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 Тулупов Роман Иванович (ФИО)

Москва, 2023

Содержание

[Содержание 2](#_Toc133336553)

[Введение 3](#_Toc133336554)

[Основная часть 4](#_Toc133336555)

[1. Аналитическая часть 4](#_Toc133336556)

[1.1. Постановка задачи. 4](#_Toc133336557)

[1.2. Описание используемых методов 5](#_Toc133336558)

[1.3. Разведочный анализ данных 10](#_Toc133336559)

[2. Практическая часть 14](#_Toc133336560)

[2.1. Предобработка данных 14](#_Toc133336561)

[2.2. Разработка и обучение модели 17](#_Toc133336562)

[2.3. Тестирование модели 19](#_Toc133336563)

[2.4. Написать нейронную сеть, которая будет рекомендовать соотношение матрица - наполнитель. 19](#_Toc133336564)

[2.5. Разработка приложения 20](#_Toc133336565)

[2.6. Создание удаленного репозитория и загрузка результатов работы на него. 22](#_Toc133336566)

[Заключение 23](#_Toc133336567)

[Список литературы и веб ресурсы 24](#_Toc133336568)

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. Яркий пример композита - железобетон. Бетон прекрасно сопротивляется сжатию, но плохо растяжению. Стальная арматура внутри бетона компенсирует его неспособность сопротивляться сжатию, формируя тем самым новые, уникальные свойства. Современные композиты изготавливаются из других материалов: полимеры, керамика, стеклянные и углеродные волокна, но данный принцип сохраняется. У такого подхода есть и 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).

На входе имеются данные о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.).

На выходе необходимо спрогнозировать ряд конечных свойств получаемых композиционных материалов.

Кейс основан на реальных производственных задачах Центра НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана).

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

Основная часть

1. Аналитическая часть

1.1. Постановка задачи.

Для сокращения количества физических испытаний, а также для пополнения базы данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов, требуется разработать приложение для предсказания свойств материалов на основе предложенного датасета. Имеется два файла X\_bp.xlsx - с информацией о параметрах, состоящий из 1023 строк и 11 колонок и X\_nup.xlsx - с информацией о нашивке, состоящий из 1040 строк и 4 колонок. Файл X\_bp.xlsx содержит колонку с индексами, колонки с начальными параметрами:

«Соотношение матрица-наполнитель»;

«Количество отвердителя, м.%»;

«Содержание эпоксидных групп,%\_2»;

«Температура вспышки, С\_2»;

«Потребление смолы, г/м2».

и колонки с характеристиками полученного материала:

«Плотность, кг/м3»;

«модуль упругости, ГПа»;

«Поверхностная плотность, г/м2»;

«Модуль упругости при растяжении, ГПа»;

«Прочность при растяжении, МПа».

Файл X\_nup.xlsx содержит колонку с индексами и колонки с информацией о нашивке:

«Угол нашивки, град»;

«Шаг нашивки»;

«Плотность нашивки».

Файлы объединяются по индексу в датасет «full\_df» тип объединения INNER. Объем полученной выборки составляет 1023 строки и 13 признаков, из них 8 являются входными переменными и 5 выходными переменными. В данном задании колонки «Соотношение матрица-наполнитель», «Модуль упругости при растяжении, ГПа», «Прочность при растяжении, МПа» являются целевыми.

Изображение выглядит как стол

Автоматически созданное описание

Пропусков нет, разброс данных по всем колонкам достигает 5 порядков, отрицательных значений нет. «Шаг нашивки» принимает всего два значения 0 и 90. Медиана и средняя близки по значению для всех колонок кроме поверхностной плотности, где распределение, видимо, немного смещено.

1.2. Описание используемых методов

Данная задача является задачей регрессии, буду использовать библиотеку sklearn и следующие модели:

* Линейная регрессия LinearRegression();
* Метод опорных векторов SVR();
* Случайный лес RandomForestRegressor();
* Градиентный бустинг GradientBoostingRegressor();
* Лассо регрессор Lasso().

Линейная регрессия – один из самых простых инструментов статистического моделирования, но именно в его простоте и заключается его эффективность. Прогнозируемое значение целевой переменной выражается в виде суммы постоянного смещения и взвешенной суммы исходных переменных.

