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 Григорьев Сергей Алексеевич

Москва, 2023

Содержание

Введение 3

1. Аналитическая часть
   1. Постановка задачи 5
   2. Описание используемых методов 7

1.3. Разведочный анализ данных 11

2. Практическая часть

2.1. Предобработка данных 12

2.2. Разработка и обучение модели 18

2.3. Тестирование модели 27

2.4. Написать нейронную сеть, которая будет рекомендовать

соотношение матрица 29

2.5. Разработка приложения 35

2.6. Создание удаленного репозитория и загрузка результатов

работы на него 38

Заключение 40

Список используемой литературы 41

# **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.

Композиционные материалы представляют собой многофазные системы, которые состоят из двух и более компонентов, сохраняющих индивидуальность (структуру и свойства) своего вещества в составе композита. Компонент, непрерывный в объеме композита, является матрицей или связующим элементом. Упрочняющие или армирующие элементы распределены в матрице в определенном порядке. Наполненные композиты содержат в матрице наполнители – дисперсные частицы неорганических и органических веществ, которые могут находиться в любой фазе. Разработка новых материалов имеет жизненно важное значение для решения насущных проблем в разных промышленных областях. Пространство гипотетических материалов, подлежащих рассмотрению, невероятно велико, и только небольшая часть возможных соединений может быть проверена экспериментально.

Уровень сложности разработки композитов очень высок из-за огромного количества компонентов и условий обработки, поскольку изменение одного фактора может иметь множество непредвиденных эффектов из-за их взаимосвязи. Традиционный подход к разработке новых материалов происходит в режиме «проб и ошибок», что приводит к проблемам низкой эффективности, высокой стоимости и неустойчивости полученных результатов.

Между тем, многочисленные экспериментальные и вычислительные испытания накапливают огромные объемы многомерных, сложных и недостаточно изученных данных, которые могут скрывать важные правила «структура-свойства». Машинное обучение может помочь выявить подобные связи, а также предоставить возможность тестировать и оптимизировать несколько показателей одновременно, тем самым ускоряя процесс обнаружения и оптимизации конструирования материалов, спрогнозировав состав с желаемыми характеристиками и уникальными свойствами.

Внедрение композиционных материалов обусловлено стремлением использовать их преимущества по сравнению с традиционно используемыми металлами и сплавами.

Для достижения определенных характеристик требуется большое количество различных комбинированных тестов, что делает насущной задачу прогнозирования успешного решения, снижающего затраты на разработку новых материалов и затраты на рабочую силу. 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. Работа основан на реальных производственных задачах Центра НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана)

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

В процессе исследовательской работы были разработаны несколько моделей, способные с высокой вероятностью прогнозировать модули упругости при растяжении и прочности при растяжении, а также была создана нейронная сеть, которая предлагает расчет соотношение «матрица - наполнитель» на основе вводимых данных.

Для визуализации работы модели предсказания соотношение «матрица - наполнитель» было создано пользовательское веб - приложение на фреймворке Flask.

1. **Аналитическая часть**
   1. **Постановка задачи**

Цель работы - разработать регрессионные модели для прогноза «Модуля упругости при растяжении» и «Прочности при растяжении» и написать нейронную сеть для прогнозирования «Соотношение матрица-наполнитель».

Полный текст задания по настоящей работе приведен в Приложении А.

Для решения поставленных задач были использованы методы машинного обучения.

* 1. **Описание используемых методов**

Машинное обучение ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) machine learning, ML) — класс методов [искусственного интеллекта](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82), характерной чертой которых является не прямое решение задачи, а обучение за счёт применения решений множества сходных задач. Для построения таких методов используются средства [математической статистики](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0), [численных методов](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B), [математического анализа](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7), [методов оптимизации](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), [теории вероятностей](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D0%BE%D1%8F%D1%82%D0%BD%D0%BE%D1%81%D1%82%D0%B5%D0%B9), [теории графов](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%BE%D0%B2), различные техники работы с [данными в цифровой форме](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5_(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)).

Постановка задачи по прогнозированию ряда конечных свойств получаемых композиционных материалов решается задачей регрессии.

Задача регрессии в машинном обучении — это предсказание одного параметра (Y) по известному параметру X, где X — набор параметров, характеризующий наблюдение.

Для решения задач были использованы следующие регрессионные методы машинного обучения:

1. lr = LinearRegression() - линейная регрессия

2. knr = KNeighborsRegressor() - метод ближайших соседей

3. rfr = RandomForestRegressor() - случайный лес

4. svr = SVR() - метод опорных векторов

5. mlpr = MLPRegressor()- многослойный перцептрон - обучение с 'учителем'

6. smlr = sm.OLS() - statsmodels.regression.linear\_model.OLS()статистическая линейная регрессия - регрессия методом наименьших квадратов

Для оценки работы моделей используется разнообразные метрики оценки.

- MAPE - Mean Absolute Percentage Error - Средняя абсолютная процентная ошибка средняя разница между прогнозируемым значением и фактическим значением.

- MAE - mean\_absolute\_error - Средняя абсолютная ошибка. Это среднее абсолютных разностей между фактическими целевыми значениями и прогнозами. Чем ниже MAE для данной модели, тем точнее модель способна предсказать фактические значения.

- MSE - mean\_squared\_error - Среднеквадратическая ошибка. Эта метрика сообщает среднеквадратичную разницу между прогнозируемыми значениями и фактическими значениями в наборе данных. Чем ниже MSE, тем лучше модель соответствует набору данных

- MaxER - max\_error - Максимальная ошибка

- RMSE - Root Mean Squared Error- Корень из среднеквадратичной ошибки. Эта метрика сообщает нам квадратный корень из средней квадратичной разницы между прогнозируемыми значениями и фактическими значениями в наборе данных. Эта метрика измеряется в тех же единицах, что и переменная ответа. Чем ниже RMSE, тем лучше модель соответствует набору данных.

- R2 - Коэффициент детерминации - функция оценки регрессии. Он показывает долю [дисперсии](https://wiki.loginom.ru/articles/variance.html) зависимой переменной, объяснённой с помощью регрессионной модели Если значение R2 равно 1, это означает, что модель идеальна, а если ее значение равно 0, это означает, что модель будет плохо работать с неизвестным набором данных.

- SCORES - Оценка кросс-валидации

- MCVS - Среднее значение оценки кросс-валидации

- StdDS - Статистическая ошибка оценки кросс-валидации

Преимущества и недостатки некоторых приведённых метрик.

| **Мера** | **Сильные стороны** | **Слабые стороны** |
| --- | --- | --- |
| MSE | Позволяет подчеркнуть большие отклонения, простота вычисления. | Имеет тенденцию занижать качество модели, чувствительна к выбросам. Сложность интерпретации из-за квадратичной зависимости. |
| RMSE | Простота интерпретации, поскольку измеряется в тех же единицах, что и целевая переменная. | Имеет тенденцию занижать качество модели, чувствительна к выбросам. |
| MAPE | Является безразмерной величиной, поэтому её интерпретация не зависит от предметной области. | Нельзя использовать для наблюдений, в которых значения выходной переменной равны нулю. |
| R-квадрат | Универсальность, простота интерпретации. | Возрастает даже при включении в модель бесполезных переменных. Плохо работает, когда входные переменные зависимы. |

Линейная регрессия

Линейная регрессия (LinearRegression) - модель зависимости переменной x от одной или нескольких других переменных (факторов, регрессоров, независимых переменных) с линейной функцией зависимости. Линейная регрессия относится к задаче определения «линии наилучшего соответствия» через набор точек данных и стала простым предшественником нелинейных методов, которые используют для обучения нейронных сетей.

Модель линейной регрессии является часто используемой и наиболее изученной в эконометрике. А именно изучены свойства оценок параметров, получаемых различными методами при предположениях о вероятностных характеристиках факторов, и случайных ошибок модели. Предельные (асимптотические) свойства оценок нелинейных моделей также выводятся исходя из аппроксимации последних линейными моделями. С эконометрической точки зрения более важное значение имеет линейность по параметрам, чем линейность по факторам модели.

Преимущества: хорошо изучены; очень быстрые, могут работать на очень больших выборках; практически вне конкуренции, когда признаков очень много (от сотен тысяч и более), и они разреженные (хотя есть еще факторизационные машины); коэффициенты перед признаками могут интерпретироваться (при условии, что признаки масштабированы; легко интерпретируем; имеет меньшую сложность по сравнению с другими алгоритмами.

Недостатки: плохо работают в задачах, в которых зависимость ответов от признаков сложная, нелинейная; моделирует только прямые линейные зависимости; требует прямую связь между зависимыми и независимыми переменными; выбросы оказывают огромное влияние, а границы линейны.

Линейная регрессия (Linear regression) — это алгоритм машинного обучения, основанный на контролируемом обучении, рассматривающий зависимость между одной входной и выходными переменными. Это один из самых простых и эффективных инструментов статистического моделирования. Она определяет зависимость переменных с помощью линии наилучшего соответствия. Модель регрессии создаёт несколько метрик. R-квадрат, или коэффициент детерминации, 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 процентов дисперсии данных. Оставшиеся отклонения не имеют объяснения. Чем ближе R-квадрат к единице, тем лучше.

