ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
по курсу
«Data Science PRO»

Слушатель: Филимонов Фёдор Игоревич

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

План работы

1. Ознакомление с поставленной задачей (Прогнозирование конечных свойств новых материалов (композиционных материалов)
2. Разбивка общей задачи, на под задачи.
3. Описание работы
4. Заключение

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Ознакомление с поставленной задачей

Тема данной работы - прогнозирование конечных свойств новых материалов (композиционных материалов).

При ознакомление с задачей были установлены следующие требования:
1. Провести разведочный анализ предложенных данных;
2. Провести предобработку данных;
3. Обучить нескольких моделей для прогноза нескольких параметров («модуля упругости при растяжении» и «прочности при растяжении»);
4. Написать нейронную сеть, которая будет рекомендовать параметр «соотношение матрица-наполнитель»;
5. Разработать приложение, которое будет выдавать прогноз, по одной из обученной модели.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разбивка общей задачи, на под задачи.

Самым важным аспектом в данной работе, для меня было это не пытается охватить всю задачу сразу и приступит к её решению. А поделить её на несколько подзадач и решать их по очереди. Для дисциплинированния себя в этом вопросе, после составления плана (который корректировался в процессе) была составлена диаграмма Ганта, которая помогала мне держаться во временных рамках (конечно же она тоже корректировалась в процессе работы). Правильная подготовка к работе первый шаг к успешному решению задачи. В ходе выполнения работы были использованы методы и навыки приобретённые за время курса.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Список задач и диаграмма Ганта

План работы:

1. Загрузка данных из исходных excel таблиц (X_bp и X_pup)
2. Ознакомление с данными, кол-во строк и столбцов, вывод названия столбцов, вывод нескольких строк таблиц.
3. Удаление неинформативных данных (столбец)
4. Производим объединение двух таблиц по индексу, тип объединения — INNER
5. Проверяем типы данных по каждому столбцу
6. Проверяем пропущенные значений
7. Проверяем на наличие дубликатов
8. Поиск уникальных значений
9. Ознакомление с описательной статистикой данных.
10. Согласно заданию, получаем среднее, медианное значение для каждой колонки отдельно
11. Визуализация корреляционной матрицы с помощью тепловых карт используя методы "Пирсона", "Кендалл", "Спирмена"
12. Создаём палитру для единообразной визуализации данных
13. Построение гистограмм распределения
14. Построение коробочной диаграммы
15. Определяем кол-во выбросов тремя методами (Метод межквартального размаха (IQR), Метод стандартного отклонения, Метод Z-оценка.)
16. Удаляем выбросы методом (IQR)
17. Проверяем, что осталось от данных.
18. Проводим нормализацию несколькими методами: MinMaxScaler, Normalizer, MaxAbsScaler, RobustScaler, сравниваем их точность и выберем лучший результат
19. Сохраняем подготовленные данные (очищенные и нормализованные) двумя файлами
20. Начинаем процесс создания и обучения моделей согласно заданию. Задача: 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.
21. Определяем какие методы будут использоваться для обучения моделей: 'Линейная регрессия', 'Хребет', 'Опорные вектора', 'Случайный лес', 'Градиентный бустинг '
22. Определяем гиперпараметры для каждого метода
23. Начинаем поиск оптимальных гиперпараметров через GridSearchCV 10 блоков (согласно задаче)
24. Обучение моделей, получение оценок (R^2, MAE, MSE, MAPE)
25. Построение графиков сравнения тестовых и обученных данных для каждой модели
26. Построение нейронной сети, которая будет рекомендовать соотношение матрица-наполнитель.
27. Определение кол-во слоёв, нейронов, настройка раннего прекращения обучения
28. Обучение модели, получение оценки и построение графика сравнения тестовой и обученной модели.
29. Сохранение модели
30. Разработка приложения с графическим интерфейсом.
31. Т.к. в первоначальных вариантах обучения моделей я использовал нормализованные данные (лежат в диапазоне от 0 до 1), при тестировании графического приложения я не мог получить желаемого результата. Было принято решение взять подготовленные данные до нормализации и повторить операцию по созданию нейронной сети, сохранению её модели для дальнейшей работы.
32. Создание приложения с помощью tkinter
33. Добавление функции определения ошибки (не верно введённых данных), а также «подсказку» в каких диапазонах нужно указывать значения для получения результата.

