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 Кургаева Елена Сергеевна

Москва, 2022

**Содержание**

Введение

1. Аналитическая часть
   1. Постановка задачи
   2. Описание используемых методов
   3. Разведочный анализ данных
2. Практическая часть
   1. Предобработка данных
   2. Разработка и обучение модели
   3. Тестирование модели
   4. Нейронная сеть
   5. Разработка приложения
   6. Создание репозитория

Заключение

Библиографический список

# Введение

Развитие современных технологий требует новых конструкционных материалов, превосходящих по своим прочностным, упругим и другим свойствам традиционные. Здесь приходят на помощь композиты - искусственно созданные материалы, состоящие из нескольких других и обладающие свойствами, которые не наблюдаются у компонентов по отдельности. Сложность получения новых композитов заключается в прогнозировании свойств будущих материалов.

Как узнать, будет ли новый материал обладать хорошими показателями именно по тем свойствам, которые важны? Сейчас, среди инструментов исследования есть такие технологии, как машинное обучение. Созданные прогнозные модели помогут сократить количество проводимых испытаний, а также пополнить базу данных материалов возможными новыми характеристиками материалов, и цифровыми двойниками новых композитов.

В рамках выпускной квалификационной работы по курсу «Data Science» мы воспроизведем исследование с анализом данных, предоставленных Центром НТИ «Цифровое материаловедение: новые материалы и вещества» (структурное подразделение МГТУ им. Н.Э. Баумана), а также создадим модель для прогноза конечных свойств новых композиционных материалов.

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

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

Композитный материал — компонентный материал, в структуру которого входит пластичная основа (матрица), армированная наполнителями, которые обладают повышенной прочностью, жесткостью и т.д. Взаимосвязь элементов разного рода приводит к образованию нового вещества, свойства которого различаются от свойств каждого из его компонентов. Меняя строение основы и наполнителя, их пропорцию, текстуру наполнителя, приобретают широкий диапазон материалов с нужным количеством свойств. Большинство композитов по своим характеристикам лучше, чем общепринятые материалы и сплавы [1].

Композиты состоят из нескольких отдельных элементов. Основная задача изготовления композиционого материала – разработать новое вещество, которое сочетает в себе характеристики ее составляющих наиболее полезным способом [2].

Современные композиты изготавливаются из таких материалов, как: 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.

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

Имеется два датасета. Первый датасет X\_bp в формате Excel содержит данные о свойствах базальтопластика, композиционного материала, в состав которого входят базальтоволокно и органическое связующее. Среди свойств базальтопластиков можно перечислить: высокие показатели термостойкости и огнестойкости; низкую степень теплопроводности; высокие прочностные свойства, которые превышают свойства стали в несколько раз; высокую степень коррозионной устойчивости; повышение срока эксплуатации объекта по сравнению с общепринятыми материалами; устойчивость к резким значительным перепадам температуры; малый вес по сравнению с металлом; прочное сцепление с бетоном; высокие показатели на деформацию.

Материалы из базальтоволокна используют для: возведение морских и портовых сооружений, магистралей; строительства теплосберегающих ограждающих конструкций; в качестве опор ЛЭП и инженерных сетей; [3].

Датасет из 1023 строк и 11 столбцов содержит следующие характеристики:

* 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.

Второй датасет X\_nup в формате Excel содержит данные о свойствах углепластика, композиционного материала, в состав которого входит большое количество слоев. Он представляет из себя полотно из углеволокна в оболочке из термореактивных полимерных смол. Свойства углекомпозита: малый вес, в несколько раз легче стали; высокий предел выносливости; высокий показатель термостойкости и огнестойкости; не проводит электричество; высокая степень коррозионной стойкости; высокий предел прочности. [3].

Датасет из 1040 строк и 4 столбцов содержит следующие характеристики:

* Угол нашивки, град;
* Шаг нашивки;
* Плотность нашивки.

