ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по курсу

«Data Science»

Тема: «Прогнозирование конечных свойств новых материалов (композиционных материалов)»

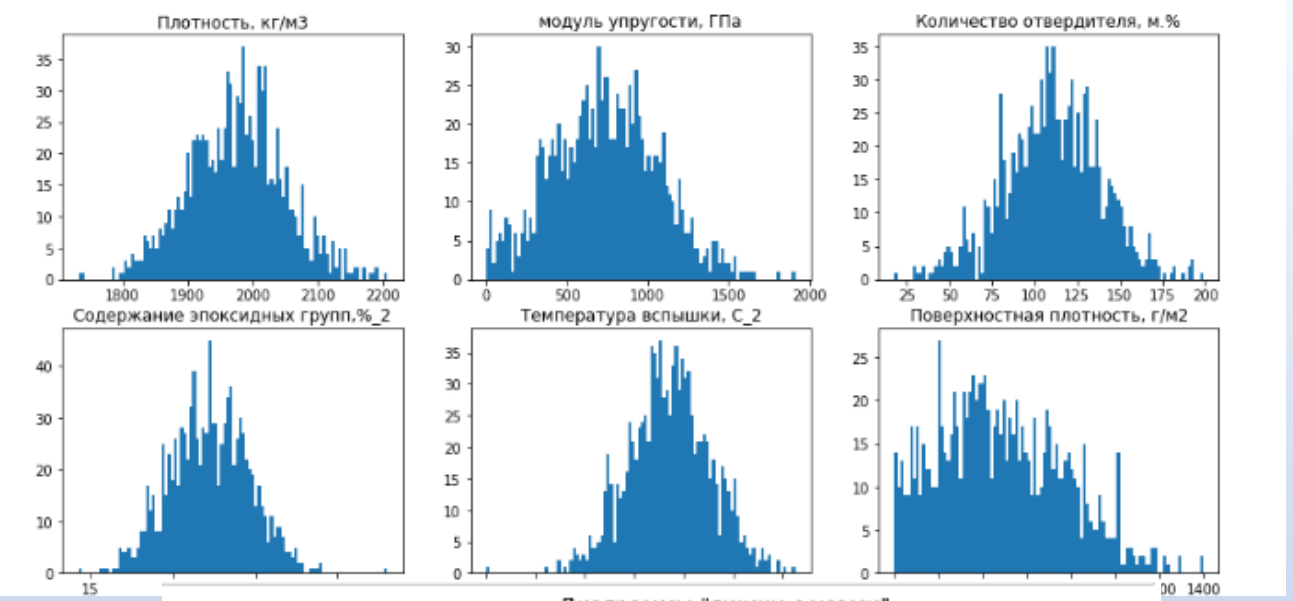
Слушатель Васильева Оксана Валерьевна

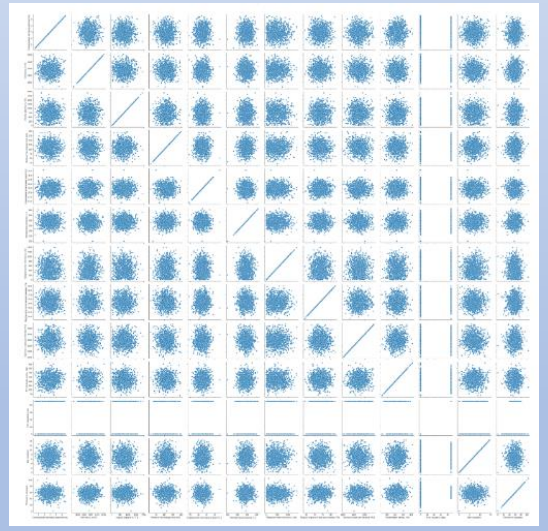
Задание

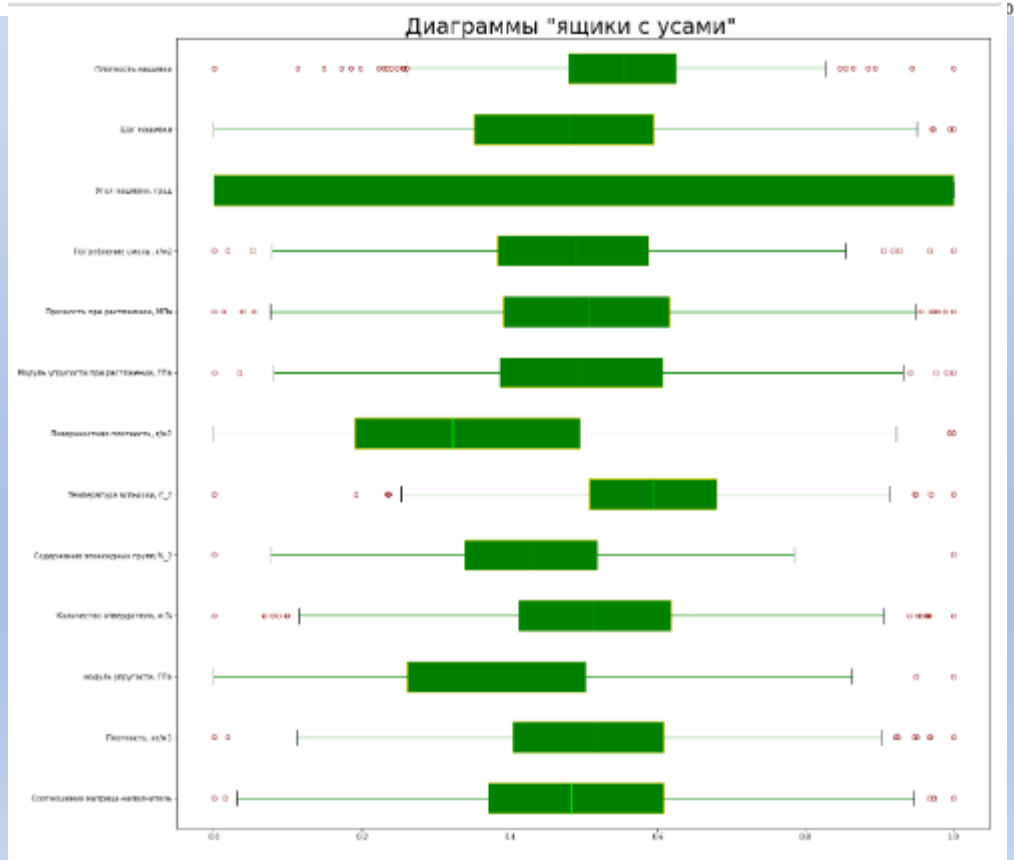
1) Провести разведочный анализ предложенных данных. Необходимо нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек. Необходимо также для каждой колонке получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков.
2) Обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении. При построении модели необходимо 30% данных оставить на тестирование модели, на остальных происходит обучение моделей. При построении моделей провести поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10.
3) Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель.
4) Разработать приложение с графическим интерфейсом или интерфейсом командной строки, которое будет выдавать прогноз, полученный в задании 4 или 5 (один или два прогноза, на выбор учащегося).
5) Оценить точность модели на тренировочном и тестовом датасете.
6) Создать репозиторий в GitHub / GitLab и разместить там код исследования. Оформить файл README.

Разведочный анализ данных

- Построение гистограмм, ящиков с усами, диаграмм рассеяния







Разведочный анализ

- Необходимо также для каждой колонки получить среднее, медианное значение
- Провести анализ и исключение выбросов
- Проверить наличие пропусков

```
df.duplicated().sum()
```

```
0
```