План организации ежегодной инвентаризации СК

Старт 9 дек 24 | Дней 12 | Завершение 21 дек 24 | Готов на 67% | 1 день

Описание	Старт	Дней	Финиш	Статус	%	Комментарии
Начало работы по подготовке ВКР	07.12.24	1	08.12.24	closed	100%	Ознакомится с задачей и тебованиям к ВКР.
Изучить тему "Компазитов"	07.12.24	2	09.12.24	closed	100%	Почитать в интернете, что это за "зверь" такой, какие проблемы сейчас актуальны по данном у вопросу.
Приступаем к написанию кода (загрузка DF и начинаем его "крутить")	09.12.24	1	10.12.24	closed	100%	Начать с подготовки данных: Загрузить, ознакомится с их содержанием, (читай/смотри ноутбуки лекций)
Обучаем модели.	10.12.24	1	11.12.24	closed	100%	Почитать про нормализацию и выбросы, определить методы и сделать
Построение нейро сети	12.12.24	2	14.12.24	closed	100%	Всё то же самое
Разработка приложения	13.12.24	3	16.12.24	closed	100%	Выбрать: модель на которой будешь делать приложение, Вы брать графический или текстовый (Внимание: помни лучшее враг хорошего)
Написать Записку	17.12.24	2	19.12.24	in progress	0%	Описать свою проделанную работу
Сделать призентацию	17.12.24	2	19.12.24	in progress	0%	Визуализировать и сжато предоставить информацию о проделанной работе
Отправить файлы до 19/12/24	18.12.24	1	19.12.24	open	0%	Почто reception@edubmstu.ru
ЗАЩИТА!!!!!	20.12.24	1	21.12.24	open	0%	КРЕПИСЬ!

Трекер вопросов

Вопрос	Статус	Дата открытия	Дата закрытия	Комментарии
Разработка приложения	**Сделано**	13.12.24	16.12.24	Выбрать: модель на которой будешь делать приложение, Вы брать графический или текстовый (Внимание: помни лучшее враг хорошего)
Написать Записку	**В процессе**	17.12.24	19.12.24	Описать свою проделанную работу
Сделать призентацию	**В процессе**	17.12.24	19.12.24	Визуализировать и сжато предоставить информацию о проделанной работе
Отправить файлы до 19/12/24	**Не сделано**	18.12.24	19.12.24	Почто reception@edubmstu.ru
ЗАЩИТА!!!!!	**Не сделано**	20.12.24	21.12.24	КРЕПИСЬ!

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Описание работы (объединение данных и их анализ)

Выводим часть таблицы по обоим файлам для ознакомления с информацией содержащихся в них

```
DS1.head(4) # вывод первых 4 строк
```

✓ 0.0s Python

	Unnamed: 0	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	0	1.857143	2030.0	738.736842	30.0	22.267857	100.000000	210.0	70.0	3000.0	220.0
1	1	1.857143	2030.0	738.736842	50.0	23.750000	284.615385	210.0	70.0	3000.0	220.0
2	2	1.857143	2030.0	738.736842	49.9	33.000000	284.615385	210.0	70.0	3000.0	220.0
3	3	1.857143	2030.0	738.736842	129.0	21.250000	300.000000	210.0	70.0	3000.0	220.0

Производим объеденение двух таблиц по индексу, тип объединения — INNER. озакомления.

```
DS3=pd.merge(DS1,DS2,left_index = True, right_index = True, how = 'inner')
```

✓ 0.0s

```
DS3.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
Index: 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   int64
 11  Шаг нашивки                       1023 non-null   float64
 12  Плотность нашивки                 1023 non-null   float64
dtypes: float64(12), int64(1)
memory usage: 111.9 KB
```

```
DS3.nunique() # поиск уникальных значений
```

✓ 0.0s

```
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
```

```
DS3.isnull().sum() # Проверка наличия пропущенных
```

✓ 0.0s

```
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
```

Объединение данных:
- Импортируем необходимые библиотеки;
- Загружаем файлы;
- Ознакомимся с содержимым;
- Произведём объединение двух файлов.

Разведочный анализ данных:
- Изучим информацию о датасете;
- Проверим типы данных в каждом столбце;
- Проверим пропуски;
- Проверим уникальные значения.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Описание работы (ознакомимся с описательной статистикой наших данных.)