Для решения задачи датасеты были объединены по индексу, а количество строк приведено к наименьшему единому значению – 1023. Выходными переменными по объединенному датасету являются:

* Соотношение матрица-наполнитель;
* Модуль упругости при растяжении, ГПа;
* Прочность при растяжении, Мпа.

При этом, будучи выходной, переменная исключается из числа входных.

При проведении анализа данных была получена такая информация о данных, как: отсутствие пропусков, сводка числовых значений (среднее значение, стандартное отклонение, минимальное и максимальное значения, первый, второй и третий процентиль по каждой характеристике), количество уникальных значений каждого столбца и небольшое количество выбросов, которые были удалены.

* 1. **Описание используемых методов**

Машинное обучение (Machine Learning, ML) – это раздел теории искусственного интеллекта, предметом которого является поиск методов решения задач путем обучения в процессе решения сходных задач. Для построения таких методов используются средства алгебры, математической статистики, дискретной математики, теории оптимизации, численных методов, и других разделов математики. [4]

Системы машинного обучение могут быть классифицированы согласно объему и типу контроля, которым они подвергаются во время обучения. Есть четыре главных категории: обучение с учителем, обучение без учителя, частичное обучение и обучение с подкреплением.

Обучение с учителем (supervised learning) предполагает, что необходимо найти функциональную зависимость результатов от входов и построить алгоритм, на входе принимающий описание объекта и на выходе выдающий ответ. Функционал качества, как правило, определяется через среднюю ошибку ответов алгоритма по всем объектам выборки.

К обучению с учителем относятся следующие алгоритмы:

1. k ближайших соседей (k-nearest neighbors)
2. линейная регрессия (linear regression)
3. логистическая регрессия (logistic regression)
4. метод опорных векторов (Support Vector Machine - SVM)
5. деревья принятия решений (decision tree) и случайные леса (random forest)
6. нейронные сети (neural network)

Обучение без учителя (unsupervised learning) - ответы не задаются, и нужно искать зависимости между объектами. Сюда входят методы кластеризации, поиска ассоциативных правил, фильтрации выбросов, построения доверительной области, сокращения размерности и заполнения пропущенных значений.

К неклассическим, но весьма популярным методам относят обучение с подкреплением, в частности, генетические алгоритмы, и искусственные нейронные сети. В качестве входных объектов выступают пары «ситуация, принятое решение», а ответами являются значения функционала качества, который характеризует правильность принятых решений (реакцию среды). Эти методы успешно применяются для формирования инвестиционных стратегий, автоматического управления технологическими процессами, самообучения роботов и других подобных задач.

В представленной работе перед нами стояло три основные задачи, две из которых являются задачами регрессии и одна – обучение модели с помощью нейронной сети для создания рекомендательной системы.

Задача регрессии в машинном обучении – это предсказание параметра Y по известному параметру Х, то есть модель прогнозирует значение метки по набору связанных компонентов. Метка здесь может принимать любое значение, а не просто выбирается из конечного набора значений, как в задачах классификации. Алгоритмы регрессии моделируют зависимость меток от связанных компонентов, чтобы определить закономерности изменения меток при разных значениях компонентов. На вход алгоритма регрессии подается набор примеров с метками известных значений. Результатом работы алгоритма регрессии является функция, которая умеет прогнозировать значения метки для любого нового набора входных компонентов.

Для решения задачи регрессии были использованы следующие методы:

1. Линейная регрессия
2. Случайный лес
3. Градиентный бустинг
4. Метод k ближайших соседей

Линейная регрессия (Linear regression) — 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.

Цель линейной регрессии — поиск линии, которая наилучшим образом соответствует точкам реальных значений на входе. Общее уравнение для прямой в формуле (1)

f (x) = m⋅ x + b, (1)

где m - наклон линии;  
 b - его y-сдвиг.

Таким образом, решение линейной регрессии определяет значения для m и b, так что f (x) приближается как можно ближе к y.