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.

Метод ближайших соседей

Метод - К-ближайших соседей (kNN - k Nearest Neighbours) ищет ближайшие объекты с известными значения целевой переменной и основывается на хранении данных в памяти для сравнения с новыми элементами. Алгоритм находит расстояния между запросом и всеми примерами в данных, выбирая определенное количество примеров (k), наиболее близких к запросу, затем голосует за наиболее часто встречающуюся метку (в случае задачи классификации) или усредняет метки (в случае задачи регрессии).

Преимущества: прост в реализации и понимании полученных результатов; имеет низкую чувствительность к выбросам; не требует построения модели; допускает настройку нескольких параметров; позволяет делать дополнительные допущения; универсален; находит лучшее решение из возможных; решает задачи небольшой размерности.

Недостатки метода: замедляется с ростом объёма данных; не создаёт правил; не обобщает предыдущий опыт; основывается на всем массиве доступных исторических данных; невозможно сказать, на каком основании строятся ответы; сложно выбрать близость метрики; имеет высокую зависимость результатов классификации от выбранной метрики; полностью перебирает всю обучающую выборку при распознавании; имеет вычислительную трудоёмкость.

Случайный лес

Случайный лес (RandomForest) - алгоритм [машинного обучения](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), предложенный [Лео Брейманом](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B5%D0%B9%D0%BC%D0%B0%D0%BD,_%D0%9B%D0%B5%D0%BE) и [Адель Катлер](https://en.wikipedia.org/wiki/Adele_Cutler), заключающийся в использовании комитета (ансамбля) [решающих деревьев](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_%D0%BF%D1%80%D0%B8%D0%BD%D1%8F%D1%82%D0%B8%D1%8F_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B9). Алгоритм сочетает в себе две основные идеи: метод [бэггинга](https://ru.wikipedia.org/wiki/%D0%91%D1%8D%D0%B3%D0%B3%D0%B8%D0%BD%D0%B3" \o "Бэггинг) Бреймана, и [метод случайных подпространств](https://en.wikipedia.org/wiki/Random_subspace_method), предложенный [Тин Кам Хо](https://en.wikipedia.org/wiki/Tin_Kam_Ho). Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого [ансамбля](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D1%81%D0%B0%D0%BC%D0%B1%D0%BB%D1%8C_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%BE%D0%B2_(%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD)) [решающих деревьев](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_%D0%BF%D1%80%D0%B8%D0%BD%D1%8F%D1%82%D0%B8%D1%8F_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B9), каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.

Преимущества: способность эффективно обрабатывать данные с большим числом признаков и классов; нечувствительность к масштабированию (и вообще к любым монотонным преобразованиям) значений признаков; одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки; существуют методы построения деревьев по данным с пропущенными значениями признаков; cсуществуют методы оценивания значимости отдельных признаков в модели; внутренняя оценка способности модели к обобщению (тест по неотобранным образцам); высокая параллелизуемость и масштабируемость.

Недостатки: большой размер получающихся моделей; требуется большой объем памяти для хранения модели; построение занимает много времени; сложно интерпретируемый; не обладает возможностью экстраполяции; может недо обучаться; трудоёмко прогнозируемый; иногда работает хуже, чем линейные методы.

Преимущества: задача выпуклого квадратичного программирования хорошо изучена и имеет единственное решение; метод опорных векторов эквивалентен двухслойной нейронной сети, где число нейронов на скрытом слое определяется автоматически как число опорных векторов; принцип оптимальной разделяющей гиперплоскости приводит к максимизации ширины разделяющей полосы, а следовательно, к более уверенной классификации; при правильной работе модели, построенной на тестовом множестве, вполне возможно применение данного метода на реальных данных; достаточно небольшого набора данных.

Недостатки: неустойчивость к шуму: выбросы в исходных данных становятся опорными объектами-нарушителями и напрямую влияют на построение разделяющей гиперплоскости; не описаны общие методы построения ядер и спрямляющих пространств, наиболее подходящих для конкретной задачи; нет отбора признаков; необходимо подбирать константу C при помощи кросс-валидации; подходит только для решения задач с 2 классами; параметры модели сложно интерпретировать.

Многослойный персептрон

Многослойный персептрон (MLPRegressor) — это алгоритм обучения с учителем, который изучает функцию обучением на наборе данных. Это искусственная нейронная сеть, имеющая 3 или более слоёв персептронов. Эти слои - один входной слой, 1 или более скрытых слоёв и один выходной слой персептронов.

Преимущества: построение сложных разделяющих поверхностей; возможность осуществления любого отображения входных векторов в выходные; легко обобщает входные данные; не требует распределения входных векторов; изучает нелинейные модели.

Недостатки: имеет невыпуклую функцию потерь; разные инициализации случайных весов могут привести к разной точности проверки; требует настройки ряда гиперпараметров; чувствителен к масштабированию функций.

Статистическая линейная регрессия

Статистическая линейная регрессия sm.OLS() – statsmodels regression linear\_model - регрессия методом наименьших квадратов

Остатки (remains) — это то же самое, что и ошибка, т.е. отклонение предсказания от истинного значения.

R-squared — это коэффициент детерминации.

F-statistic - критерий Фишера проверки гипотезы о равенстве дисперсий.

В данном случае сравнивается модель с нулевыми параметрами (кроме интерсепта) с текущей.

Prob. (F-statistic) - вероятность того, что коэффициенты при всех переменных равны нулю.

Log-likelihood - насколько хорошо данные описываются моделью.

AIC BIC ак критерий для отбора модели, который определяет на сколько сильно модель переобучена.

Чем меньше значение - тем лучше модель.

Р>|т|. Отдельные p-значения говорят нам, является ли каждая предикторная переменная статистически значимой.

Intercept - все переменные обнуляются и подбирается константа, наилучшим образом описывающая данные. Значение У при Х=0

std err - дисперсия коэффициента по точкам данных.

t-stat - статистика t-Стьюдента, чем больше — тем лучше измерен коэффициент (чем меньше стандартное отклонение, тем лучше коэффициент описывает зависимость). Обычно t-критерий позволяет нам оценить важность различных предикторов, предполагая, что остатки модели нормально распределены около нуля. Если остатки не ведут себя таким образом, то это говорит о наличии некоторой нелинейности между переменными и о том, что их t-тесты не следует использовать для оценки важности отдельных предикторов.

Нейронная сеть

Нейронная сеть - математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. Первой такой попыткой были нейронные сети У. Маккалока и У. Питтса. После разработки алгоритмов обучения получаемые модели стали использовать в практических целях: в задачах прогнозирования, для распознавания образов, в задачах управления и др.

Нейронная сеть представляет собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Такие процессоры обычно довольно просты (особенно в сравнении с процессорами, используемыми в персональных компьютерах). Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. И, тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие по отдельности простые процессоры вместе способны выполнять довольно сложные задачи.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения — одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искажённых данных.

Входные данные передаются в эти нейроны в виде линейной комбинации со множеством переменных. Значение, умножаемое на каждую функциональную переменную, называется весом. Затем к этой линейной комбинации применяется нелинейность, что даёт нейронной сети возможность моделировать сложные нелинейные отношения.

Технически обучение заключается в нахождении коэффициентов связей между нейронами. У нейрона есть функция активации, которая определяет выходное значение нейрона. Она используется для того, чтобы ввести нелинейность в нейронную сеть.

У полносвязной нейросети выход каждого нейрона подается на вход всем нейронам следующего слоя. У нейросети имеется:

- входной слой — его размер соответствует входным параметрам;

- скрытые слои — их количество и размерность определяем специалист;

- выходной слой — его размер соответствует выходным параметрам.

Прямое распространение – это процесс передачи входных значений в нейронную сеть и получения выходных данных, которые называются прогнозируемым значением.

Прогнозируемое значение сравниваем с фактическим с помощью функции потери. В методе обратного распространения ошибки градиенты (производные значений ошибок) вычисляются по значениям весов в направлении, обратном прямому распространению сигналов. Значение градиента вычитают из значения веса, чтобы уменьшить значение ошибки. Таким образом происходит процесс обучения. Обновляются веса каждого соединения, чтобы функция потерь минимизировалась.

Нейронные сети применяются для решения задач регрессии, классификации, распознавания образов и речи, компьютерного зрения и других. На настоящий момент это самый мощный, гибкий и широко применяемый инструмент в машинном обучении.

Выбор параметров нейронной сети.

Количество слоев.

Элементы нейронной сети Входной слой: этот слой принимает входные функции. Он предоставляет информацию из внешнего мира в сеть, на этом уровне вычисления не выполняются, узлы здесь просто передают информацию (функции) скрытому слою.

Скрытый слой / слои — это слой не подвергается воздействию внешнего мира, он являются частью абстракции, предоставляемой любой нейронной сетью. Скрытый слой выполняет все виды вычислений над объектами, введенными через входной слой, и передает результат на выходной слой.

