Состав проекта.

Проект разрабатывался и тестировался в среде PostgreSQL 15.2, pgAdmin 4 ver. 6.2 и IDLE Python 3.11.5 и состоит из отдельных скриптов SQL-запросов и python-скриптов, каждый из которых выполняет отдельную подзадачу. SQL-скрипты размещены в каталоге src\sql, a python-скритпты – в каталоге src\python.

Наименование и назначение SQL-скриптов:

1. Create\_table\_X\_bp.sql - создает таблицу базы данных для импорта в нее данных файла X\_bp.xlsx.
2. X\_bp.csv – экспортированные данные из X\_bp.xlsx в текстовый формат с разделителем «;» между полями.
3. X\_bp4pg.csv – обработанный файл X\_bp.csv, у которого удален заголовок и заменен разделитель десятичных разрядов с «,» на «.»и пригодный для импортирования в PostreSQL средствами pgAdmin.
4. Create\_table\_X\_nup.sql - создает таблицу базы данных для импорта в нее данных файла X\_nup.xlsx.
5. X\_nup.csv – экспортированные данные из X\_nup.xlsx в текстовый формат с разделителем «;» между полями.
6. X\_nup4pg.csv – обработанный файл X\_nup.csv, у которого удален заголовок и заменен разделитель десятичных разрядов с «,» на «.» и пригодный для импортирования в PostreSQL средствами pgAdmin.
7. CountInX\_bp\_Xnup.sql – подсчитывает количество строк в таблицах X\_bp и X\_nup.
8. INNER\_JOIN.sql – производит INNER JOIN соединение таблиц X\_bp и X\_nup по индексному полю в новую таблицу full\_X\_bp.
9. ProverkaNaNull.sql – производит проверку таблиц на пустые данные.
10. statisticsbyfull.sql - вычисляет параметры описательной статистики (среднее, медиану, максимальное, минимальное и стандарное отклонение) из результирущего множества.
11. OtsevDataBy3Sigma.sql - отсеивает выбросы методом 3 сигм.
12. OtsevDataByInterval.sql – отсеивает выбросы методом межквартильных расстояний.

Наименование и назначение python-скриптов:

1. 1\_PrimaryAnalizeInputData.py - загружает файлы X\_bp.xlsx и X\_nup.xlsx в pandas.DataFrame, выводит по ним общие сведения, выполняет соединение таблиц методам INNER JOIN, рассчитывает параметры описательной статистики по объединенной таблице, формирует таблицу результатов. Кроме этого выполняет сохранение минимального, максимального и среднего значения для каждого поля.
2. minmaxmeans.json - сохраненные значения минимум, максимум среднее для каждого поля. Требуется для работы разработанного консольного приложения.
3. 2\_BuildDisplotAndBoxplot\_1.py – построение графиков гистограмм распределения и диаграмм «ящик с усами».
4. 3\_BuildPearGrid\_1.py - построение попарных графиков рассеяния точек.
5. 4\_OtsevBy3SigmAndIntervals.py – отсев выбросов методом 3 сигм и межквартильных расстояний.
6. 5\_OtsevBy3SigmAndBuildsGrafs.py – отсев выбросов методом 3 сигм, построение графиков по отсеянному множеству. Сохранение очищенной коллекции в файл clearing\_X\_bp\_nup.xlsx.
7. clearing\_X\_bp\_nup.xlsx – очищенные от выбросов данные, результат работы скрипта 5\_OtsevBy3SigmAndBuildsGrafs.py.
8. 6\_normalizedata.py – нормализация данных, сохранение нормализаванной коллекции в файл normalized\_data.xlsx. Кроме этого сохраняет коэффициенты и дельты для каждого поля, с которыми проводилась нормализация.
9. mnojanddelta.json – содержит сохраненные коэффициенты и дельты для каждого поля, с которыми проводилась нормализация. Результат работы скрипта 6\_normalizedata.py. Необходим для работы пользовательского приложения.
10. normalized\_data.xlsx – нормализованные данные, результат работы скрипта 6\_normalizedata.py.
11. 7\_CreateDataFor3Models.py – создает 3 датасета для работы моделей для нахождения модуля упругости при растяжении, прочности при растяжении и соотношения матрица-наполнитель.
12. 9\_getAllgiperparams.py – запрашивает все гиперпараметры каждой функции регрессии из библиотеки sklearn.linear, используемой в проекте.
13. 9.1\_podbor\_parameters\_DecisionTreeRegressor.py – подбор гиперпараметров для DecisionTreeRegressor для всех 3-х датасетов.
14. 9.1\_podbor\_parameters\_KNeighborsRegressor.py – подбор гиперпараметров для KNeighborsRegressor для всех 3-х датасетов.
15. 9.1\_podbor\_parameters\_Lasso.py – подбор гиперпараметров для Lasso для всех 3-х датасетов.
16. 9.1\_podbor\_parameters\_svm.py – подбор гиперпараметров для SVR для всех 3-х датасетов.
17. 9.1\_podbor\_parameters\_Ridge.py – подбор гиперпараметров для Ridge для всех 3-х датасетов.
18. 9.1\_podbor\_parameters\_GradientBoosting.py – подбор гиперпараметров для GradientBoosting для всех 3-х датасетов.
19. 9.1\_podbor\_parameters\_RandomForestRegressor.py – подбор гиперпараметров для RandomForestRegressor для всех 3-х датасетов.
20. 9\_examine\_mat\_nap.py – испытание всех моделей на 3-x датасетах, с настройками по умалчанию и оптимальных, на обучающем множестве и тестовом, сбор статистики качества работы модели по разным оценочным метрикам.
21. 12.1\_keras\_\_modupr.py – построение и обучение модели для определения модуля упругости в tensorflow и сохранение обученной модели на диске в виде файла moduprmodel.keras.
22. moduprmodel.keras – сохраненная модель, необходима для работы приложения.
23. 12.2\_keras\_modproch.py – построение и обучение модели для определения модуля прочности в tensorflow и сохранение обученной модели на диске в виде файла modprochrastmodel.keras.
24. modprochrastmodel.keras – сохраненная модель, необходима для работы приложения.
25. 12.3\_keras\_\_mat\_nap.py – построение и обучение модели для определения соотношения матрица - наполнитель в tensorflow и сохранение обученной модели на диске в виде файла mat\_napmodel.keras.
26. mat\_napmodel.keras – сохраненная модель, необходима для работы приложения.
27. 13\_consoleprogramms.py – консольная программа для предсказания модуля упругости при растяжении и прочности при растяжении, рекомендации соотношения матрица-наполнитель.