Случайный лес (Random Forest) — 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 (в задачах регрессии их ответы усредняются). Благодаря своей гибкости Random Forest применяется для решения практически любых проблем в области машинного обучения. Сюда относятся классификации (RandomForestClassifier) и регрессии (RandomForestRegressor), а также более сложные задачи, вроде отбора признаков, поиска выбросов/аномалий и кластеризации.

Градиентный бустинг – это продвинутый алгоритм машинного обучения для решения задач классификации и регрессии. Он строит предсказание в виде ансамбля слабых предсказывающих моделей, которыми в основном являются деревья решений. Из нескольких слабых моделей в итоге мы собираем одну, но уже эффективную. Общая идея алгоритма – последовательное применение предиктора (предсказателя) таким образом, что каждая последующая модель сводит ошибку предыдущей к минимуму.

К-ближайших соседей (k-nearest neighbors) – это простой алгоритм машинного обучения с учителем, который можно использовать для решения задач классификации и регрессии. Он прост в реализации и понимании, но имеет существенный недостаток – значительное замедление работы, когда объем данных растет.

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).

Для решения задачи создания рекомендательной системы был использован метод многослойного персептрона.

Рекомендательные системы — это комплекс алгоритмов, программ и сервисов, задача которого предсказать на основе информации о профиле и активности пользователя, что его может заинтересовать. Грамотно подобранная рекомендательная система позволяет сократить время поиска нужного товара или услуги, удержать пользователя на сервисе, повысить вероятность совершения сопутствующих целевых действий, например, покупки. В рамках данной задачи рекомендательной системой является модель машинного обучения, рассчитывающая соотношение матрица-наполнитель на основании входных данных из датасета.

Многослойный персептрон — частный случай [персептрона Розенблатта](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%86%D0%B5%D0%BF%D1%82%D1%80%D0%BE%D0%BD), в котором один [алгоритм](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) [обратного распространения ошибки](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BE%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D1%80%D0%B0%D1%81%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B8) обучает все слои. Название по историческим причинам не отражает особенности данного вида персептрона, то есть не связано с тем, что в нём имеется несколько слоёв (так как несколько слоёв было и у персептрона Розенблатта). Особенностью является наличие более чем одного обучаемого слоя (как правило — два или три). Необходимость в большом количестве обучаемых слоёв отпадает, так как теоретически единственного скрытого слоя достаточно, чтобы перекодировать входное представление таким образом, чтобы получить [линейную разделимость](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D0%BF%D0%B0%D1%80%D0%B0%D0%B1%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C) для выходного представления. Существует предположение, что, используя большее число слоёв, можно уменьшить число элементов в них, то есть суммарное число элементов в слоях будет меньше, чем если использовать один скрытый слой. Это предположение успешно используется в технологиях [глубокого обучения](https://ru.wikipedia.org/wiki/%D0%93%D0%BB%D1%83%D0%B1%D0%BE%D0%BA%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) и имеет обоснование. [6].

Все вышеперечисленные задачи в данной работе решены на языке Python с использованием библиотек Pandas, NumPy, Matplotlib, Seaborn и Tensorflow.

Python — [высокоуровневый язык программирования](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%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%D1%8F) общего назначения с [динамической](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) [строгой](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D0%BE%D0%B3%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости [кода](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) и его качества, а также на обеспечение переносимости написанных на нём программ. [7].

Pandas — это библиотека машинного обучения, представляющая структуры данных высокого уровня и большой диапазон инструментов для анализа. Отличительной чертой Pandas считается возможность переводить сложнейшие операции с информацией, используя всего одну либо две команды. Данная библиотека содержит массу способов для объединения данных, их группировки и фильтрации.

**К особенностям Pandas относятся:**

* возможность упростить манипуляции данными;
* поддержка сортировки, визуализации и прочих опций.

Pandas обеспечивает широкую гибкость, функциональность, если эксплуатировать ее с иными библиотеками. Кроме того, пользователи могут применять ее в различных операциях, приложениях для сортировки данных и т. д.

