МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по курсу

«Data Science»

Слушатель Марченко П. А.

Москва, 2022

Содержание

Введение.

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. Благодаря своим характеристикам композиционные материалы нашли широкое применение в различных областях промышленности, они применяются в строительстве, автомобилестроении, авиастроении и других отраслях промышленности. У композитов есть недостаток: даже если мы знаем характеристики исходных компонентов, определить характеристики композита, состоящего из этих компонентов, достаточно проблематично. Для решения этой проблемы есть два пути: физические испытания образцов материалов, или прогнозирование характеристик. Суть прогнозирования заключается в симуляции представительного элемента объема композита, на основе данных о характеристиках входящих компонентов (связующего и армирующего компонента).

В предложенном Вашему вниманию проекте, я построил модели для прогноза Модуля упругости при растяжении и Прочности при растяжении, спроектировал нейронную сеть для прогноза Соотношения матрица-наполнитель и разработал web-приложение, в котором с помощью обученных нейронные сетей можно прогнозировать конечные свойства композиционных материалов.

1. Аналитическая часть

1.1. Постановка задачи

У композиционных материалов есть недостаток: даже если мы знаем характеристики исходных компонентов, определить характеристики композита, состоящего из этих компонентов, достаточно проблематично. Для решения этой проблемы есть два пути: физические испытания образцов материалов, или прогнозирование характеристик. Для проведения физических испытаний образцов тратится много денежных средств и времени. Можно попробовать решить задачу определения конечных свойств композитов путем прогнозирования этих свойств с помощью обученных моделей или нейронных сетей. Суть прогнозирования заключается в симуляции представительного элемента объема композита, на основе данных о характеристиках входящих компонентов (связующего и армирующего компонента).

Созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

Для выполнения проекта были даны два файла в формате excel-таблиц. Первый файл содержит в себе физические характеристики базальтопластика X_bp (табл. 1), второй файл содержит геометрические характеристики нашивки углепластика X_nup (табл. 2).

Таблица 1. Файл X_bp

	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
0	1,85714	2030	738,73	30	22,26	100	210	70	3000	220
1	1,85714	2030	738,73	50	23,75	284,615	210	70	3000	220
2	1,85714	2030	738,73	49,9	33	284,615	210	70	3000	220
3	1,85714	2030	738,73	129	21,25	300	210	70	3000	220
	…	…	…	…	…	…	…	…	…	…

Таблица 2. Файл X_nup

	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0	0	4	57
1	0	4	60
2	0	4	70
3	0	5	47
…	…	…	…

В файле X_bp 1023 элемента, в файле X_nup 1040 элементов. Таблицы были объединены в одну внутренним соединением по индексу элемента. После объединения таблиц, количество элементов в рассматриваемом датасете составило 1023, а количество параметров равно 13. Далее было проверено наличие пропусков и дубликатов в рассматриваемом датасете - отсутствуют.

Рисунок 1. Проверка датасета на дубликаты

```
1 df.duplicated().sum()
✓ 0.1s                                                    MagicPython

0
```

Рисунок 2. Проверка датасета на пропуски

```
1 df.isnull().sum()
✓ 0.9s

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                             0
Шаг нашивки                              0
Плотность нашивки                        0
dtype: int64
```

Все данные в датасете принадлежат одному типу данных – число с плавающей точкой. В переменной угол нашивки всего 2 значения 0.0 и 90.0 градусов, столбец с данной переменной был преобразован к значениям 0 = 0.0 град и 1 = 90.0 град, а тип данных изменен на int32.

Рисунок 3. Тип данных в датасете

```
1  df.info()
✓ 0.1s
<class 'pandas.core.frame.DataFrame'>
Float64Index: 1023 entries, 0.0 to 1022.0
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                          1023 non-null   int32
 11  Шаг нашивки                           1023 non-null   float64
 12  Плотность нашивки                     1023 non-null   float64
dtypes: float64(12), int32(1)
memory usage: 107.9 KB
```

1.2. Описание используемых методов

- **Метод k-ближайших соседей**

Метод k-ближайших соседей (англ. *k-nearest neighbors algorithm*, k-NN) —метрический алгоритм для автоматической классификации объектов или регрессии.

В случае использования метода для классификации объект присваивается тому классу, который является наиболее распространённым среди соседей данного элемента, классы которых уже известны. В случае использования метода для регрессии, объекту присваивается среднее значение по ближайшим к нему объектам, значения которых уже известны.

- **Стохастический градиентный спуск**

Стохастический градиентный спуск (англ. *stochastic gradient descent*) –– оптимизационный алгоритм, отличающийся от обычного градиентного спуска тем, что градиент оптимизируемой функции считается на каждом шаге не как сумма градиентов от каждого элемента выборки, а как градиент от одного, случайно выбранного элемента.

- **Линейная регрессия**

Линейная регрессия (англ. Linear regression) — используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной y от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) x с линейной функцией зависимости.

- **Случайный лес**

Random forest (с англ. — «случайный лес») — алгоритм машинного обучения, заключающийся в использовании комитета (ансамбля) решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга, и метод случайных подпространств. Алгоритм

применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.

- **Многослойный перцептрон**

  Многослойный персептрон — это класс искусственных нейронных сетей прямого распространения, состоящих как минимум из трех слоёв: входного, скрытого и выходного. За исключением входных, все нейроны использует нелинейную функцию активации. При обучении MLP используется обучение с учителем и алгоритм обратного распространения ошибки.

1.3. Разведочный анализ данных

Разведочный анализ данных (англ. *Exploratory data analysis, EDA*) — анализ основных свойств данных, нахождение в них общих закономерностей, распределений и аномалий, построение начальных моделей, зачастую с использованием инструментов визуализации.

В проекте были использованы следующие методы разведочного анализа данных:

- Визуальный анализ гистограмм
- Визуальный анализ диаграмм размаха («ящик с усами»)
- Проверка нормальности распределения по критерию Пирсона
- Анализ попарных графиков рассеяния переменных
- Корреляционный анализ с целью поиска коэффициентов

2. Практическая часть

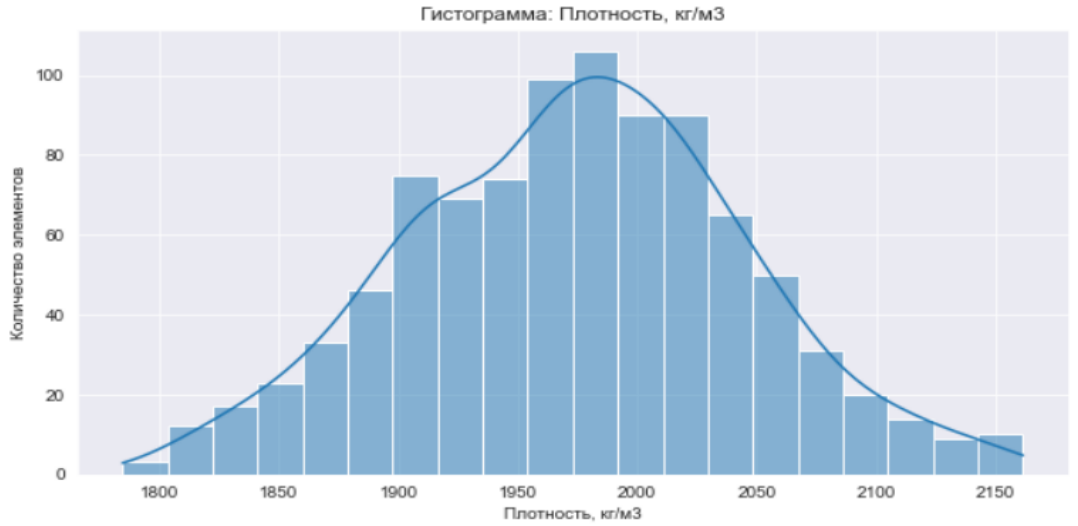
2.1. Предобработка данных

Для каждого параметра были построены гистограммы и диаграммы размаха, а также указаны минимальное, максимальное, среднее и медианное значения.