наименование столбца	Значение
ount	количество значений
mean	среднее значение
std	стандартное отклонение
min	минимум
25%	верхнее значение первого квартиля
50%	медиана
75%	верхнее значение третьего квартиля
max	максимум

```
description = DS3.describe() # Получение описательной статистики

description.T # Поворот таблицы
```
✓ 0.0s

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660	5.591742
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375	2207.773481
модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	500.047452	739.664328	961.812526	1911.536477
Количество отвердителя, м.%	1023.0	110.570769	28.295911	17.740275	92.443497	110.564840	129.730366	198.953207
Содержание эпоксидных групп,%_2	1023.0	22.244390	2.406301	14.254985	20.608034	22.230744	23.961934	33.000000
Температура вспышки, С_2	1023.0	285.882151	40.943260	100.000000	259.066528	285.896812	313.002106	413.273418
Поверхностная плотность, г/м2	1023.0	482.731833	281.314690	0.603740	266.816645	451.864365	693.225017	1399.542362
Модуль упругости при растяжении, ГПа	1023.0	73.328571	3.118983	64.054061	71.245018	73.268805	75.356612	82.682051
Прочность при растяжении, МПа	1023.0	2466.922843	485.628006	1036.856605	2135.850448	2459.524526	2767.193119	3848.436732
Потребление смолы, г/м2	1023.0	218.423144	59.735931	33.803026	179.627520	219.198882	257.481724	414.590628
Угол нашивки, град	1023.0	44.252199	45.015793	0.000000	0.000000	0.000000	90.000000	90.000000
Шаг нашивки	1023.0	6.899222	2.563467	0.000000	5.080033	6.916144	8.586293	14.440522
Плотность нашивки	1023.0	57.153929	12.350969	0.000000	49.799212	57.341920	64.944961	103.988901

```
DS3.mean() # среднее значение в отдельной колонке
```
✓ 0.0s

```
Соотношение матрица-наполнитель          2.930366
Плотность, кг/м3                      1975.734888
модуль упругости, ГПа                  739.923233
Количество отвердителя, м.%            110.570769
Содержание эпоксидных групп,%_2         22.244390
Температура вспышки, С_2               285.882151
Поверхностная плотность, г/м2          482.731833
Модуль упругости при растяжении, ГПа    73.328571
Прочность при растяжении, МПа         2466.922843
Потребление смолы, г/м2                218.423144
Угол нашивки, град                      44.252199
Шаг нашивки                              6.899222
Плотность нашивки                       57.153929
dtype: float64
```

```
DS3.median()# медиана в отдельной колонке
```
✓ 0.0s

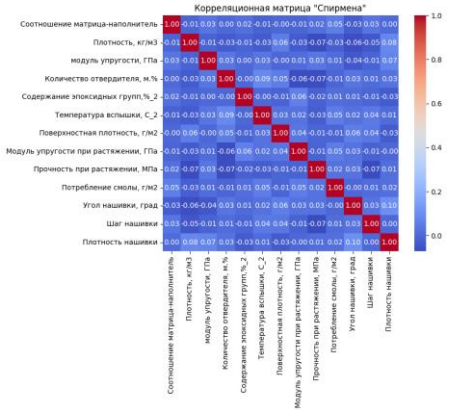
```
Соотношение матрица-наполнитель          2.906878
Плотность, кг/м3                      1977.621657
модуль упругости, ГПа                  739.664328
Количество отвердителя, м.%            110.564840
Содержание эпоксидных групп,%_2         22.230744
Температура вспышки, С_2               285.896812
Поверхностная плотность, г/м2          451.864365
Модуль упругости при растяжении, ГПа    73.268805
Прочность при растяжении, МПа         2459.524526
Потребление смолы, г/м2                219.198882
Угол нашивки, град                       0.000000
Шаг нашивки                              6.916144
Плотность нашивки                       57.341920
dtype: float64
```

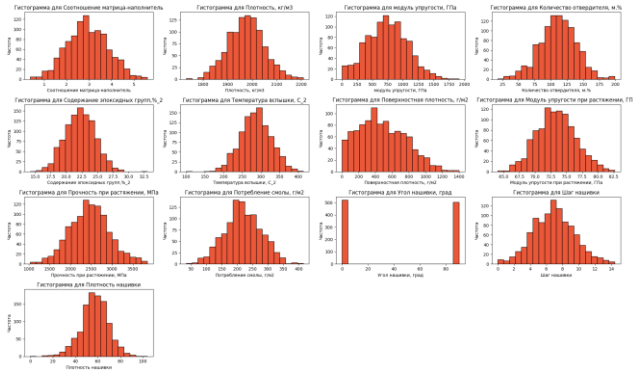
Описательная статистика:
•Получаем значения по датасету.

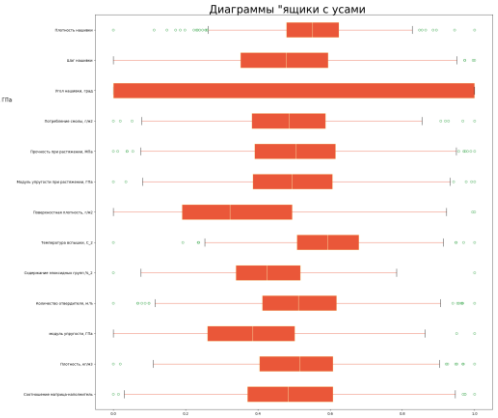
Первое выполнение задания:
•Для каждой колонке получаем среднее, медианное значение.