Выходной уровень: этот уровень выводит информацию, полученную сетью, во внешний мир.

В настоящее время нет никаких жестких правил ни для выбора количества скрытых слоев, ни для выбора количества нейронов в них. Хотя существуют ограничения, помогающие принимать решения. Критерии выбора количество слоев.

Если функция определена на конечном множестве точек, непрерывна и определена на компактной области, то 3-ехслойный перцептрон способен ее апроксимировать.

Остальные функции, которым могут быть обучены нейронные сети, могут быть апроксимированы 4-ехслойным перцетроном. Таким образом, теоретически максимальное количество слоев, которое необходимо – четыре, или два скрытых слоя.

Количество нейронов в скрытых слоях.

Слишком малое количество – и сеть не сможет обучиться. Слишком большое повлечет за собой увеличение времени обучения сети до фактически нереального значения. Также это может привести к переобученности сети (overfitting), проявляющейся в том, что сеть будет прекрасно работать на обучающей выборке, но очень плохо на входных примерах, не входящих в нее.

Это происходит из-за того, что сеть будет обладать избыточными способностями к обучению и наряду со значительными для данной задачи факторами будет учитывать черты, характерные лишь для данной обучающей выборки.

Однако, существуют эвристические правила выбора количества нейронов в скрытых слоях. Одним из таких правил является правило геометрической пирамиды (geometric pyramid rule).

Для 3-ех слойного перцептрона, по этому правилу, число нейронов скрытого слоя вычисляется по следующей формуле: k = sqrt(n\*m)  
где k – число нейронов в скрытом слое, n – число нейронов во входном слое; m – число нейронов в выходном слое.

Для 4-ехслойного перцептрона число нейронов вычисляется несколько сложнее: r = sqrt^3(n/m)( кубический корень) где к1= m \* r2 - число нейронов в первом скрытом слое; к2 = m \* r - число нейронов во втором скрытом слое.

Преимущества: высокая точность классификации; сильная способность параллельной распределенной обработки, сильная распределенная память и способность к обучению; он обладает высокой устойчивостью и отказоустойчивостью к шумовым нервам и может полностью аппроксимировать сложные нелинейные отношения; с функцией ассоциативной памяти.

Недостатки: нейронные сети требуют большого количества параметров, таких как топология сети, начальные значения весов и пороговых значений; невозможно наблюдать процесс обучения между ними, а полученные результаты трудно объяснить, что повлияет на достоверность и приемлемость результатов; время обучения слишком велико и может даже не достичь цели обучения.

# 

# 1.3 Описание используемых библиотек

Подготовка среды для проекта ENV.

При начале работы над проектом необходимо было установить несколько пакетов библиотек Python таких как tensorflow, seaborn и др. Выяснилось, что невозможно установить пакеты через PIP install, так как win 10 не воспринимает путь к файлу длиннее 260 символов. Пришлось написать программу по изменению Политики доступа для win 10 с расширением .bat. Однако это тоже не дало результатов. Пришлось с помощью изменения regedit вносить изменения в ключ Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem, что привело к возможности записывать блинные пути к файлу.

Задачи решались с использованием следующих библиотек Python:

* 1. NumPy. Общедоступный модуль для Python, который предоставляет общие математические и числовые операции в виде пре-скомпилированных, быстрых функций, обеспечивает функционал, который можно сравнить с функционалом MatLab. NumPy (Numeric Python) предоставляет базовые методы для манипуляции с большими массивами и матрицами.
  2. Matplotlib. Библиотека на языке программирования [Python](https://ru.wikipedia.org/wiki/Python) для визуализации данных [двумерной (2D) графикой](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0#%D0%94%D0%B2%D1%83%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0_(2D)) ([3D графика](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0#%D0%A2%D1%80%D1%91%D1%85%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0) также поддерживается). Получаемые изображения могут быть использованы в качестве иллюстраций в публикациях.
  3. [Pandas](http://pandas.pydata.org/). Данная библиотека делает Python мощным инструментом для анализа данных. Пакет дает возможность строить сводные таблицы, выполнять группировки, предоставляет удобный доступ к табличным данным.
  4. Scikit-learn. Один из наиболее широко используемых пакетов Python для Data Science и Machine Learning. Он содержит функции и алгоритмы для машинного обучения: классификации, прогнозирования или разбивки данных на группы.
  5. [Seaborn](https://seaborn.pydata.org/). Библиотека для создания статистических графиков на Python. Она основывается на [matplotlib](https://matplotlib.org/) и тесно взаимодействует со [структурами данных pandas](https://livecodestream.dev/post/how-to-work-with-pandas-in-python/).
  6. TensorFlow. Открытая [программная библиотека](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) для [машинного обучения](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), разработанная компанией [Google](https://ru.wikipedia.org/wiki/Google_(%D0%BA%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F)) для решения задач построения и тренировки [нейронной сети](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия.
  7. Math является самым базовым математическим модулем Python. Охватывает основные математические операции, такие как сумма, экспонента, модуль и так далее. Эта библиотека не используется при работе со сложными математическими операциями, такими как умножение матриц.
  8. Pickle реализует мощный алгоритм сериализации и десериализации объектов Python. Широко применяется для сохранения и загрузки сложных объектов в Python как поток байтов.

# 1.4. Методы разведочного анализа данных

Цель разведочного анализа - получение первоначальных представлений о характерах распределений переменных исходного набора данных, формирование оценки качества исходных данных (наличие пропусков, выбросов), выявление характера взаимосвязи между переменными с целью последующего выдвижения гипотез о наиболее подходящих для решения задачи моделях машинного обучения.

В качестве инструментов разведочного анализа используется: оценка статистических характеристик датасета; гистограммы распределения каждой из переменной (несколько различных вариантов); диаграммы ящика с усами (несколько интерактивных вариантов); попарные графики рассеяния точек (несколько вариантов); тепловая карта (несколько вариантов); описательная статистика для каждой переменной; анализ и полное исключение выбросов (5 повторных итераций); проверка наличия пропусков и дубликатов; корреляционный анализ для установления статистических зависимостей между переменными.

Корреляционный анализ - статистический метод изучения взаимосвязи между двумя и более случайными величинами. В качестве случайных величин в эмпирических исследованиях выступают значения переменных, измеряемые свойства исследуемых объектов наблюдения. Суть корреляционного анализа заключается в расчете коэффициентов корреляции. Коэффициенты корреляции могут принимать, как правило, положительные и отрицательные значения. Знак коэффициента корреляции позволяет интерпретировать направление связи, а абсолютное значение – силу связи.

Способ расчета коэффициентов корреляции зависит от шкал измерения переменных, между которыми исследуется взаимосвязь. Для переменных, измеряемых в количественной шкале (интервальной шкале или шкале отношений), рассчитывают ковариацию или корреляционный момент, а на его основе линейный коэффициент корреляции (коэффициент корреляции Пирсона).

Для оценки силы и направления связи между переменными, измеренными в порядковой шкале, используются непараметрические ранговые коэффициенты корреляции.

Прежде чем передать данные в работу моделей машинного обучения, необходимо обработать и очистить их. Очевидно, что «грязные» и необработанные данные могут содержать искажения и пропущенные значения – это ненадёжно, поскольку способно привести к крайне неверным результатам по итогам моделирования. Но безосновательно удалять что-либо тоже неправильно. Именно поэтому сначала набор данных надо изучить.

После обнаружения выбросов данные, значительно отличающиеся от выборки, будут полностью удалены. Для расчёта этих данных мы будем использовать методы трех сигм и межквартильного расстояния.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**



я исследовательской работы были даны 2 файла: X\_bp.xlsx (с данными о параметрах базальтопластика, состоящий из 1023 строки и 11 столбцов) и X\_nup.xlsx (данными нашивок углепластика, состоящий из 1040 строк и 4 столбцов).

Рисунок 1 - Пример начала работы с файлом X\_bp.xlsx

Рисунок 2 - Пример начала работы с файлом X\_nup.xlsx

Объединение делаем по индексу, тип объединения INNER.

Рисунок 3 – Объединение файлов

**1.3. Разведочный анализ данных**

Прежде чем передать данные в работу моделей машинного обучения, необходимо обработать и очистить их. Очевидно, что «грязные» и необработанные данные могут содержать искажения и пропущенные значения – это ненадёжно, поскольку способно привести к крайне неверным результатам по итогам моделирования. Но безосновательно удалять что-либо тоже неправильно. Именно поэтому сначала набор данных надо изучить.

Цель разведочного анализа - получение первоначальных представлений о характерах распределений переменных исходного набора данных, формирование оценки качества исходных данных (наличие пропусков, выбросов), выявление характера взаимосвязи между переменными с целью последующего выдвижения гипотез о наиболее подходящих для решения задачи моделях машинного обучения.