[NumPy](https://blog.skillfactory.ru/glossary/numpy/) — основная библиотека Python, которая упрощает работу с векторами и матрицами. Содержит готовые методы для разных математических операций: от создания, изменения формы, умножения и расчета детерминант матриц до решения линейных уравнений и сингулярного разложения. Это значительно повышает производительность и, соответственно, ускоряет время выполнения работы.

[Matplotlib](https://blog.skillfactory.ru/glossary/matplotlib/) — низкоуровневая библиотека для создания двумерных диаграмм и графиков. С ее помощью можно отображать широкий спектр визуализаций: линейные и точечные диаграммы, диаграммы с областями, гистограммы, круговые диаграммы, диаграммы «стебель-листья», контурные графики, поля векторов и спектрограммы.

Seaborn — [библиотека](https://seaborn.pydata.org/) более высокого уровня, чем matplotlib. С ее помощью проще создавать специфическую визуализацию: тепловые карты, временные ряды и скрипичные диаграммы.

TensorFlow — это библиотека AI, которая помогает разработчикам создавать крупномасштабные нейронные сети со многими слоями, используя графики потоков данных. TensorFlow также облегчает построение моделей глубокого обучения, продвигает современную технологию ML / AI и позволяет легко развертывать приложения на базе ML. TensorFlow достаточно эффективен, когда дело доходит до классификации, восприятия, понимания, обнаружения, прогнозирования и создания данных.

[Scikit-learn](https://scikit-learn.org/stable/) – библиотека, которая основана на NumPy и SciPy. В ней есть алгоритмы для машинного обучения и интеллектуального анализа данных: кластеризации, регрессии и классификации. Особенностями Scikit-Learn являются:

возможность извлечения элементов из текстов и картинок, перекрестная проверка – множество различных методов проверки точности контролируемой модели на невидимой информации, множество алгоритмов машинного обучения и возможность осуществления дорогостоящих задач (кластеризация, факторный анализ и т. д.).

* 1. **Разведочный анализ данных**

Для разведочного анализа были использованы следующие методы Python:

DF.info(), изображенный на рисунке 1 – данный метод выводит общую информацию о датасете: количество строк и столбцов, количество значений, название переменных, тип данных.

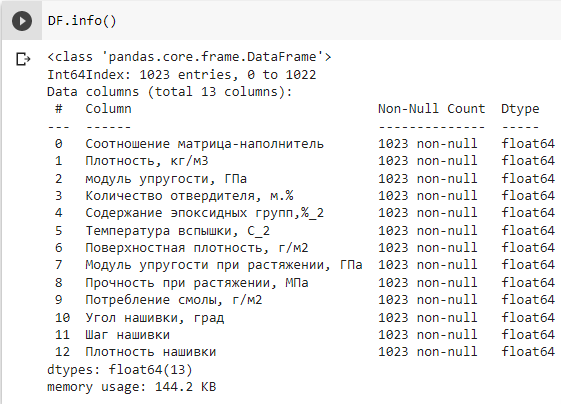


Рисунок 1 – Общая информация о датасете

* DF.nunique() на рисунке 2 – количество уникальных значений в каждом столбце

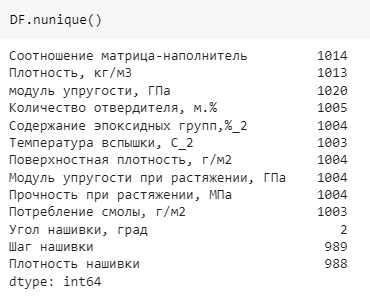


Рисунок 2 – Количество уникальных значений датасета

* DF.isnull().sum() – проверка на наличие пропусков. Пропусков не обнаружено.
* DF.describe().round(2).T - Описательная статистика, на рисунке 3 можно увидеть выводимые показатели (количество значений, среднее значение, стандартное отклонение, минимум, верхнее значение первого квартиля, медиана, верхнее значение третьего квартиля, максимум по каждому столбцу)



Рисунок 3 – Описательная статистика

Также, в рамках разведочного анализа были построены гистограммы распределения для каждого параметра. Для большинства характеристик распределение близко к нормальному, один из примеров видно на рисунке 4 - 'Температура вспышки, С\_2'.