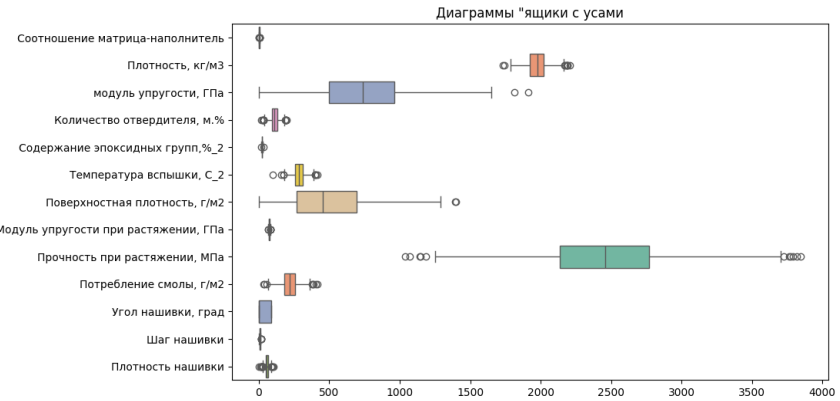
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

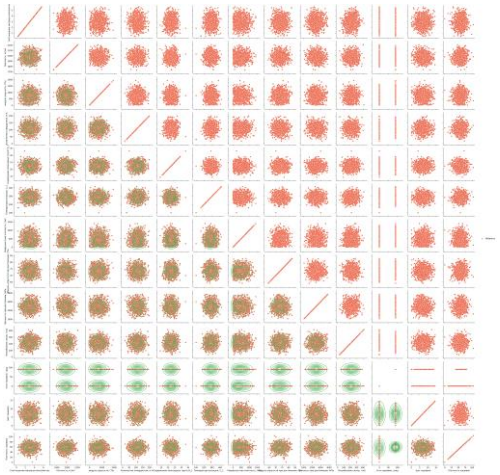
Описание работы (визуализация наших данных.)











Визуализация данных до нормализации и очистки от выбросов:

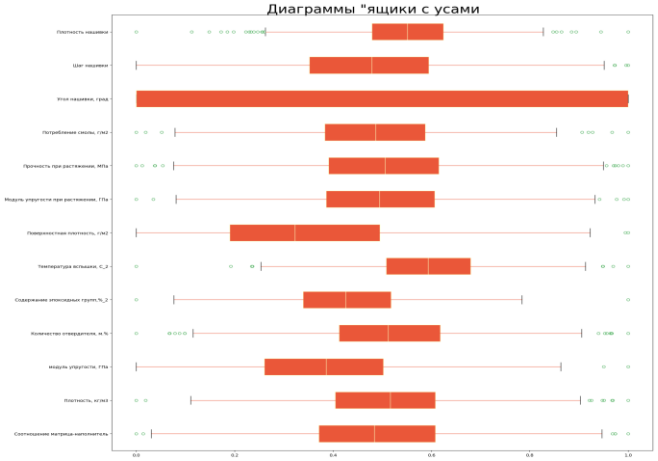
- Пример тепловых карт корреляций;
- Гистограммы распределения;
- Примеры коробочной диаграммы ;
- Пример попарного графика рассеяния точек.

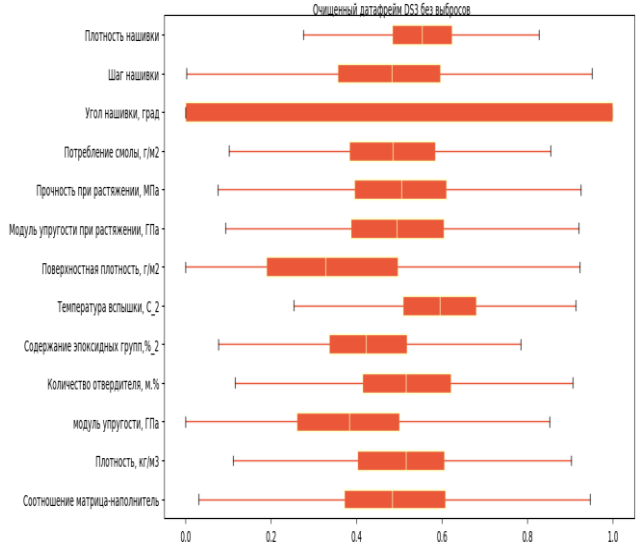
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Описание работы (выявляем выбросы)