В качестве инструментов разведочного анализа используется: оценка статистических характеристик датасета; гистограммы распределения каждой из переменной (несколько различных вариантов); диаграммы ящика с усами (несколько интерактивных вариантов); попарные графики рассеяния точек (несколько вариантов); график «квантиль-квантиль»; тепловая карта (несколько вариантов); описательная статистика для каждой переменной; анализ и полное исключение выбросов; проверка наличия пропусков и дубликатов.

После обнаружения выбросов данные, значительно отличающиеся от выборки, будут полностью удалены. Для расчёта этих данных мы будем использовать методы трех сигм и межквартильного расстояния.

**2. Практическая часть**

**2.1. Предобработка данных**

Необходимо провести разведочный анализ предложенных данных. Нарисуем гистограммы распределения каждой из переменных, диаграммы ящика с усами, попарные графики рассеяния точек.

Рисунок 4 – Гистограммы распределения переменных.

Ящики с усами

Построив тепловую карту, видим, что между параметрами очень слабая корреляция. Чем ближе корреляция к 1 или -1, тем она выше, значит, есть влияние и зависимость одних параметров и других. В нашем случае самая высокая корреляция 0.11 обнаруживается между углом нашивки и плотностью нашивки.

Рисунок 5 – Тепловая карта корреляции

Построим попарные графики рассеяния точек. Распределение каждой переменной показано в виде гистограммы в прямоугольниках по диагонали. Во всех остальных полях отображается диаграмма рассеяния отношений между каждой парной комбинацией переменных. На данном графике также видим отсутствие четкой зависимости между переменными.

Рисунок 6 – Попарные графики рассеяния точек

Необходимо также для каждой колонки получить среднее, медианное значение. Видим, что сильных отклонений не наблюдается. Значит, выбросы практически не влияют на данные.

Рисунок 7 – Среднее и медианное значения для каждой колонки

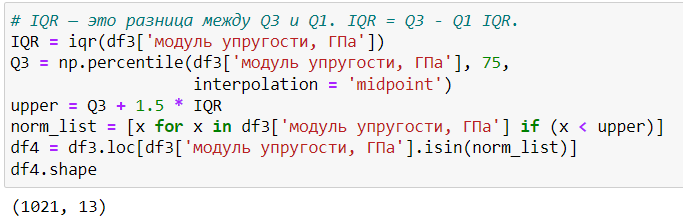
Произведем предобработку данных: удалим выбросы, сделаем нормализацию. Выбросы в данных - это значения, которые значительно отличаются от закономерностей и трендов других значений.

Рисунок 8 – Удаление выбросов

Рисунок 9 – Результаты удаления выбросов

Удалили выбросы из четырех колонок, где они были особенно заметны. В нашем датасете после удаления выбросов осталось 990 строк. Этого достаточно для дальнейшей работы. С выбросами мы удалили всего около 3 процентов данных (33 строки).

В датасете есть значения, которые исчисляются в единицах, а есть значения, исчисляемые в тысячах. Чтобы с ними было удобно работать, нужно провести нормализацию — привести различные данные в самых разных единицах измерения и диапазонах значений к единому виду, который позволит сравнивать их между собой или использовать для расчёта схожести объектов.

Рисунок 10 – Пример нормализации данных

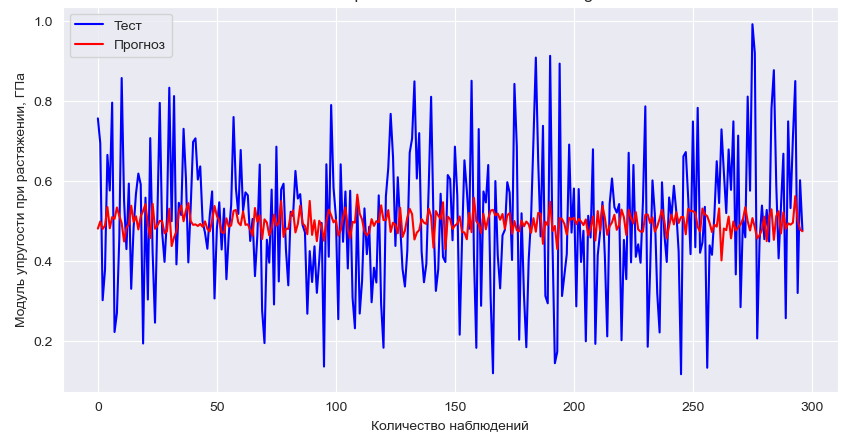
Рисунок 11 – Данные после нормализации

**2.2. Разработка и обучение модели**

Необходимо обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении. При построении модели необходимо 30 процентов данных оставить на тестирование модели, на остальных происходит обучение моделей

Линейная регрессия (Linear regression) — это алгоритм машинного обучения, основанный на контролируемом обучении, рассматривающий зависимость между одной входной и выходными переменными.

Рисунок 12 – Построение модели Линейной регрессии

Рисунок 13 – Тестовые и прогнозные значения LinearRegression

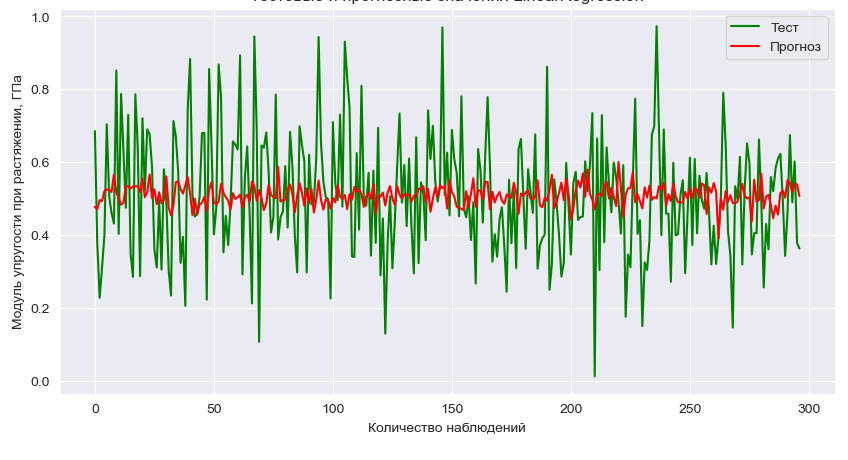
для Модуля упругости при растяжении

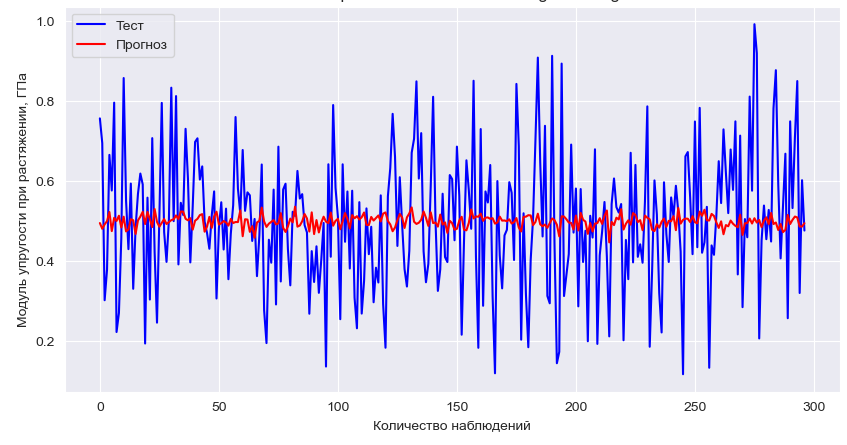
Рисунок 14 – Тестовые и прогнозные значения LinearRegression

для Прочности при растяжении

нии

Метод ближайших соседей - К-ближайших соседей (kNN - k Nearest Neighbours) ищет ближайшие объекты с известными значения целевой переменной и основывается на хранении данных в памяти для сравнения с новыми элементами.

Рисунок 18 – Построение модели KNeighborsRegressor

Рисунок 19 – Тестовые и прогнозные значения KNeighborsRegressor

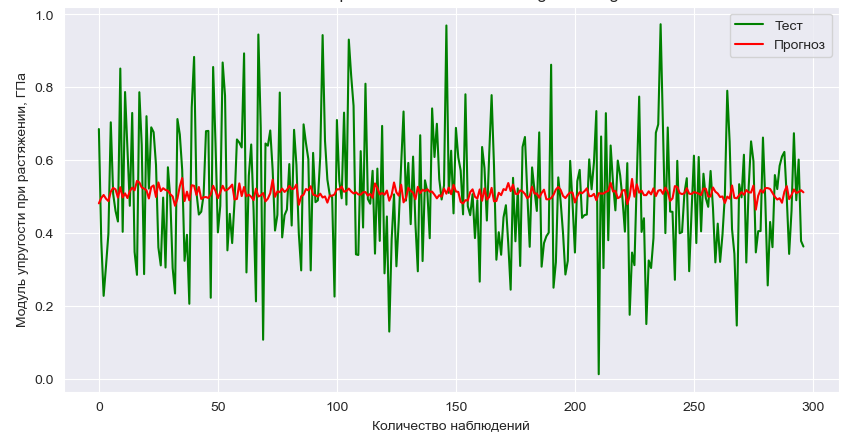
для Модуля упругости при растяжении

Рисунок 20 – Тестовые и прогнозные значения KNeighborsRegressor

для Прочности при растяжении

Случайный лес (RandomForest) — это множество решающих деревьев. Универсальный алгоритм машинного обучения с учителем, представитель ансамблевых методов.  Если точность дерева решений оказалось недостаточной, мы можем множество моделей собрать в коллектив.