Рисунок 4. Гистограмма: Соотношение матрица-наполнитель

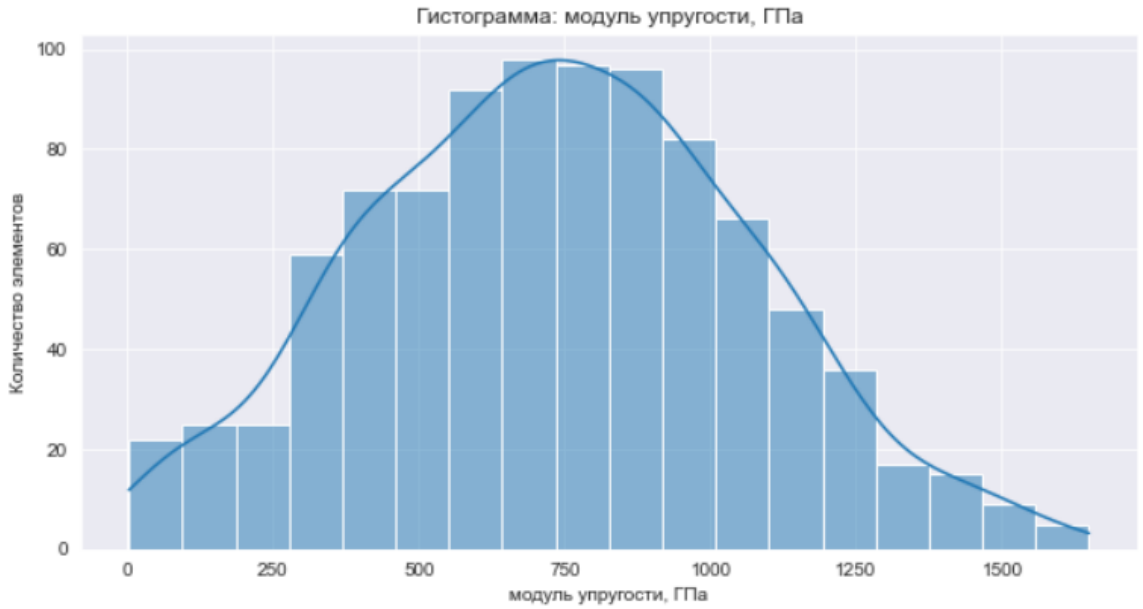

```
Минимальное значение: 0.547
Максимальное значение: 5.314
Среднее значение: 2.926
Медианное значение: 2.905
```

Рисунок 5. Гистограмма: Плотность, кг/м3




```
Минимальное значение: 1784.482
Максимальное значение: 2161.565
Среднее значение: 1974.04
Медианное значение: 1977.258
```

Рисунок 6. Гистограмма: Модуль упругости, Гпа




```
Минимальное значение: 2.437
Максимальное значение: 1649.416
Среднее значение: 738.248
Медианное значение: 738.737
```

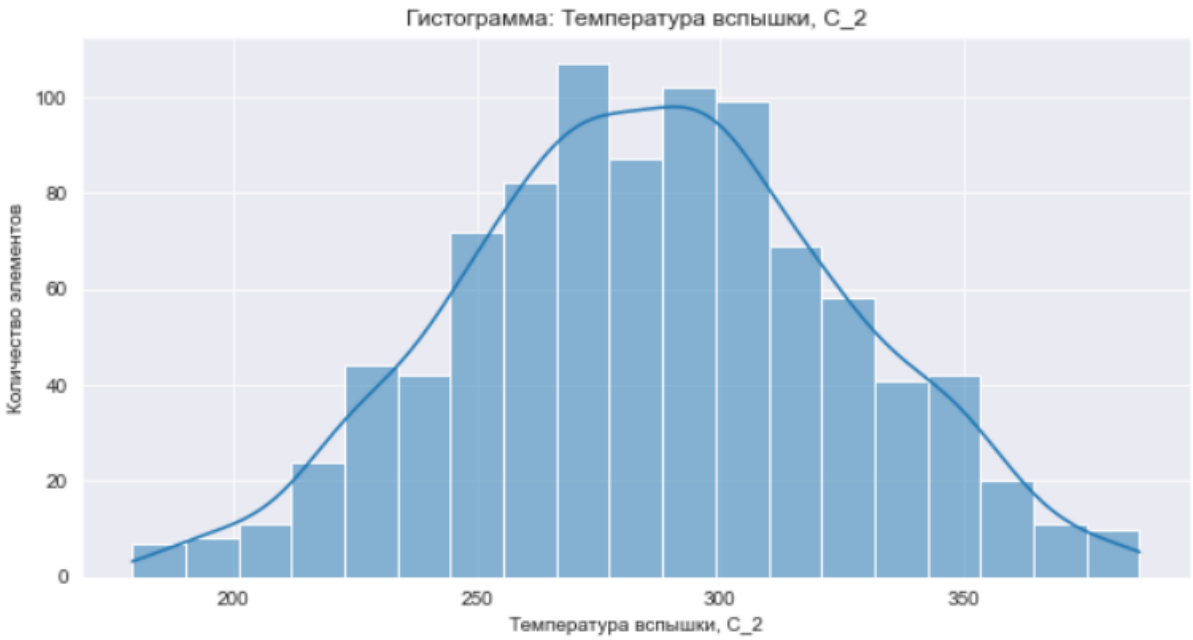
Рисунок 7. Гистограмма: Количество отвердителя, м.%


```
Минимальное значение: 38.669
Максимальное значение: 181.828
Среднее значение: 110.916
Медианное значение: 111.113
```

Рисунок 8. Гистограмма: Содержание эпоксидных групп, %_2


```
Минимальное значение: 15.696
Максимальное значение: 28.955
Среднее значение: 22.209
Медианное значение: 22.185
```

Рисунок 9. Гистограмма: Температура вспышки С_2




```
Минимальное значение: 179.374
Максимальное значение: 386.068
Среднее значение: 286.04
Медианное значение: 286.024
```

Рисунок 10. Гистограмма: Поверхностная плотность, г/м2


```
Минимальное значение: 0.604
Максимальное значение: 1291.34
Среднее значение: 482.994
Медианное значение: 457.732
```

Рисунок 11. Гистограмма: Модуль упругости при растяжении, ГПа


```
Минимальное значение: 65.553
Максимальное значение: 81.417
Среднее значение: 73.305
Медианное значение: 73.259
```