Достоинства метода: быстр и прост в реализации; легко интерпретируем; имеет меньшую сложность по сравнению с другими алгоритмами.

Недостатки метода: моделирует только прямые линейные зависимости; требует прямую связь между зависимыми и независимыми переменными; выбросы оказывают огромное влияние, а границы линейны.

В качестве метрики прежде всего используется R2 или коэффициент детерминации, который позволяет измерить, насколько модель может объяснить дисперсию данных. Если R-квадрат равен 1, это значит, что модель описывает все данные. Если же R-квадрат равен 0.5, модель объясняет лишь 50% дисперсии данных. Оставшиеся отклонения не имеют объяснения. Чем ближе R2 к единице, тем лучше.

Метод опорных векторов – это бинарный линейный классификатор. Хорошо работает на небольших датасетах. Алгоритм помечает каждый объект, как принадлежащий к одной из двух категорий, строит модель, которая определяет новые наблюдения в одну из категорий. Модель метода опорных векторов – отображение данных точками в пространстве, так что между наблюдениями отдельных категорий имеется разрыв. Каждый объект данных представляется как вектор (точка) в p-мерном пространстве. Он создаёт линию или гиперплоскость, которая разделяет данные на классы.

Достоинства метода: для классификации достаточно небольшого набора данных; при правильной работе модели, построенной на тестовом множестве, вполне возможно применение данного метода на реальных данных; эффективен при большом количестве гиперпараметров; способен обрабатывать случаи, когда гиперпараметров больше, чем количество наблюдений; существует возможность гибко настраивать разделяющую функцию; алгоритм максимизирует разделяющую полосу, которая, как подушка безопасности, позволяет уменьшить количество ошибок классификации.

Недостатки метода: неустойчивость к шуму; для больших наборов данных требуется долгое время обучения; параметры модели сложно интерпретировать.

Случайный лес — универсальный алгоритм машинного обучения, суть которого состоит в использовании ансамбля решающих деревьев. Само по себе решающее дерево предоставляет крайне невысокое качество классификации, но из-за большого их количества результат значительно улучшается. Также это один из немногих алгоритмов, который можно использовать в абсолютном большинстве задач.

Достоинства метода: имеет высокую точность предсказания, которая сравнима с результатами градиентного бустинга; не требует тщательной настройки параметров, хорошо работает из коробки; практически не чувствителен к выбросам в данных из-за случайного семплирования (random sample); не чувствителен к масштабированию и к другим монотонным преобразованиям значений признаков; редко переобучается; способен эффективно обрабатывать данные с большим числом признаков и классов; хорошо работает с пропущенными данными; одинаково хорошо обрабатывает как непрерывные, так и дискретные признаки.

Недостатки метода: для реализации алгоритма случайного дерева требуется значительный объем вычислительных ресурсов; большой размер моделей; построение случайного леса отнимает больше времени, чем деревья решений или линейные алгоритмы; алгоритм склонен к переобучению на зашумленных данных; нет формальных выводов, которые используются для оценки важности переменных; в отличие от более простых алгоритмов, результаты случайного леса сложнее интерпретировать; когда в выборке очень много разреженных признаков, алгоритм работает хуже, чем линейные методы; в отличие от линейной регрессии, Random Forest не обладает возможностью экстраполяции; это можно считать и плюсом, так как в случае выбросов не будет экстремальных значений; если данные содержат группы признаков с корреляцией, которые имеют схожую значимость для меток, то предпочтение отдается небольшим группам перед большими, что ведет к недообучению; процесс прогнозирования с использованием случайных лесов очень трудоемкий по сравнению с другими алгоритмами.

Градиентный бустинг — это ансамбль деревьев решений, обученный с использованием градиентного бустинга. В основе данного алгоритма лежит итеративное обучение деревьев решений с целью минимизировать функцию потерь. Основная идея градиентного бустинга: строятся последовательно несколько базовых классификаторов, каждый из которых как можно лучше компенсирует недостатки предыдущих. Финальный классификатор является линейной композицией этих базовых классификаторов.