Рисунок 21 – Построение модели RandomForestRegressor

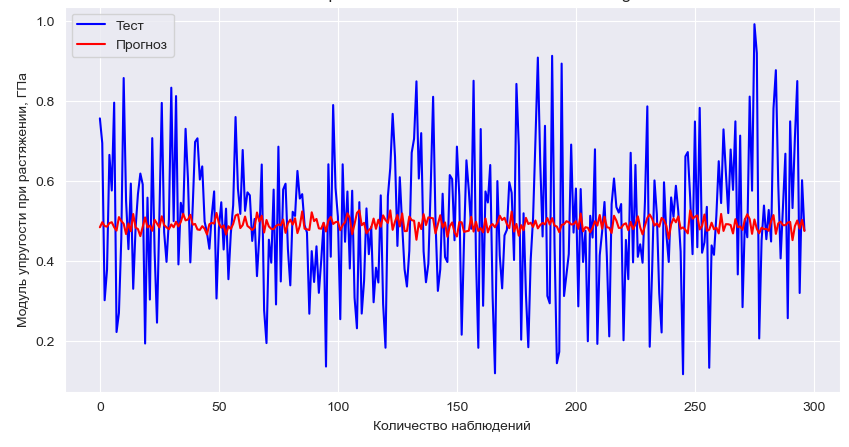


Рисунок 22 – Тестовые и прогнозные значения RandomForestRegressor

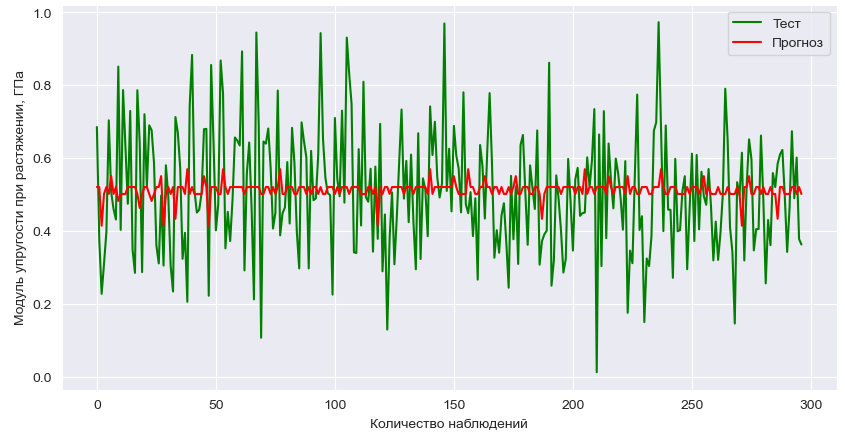
для Модуля упругости при растяжении

Рисунок 23 – Тестовые и прогнозные значения RandomForestRegressor

для Прочности при растяжении

**2.3. Тестирование модели**

После обучения моделей была проведена оценка точности этих моделей на обучающей и тестовых выборках.

В качестве метрик для оценки качества моделей использовались: Mean Absolute Error (MAE), Mean Squared Error (MSE) score и оценка R2.

Mean Absolute Error (MAE): метрика измеряет среднюю сумму абсолютной разницы между фактическим значением и прогнозируемым значением. Чем ниже MAE для данной модели, тем точнее модель способна предсказать фактические значения.

Mean Squared Error (MSE): измеряет среднюю сумму квадратной разности между фактическим значением и прогнозируемым значением для всех точек данных. Выполняется возведение во вторую степень, поэтому отрицательные значения не компенсируют положительными. Чем ниже значение MSE, тем лучше модель способна точно предсказывать значения.

Оценка R2: один из показателей оценки эффективности моделей машинного обучения на основе регрессии. Также метрика известна как коэффициент детерминации.

Таблица 1. Результаты построения и обучения моделей для Модуля упругости при растяжении

Для предсказания Модуля упругости при растяжении лучший результат показала модель KNeighborsRegressor: MAE и MSE минимальны, а R2 максимален из всех полученных результатов. В целом, все модели плохо справились с задачей.

Таблица 2. Результаты построения и обучения моделей для Прочности при растяжении

Для предсказания Прочности при растяжении лучший результат также показала модель KNeighborsRegressor: MSE минимально, а R2 максимален из всех полученных результатов. В целом, все модели плохо справились с задачей.

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

Обучение нейронной сети — это такой процесс, при котором происходит подбор оптимальных параметров модели, с точки зрения минимизации функционала ошибки.

Рисунок 27 – 1 вариант нейронной сети

Рисунок 28 – График потерь первой модели

активационной функцией Sigmoid и оптимайзером Adam. Хотя после 20-й эпохи сеть начала переобучаться, уменьшение количества эпох не улучшило результатов.

**2.5. Разработка приложения**

Разработано веб-приложение для рекомендательной системы «Соотношение матрица-наполнитель». Приложение разработано в среде разработки VSCode. Для разработки приложения был использован веб-фреймворк Flask. На рисунке \_\_\_\_\_ показан интерфейс приложения. Пользователь вводит 12 параметров в реальных физических значениях в соответствующие окна ввода, с учетом рекомендованных интервалов значений, и при нажатии кнопки «Рассчитать значение» Приложение выводит ниже расчетное значение показателя «Соотношение матрица-наполнитель».



Рисунок 37 - Пример результата работы приложения

Приложение успешно работает на локальном хосте и показывает результат прогноза для параметра «соотношения матрица – наполнитель».

Данное приложение — это основной файл Flask app.py, расположен в папке Diploma\_block5\_flask. Код основного файла app.py Приложения приведен на рис. \_\_

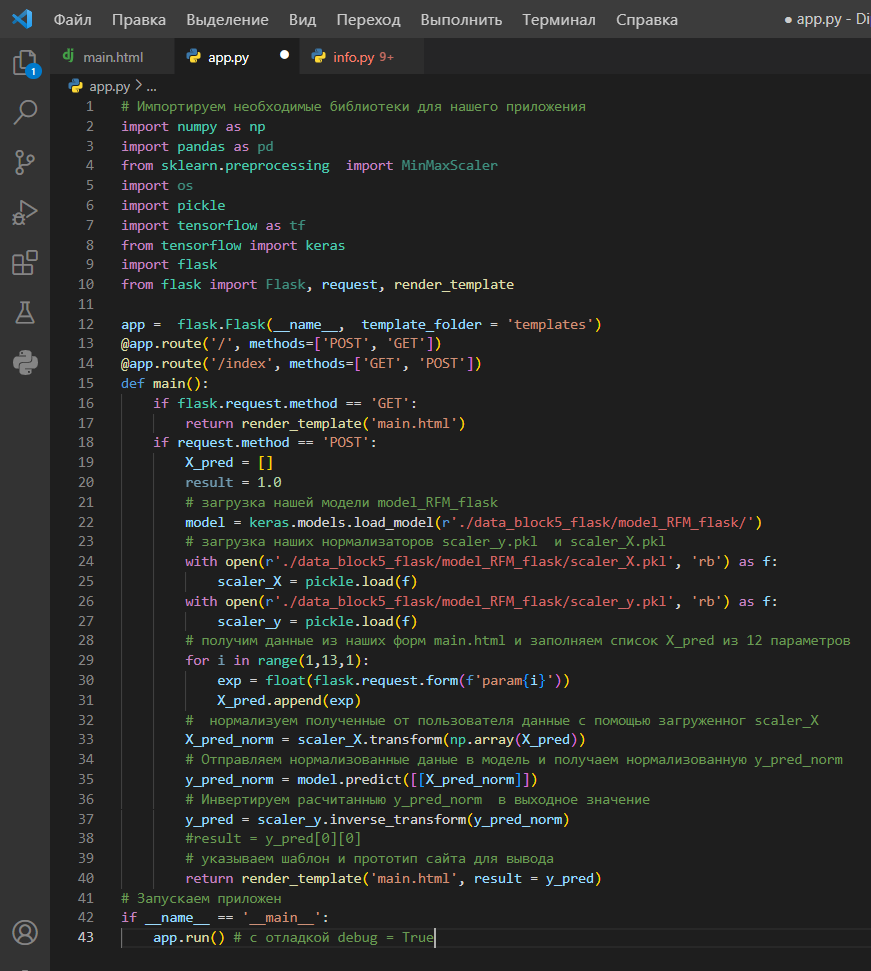


Рисунок \_\_ - Код основного файла app.py Приложения

Остальные файлы проекта flask расположены в папке Diploma\_block5\_flask, в папке templates находится файл main.html с шаблоном страницы и в папке data\_block5\_flask папка model\_RFM\_flask c сохранённой моделью для данных и с файлами scaler\_X.pkl и scaler\_y.pkl – нормализаторами для передачи физических данных в сохраненную модель в нормализованном виде и получения предсказанного значения параметра Соотношение матрица-наполнитель обратно из нормализованного значения в физически значимое и ввод его в Результаты расчета.