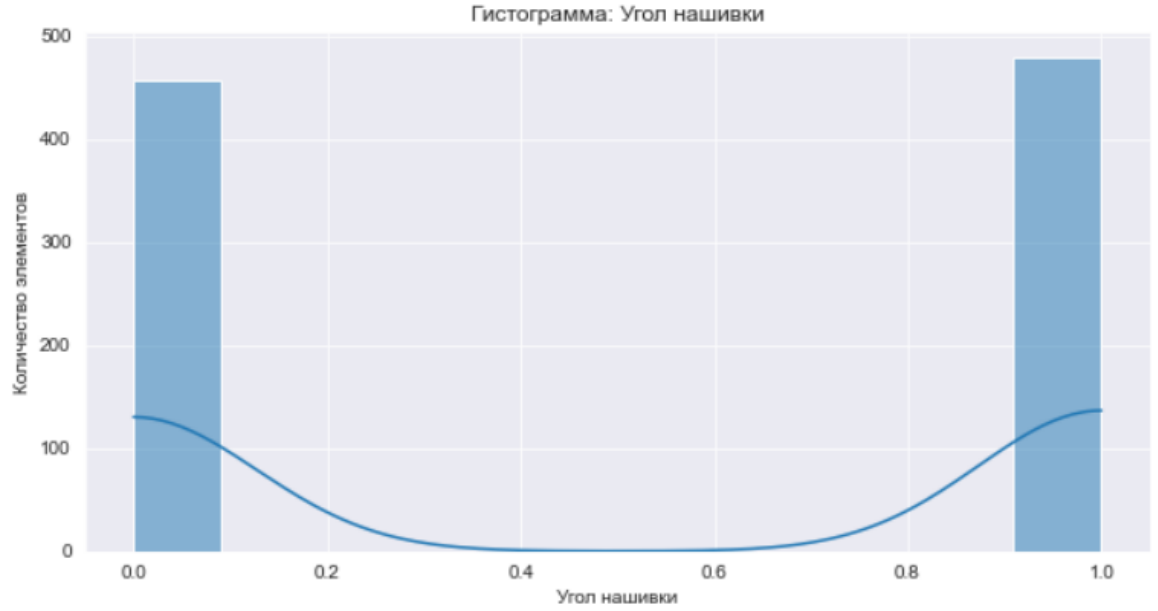
Рисунок 12. Гистограмма: Прочность при растяжении, МПа


```
Минимальное значение: 1250.393
Максимальное значение: 3705.673
Среднее значение: 2467.489
Медианное значение: 2457.96
```

Рисунок 13. Гистограмма: Потребление смолы, г/м2

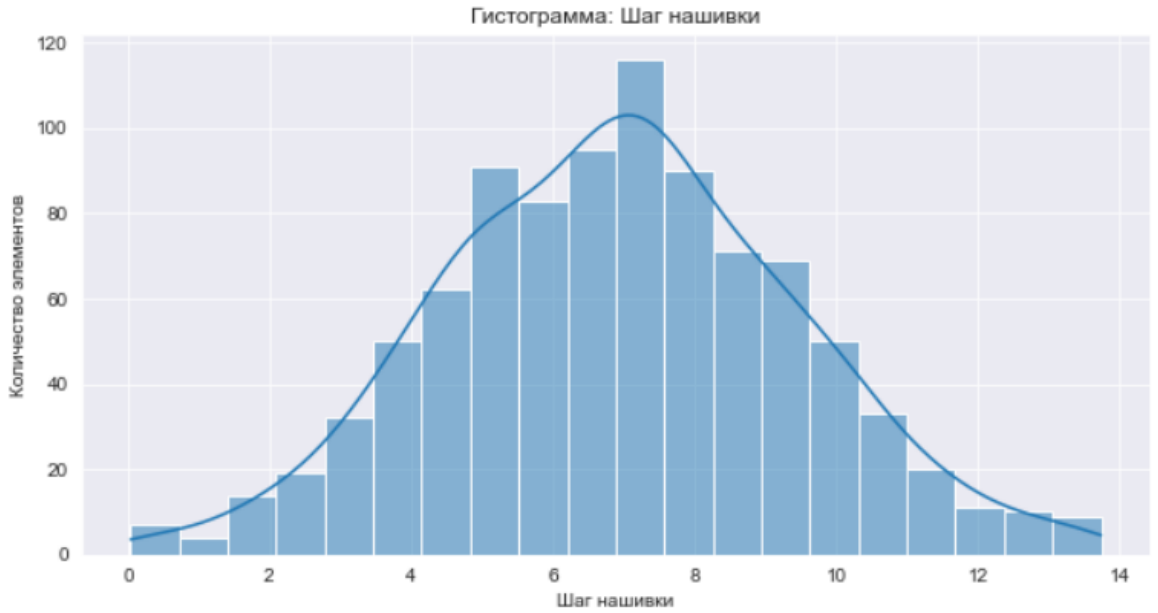

```
Минимальное значение: 63.686
Максимальное значение: 359.052
Среднее значение: 217.613
Медианное значение: 218.389
```

Рисунок 14. Гистограмма: Угол нашивки



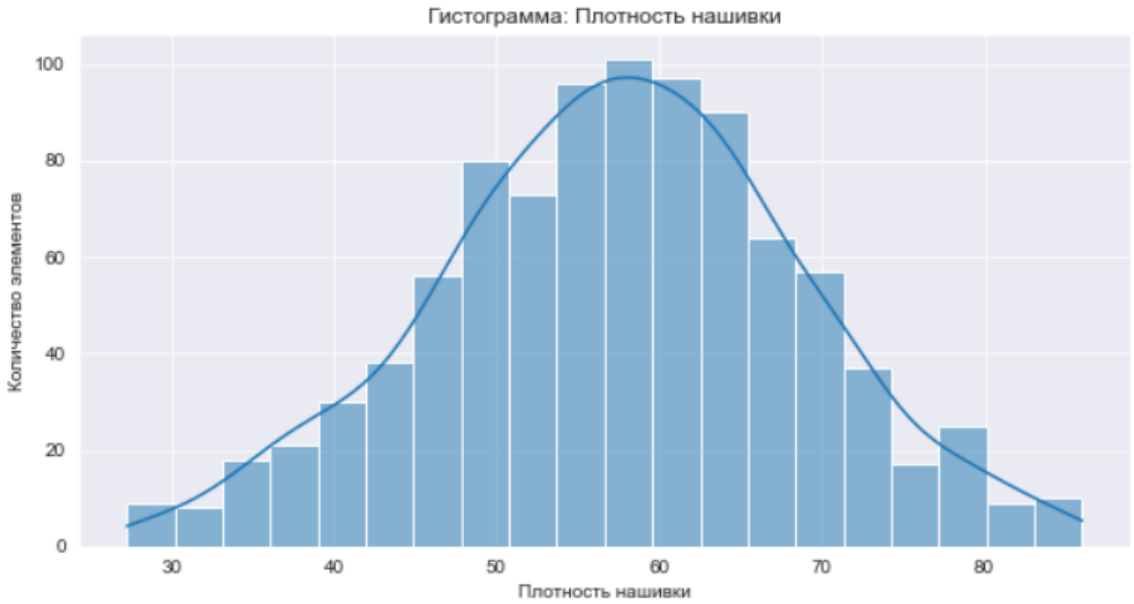

```
Минимальное значение: 0.0
Максимальное значение: 1.0
Среднее значение: 0.512
Медианное значение: 1.0
```

Рисунок 15. Гистограмма: Шаг нашивки




```
Минимальное значение: 0.038
Максимальное значение: 13.732
Среднее значение: 6.916
Медианное значение: 6.943
```

Рисунок 16. Гистограмма: Плотность нашивки




```
Минимальное значение: 27.273
Максимальное значение: 86.012
Среднее значение: 57.452
Медианное значение: 57.584
```

На гистограммах можно увидеть, что распределения параметров являются нормальными или близкими к нормальному.

Распределения переменных были проверены на нормальность по критерию Пирсона.