Достоинства метода: новые алгоритмы учатся на ошибках предыдущих; требуется меньше итераций, чтобы приблизиться к фактическим прогнозам; наблюдения выбираются на основе ошибки; прост в настройке темпа обучения и применения; легко интерпретируем.

Недостатки метода: необходимо тщательно выбирать критерии остановки, иначе это может привести к переобучению; наблюдения с наибольшей ошибкой появляются чаще; слабее и менее гибок чем нейронные сети.

Лассо регрессор — это линейная модель, которая оценивает разреженные коэффициенты. Это простой метод, позволяющий уменьшить сложность модели и предотвратить переопределение, которое может возникнуть в результате простой линейной регрессии. Данный метод вводит дополнительное слагаемое регуляризации в оптимизацию модели. Это даёт более устойчивое решение. В регрессии лассо добавляется условие смещения в функцию оптимизации для того, чтобы уменьшить коллинеарность и, следовательно, дисперсию модели. Но вместо квадратичного смещения, используется смещение абсолютного значения. Лассо регрессия хорошо прогнозирует модели временных рядов на основе регрессии, таким как авторегрессии.

Достоинства метода: легко полностью избавляется от шумов в данных; быстро работает; не очень энергоёмко; способно полностью убрать признак из датасета; доступно обнуляет значения коэффициентов.

Недостатки метода: выбор модели не помогает и обычно вредит; часто страдает качество прогнозирования; выдаёт ложное срабатывание результата; случайным образом выбирает одну из коллинеарных переменных; не оценивает правильность формы взаимосвязи между независимой и зависимой переменными; не всегда лучше, чем пошаговая регрессия.

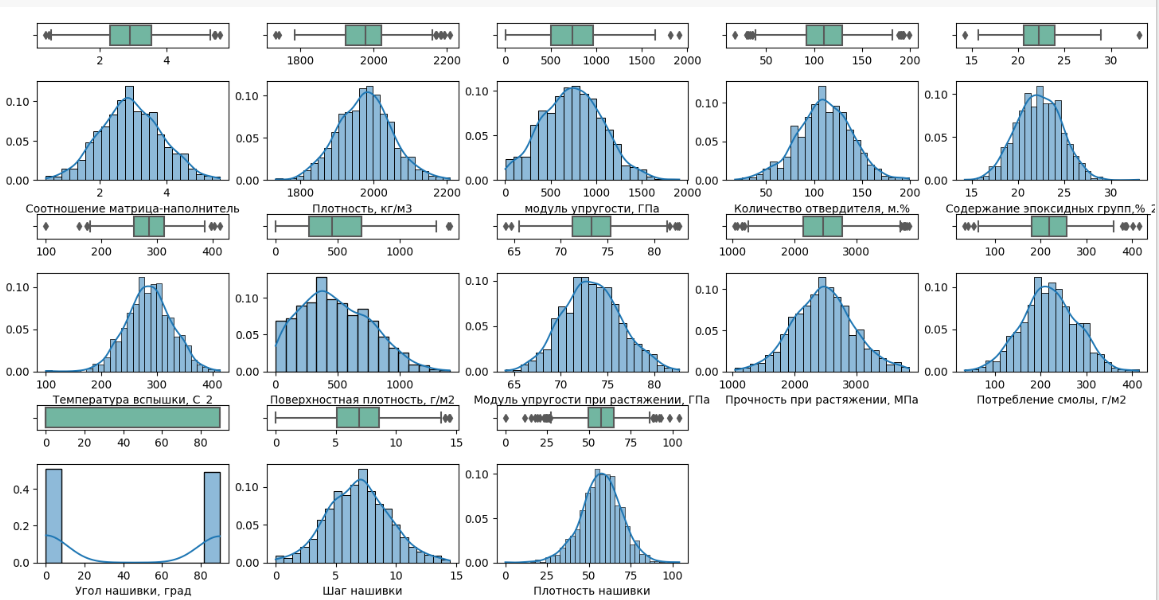
1.3. Разведочный анализ данных

Дублирующие записи не только искажают статистические показатели датасета, но и снижают качество обучения модели.