При запуске приложения пользователь переходит на страницу сайта Приложения , расположенную на локальном хосте: <http://127.0.0.1:5000/>. Код 200 свидетельствует о нормальной работе приложения.

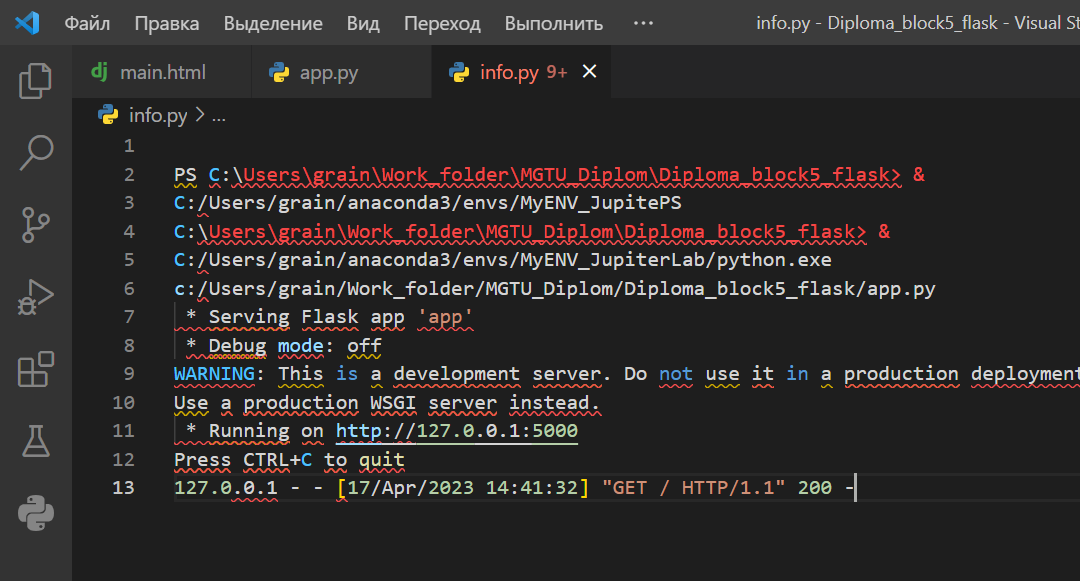


Рисунок 39 - Ссылка для открытия страницы на локальном сервере

**2.6. Создание удалённого репозитория и загрузка**

**результатов работы на него**

Репозиторий был создан на github.com по адресу: <https://github.com/Grain1963/MHTS_Diploma>