Рисунок 17. Проверка распределений на нормальность

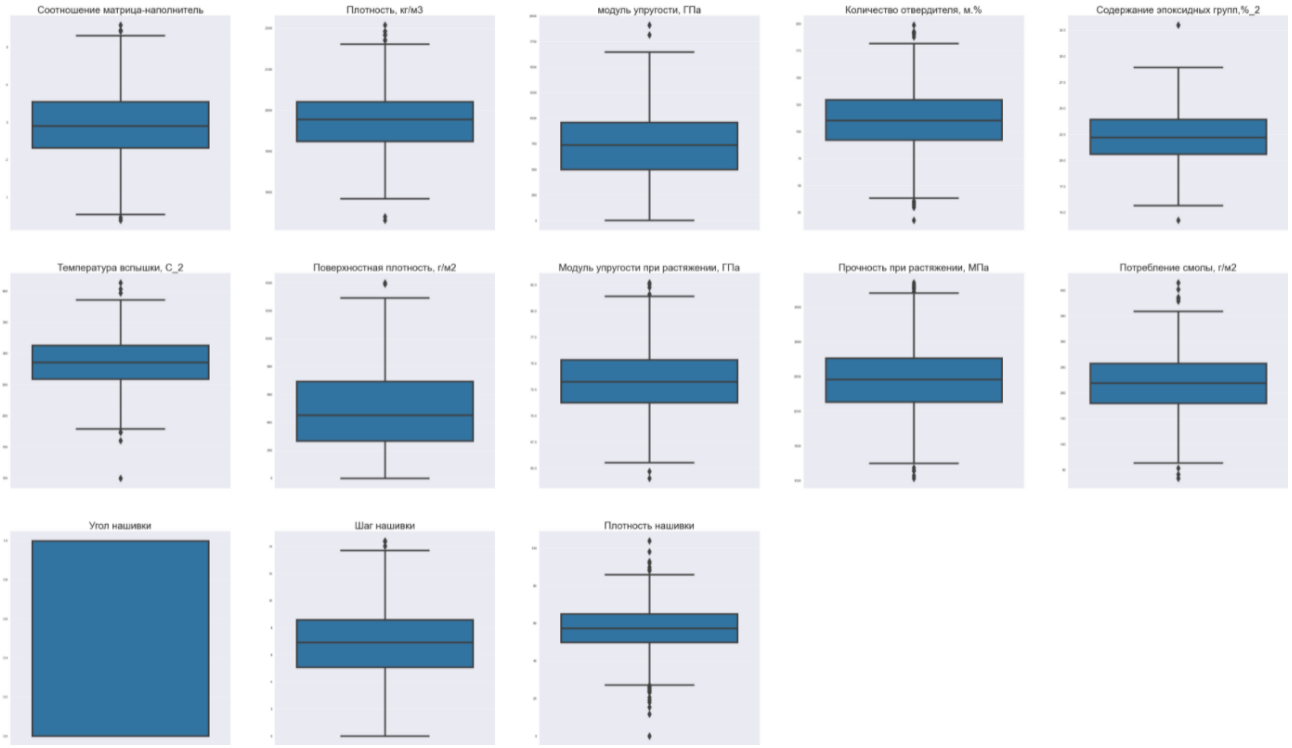
```python
import scipy
accepted_list = []
rejected_list = []

for col in df.columns:
    alpha = 0.05
    stat, p = scipy.stats.normaltest(df[col]) # Критерий согласия Пирсона
    #print('Statistics=%.3f, p-value=%.3f' % (stat, p))
    if p > alpha:
        accepted_list.append(col)
    else:
        rejected_list.append(col)
print(f'Наименование колонок с данными С нормальным распределением: {accepted_list}')
print(f'Наименование колонок с данными БЕЗ нормального распределения : {rejected_list}')
```

✓ 0.1s MagicPython

```
Наименование колонок с данными С нормальным распределением: ['Соотношение матрица-наполнитель', 'Плотность, кг/м3', 'модуль упругости, ГПа', 'Количество отвердителя, м.%', 'Содержание эпоксидных групп,%_2', 'Температура вспышки, С_2', 'Модуль упругости при растяжении, ГПа', 'Прочность при растяжении, МПа', 'Потребление смолы, г/м2', 'Шаг нашивки']
Наименование колонок с данными БЕЗ нормального распределения : ['Поверхностная плотность, г/м2', 'Угол нашивки', 'Плотность нашивки']
```

Рисунок 18. Диаграммы размаха переменных до удаления выбросов



На диаграммах размаха видно наличие выбросов. Выбросами являются точки, превышающие 1,5 межквартильного расстояния. *Межквартильное расстояние — это разница между 1-м и 3-м квартилями, т.е. между 25-м и 75-м процентилями.*

Данные, выходящие за пределы 1,5 межквартильных расстояния, были заменены на пустые значения и посчитаны. Их кол-во оказалось не большим и было принято решение удалить строки, содержащие выбросы.