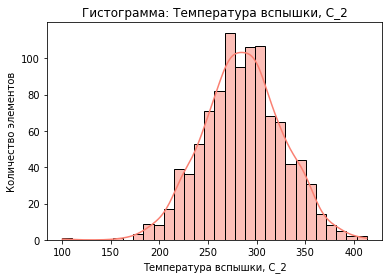


Рисунок 4 – Пример нормального распределения

Исключением оказались показатели:

"Поверхностная плотность, г/м2" (нормальное распределение со смещением вправо, вероятнее всего было проведено большое количество измерений с меньшим показателем поверхностной плотности).

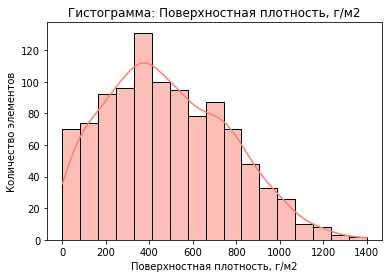


Рисунок 5 - Нормальное распределение со смещением вправо

"Угол нашивки, град" (дискретное распределение, т.к. колонка содержит всего два уникальных значения).

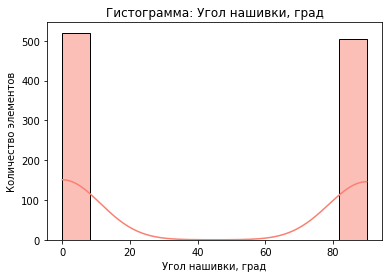


Рисунок 6 - Дискретное распределение

Были построены диаграммы размаха «ящик с усами» - удобный способ визуального представления групп числовых данных через квартили, который позволяет сделать наблюдения о выбросах, симметричности и сгруппированности данных, а также, о их смещении.

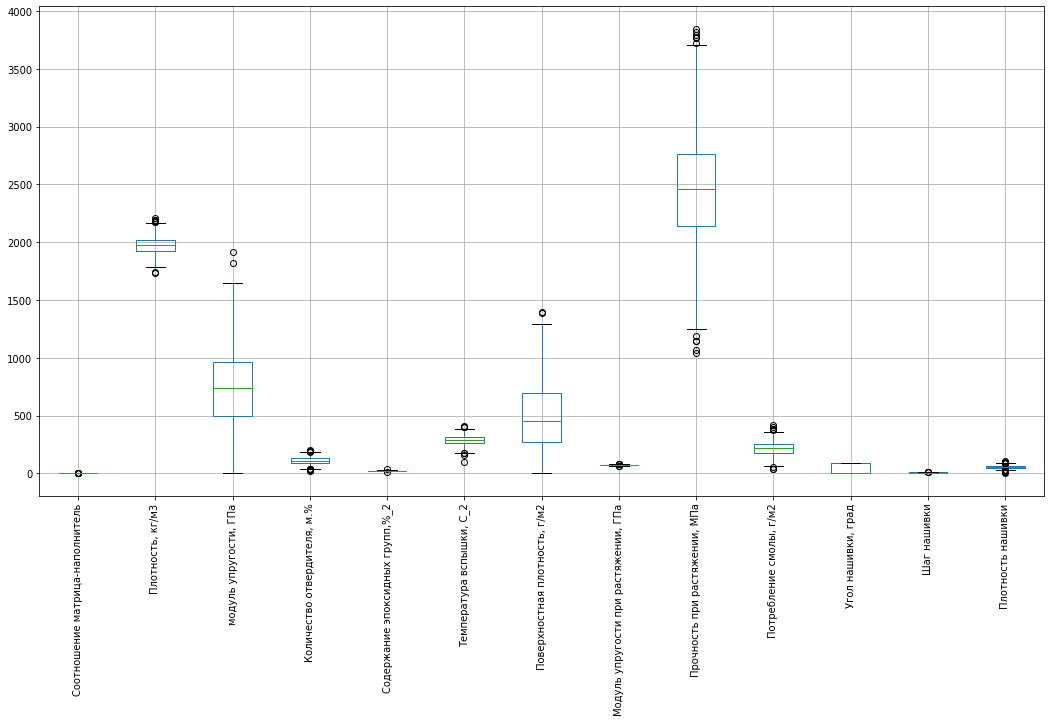


Рисунок 7 - Диаграммы размаха «ящик с усами»