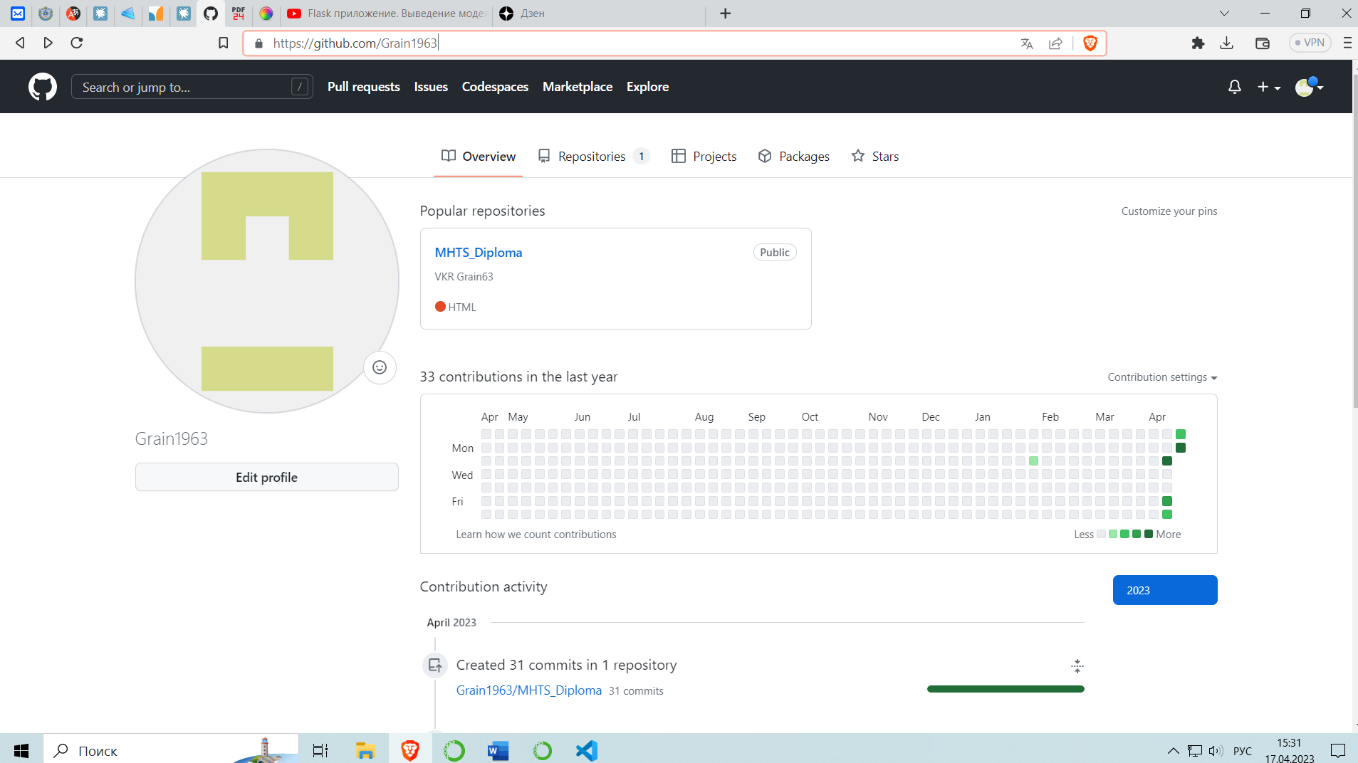


Рисунок \_ - Часть страницы на github.com

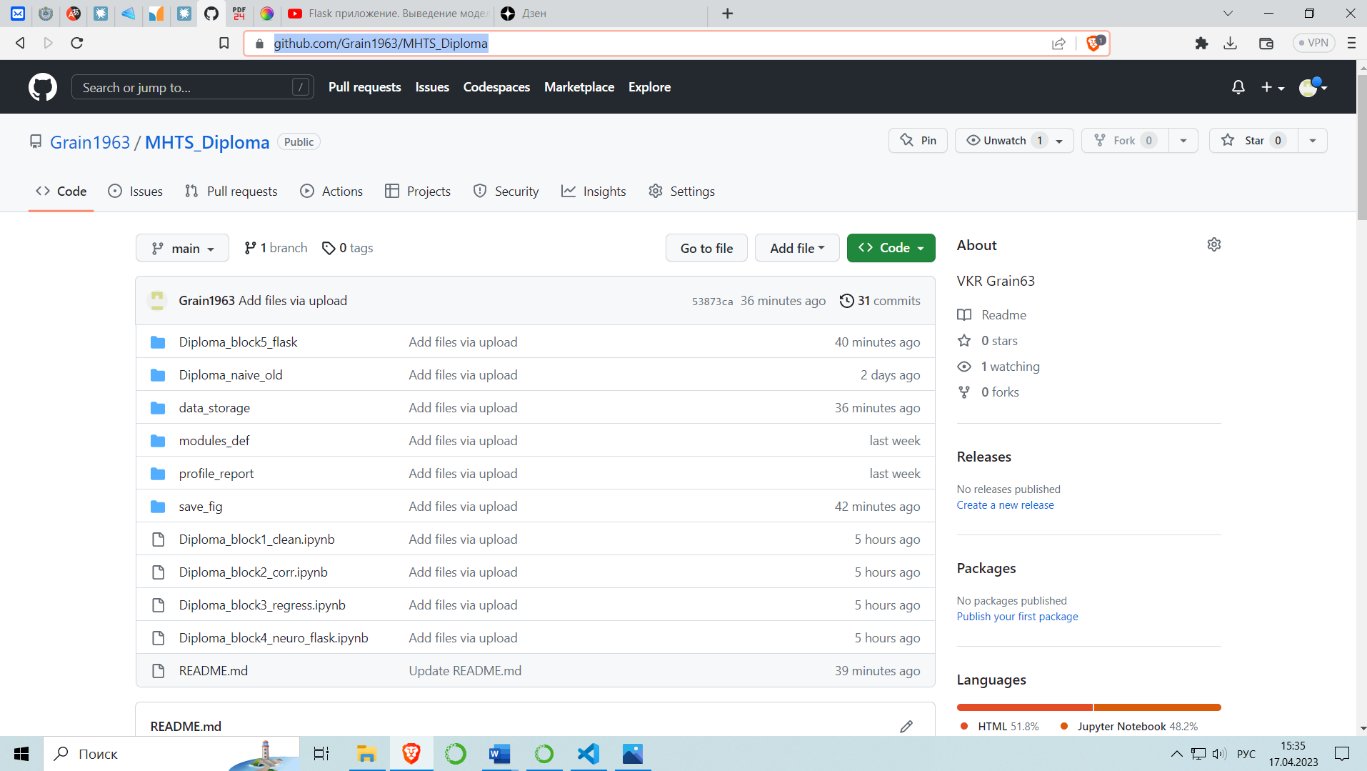


Рисунок \_ - Структура репозитария на github.com

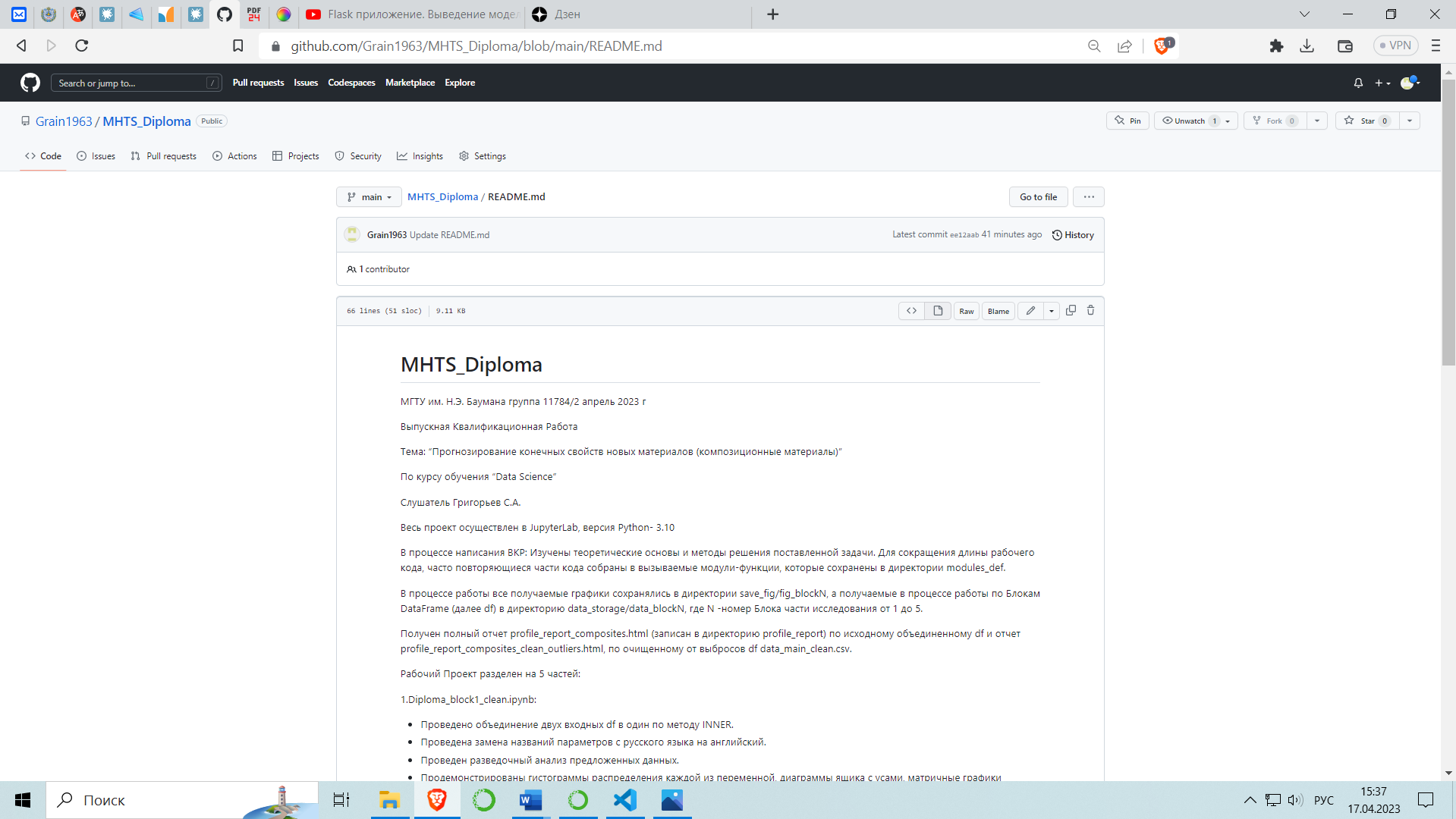


Рисунок 41 - Часть созданного описательного файла README

**Заключение**

В ходе выполнения данной работы были изучены теоретические основы методов машинного обучения, изучены основные библиотеки Python, как одного из основных инструментов для работы аналитика данных.

На основании практической задачи были изучены и реализованы:

- разведочный анализ данных;

- предобработка данных;

- построение регрессионных моделей и модели нейронной сети;

- визуализация модели и оценка качества прогноза;

- разработка и тестирование веб-приложения.

Данная исследовательская работа позволяет сделать некоторые основные выводы по теме. Распределение полученных данных в объединённом датасете близко к нормальному, но коэффициенты корреляции между парами признаков стремятся к нулю.

Результаты построения и обучения моделей не дали положительного результата. Возможные причины неудовлетворительной работы моделей:

- нечеткая постановка задачи, отсутствие дополнительной информации о зависимости между входными датасетами, что привело к неверному объединению датасетов;

- исследование проводилось на предварительно обработанных датасетах.

- возможно, на исходных датасетах можно было бы получить более качественные регрессионные модели;

- недостаток знаний и опыта, были испробованы не все возможные методы прогноза.

Был сделан вывод, что невозможно определить из свойств материалов соотношение «матрица – наполнитель». Данный факт не указывает на то, что прогнозирование характеристик композитных материалов на основании предоставленного набора данных невозможно, но может указывать на недостатки базы данных, подходов, использованных при прогнозе, необходимости пересмотра инструментов для прогнозирования.

Необходимы дополнительные вводные данные, получение новых результирующих признаков в результате математических преобразований, релевантных доменной области, консультации экспертов предметной области, новые исследования, работа эффективной команды, состоящей из различных учёных.

Учитывая отсутствие корреляции между признаками, делаем вывод, что текущим набором алгоритмов задача не решается, возможно, решается трудно или не решается совсем.

Разработанные модели при определенных условиях и дальнейшей доработки могут быть использованы для переобучения и решения аналогичных задач на новых входных данных.

В процессе написания ВКР:

Для сокращения длины рабочего кода, часто повторяющиеся части кода собраны в вызываемые модули-функции, которые сохранены в директории modules\_def.

В процессе работы все получаемые графики сохранялись в директории save\_fig/fig\_blockN, а получаемые в процессе работы по Блокам DataFrame (далее df) в директорию data\_storage/data\_blockN, где N -номер Блока части исследования от 1 до 5.

Получены полные отчеты по двум датасетам - по исходному объединенному дата сету и по очищенному от выбросов, причем это сделано двумя разными методами: ProfileReport и SweetViz. Отчеты сформированы и представлены в директории profile\_report в исходных html и PDF форматах

Рабочий Проект разделен на 5 частей:

1.Diploma\_block1\_clean.ipynb:

Проведено объединение двух входных df в один по методу INNER.

Проведена замена названий параметров с русского языка на английский.

Проведен разведочный анализ предложенных данных.

Продемонстрированы гистограммы распределения каждой из переменной, диаграммы ящика с усами, матричные графики рассеяния точек.

Для каждого параметра получены среднее, медианное значение, проверено наличие пропусков, проведен итерационный анализ и исключение выбросов несколькими методами.

Осуществлена замена “подозрительных значений” на средние значения ближайших 5 соседей.

По результатам работы в Блоке № 1 для дальнейшего исследования создан очищенный от выбросов df data\_main\_clean.csv, который записан в папку data\_block1\_clean, которая входит в директорию верхнего уровня data\_storage.

Все полученные графики сохранены в директории save\_fig/ fig\_block1\_clean.

Diploma\_block2\_corr.ipynb:

Проведено исследование корреляционных матриц очищенного от выбросов и подозрительных значений df в зависимости от количества значений в выборке.

-Сделан вывод, что при рассмотрении полного df корреляция отсутствует практически полностью, однако при значениях N = 100 и 200 еще наблюдается некоторая корреляционная зависимость между параметрами, с увеличением числа выборки зависимость падает до около нулевых значений.

Проведен графический анализ использованием регрессионной прямой по каждому параметру с целевыми переменными “Модуля упругости при растяжении” и “Прочности при растяжении”.

На основании теоретических исследований, приведенных в научной литературе по данной теме введен новый параметр ALFA как функция Модуля упругости при растяжении и Прочности при растяжении.

Произведен выбор фитчей для наших целевых параметров, что позволит сократить вычислительные ресурсы и увеличить точность расчетов.