Рисунок 19. Диаграммы размаха переменных после удаления выбросов

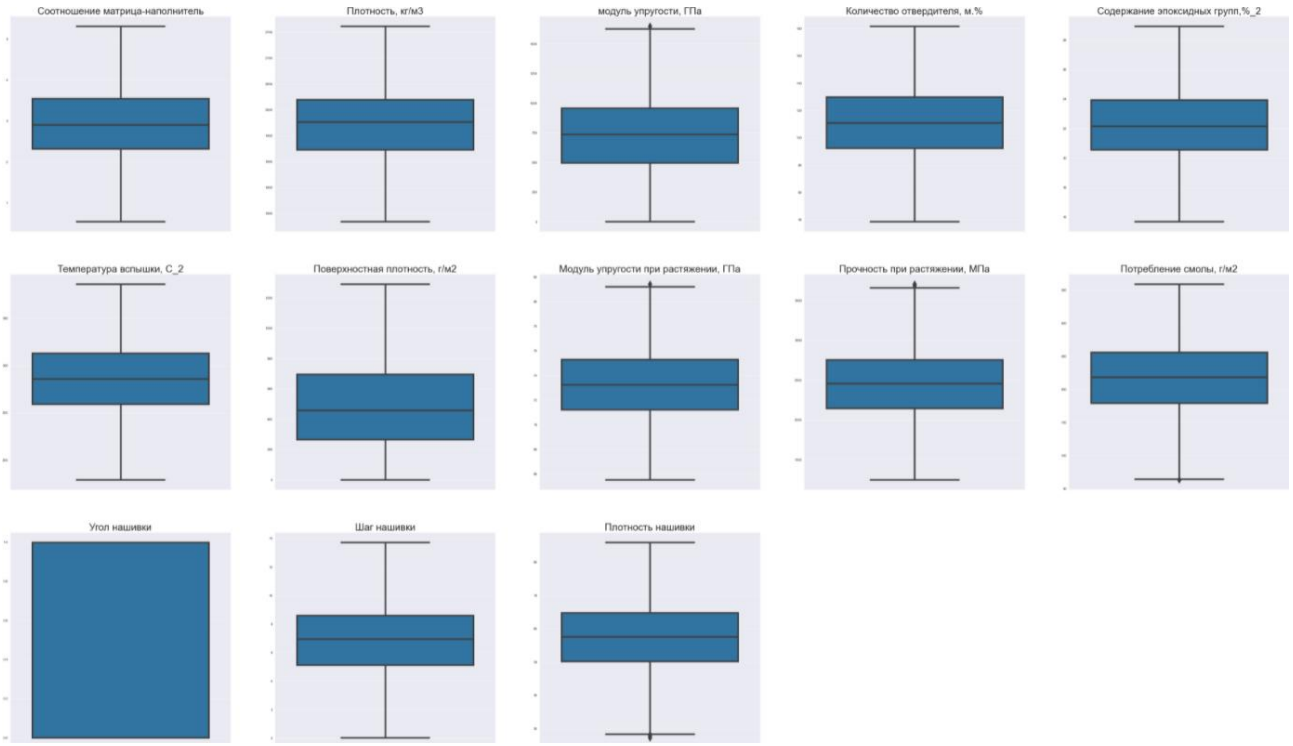


Рисунок 20. Описательная статистика датасета

```
1  df.describe()
✓ 0.9s
```

	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
count	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000
mean	2.930366	1975.734888	739.923233	110.570769	22.244390	285.882151	482.731833	73.328571	2466.922843	218.423144
std	0.913222	73.729231	330.231581	28.295911	2.406301	40.943260	281.314690	3.118983	485.628006	59.735931
min	0.389403	1731.764635	2.436909	17.740275	14.254985	100.000000	0.603740	64.054061	1036.856605	33.803026
25%	2.317887	1924.155467	500.047452	92.443497	20.608034	259.066528	266.816645	71.245018	2135.850448	179.627520
50%	2.906878	1977.621657	739.664328	110.564840	22.230744	285.896812	451.864365	73.268805	2459.524526	219.198882
75%	3.552660	2021.374375	961.812526	129.730366	23.961934	313.002106	693.225017	75.356612	2767.193119	257.481724
max	5.591742	2207.773481	1911.536477	198.953207	33.000000	413.273418	1399.542362	82.682051	3848.436732	414.590628

После очистки датасета от выбросов была проведена нормализация данных. Данные были приведены к общей шкале, в которой минимальное значение параметра принимало 0, а максимальное – 1.

Нормализация данных была выполнена функцией MinMaxScaler из библиотеки sklearn.

Данные в нормализованном виде понадобятся для построения графиков попарного рассеяния параметров, построения матрицы корреляции и при построении нейронной сети.

Рисунок 21. Нормализация данных и описательная статистика

```
1  from sklearn.preprocessing import MinMaxScaler
✓ 0.3s
```

```
1  min_max_scaler = MinMaxScaler()
2  df_norm = pd.DataFrame(min_max_scaler.fit_transform(df), columns = df.columns, index=df.index)
✓ 0.4s
```

```
1  df_norm.describe()
✓ 0.9s
```

	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
count	936.000000	936.000000	936.000000	936.000000	936.000000	936.000000	936.000000	936.000000	936.000000	936.000000
mean	0.498933	0.502695	0.446764	0.504664	0.491216	0.516059	0.373733	0.488647	0.495706	0.521141
std	0.187489	0.187779	0.199583	0.188865	0.180620	0.190624	0.217078	0.191466	0.188915	0.195781
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.372274	0.368517	0.301243	0.376190	0.367716	0.386128	0.205619	0.359024	0.365149	0.392067
50%	0.494538	0.511229	0.447061	0.506040	0.489382	0.515980	0.354161	0.485754	0.491825	0.523766
75%	0.629204	0.624999	0.580446	0.637978	0.623410	0.646450	0.538683	0.615077	0.612874	0.652447
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

График попарной зависимости переменных (рис. 22) был построен для всех параметров кроме угла нашивки. На графике зеленым выделены точки для угла нашивки равного 0 градусов, а оранжевым для угла нашивки равного 90 градусам. Согласно графику ни для одной пары переменных не прослеживается какая-либо зависимость. Это подтверждается матрицей корреляции (рис. 23), в которой максимальное значение корреляции равно 0.093 между параметрами Плотность нашивки и угол нашивки. Корреляция близка к 0, что говорит о том, что переменные не связаны между собой.

Рисунок 22. График попарной зависимости переменных

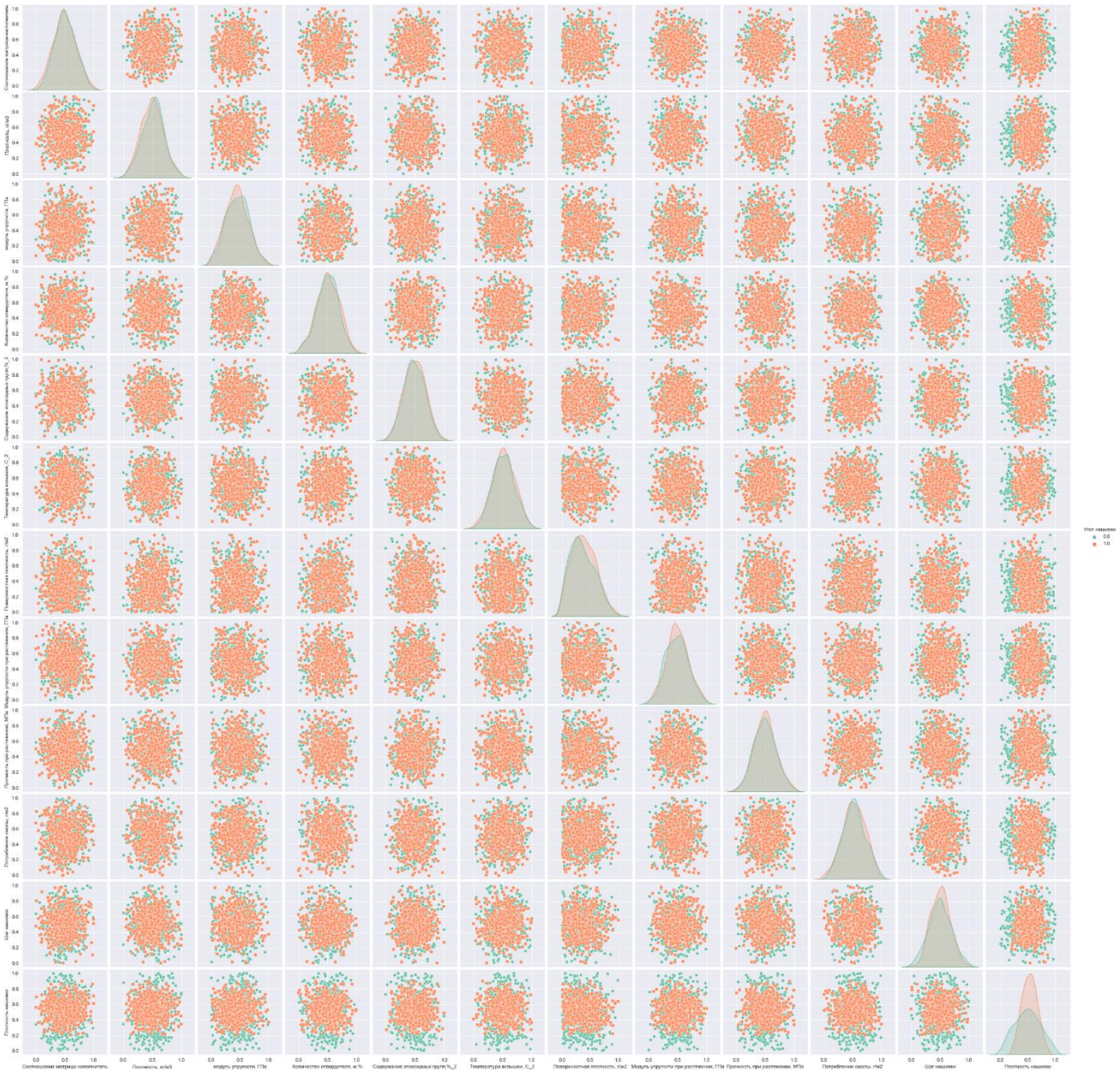
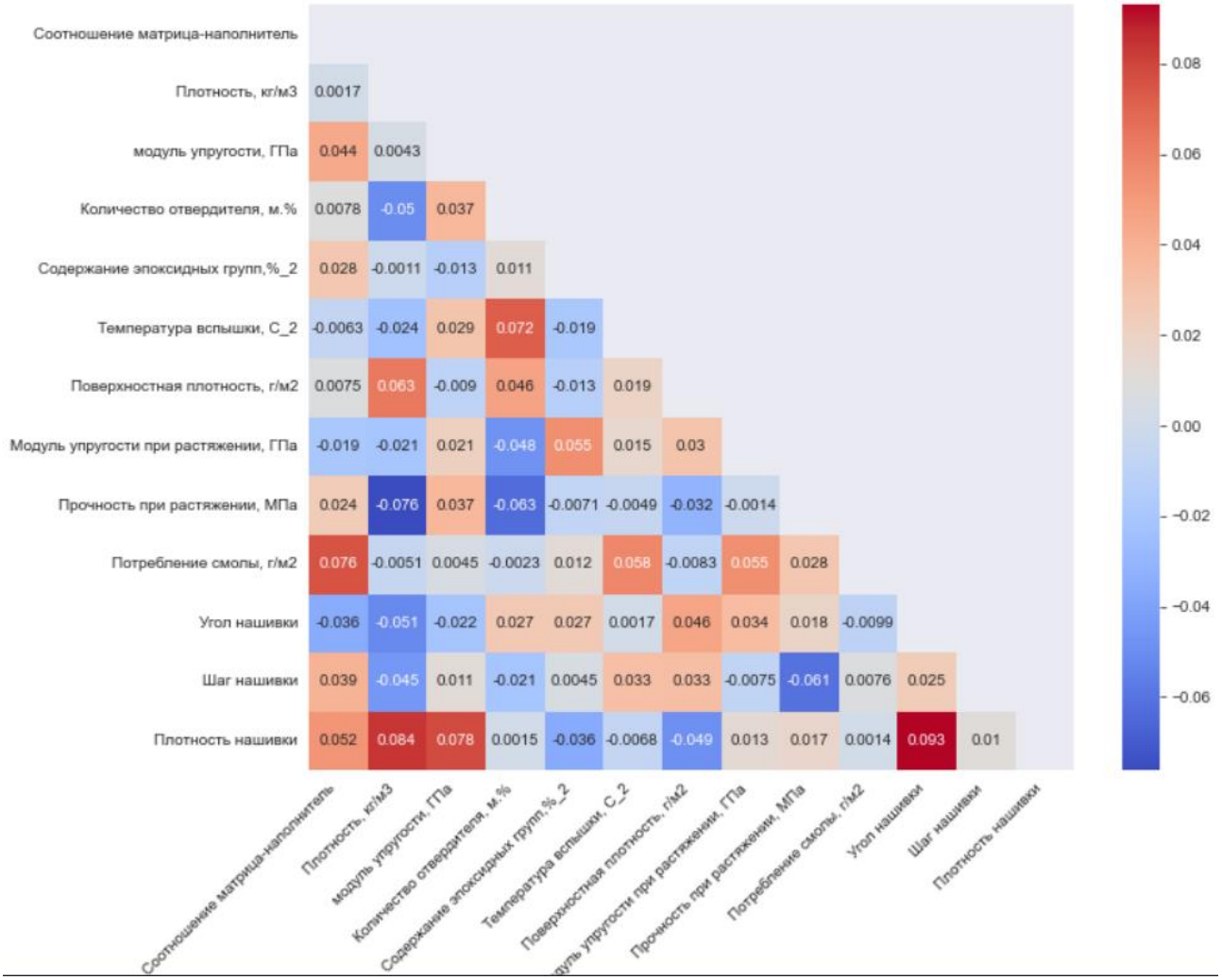


