно получим файлы сохраненных моделей moduprmodel.keras, modprochrastmodel.keras и mat_napmodel.keras.



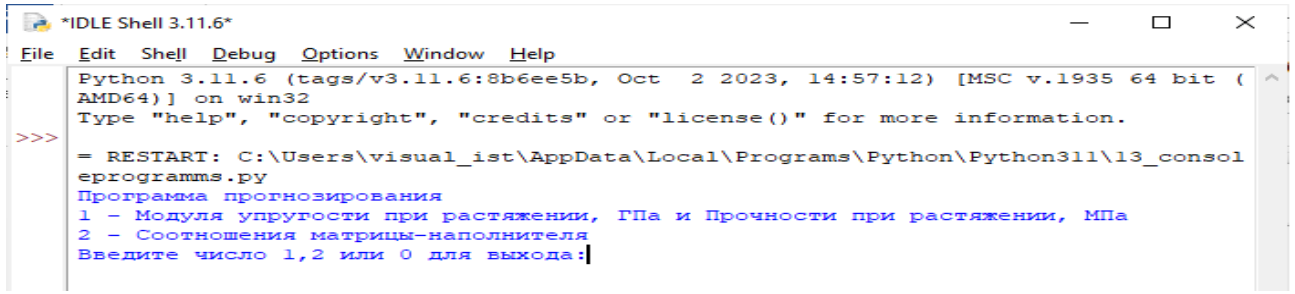


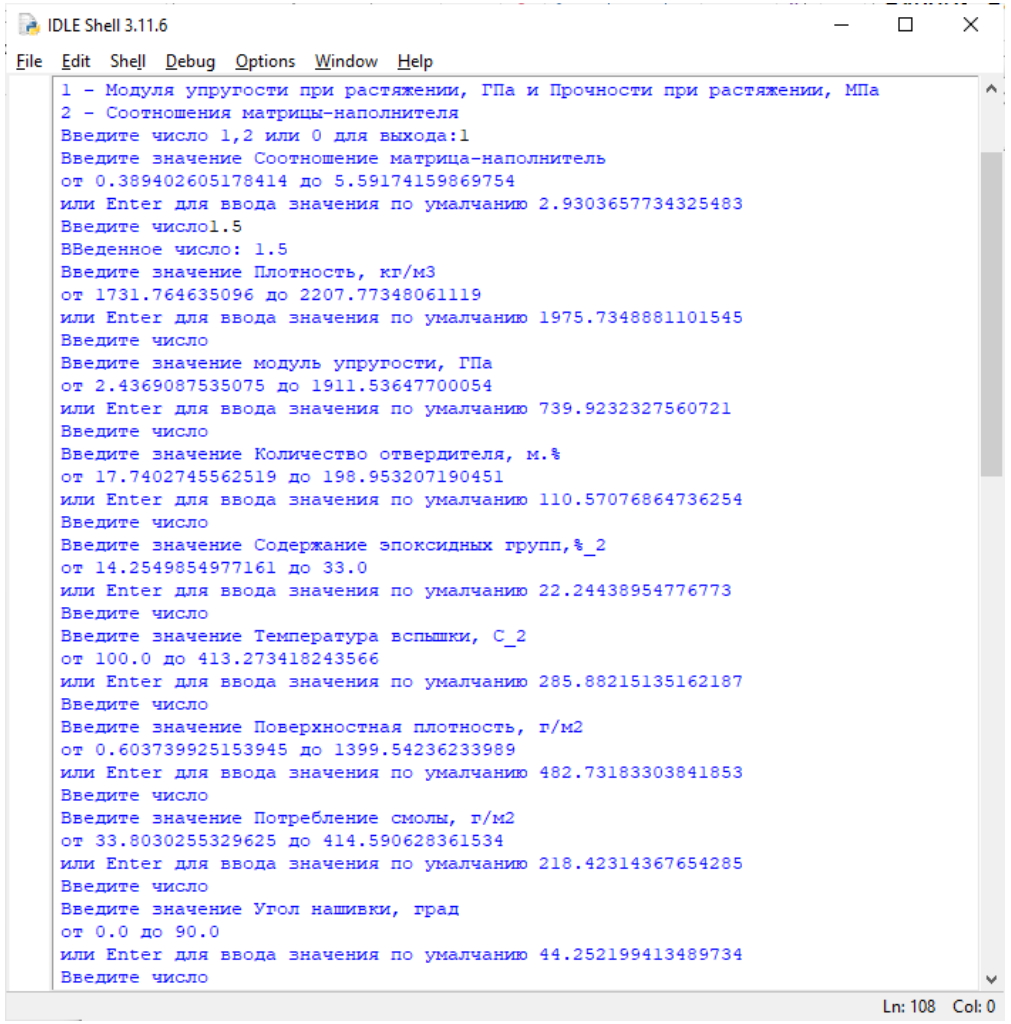
Ход обучения модели

На графиках обучения видно, что с этапами обучения (эпохами) количество ошибок уменьшается, и когда результат практически не улучшается заканчивается обучение функцией ранней остановки.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Интерфейс приложения