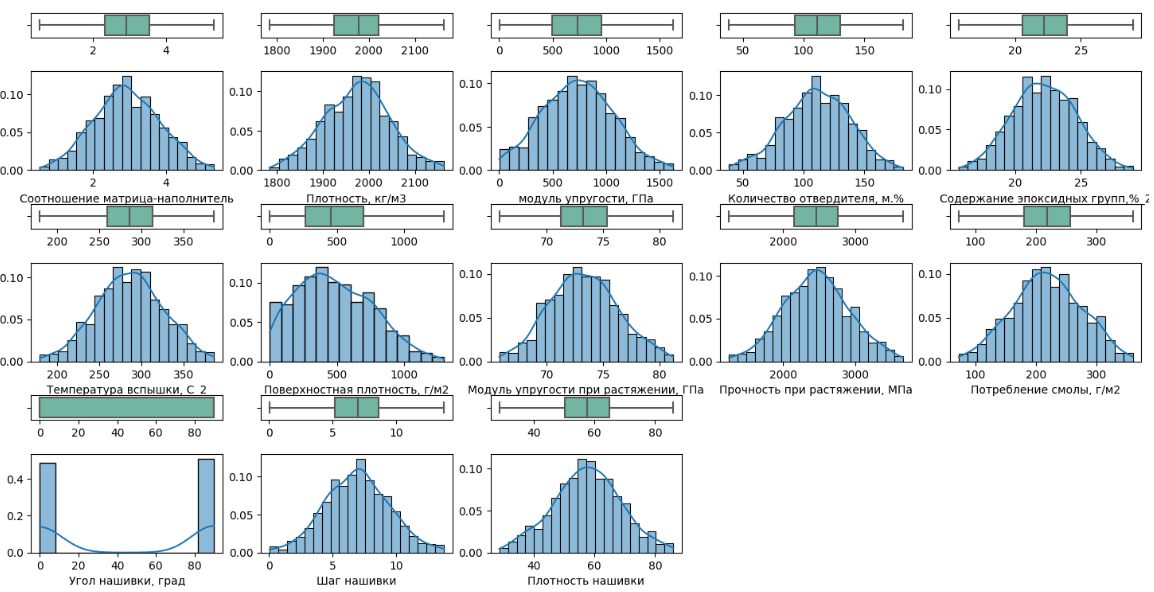
Изображение выглядит как текст

Автоматически созданное описание

Для большей наглядности объединил histplot и boxplot в один комбинированный график, и построил для каждого признака



Распределение по большинству признаков близко к нормальному. Есть выбросы. Существует два распространенных способа удалить выбросы: метод межквартильного диапазона и метод трех сигм. Первый реализован в функции method\_iq(), второй – в method\_3s(). После трехкратного применения method\_iq() от выбросов удалось избавится.



Для того, чтобы сравнить переменный друг с другом, построю тепловую карту корреляций и попарные графики рассеяния точек для всех признаков.

Изображение выглядит как диаграмма

Автоматически созданное описание

Максимальная корреляция между плотностью нашивки и плотностью составляет 0.086, корреляция между всеми параметрами очень близка к 0, значит нет корреляционной связи между переменными

Изображение выглядит как стол

Автоматически созданное описание

На попарных графиках распределения не видно корреляции между признаками.

Среднее, медианное значение для каждой колонки:

Изображение выглядит как стол

Автоматически созданное описание

Среднее и медианное значения по всем колонкам приблизительно равны за исключением поверхностной плотности. Здесь отклонение составляет 5.2%. Ранее на гистограмме можно было видеть на этом признаке смещение распределения влево. Для исправления ситуации можно извлечь корень.

2. Практическая часть

2.1. Предобработка данных

Как было замечено ранее данные имеют разный масштаб. Проведу нормализацию, построю графики распределения до и после нормализации, посмотрю максимальные и минимальные значения.

Изображение выглядит как диаграмма

Автоматически созданное описание

Так выглядят исходные данные. Наблюдается большой разброс.