Рисунок 23. Матрица корреляции переменных



2.2. Разработка и обучение модели

Согласно поставленной задаче, данные были разделены на обучающую и тестовую выборки в соотношении 70/30. Для каждой модели был создан словарь с гиперпараметрами. С помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10, были найдены лучшие гиперпараметры для каждой модели. Каждая модель с лучшими гиперпараметрами давала свои прогнозы на тестовой выборке. В качестве параметров оценки моделей были выбраны: Средняя абсолютная ошибка (MAE) и коэффициент детерминации (R^2).

В ходе выполнения проекта в качестве моделей для прогноза модуля упругости при растяжении и прочности при растяжении были использованы:

- Метод k-ближайших соседей
- Стохастический градиентный спуск
- Линейная регрессия
- Случайный лес
- Многослойный перцептрон из библиотеки sklearn

2.3. Тестирование модели

Первым шагом объявляется «пустая» модель, т.е. модель со всеми стандартными значениями параметров. Далее создается сетка гиперпараметров, в данном случае словарь с названием гиперпараметра и его возможными значениями. Следующим шагом объявленная модель и сетка гиперпараметров подаются в поиск по сетке и объявляется кол-во блоков равное 10, т.е. тренировочные данные делятся на 10 равных частей. Далее, для каждого возможного сочетания значений гиперпараметров, метод считает ошибку и в конце выбирает сочетание, при котором ошибка минимальна. С помощью лучшей модели прогнозируются значения на тестовой выборке и результаты Средней абсолютной ошибки и коэффициента детерминации заносятся в сводную таблицу (табл. 3).

Таблица 3. Сводная таблица построенных моделей

	Model	MAE	R2 score
Модуль упругости при растяжении	**KNeighborsRegressor_upr**	**2.513734**	**0.002**
Модуль упругости при растяжении	MLPRegressor_upr	2.524963	-0.001
Модуль упругости при растяжении	RandomForestRegressor_upr	2.537739	-0.009
Модуль упругости при растяжении	LinearRegression_upr	2.546419	-0.021
Модуль упругости при растяжении	SGDRegressor_upr	3.196386	-0.664
Прочность при растяжении	**RandomForestRegressor_pr**	**367.043280**	**-0.005**
Прочность при растяжении	MLPRegressor_pr	368.202754	-0.011
Прочность при растяжении	KNeighborsRegressor_pr	368.730938	-0.022
Прочность при растяжении	LinearRegression_pr	370.542618	-0.021
Прочность при растяжении	SGDRegressor_pr	393.361501	-0.110

Для прогноза модуля упругости при растяжении лучше всего показала результаты модель метода «К ближайших соседей» со Средней абсолютной ошибкой = 2.51 Гпа, а для прогноза прочности при растяжении лучше всего сработала модель метода «Случайный лес» со Средней абсолютной ошибкой = 367.04 Мпа.

При этом, во всех использованных моделях коэффициент детерминации очень близок к 0, это говорит о том, что результат использования моделей не точнее использования для прогноза среднего значения прогнозируемого параметра.

2.4. Нейронная сеть, которая рекомендует соотношение матрица-наполнитель.

Нейронная сеть была написана с помощью библиотеки Keras на языке программирования Python.

Характеристики нейронной сети для прогноза соотношения матрица-наполнитель:

- Последовательная модель (sequential) нейронной сети
- Модель состоит из 5 скрытых Dense слоев, количество нейронов в которых равно 128, 128, 64, 32, 16 и выходного слоя с одним нейроном. Функция активации слоев – selu (Масштабная экспоненциальная линейная единица).
- Используются слои Batch-Normalization (Пакетная нормализация (англ. batch-normalization) — метод, который позволяет повысить производительность и стабилизировать работу искусственных нейронных сетей. Суть данного метода заключается в том, что некоторым слоям нейронной сети на вход подаются данные, предварительно обработанные и имеющие нулевое математическое ожидание и единичную дисперсию.)