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375
модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	500.047452	739.664328	961.812526
Количество отвердителя, м.%	1023.0	110.570769	28.295911	17.740275	92.443497	110.564840	129.730366
Содержание эпоксидных групп,%_2	1023.0	22.244390	2.406301	14.254985	20.608034	22.230744	23.961934
Температура вспышки, С_2	1023.0	285.882151	40.943260	100.000000	259.066528	285.896812	313.002106
Поверхностная плотность, г/м2	1023.0	482.731833	281.314690	0.603740	266.816645	451.864365	693.225017
Модуль упругости при растяжении, ГПа	1023.0	73.328571	3.118983	64.054061	71.245018	73.268805	75.356612
Прочность при растяжении, МПа	1023.0	2466.922843	485.628006	1036.856605	2135.850448	2459.524526	2767.193119
Потребление смолы, г/м2	1023.0	218.423144	59.735931	33.803026	179.627520	219.198882	257.481724
Угол нашивки, град	1023.0	44.252199	45.015793	0.000000	0.000000	0.000000	90.000000
Шаг нашивки	1023.0	6.899222	2.563467	0.000000	5.080033	6.916144	8.586293
Плотность нашивки	1023.0	57.153929	12.350969	0.000000	49.799212	57.341920	64.944961

```
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, 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
```

```
df.isnull().sum()
```

```
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
dtype: int64
```

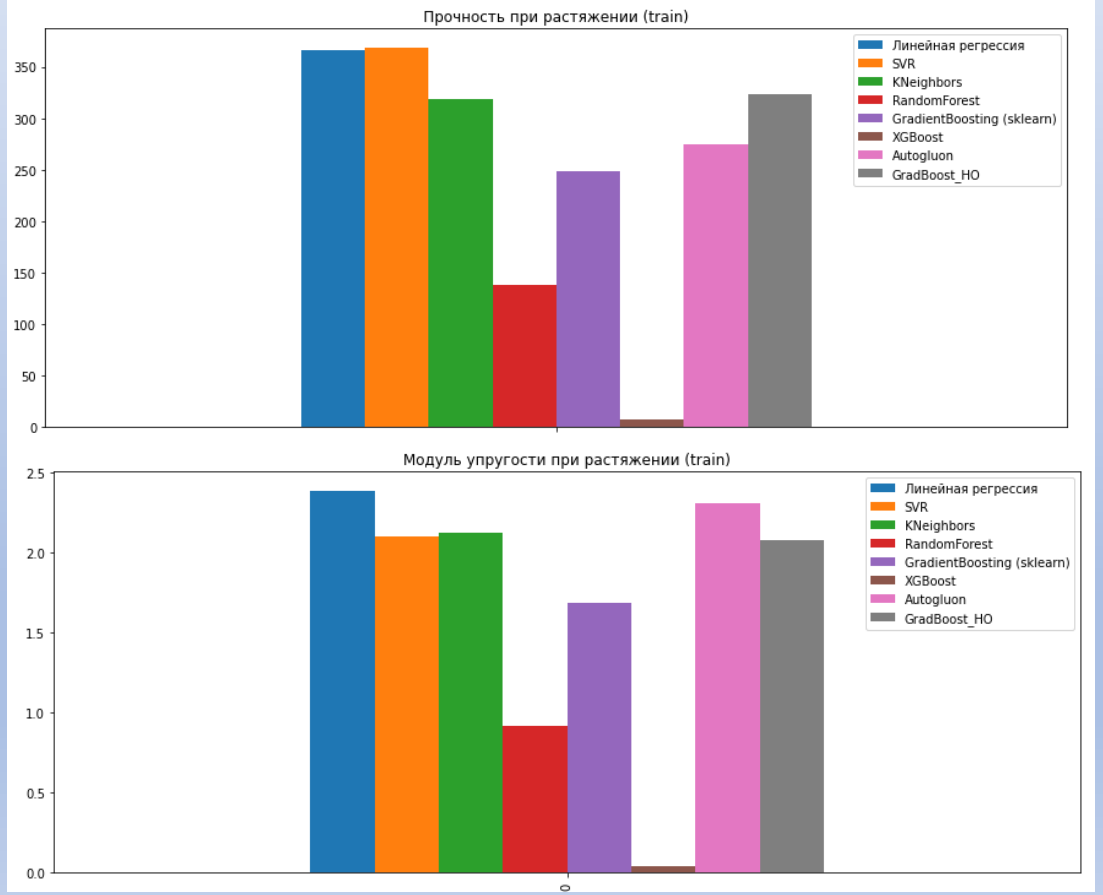
Удалим выбросы.

```
df=df.dropna()
```

Разработка и обучение моделей

- Линейная регрессия;
- Метод случайного леса;
- Метод опорных векторов;
- Метод k-ближайших соседей;
- Градиентный бустинг (библиотека sklearn) до и после подбора гиперпараметров;
- Градиентный бустинг (библиотека xgboost);
- autogluon