Изображение выглядит как диаграмма

Автоматически созданное описание

После применения MinMaxScaler значения всех переменных укладываются в диапазон от 0 до 1

Изображение выглядит как диаграмма

Автоматически созданное описание

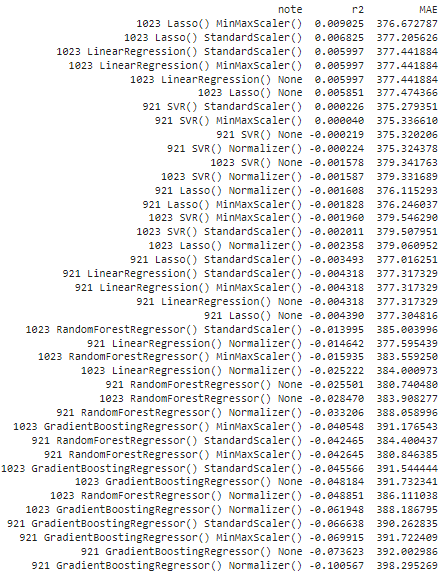
StandardScaler делает среднее значение наблюдаемых значений равным 0, а стандартное отклонение – 1

Изображение выглядит как диаграмма

Автоматически созданное описание

Normalizer приводит каждую строку к норме 1

Чтобы убить всех зайцев сразу, обошел в цикле датасеты (с выбросами и без), модели, препроцессоры. Все свел в одну таблицу, отсортировал по r2



Lasso MinMaxScaler на не очищенном датасете r2 = 0.009025 - лучший результат.

LinearRegression с выбросами r2 = 0.005997 StandardScaler и MinMaxScaler на результат не повлияли он такой же, как и вообще без препроцессинга, Normalizer результат ухудшил.

SVR лучше отработал на чистом датасете без выбросов в связке с StandardScaler r2 = 0.000226

RandomForestRegressor StandardScaler на не очищенном датасете r2 = -0.013995

GradientBoostingRegressor MinMaxScaler на не очищенном датасете r2 = -0.040548

Продолжу работу с вышеперечисленными конфигурациями.

2.2. Разработка и обучение модели

Чтобы найти лучшую модель, воспользуюсь 2 методами RandomizedSearchCV для начального приближения параметров и GridSearchCV для уточнения параметров. Во избежание повторения кода случайный поиск гиперпараметров и визуализацию результатов поиска вынес в функцию rs\_viz.

Вот результат её работы на примере метода опорных векторов

Изображение выглядит как диаграмма

Автоматически созданное описание

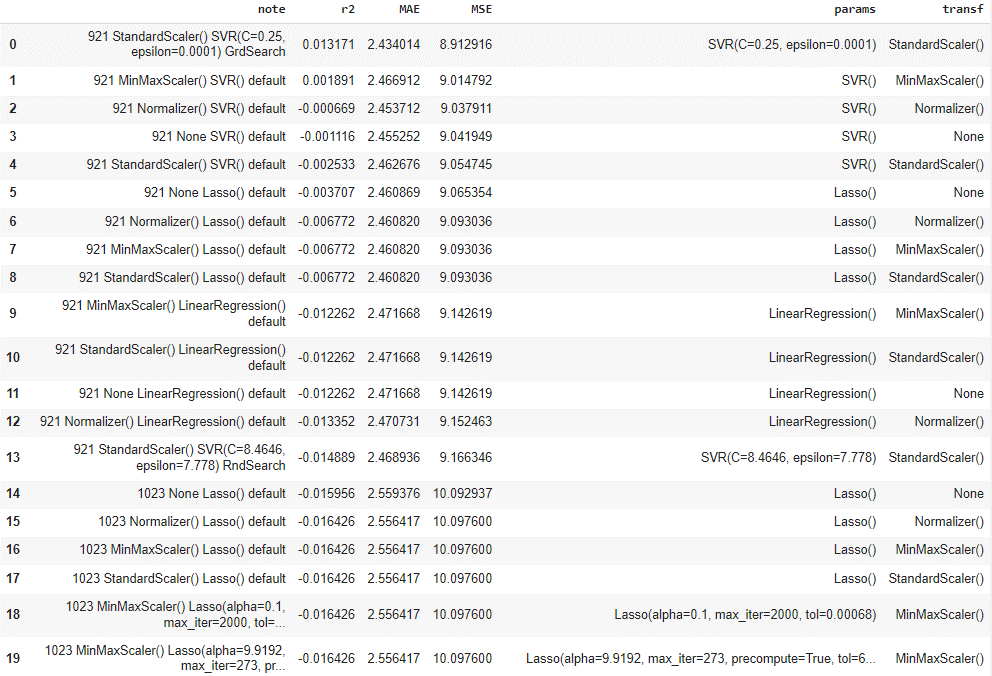
Видно, что оценка улучшается при приближении параметра «С» к нулю.