По диаграмме размаха можно сделать вывод о том, что выбросы есть по всем характеристикам, кроме "Угол нашивки, град".

Следующим шагом в работе с данными было построение попарных графиков рассеяния точек, Присвоив каждой оси переменную, можно определить, существуют ли отношения или корреляция между этими двумя переменными. Отображаемые на диаграммах рассеяния паттерны позволяют увидеть разные типы корреляции. Среди них могут быть: **положительная** (оба значения увеличиваются), **отрицательная** (одно значение увеличивается, в то время как второе уменьшается), **нулевая** (отсутствие корреляции), **линейная, экспоненциальная**и**подковообразная**. Сила корреляции определяется по тому, насколько близко расположены друг от друга точки на графике. Данный график показал очень слабую зависимость между переменными датасета. Также имеем возможность еще раз увидеть наличие некоторого количества выбросов – точки на графике, которые значительно удалены от общего кластера.

Тепловая карта коэффициентов корреляции на рисунке 8 хорошо демонстрирует наличие слабой зависимости переменных в датасете. Все полученные коэффициенты находятся в промежутке значений от - 0,25 до 0,25. Т.е. скорее всего представленный детасет был предобработан и оставшиеся переменные являются независимыми, каждая из них несет свою функцию при влиянии на композитный материал.