	Column	IQR	Standard Deviation	Z-Score
0	Соотношение матрица-наполнитель	6	0	0
1	Плотность, кг/м3	9	3	3
2	модуль упругости, ГПа	2	2	2
3	Количество отвердителя, м.%	14	2	2
4	Содержание эпоксидных групп,%_2	2	2	2
5	Температура вспышки, С_2	8	3	3
6	Поверхностная плотность, г/м2	2	2	2
7	Модуль упругости при растяжении, ГПа	6	0	1
8	Прочность при растяжении, МПа	11	0	0
9	Потребление смолы, г/м2	8	3	3
10	Угол нашивки, град	0	0	0
11	Шаг нашивки	4	0	0
12	Плотность нашивки	21	7	7

```
<class 'pandas.core.frame.DataFrame'>
Index: 921 entries, 1 to 1022
Data columns (total 13 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Соотношение матрица-наполнитель     921 non-null    float64
 1   Плотность, кг/м3                    921 non-null    float64
 2   модуль упругости, ГПа               921 non-null    float64
 3   Количество отвердителя, м.%         921 non-null    float64
 4   Содержание эпоксидных групп,%_2     921 non-null    float64
 5   Температура вспышки, С_2            921 non-null    float64
 6   Поверхностная плотность, г/м2       921 non-null    float64
 7   Модуль упругости при растяжении, ГПа  921 non-null  float64
 8   Прочность при растяжении, МПа       921 non-null    float64
 9   Потребление смолы, г/м2             921 non-null    float64
 10  Угол нашивки, град                  921 non-null    int64
 11  Шаг нашивки                         921 non-null    float64
 12  Плотность нашивки                   921 non-null    float64
dtypes: float64(12), int64(1)
```





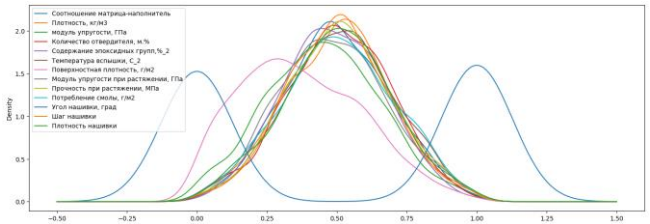
Визуализация данных до нормализации и очистки от выбросов:

- Определяем кол-во выбросов тремя методами (Метод межквартильного размаха (IQR), Метод стандартного отклонения, Метод Z-оценка.)
- Визуальный контроль через «ящик с усам»
- Проведение очистки от выбросов до полного устранения и получения конечного результата (было удалено приблизительно 10% данных)

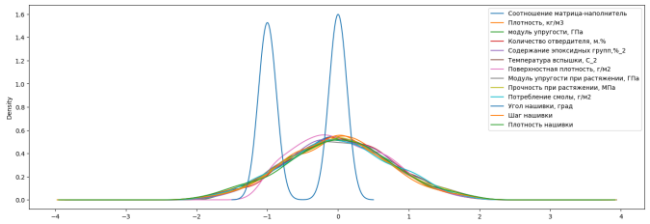
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Описание работы (Нормализация данных)

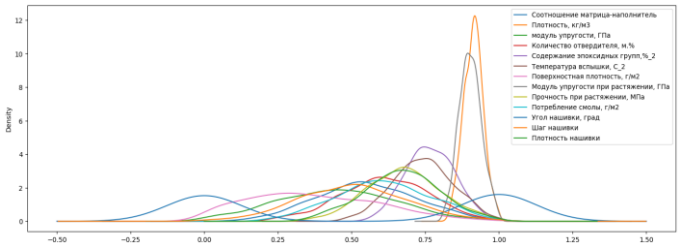
```
# Нормализация данных MinMaxScaler
scaler_minmax = MinMaxScaler()
DS3_normalized_1= pd.DataFrame(scaler_minmax.fit_transform(DS3_cleaned), columns=DS3_cleaned.columns)
```
✓ 0.0s