Во второй попытке можно выбрать диапазон значений для параметра «С» ближе к нулю и повторить процедуру. Следующим этапом подбираю параметры по сетке GridSearchCV, учитывая результаты предыдущего. Оценки помещаю в таблицу est\_df. Произведя эти действия для каждой модели, получаю для 'Прочность при растяжении, МПа' следующий результат:

Изображение выглядит как текст

Автоматически созданное описание

а для 'Модуль упругости при растяжении, ГПа' такой:



Сохраняю две лучшие модели для дальнейшего использования в приложении.

2.3. Тестирование модели

Для модели 'Прочность при растяжении, МПа' ошибки на тренировочной и тестирующей части выборки составили:

r2 на тестовой: 0.008335652932277449

r2 на тренировочной: 0.02040802593721114

Для модели 'Модуль упругости при растяжении, ГПа' ошибки на тренировочной и тестирующей части выборки составили:

r2 на тестовой: -0.029439681645599514

r2 на тренировочной: 0.07053092036691377

По сути, ни одна из моделей не смогла обучиться.

Результат стабильно отвратительный, но не надо отчаиваться.

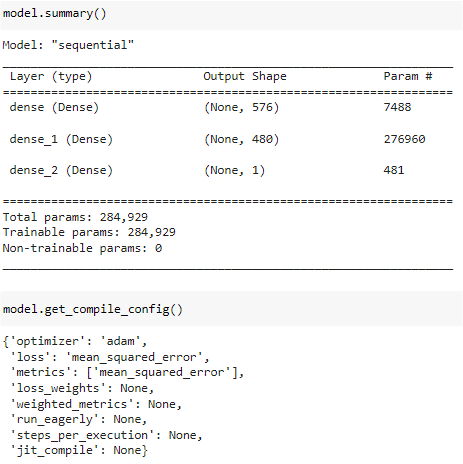
2.4. Написать нейронную сеть, которая будет рекомендовать соотношение матрица - наполнитель.

После экспериментов с построением сети вручную, решил воспользоваться инструментом kerastuner, который позволяет автоматически подбирать параметры нейросети, стремясь найти наилучшие. Для создания моделей реализована функция build\_model. По итогам работы kerastuner, была выбрана модель со следующими результатами на тестовой и тренировочной выборках:

Изображение выглядит как текст

Автоматически созданное описание

Вот её архитектура:



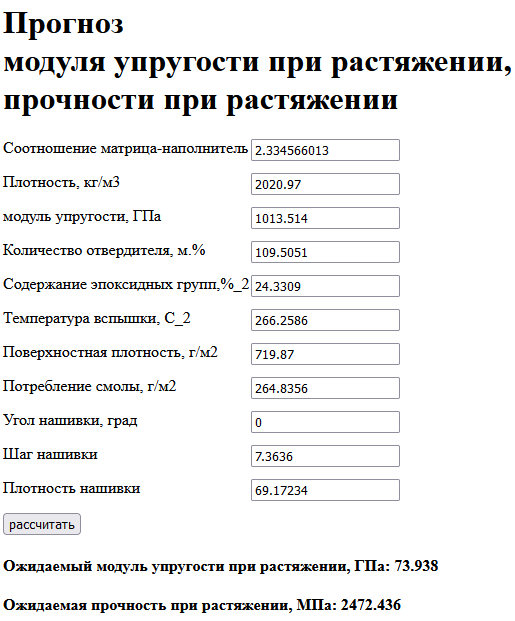
Модель сохранена в файл 'Соотношение матрица-наполнитель' и может быть использована для дальнейшего обучения или при разработке приложения.

2.5. Разработка приложения

Для разработки приложения composit.py использовал библиотеку Flask.