- MAE на тренировочной выборке



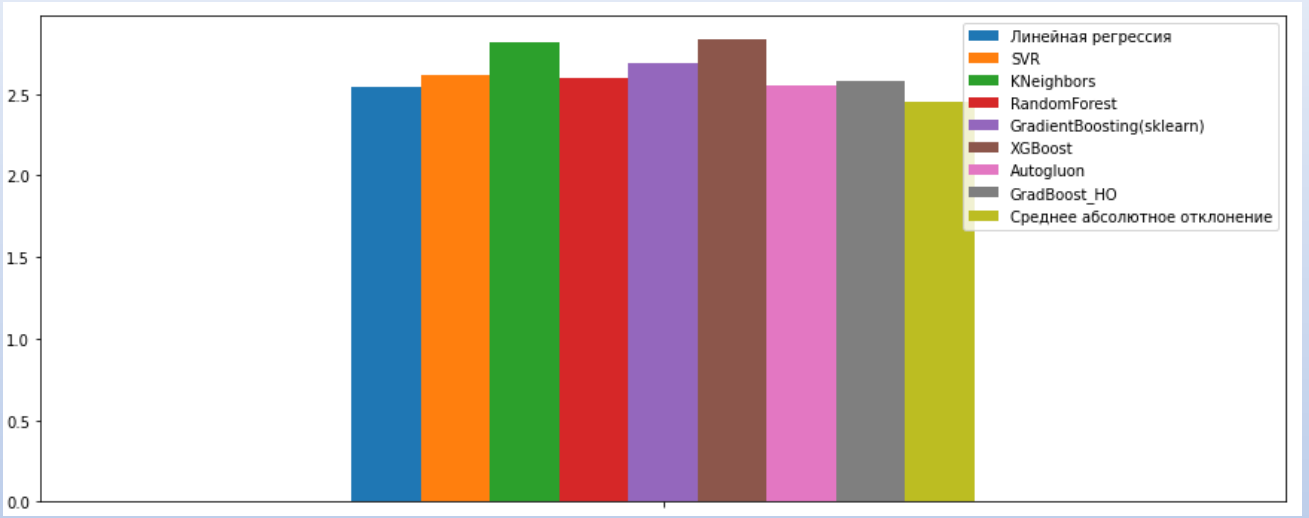
Гиперпараметрическая оптимизация градиентного бустинга

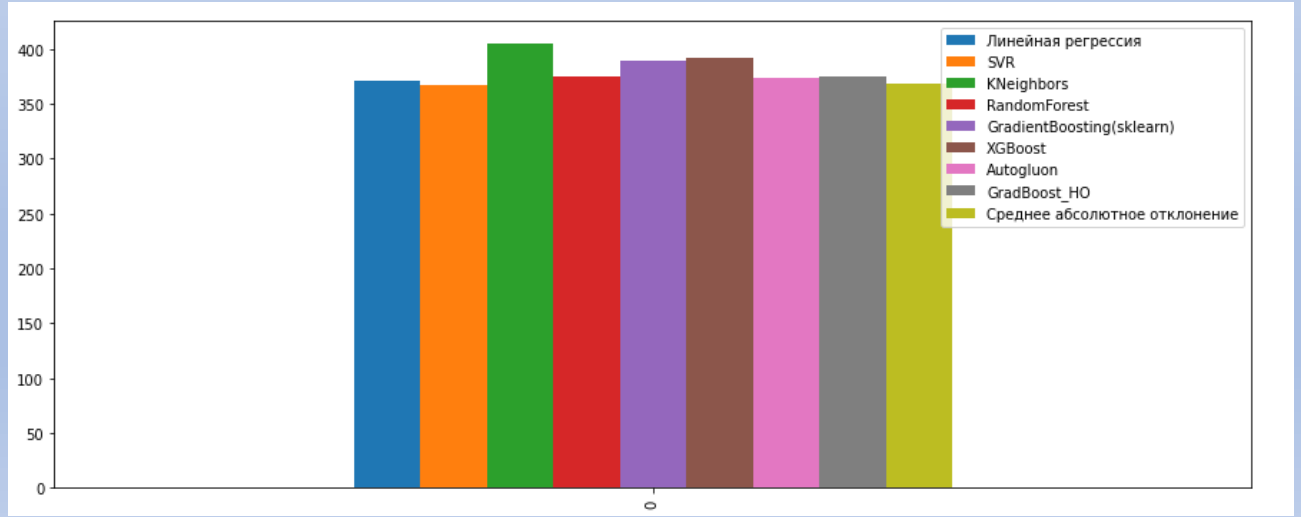
- Функция ошибки(loss),
- Количество (n_estimators) и глубина деревьев (max_depth),
- Минимальный размер листа (min_samples_leaf),
- Минимальное ветвление на узел (min_samples_split),
- Функция разделения (max_features)

```
loss = ['ls', 'absolute_error', 'huber']
#деревья
n_estimators = [100, 500, 900, 1100, 1500]
# Глубина деревьев
max_depth = [2, 3, 5, 10, 15]
# минимальный размер листа
min_samples_leaf = [1, 2, 4, 6, 8]
# Минимальное количество ветвлений на узел
min_samples_split = [2, 4, 6, 10]
# Размер делящегося множества
max_features = ['auto', 'sqrt', 'log2', None]

# Сетка гиперпарраметров
hyperparameter_grid = [{'loss': loss},
                       {'n_estimators': n_estimators},
                       {'max_depth': max_depth},
                       {'min_samples_leaf': min_samples_leaf},
                       {'min_samples_split': min_samples_split},
                       {'max_features': max_features}]
```