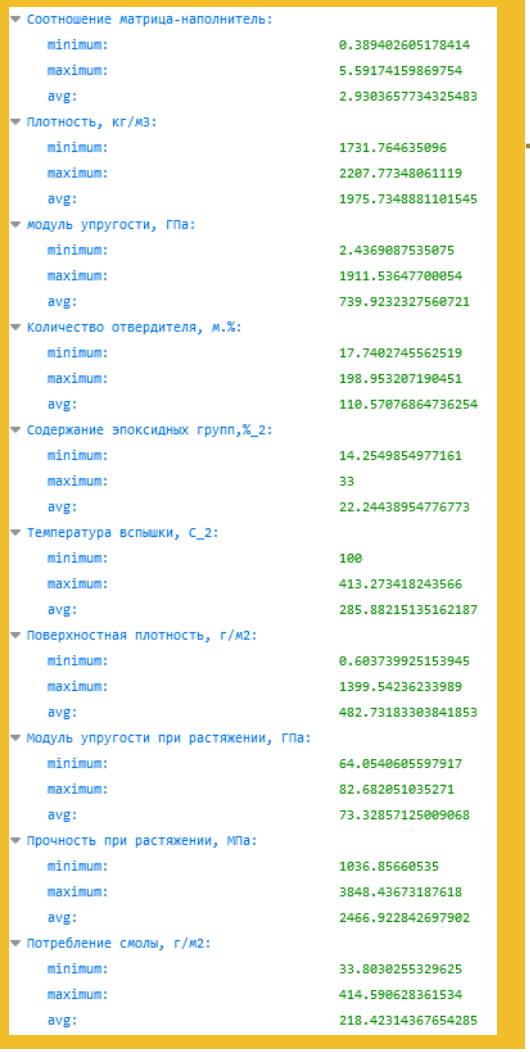




ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разработка приложения

Json



Подсказки по предельным значениям и значения по умолчанию

INPUT

Нормализация данных

LOAD

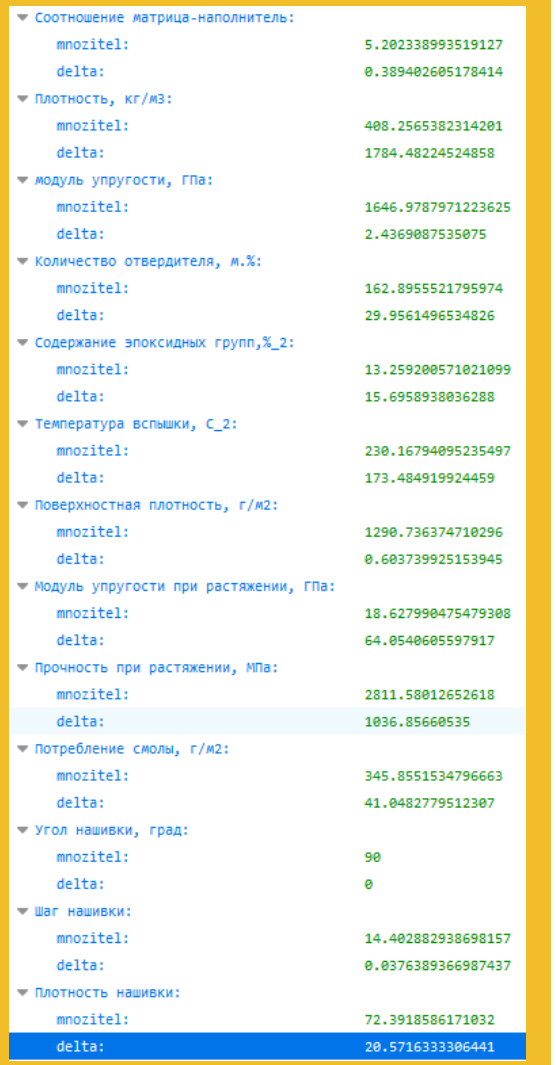
Сохраненные модели

PREDICT

Денормализация данных

Прогнозированные свойства

Json



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

do.bmstu.ru