Flask — это небольшой и легкий веб-фреймворк, написанный на языке Python, предлагающий полезные инструменты и функции для облегчения процесса создания веб-приложений с использованием Python. Он обеспечивает гибкость и является более доступным фреймворком для новых разработчиков, так как позволяет создать веб-приложение быстро, используя только один файл Python. Flask — это расширяемая система, которая не обязывает использовать конкретную структуру директорий и не требует сложного шаблонного кода перед началом использования.

Приложение composit.py находится в папке app. Для запуска можно использовать сервер с поддержкой Flask или локальный компьютер с предварительно установленными зависимостями. Интерфейс приложения доступен в браузере по адресу hostname:5000/. Выглядит следующим образом:

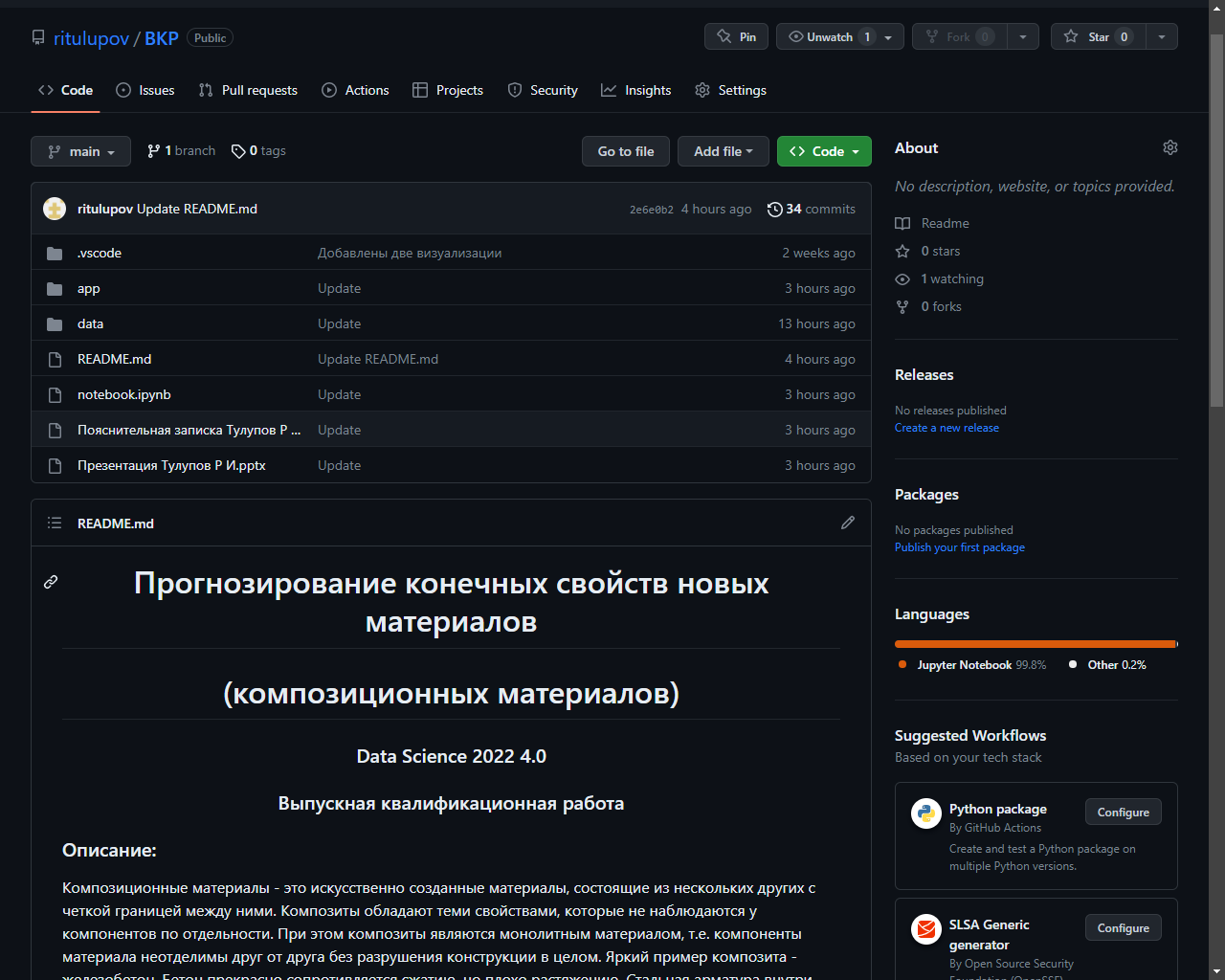


Для того, чтобы рассчитать модуль упругости при растяжении и прочность при растяжении необходимо заполнить все поля формы и нажать кнопку «рассчитать». Результат отобразится под кнопкой.