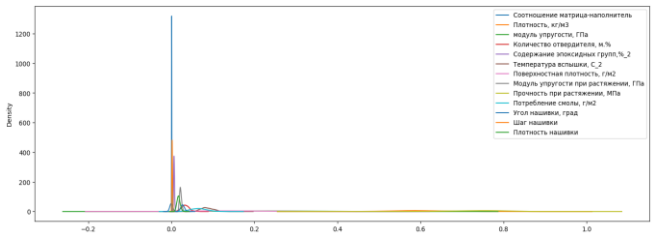
```
# Нормализация данных RobustScaler
scaler_robust = RobustScaler()
DS3_normalized_2 = pd.DataFrame(scaler_robust.fit_transform(DS3_cleaned), columns=DS3_cleaned.columns)
```
✓ 0.0s



```
scaler_maxabs = MaxAbsScaler()
DS3_normalized_3 = pd.DataFrame(scaler_maxabs.fit_transform(DS3_cleaned), columns=DS3_cleaned.columns)
```
✓ 0.0s

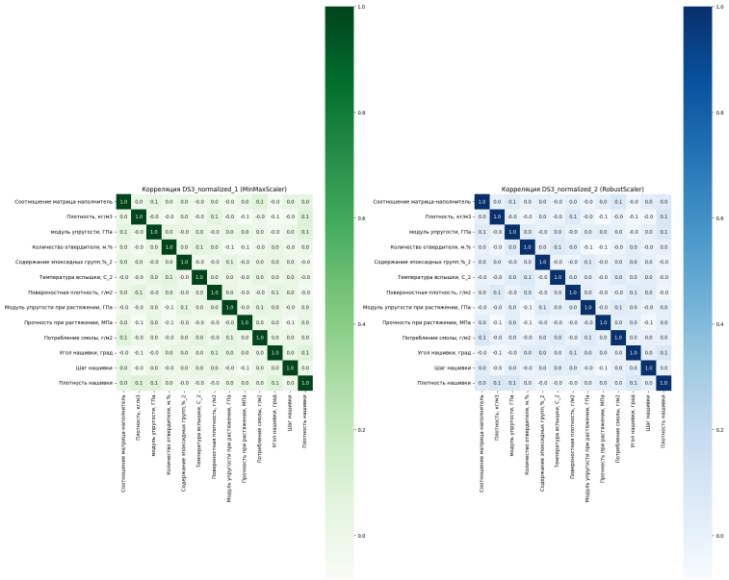


```
normalizer = Normalizer(norm='l2')
DS3_normalized_4 = pd.DataFrame(normalizer.fit_transform(DS3_cleaned), columns=DS3_cleaned.columns)
```
✓ 0.0s



Нормализация очищенных данных:

- Мы опробуем несколько вариантов НОРМАЛИЗАЦИИ: MinMaxScaler, Normalizer, MaxAbsScaler, RobustScaler, сравним их точность и выберем лучший результат.



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Описание работы (процесс создания и обучения моделей)

```python
# Создаём словарь для хранения моделей и их результатов
models = {
    'Линейная регрессия': LinearRegression(),
    'Хребет': Ridge(),
    'Опорные вектора': SVR(),
    'Случайный_лес': RandomForestRegressor(),
    'Градиентный бустинг ': GradientBoostingRegressor()}
```

✓ 0.0s

```python
#Запускаем поиск через GridSearchCV 10 блоков, если таковые есть
best_models = {}
for name, model in models.items():
    if name in param_grids:  # Если есть гиперпараметры для поиска
        grid = GridSearchCV(model, param_grids[name], cv=10, scoring='neg_mean_squared_error')
        grid.fit(X_train, y_train)
        best_models[name] = grid.best_estimator_
        print(f"Лучшие параметры для {name}: {grid.best_params_}")
```