- В качестве оптимизатора нейронной сети используется SGD (стохастический градиентный спуск) с импульсом ускорения = 0.5.
- В нейронной сети используется функция ранней остановки обучения, если в течение 10ти эпох не наблюдается улучшения потерь на валидационной выборке.
- Размер батча при обучении – 64
- Кол-во эпох обучения – 100
- Размер валидационной выборки – 20% от размера обучающей выборки

Рисунок 24. Структура нейронной сети для прогноза соотношения матрица-наполнитель

```
model_mn = Sequential(X_train_mn_norm)

model_mn.add(Dense(128))
model_mn.add(BatchNormalization())
model_mn.add(LeakyReLU())
model_mn.add(Dense(128, activation='selu'))
model_mn.add(BatchNormalization())
model_mn.add(Dense(64, activation='selu'))
model_mn.add(BatchNormalization())
model_mn.add(Dense(32, activation='selu'))
model_mn.add(BatchNormalization())
model_mn.add(LeakyReLU())
model_mn.add(Dense(16, activation='selu'))
model_mn.add(BatchNormalization())
model_mn.add(Dense(1))
model_mn.add(Activation('selu'))
```

```
early_mn = EarlyStopping(monitor='val_loss', min_delta=0, patience=10, verbose=1, mode='auto')
```

```
model_mn.compile(
    optimizer=tf.optimizers.SGD(learning_rate=0.02, momentum=0.5),
    loss='mean_absolute_error')
```

```
%%time
history_mn = model_mn.fit(
    X_train_mn,
    y_train_mn,
    batch_size = 64,
    epochs=100,
    verbose=1,
    validation_split = 0.2,
    callbacks = [early_mn]
    )
```

В процессе обучения нейронной сети сработала ранняя остановка обучения на 36-ой эпохе обучения, т.к. в течение 10-ти эпох не наблюдалось улучшения результата потерь на валидационной выборке.

Рисунок 25. График потерь нейронной сети

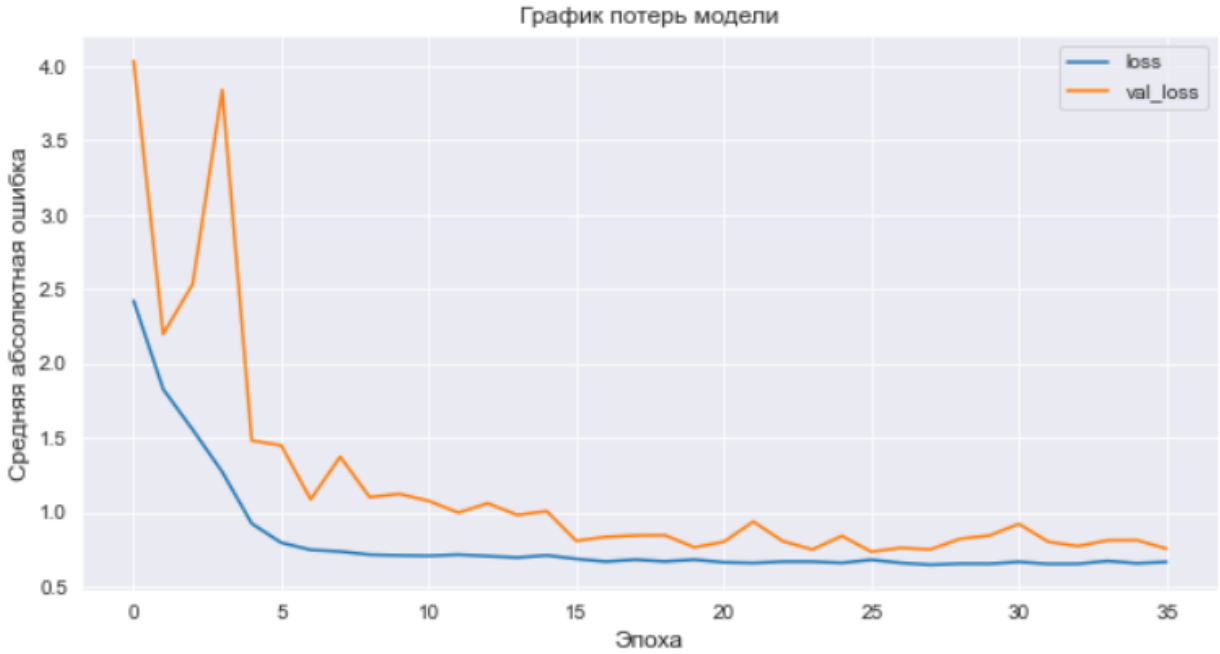


Рисунок 26. График прогнозных значений нейронной сети

На графике прогнозных значений нейронной сети можно увидеть, что построенная модель нейронной сети не так хорошо прогнозирует значения на тестовой выборке данных, хотя местами правильно предсказывает направление тренда и величину прогнозного значения.

Рисунок 27. График рассеяния тестовых и прогнозных значений

На графике рассеяния тестовых и прогнозных значений (рис. 27) можно увидеть, что нейронная сеть показывает не самый лучший результат прогноза. В идеальном варианте точки тестовых и прогнозных значений должны выстроиться в прямую линию под углом 45°.

Результат средней абсолютной ошибки нейронной сети на тестовой выборке равен 0.78 (рис. 28).

Результат средней абсолютной ошибки при прогнозе соотношения матрица-наполнитель средним значением этой переменной равен 0.74 (рис. 29).

Рисунок 28. Средняя абсолютная ошибка нейронной сети

```
1  print(f'Model MAE: {model_mn.evaluate(X_test_mn, y_test_mn)}')
✓ 0.1s

9/9 [==============================] - 0s 1ms/step - loss: 0.7805
Model MAE: 0.780497670173645
```

Рисунок 29. Средняя абсолютная ошибка среднего значения

```
  1  print(f'MAE среднего значения: {np.mean(np.abs(y_test_mn-np.mean(y_test_mn)))}')
✓ 0.1s

MAE среднего значения: Соотношение матрица-наполнитель    0.741688
```

2.5. Web-приложение на Flask

Web-приложение написано на языке программирования Python с использованием библиотеки Flask, для написания шаблонов страниц был использован язык разметки HTML. Шаблоны страниц хранятся в папке templates.

Для разработки web-приложения дополнительно были разработаны и написаны две нейронных сети для прогноза модуля упругости при растяжении и прочности при растяжении.

Характеристика нейронной сети для прогноза модуля упругости при растяжении:

- Последовательная модель (sequential) нейронной сети
- Модель состоит из 5 скрытых Dense слоев, количество нейронов в которых равно 128, 64, 64, 32, 32 и выходного слоя с одним нейроном. Функция активации слоев – leaky-relu, активация выходного слоя – elu.
- В качестве оптимизатора нейронной сети используется SGD (стохастический градиентный спуск) с импульсом ускорения = 0.9.
- Размер батча при обучении – 64
- Кол-во эпох обучения – 40
- Размер валидационной выборки – 20% от размера обучающей выборки
- MAE нейронной сети после обучения – 2.62

Характеристика нейронной сети для прогноза прочности при растяжении:

- Последовательная модель (sequential) нейронной сети
- Модель состоит из 4 скрытых Dense слоев, количество нейронов в которых равно 128, 64, 64, 32 и выходного слоя с одним нейроном.

Функция активации слоев – leaky-relu, активация выходного слоя – selu.