2.6. Создание удаленного репозитория и загрузка результатов работы на него.

Репозиторий приложения composit.py расположен по ссылке

<https://github.com/ritulupov/BKP>



[Commits on Apr 25, 2023](https://github.com/ritulupov/BKP/commits/main)

Заключение

В ходе проделанной работы стало ясно, что предложенный для анализа набор данных – крепкий орешек. Изначально распределение данных было близко к нормальному, выбросы не большие, пропусков и дубликатов нет. В процессе подготовки данные были очищены от выбросов, нормализованы. Для каждой из пяти моделей, выбранных для обучения, был проведен автоматический подбор подходящего набора данных (очищенные и не очищенные), нормализатора (MinMaxScaler, StandardScaler, Normalizer). Хочется отметить, что на линейную регрессию не повлияло применение MinMaxScaler и StandardScaler – результат оказался такой же, как и без них. Затем для каждой модели был проведен поиск оптимальных гиперпараметров рандомно и по сетке, но несмотря на все усилия результат оказался плачевным. Построение нейронной сети с автоматическим подбором гиперпараметров при помощи kerastuner, позволило выбрать лучшие параметры, но и с ними модель ничему не смогла научиться.

Судя по всему, в предоставленном для анализа датасете сокрыт какой-то секрет, который, к огромному сожалению, оказался мне не по силам. Но несмотря на это я получил хороший опыт и огромное удовольствие от проделанной работы.

Список литературы и веб ресурсы

1. Гафаров, Ф.М., Галимянов А.Ф. Искусственные нейронные сети и приложения: учеб. пособие /Ф.М. Гафаров, А.Ф. Галимянов. – Казань: Издательство Казанского университета, 2018
2. <https://machinelearningmastery.ru/bias-variance-and-regularization-in-linear-regression-lasso-ridge-and-elastic-net-8bf81991d0c5/>
3. Devpractice Team. Python. Визуализация данных. Matplotlib. Seaborn. Mayavi. - devpractice.ru
4. Andre Ye. 5 алгоритмов регрессии в машинном обучении, о которых вам сле-дует знать: <https://habr.com/ru/company/vk/blog/513842/>
5. Документация по библиотеке keras: [https://keras.io/api/.](https://keras.io/api/)
6. Документация по библиотеке matplotlib: <https://matplotlib.org/stable/users/index.html>.
7. Документация по библиотеке numpy: [https://numpy.org/doc/1.22/user/index.html#user](https://numpy.org/doc/1.22/user/index.html%23user).
8. Документация по библиотеке pandas: [https://pandas.pydata.org/docs/user\_guide/index.html#user-guide](https://pandas.pydata.org/docs/user_guide/index.html%23user-guide).
9. Документация по библиотеке scikit-learn: <https://scikit-learn.org/stable/user_guide.html>.
10. Документация по библиотеке seaborn: <https://seaborn.pydata.org/tutorial.html>.
11. Документация по библиотеке Tensorflow: <https://www.tensorflow.org/overview>
12. Иванов Д.А., Ситников А.И., Шляпин С.Д – Композиционные материалы: учебное пособие для вузов, 2019. 13 с.
13. Макс Ветков. Градиентный бустинг(AdaBoost[) https://your-scorpion.ru/gradient-boosting-adaboost]()%20https:/your-scorpion.ru/gradient-boosting-adaboost)
14. Руководство по быстрому старту в flask: – Режим доступа: <https://flaskrussian-docs.readthedocs.io/ru/latest/quickstart.html>
15. <https://github.com/ugapanyuk/ml_course_2022/>;
16. <https://wiki.loginom.ru/>;
17. <https://ai-news.ru/>;
18. <https://keras.io/api/keras_tuner/tuners/bayesian/>;