Подготовлен df data\_Train\_ML.csv из 200 случайных значений (random\_state=42) в качестве обучающего массива для регрессионного анализ в оке 3 и df data\_Test\_ML.csv для использования в качестве тестового набора данных (random\_state=50). Полученные df записаны в соответствующую директорию data\_storage/data\_block2\_corr.

Полученные в процессе работы Блока 2 графики записаны в соответствующую директорию save\_fig/fig\_block2\_corr

Diploma\_block3\_regress.ipynb:

Проведена предобработка данных, нормализация и стандартизация.

Обучены 6 типов моделей для прогноза модуля упругости при растяжении и прочности при растяжении.

Получены и сведены в 2 df: df\_error\_calc\_EMS.csv и df\_error\_calc\_SS.csv значения по 7 оценочным параметрам работы разных моделей регрессии и записаны в директорию data\_storage/data\_block3\_regress

Сделан вывод о практическом отсутствии регрессии между параметрами и целевыми параметрами.

Полученные в процессе работы Блока 3 графики записаны в соответствующую директорию save\_fig/fig\_block3\_regress.

Diploma\_block4\_neuro.ipynb:

Для работы с нейросетью был применен полный очищенный от выбросов df df data\_main\_clean.csv, который получен в Блоке №1.

Проведен обоснованный расчет количества слоев нейронной сети, количества нейронов в каждом слое и выбор активационных функции для каждого слоя.

Написана нейронная сеть, которая будет рекомендовать соотношение целевого параметра “Матрица-наполнитель”.

Оценены точности моделей на тренировочном и тестовом датасете.

Полученная и обученная модел нейронной сети была сохранена в директории data\_storage/data\_block5\_neuro/model\_RFM\_flask.

Полученные в процессе работы Блока 4 графики записаны в соответствующую директорию save\_fig/fig\_block4\_neuro.

Diploma\_block5\_flask (папка):

Блок создания приложения в модуле FLASK по прогнозированию целевого параметра Соотношение матрица-наполнитель.

Создано приложение app.py и запущено в работу. Результаты отработанного приложения приведены в конце файла. процессе работы приложения Пользователь заносит рабочие значения параметров, а Приложение сначала нормализует данные, затем передает их в сохраненную модель, а затем инвертирует расчетные значения в обычную размерность параметра “Соотношение матрица-наполнитель”.

Создан шаблон сайта - файл main.html в папке templates

В папке templates расположен снимок с экрана работающего Приложения на рабочем локальном хосте [http://127.0.0.1:5000](http://127.0.0.1:5000/)

По итогам работ над Проектом был создан репозиторий на web ресурсе GitHub, в котором выложены все материалы. Ссылка на репозиторий: <https://github.com/Grain1963/MHTS_Diploma>.

**Список используемой литературы и веб ресурсы**

1. Артеменко, С.Е. Структура и свойства базальто-, стекло- и углепластиков, сформированных по интеркаляционной технологии / С.Е. Артеменко, О.Г. Васильева, Ю.А. Кадыкова, А.Н. Леонтьев // Полимеры-2004: III Всерос. Каргинская конф. М., 2004. Т. 2. С. 162.
2. Джулли, П.: Библиотека Keras - инструмент глубокого обучения / пер. с англ. А.А. Слинкин. – ДМК Пресс, 2017. – 249 с.
3. Грас, Джоэл. Data Science. Наука о данных с нуля: пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербурr, 2021. - 416 с.
4. 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.
5. Кадыкова, Ю.А. Полимерные композиционные материалы на основе волокон различной химической природы / Ю.А. Кадыкова, А.Н. Леонтьев, О.Г. Васильева, С.Е. Артеменко // Строительные материалы, оборудование, технологии ХХ1 века. 2002. № 6. С. 10-11.
6. Николенко, С., Кадурин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. - СПб.: Питер. - 2020. - 480 с. ISBN: 978-5-4461-1537-2.
7. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2007. - 1408 с.
8. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. - М.: Горячая Линия - Телеком. - 2013. - 384 с. ISBN: 978-5-9912-0320-3.
9. Статистическая обработка данных, планирование эксперимента и случайные процессы : учебное пособие для вузов / Берикашвили В. Ш., Оськин С. П. - 2-е изд., испр. и доп. - М.: Юрайт, 2021. - 163 с.
10. Фостер Д. Генеративное глубокое обучение. Творческий потенциал нейронных сетей. - СПб.: Питер. - 2020. - 336 с. - ISBN: 978-5-4461-1566-2.
11. Документация по библиотеке keras: – Режим доступа: <https://keras.io/api/>.(дата обращения: 18.03.2023).
12. Документация по библиотеке matplotlib: – Режим доступа: <https://matplotlib.org/stable/users/index.html>. (дата обращения: 20.12.2022)
13. Документация по библиотеке numpy: – Режим доступа: <https://numpy.org/doc/1.22/user/index.html#user>. (дата обращения: 23.12.2022).
14. Документация по библиотеке pandas: – Режим доступа: <https://pandas.pydata.org/docs/user_guide/index.html#user-guide>. (дата обращения: 04.12.2022).
15. Документация по библиотеке scikit-learn: – Режим доступа: <https://scikit-learn.org/stable/user_guide.html>. (дата обращения: 05.01.2023).
16. Документация по библиотеке seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>. (дата обращения: 16.01.2023).
17. Документация по библиотеке Tensorflow: – Режим доступа: <https://www.tensorflow.org/overview> (дата обращения: 10.02.2023).
18. Иванов Д.А., Ситников А.И., Шляпин С.Д – Композиционные материалы: учебное пособие для вузов, 2019. 13 с.
19. Краткий обзор алгоритма машинного обучения Метод Опорных Векторов (SVM) – Режим доступа: <https://habr.com/ru/post/428503/> (дата обращения 07.03.2023)
20. Шитиков В.К., Мастицкий С.Э. (2017) Классификация, регрессия и другие алгоритмы Data Mining с использованием R. 351 с. − Электронная книга, адрес доступа: https://github.com/ranalytics/data-mining.
21. Кашнитский, Ю. Открытый курс машинного обучения. Тема 3. Классификация, деревья решений и метод ближайших соседей [Электронный ресурс] : – Режим доступа: https://habr.com/ru/company/ods/blog/322534/. (дата обращения: 14.03.2023).
22. Константинов, М. Краткий курс машинного обучения или как создать нейронную сеть для решения скоринг задачи [Электронный ресурс]: – Режим доступа: https://habr.com/ru/post/340792/. (дата обращения: 12.03.2023).
23. Масзанский, А. Метод k-ближайших соседей (k-nearest neighbour) [Электронный ресурс]: – Режим доступа: https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19/. (дата обращения: 09.03.2023).
24. Метод обратного распространения ошибки: математика, примеры, код [Электронный ресурс]: – Режим доступа: https://neurohive.io/ru/osnovy-data-science/obratnoe-rasprostranenie/. (дата обращения: 10.0,.2023).
25. Радченко, В. Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес [Электронный ресурс]: – Режим доступа: https://habr.com/ru/company/ods/blog/324402. (дата обращения: 05.03.2023).
26. Федоров, А. Решение задачи регрессии полносвязной нейронной сетью [Электронный ресурс]: – Режим доступа: https://www.bizkit.ru/2019/11/05/14921/. (дата обращения: 12.03.2023).
27. Функции активации в нейронных сетях [Электронный ресурс]: – Режим доступа: http://www.aiportal.ru/articles/neural-networks/activation-function.html. (дата обращения: 12.03.2022).
28. Хлевнюк, А. Основы линейной регрессии [Электронный ресурс]: – Режим доступа: https://habr.com/ru/post/514818/. (дата обращения: 03.03.2023).

# **5 Приложения**

**Приложение А**

1. **Задание**

**Тема:** Прогнозирование конечных свойств новых материалов (композиционных материалов).

**Описание:**

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.

Датасетсо свойствами композитов. Объединение делать по индексу тип объединения INNER. Датасет расположен: <https://drive.google.com/file/d/1B1s5gBlvgU81H9GGolLQVw_SOi-vyNf2/view?usp=sharing>

Требуется:

1. Изучить теоретические основы и методы решения поставленной задачи.
2. 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.
3. Провести предобработку данных (удаление шумов, нормализация и т.д.).
4. 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.
5. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель.
6. 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).
7. Оценить точность модели на тренировочном и тестовом датасете.
8. Создать репозиторий в GitHub / GitLab и разместить там код исследования. Оформить файл README.

**Приложение Б (в PDF версии)**

1. Отчет по объединенному DataFrame, полученному с помощью модуля ProfileReport в html и PDF форматах
2. Отчет по очищенному DataFrame, полученному с помощью модуля ProfileReport в html и PDF форматах
3. Отчет по объединенному DataFrame, полученному с помощью модуля SweetVIZ в html и PDF форматах
4. Отчет по очищенному DataFrame, полученному с помощью модуля SweetVIZ в html и PDF форматах