- В нейронной сети используется функция ранней остановки обучения, если в течение 20ти эпох не наблюдается улучшения потерь на валидационной выборке.
- В качестве оптимизатора нейронной сети используется SGD (стохастический градиентный спуск).
- Размер батча при обучении – 32
- Кол-во эпох обучения – 400
- Размер валидационной выборки – 20% от размера обучающей выборки
- MAE нейронной сети после обучения – 381

В разработанном веб-приложении можно спрогнозировать с помощью обученных нейронных сетей конечные свойства композиционных материалов, такие как [Матрица-наполнитель, Прочность при растяжении, Модуль упругости при растяжении], на основе введенных пользователем значений.

Пользователь изначально попадает на главную страницу (рис. 30), на которой находятся описание приложения и меню выбора прогнозируемой переменной. При нажатии на кнопку прогноза необходимой переменной пользователь попадает на новую страницу. На странице прогноза выбранной переменной (рис. 31) пользователю предлагается ввести значения параметров необходимых для прогноза, после ввода всех параметров и нажатии кнопки рассчитать пользователю в виде сообщения отображается прогнозное значение переменной, полученное с помощью прогноза нейронной сети, структура и веса которой загружаются из папки models. Также на странице прогноза выбранной переменной находятся кнопка «Сбросить»- для сброса введенных значений и кнопка возврата в главное меню.

Рисунок 30. Главная страница web-приложения

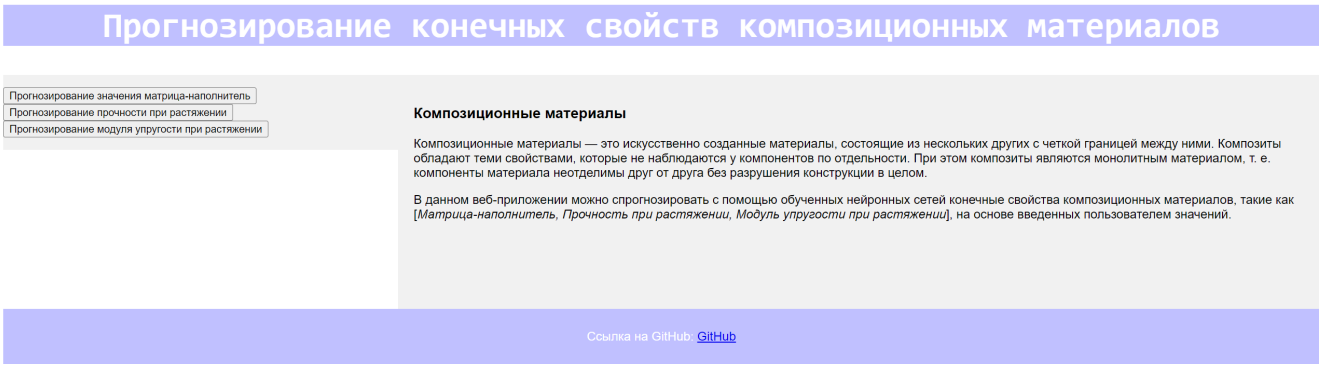
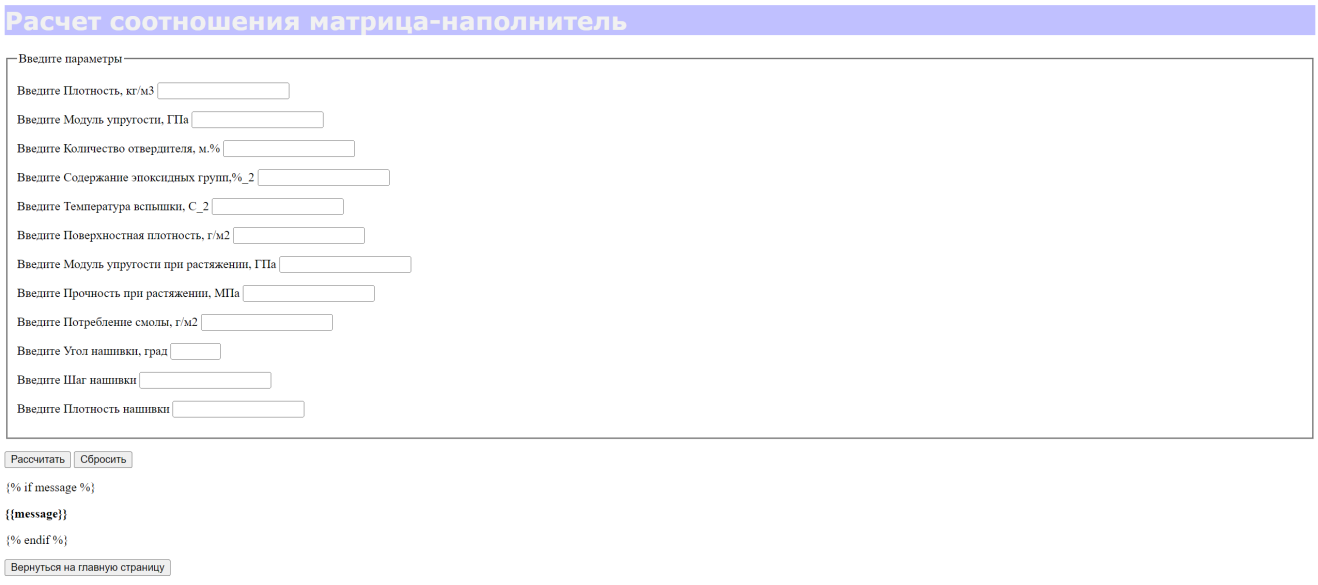


Рисунок 31. Страница прогноза выбранной переменной



2.6. Создание удаленного репозитория и загрузка результатов работы на него

Страница на GitHub - https://github.com/Turto1se/BMSTU-VKR

Репозиторий с загруженными результатами работы - https://github.com/Turto1se/BMSTU-VKR

Заключение

Цель проекта заключалась в создании прогнозных моделей, которые помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

В ходе выполнения проекта была выполнена предобработка данных (проверка на пропуски, выбросы, выполнена нормализация данных), проведен разведочный анализ данных на основании результатов которого простые зависимости между рассматриваемыми переменными не наблюдаются. Далее были разработаны и обучены модели и нейронные сети для прогноза конечных свойств композиционных материалов, а также разработано web-приложение для прогноза конечных свойств композитных материалов с помощью обученных нейронных сетей.

На основании полученных результатов можно сделать вывод, что данные находятся в сложной зависимости между собой. Простые модели и нейронные сети, разработанные при выполнении проекта, не демонстрируют приемлемого уровня точности. Для решения поставленной задачи необходимо применить комплексное решение.

Список использованной литературы

1. Грас, Джоэл. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2021. - 416 с.
2. Жерон, Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. Пер. с англ. - СпБ.: ООО "Альфа-книга': 2018. - 688 с.
3. Джулли, Пал: Библиотека Keras - инструмент глубокого обучения / пер. с англ. А. А. Слинкин.- ДМК Пресс, 2017. – 249 с.
4. https://ru.wikipedia.org/wiki/
5. https://neerc.ifmo.ru/wiki/
6. https://wiki.loginom.ru/articles/multilayered-perceptron.html
7. https://neerc.ifmo.ru/wiki/index.php?title=Batch-normalization
8. https://scikit-learn.org/stable/user_guide.html
9. https://keras.io/guides/
10. https://www.tensorflow.org/guide
11. https://e-plastic.ru/specialistam/composite/kompozicionnye-materialy/
12. https://statpsy.ru/correlation/correlation/
13. https://www.machinelearningmastery.ru/5-ways-to-detect-outliers-that-every-data-scientist-should-know-python-code-70a54335a623/