Тестирование моделей

Метод	Модуль упругости при растяжении, mae	Прочность при растяжении, mae
Линейная регрессия	2.5464192980820095	370.5426179675644
Метод опорных векторов	2.6116531333051856	366.5975512794085
Метод k-ближайших соседей	2.818761217799366	405.2389030228364
Метод случайного леса	2.5959501252029167	374.8823137442501
Градиентный бустинг (sklearn)	2.691237353851188	389.8646208989257
Градиентный бустинг (xgboost)	2.836931371656709	391.90658604954945
autogluon	2.5477529120587277	373.7778693276293
Градиентный бустинг (sklearn) с лучшими гиперпараметрами	2.5832219351980394	375.38057907539417





Разработка нейронной сети

В качестве модели выбран многослойный персептрон.

Поиск лучших гиперпараметры для модели:

- Количество слоёв,
- функции активации,
- параметр dropout,
- метод оптимизации функции,
- размер батча.

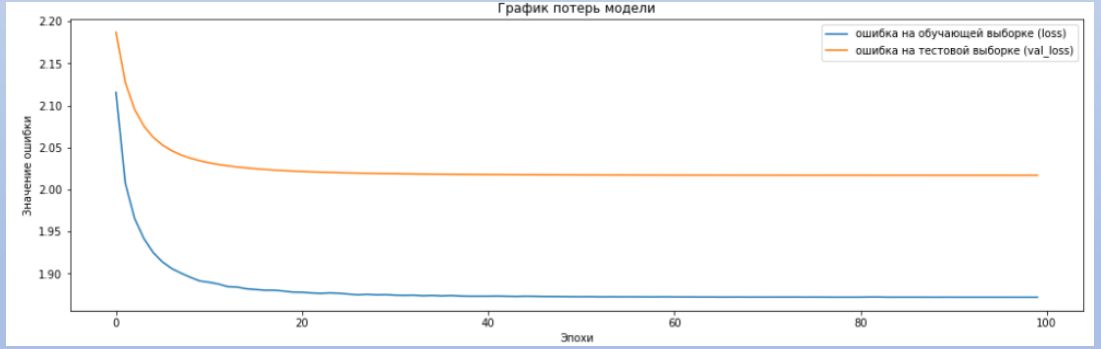
По итогу имеем очень большую ошибку.

```
# оценим модель. Результат mae для тестовой выборки
scores = model.evaluate(X_test, y_test)
scores
```