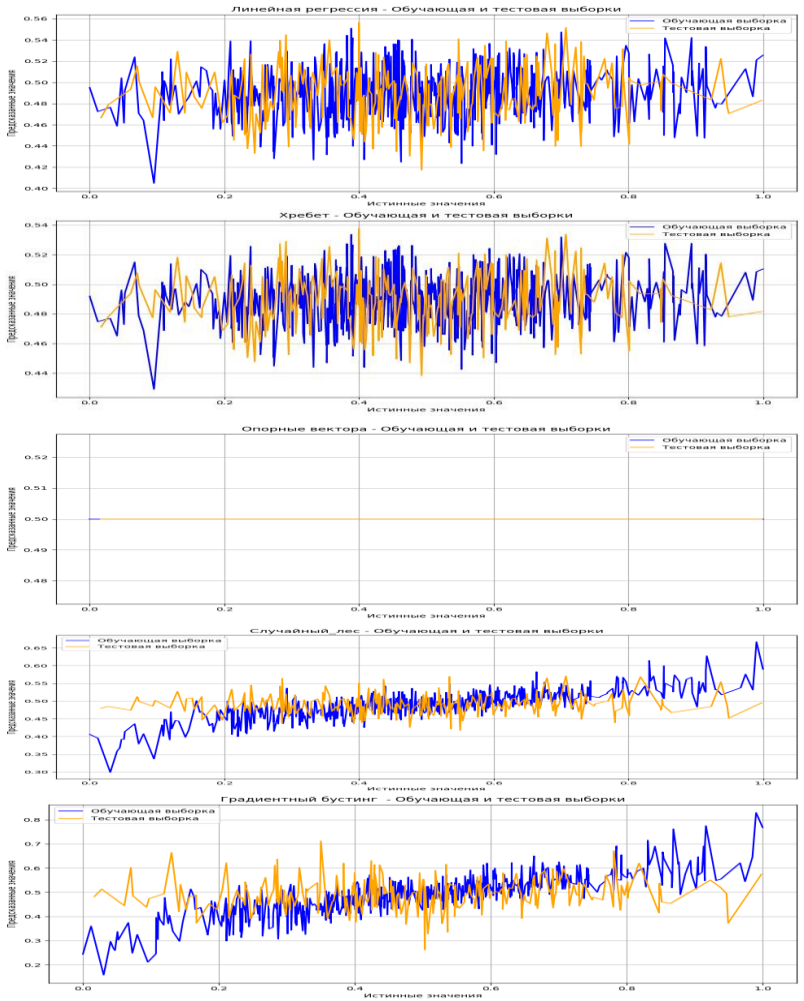
✓ 1m 35.6s

```
Лучшие параметры для Линейная регрессия: {'fit_intercept': True}
Лучшие параметры для Хребет: {'alpha': 10.0}
Лучшие параметры для Опорные вектора: {'C': 0.1, 'epsilon': 0.5, 'gamma': 'scale', 'kernel': 'linear'}
Лучшие параметры для Случайный_лес: {'max_depth': 5, 'n_estimators': 100}
```

```python
#Обучение моделей с оптимальными гипперпараметрами
model.fit(X_train, y_train)
best_models[name] = model
```

✓ 0.5s

	R^2	MAE	MSE	MAPE	Test Score
Линейная регрессия	-0.00	0.16	0.04	64.45	-0.00
Хребет	0.00	0.16	0.04	64.51	0.00
Опорные вектора	-0.01	0.16	0.04	66.38	-0.01
Случайный_лес	-0.01	0.16	0.04	65.10	-0.01
Градиентный бустинг	-0.06	0.16	0.04	66.59	-0.06



Создание и обучени:

- Сначала определили набор методов.
- Согласно задаче определили гипперпараметры и получили оптимальные по условию(cv=10)
- Обучили модели и получили оценки и визуализировали результаты.
- Для "модуля упругости при растяжении« и "прочность при растяжении« использовались одинаковые методы.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Описание работы (процесс создания нейронной сети)

```
model = Sequential()  # Создание модели

# Добавление слоев с регуляризацией и Dropout
model.add(Dense(64, activation='tanh', input_shape=(X_train.shape[1],), kernel_regularizer='l2'))
model.add(Dropout(0.3))  # Слой Dropout
model.add(Dense(64, activation='tanh', kernel_regularizer='l2'))
model.add(Dropout(0.3))  # Слой Dropout
model.add(Dense(32, activation='tanh', kernel_regularizer='l2'))
model.add(Dense(1, activation='sigmoid'))
```

✓ 0.3s

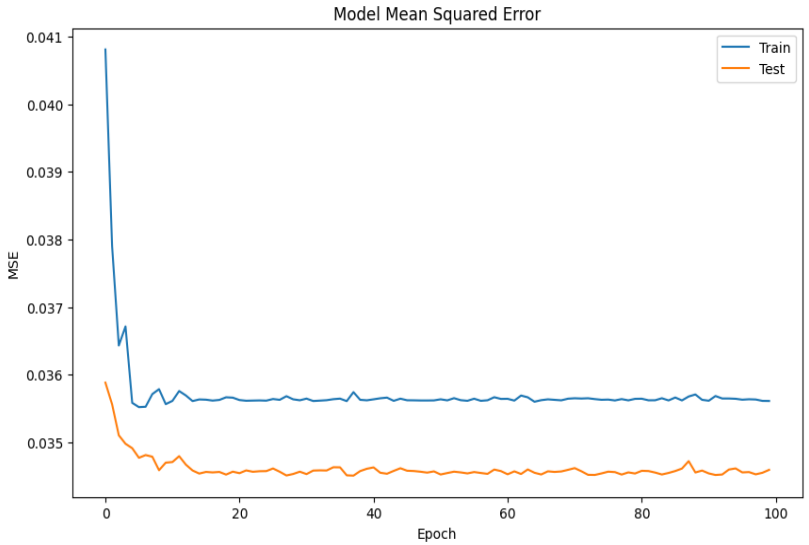
```
model.compile(optimizer='adam', loss='mse', metrics=['mse'])  # Для регрессии

early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)# Настройка раннего прекращения

history = model.fit(X_train, y_train, epochs=100, batch_size=32, validation_data=(X_test, y_test), verbose=0)
```