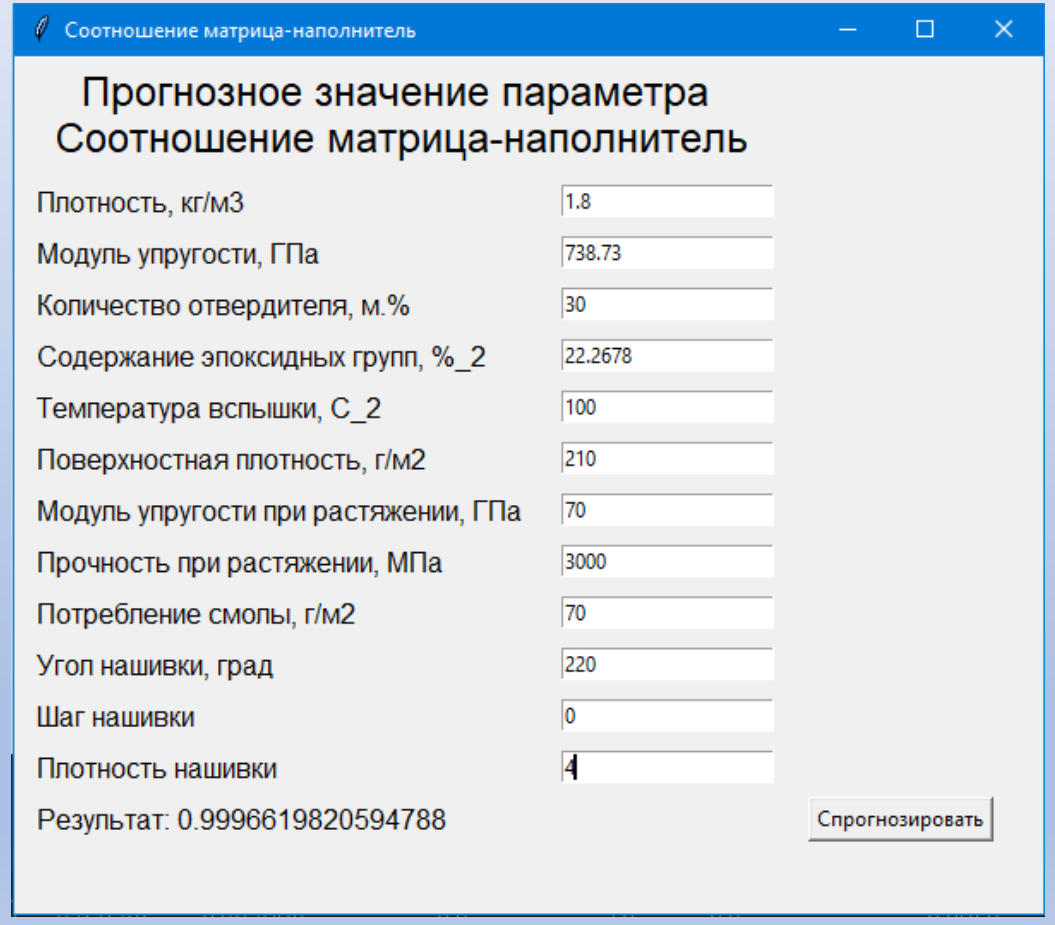
```
9/9 [==============================] - 0s 3ms/step - loss: 1.9986

1.9986492395401
```

```
# Среднее абсолютное отклонение
df['Соотношение матрица-наполнитель'].mad()
```

```
0.7159293950955805
```



Разработка приложения



- Для разработки приложения применялась библиотека Tkinter.
- Была использована нейронная сеть построенная на библиотеке Tensorflow.
- Был использован нормализатор (scaler) из предыдущего пункта. Т.к. в нейросеть должны идти нормализованные данные

Спасибо за внимание!

- При прогнозировании как модуля упругости при растяжении, так и прочности при растяжении был использован довольно широкий ряд моделей, но даже для лучших моделей ошибки достаточно значимы. Предполагаю, что для построения хорошей модели не хватает входных данных. Или же надо выбирать какие-то иные подходы.
- Прогнозирование соотношения матрица-наполнитель тоже оказалось трудновыполнимым. Ошибка тоже достаточно велика.