✓ 17.7s

```
test_loss, test_mse, = model.evaluate(X_test, y_test)
print(f'MSE: {test_mse:.4f}')
```

✓ 0.2s

```
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0345 - mse: 0.0345
MSE: 0.0346
```



Создание и обучении:

- Создаём модель из 4 слоёв
- Добавляем слои Dropout
- Делаем настройку раннего прекращения обучения
- Получаем оценку модели (MSE) и визуализируем результат

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

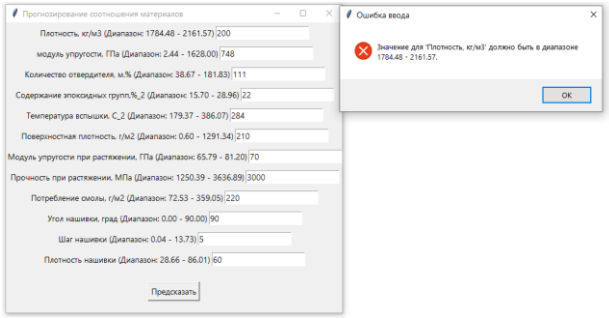
Описание работы (Создание приложения с графическим интерфейсом.)

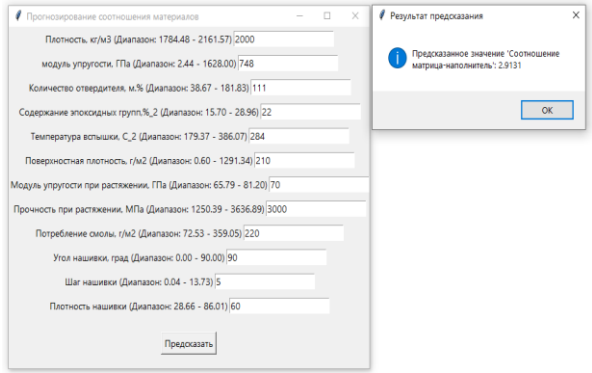
```python
# Поля ввода для каждого признака
entries = {}
for col in columns:
    frame = tk.Frame(root)
    frame.pack(pady=5)
    label = tk.Label(frame, text=f"{col} (Диапазон: {ranges[col][0]:.2f} - {ranges[col][1]:.2f})")
    label.pack(side=tk.LEFT)
    entry = tk.Entry(frame)
    entry.pack(side=tk.RIGHT)
    entries[col] = entry
```

✓ 0.0s

```python
# Функция для предсказания
def predict():
    try:
        # Сбор данных из полей ввода
        input_data = []
        for col in columns:
            value = float(entries[col].get())
            if not (ranges[col][0] <= value <= ranges[col][1]):
                raise ValueError(f"Значение для '{col}' должно быть в диапазоне {ranges[col][0]:.2f} - {ranges[col][1]:.2f}.")
            input_data.append(value)

        # Нормализация данных
        input_data = scaler.transform([input_data])

        # Предсказание
        prediction = model.predict(input_data)[0][0]
        messagebox.showinfo("Результат предсказания", f"Предсказанное значение 'Соотношение матрица-наполнитель': {prediction:.4f}")

    except ValueError as e:
        messagebox.showerror("Ошибка ввода", str(e))
    except Exception as e:
        messagebox.showerror("Ошибка", f"Произошла ошибка: {str(e)}")
```

✓ 0.0s





Создание приложения:

• Создаём приложения с использованием библиотеки tkinter

• Добавляем функцию уведомления об ошибочно введённых данных

• Добавлена «шпаргалка» с диапазонами для ввода данных.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Заключение.

Решить поставленную задачу не получилось, так как модели показали себя не с лучшей стороны. Возможными причинами неудачи могли стать как изначальный набор данных, так и методы и подходы к решению поставленной задачи. Не без основания сюда же можно отнести и мои компетенции, которые находятся на начальном этапе и отсутствия опыта. Не знание в области для которой ведутся расчёты (композитные материалы), так же могут отрицательно сказаться на решении задачи. Нужно обращается к металлургам или химикам для более глубокого понимания предмета по композитам.

Спасибо за внимание.

ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

do.bmstu.ru