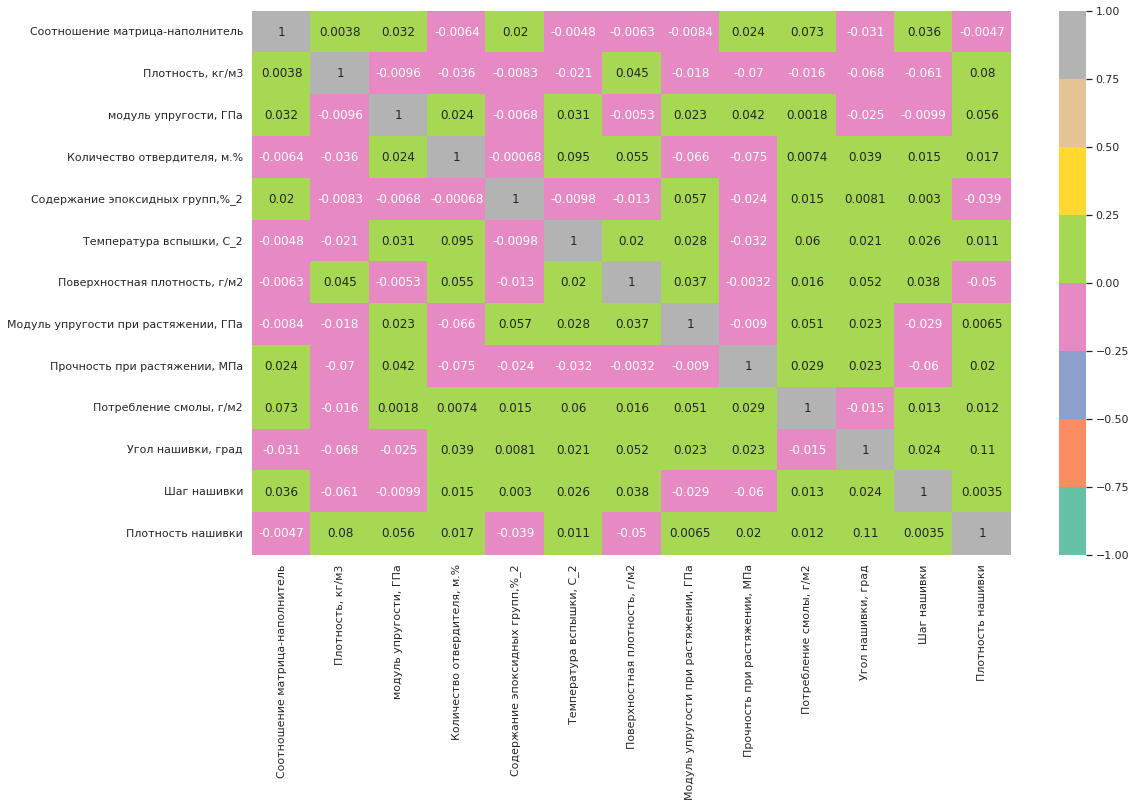


Рисунок 8 - Тепловая карта коэффициентов корреляции

1. **Практическая часть**
   1. **Предобработка данных**

Практическую часть работы начнем с удаления выбросов объединенного датасета, данное действие необходимо для большей достоверности результатов. Сравним два различных метода удаления выбросов.

Метод трех сигм - правило основано на условном предположении, что все наблюдения выборки укладываются в интервал. Результаты наблюдений, которые выходят за пределы интервала, считаются промахами и из выборки исключаются.

Общее количество выбросов в датасете данным методом составляет 24.

Метод межквартильных расстояний, графическим отображением которого является диаграмма **«ящик с усами»**, построенная выше. Для этого создается вариационный ряд, определяются пять основных характеристик, включающих в себя минимальное значение признака, первое квартильное, медианное, третье квартильное и максимальное значения, затем по формуле (2) рассчитывается межквартильное расстояние

IQR = Q3 – Q1 (2)

и на его основе по формуле (3) определяются верхняя

Q1 – 1.5 \* IQR (3)

и по аналогичной формуле (4) нижняя границы совокупности

Q3 + 1.5 \* IQR (4)

Все, что выходит за верхнюю и нижнюю границы, на графике изображается в виде отдельных точек и считается выбросами. Конечно, можно удалить все аномальные значения, и это не вызовет особых проблем при работе с исчерпывающим датасетом в несколько десятков тысяч строк. Но в случае небольшого количества данных один из вариантов решения – заменить аномальные значения на значение границ.

Общее количество выбросов в датасете данным методом составляет 93.

Так как итоговый датасет для выполнения работы очень маленького размера и есть предположение о том, что датасет был предобработан заказчиком, для удаления выбросов выберем метод трех сигм, ввиду того, что данный способ затрагивает меньшее количество данных. Удаляем 24 строки значений с выбросами.

После построения тепловой карты корреляции для очищенного от выбросов датасета видим, что ситуация практически не изменилась, полученные коэффициенты все еще находятся в промежутке значений от - 0,25 до 0,25, что говорит о слабой зависимости переменных между собой.

Поскольку минимальные и максимальные значения каждой переменной имеют большой разброс, необходимо дополнительно провести нормализацию данных, основная цель которой заключается в приведении данных к единому виду, который позволить сравнить данные между собой и произвести дальнейшие расчеты. Существует множество способов нормализации значений признаков, чтобы масштабировать их к единому диапазону и использовать в различных моделях машинного обучения. В зависимости от используемой функции, их можно разделить на 2 большие группы: линейные и нелинейные. При **нелинейной нормализации** в расчетных соотношениях используются функции логистической сигмоиды или гиперболического тангенса. В линейной нормализации изменение переменных осуществляется пропорционально, по линейному закону. Встроенный метод MinMaxScaler() библиотеки Sklearn – это линейное преобразование данных в диапазоне [0..1], где минимальное и максимальное масштабируемые значения соответствуют 0 и 1 соответственно. Ниже представлена формула (5) для расчета.

, (5)

На графике разброса данных на рисунке 9 видим, как распределены значения переменных до нормализации и после.

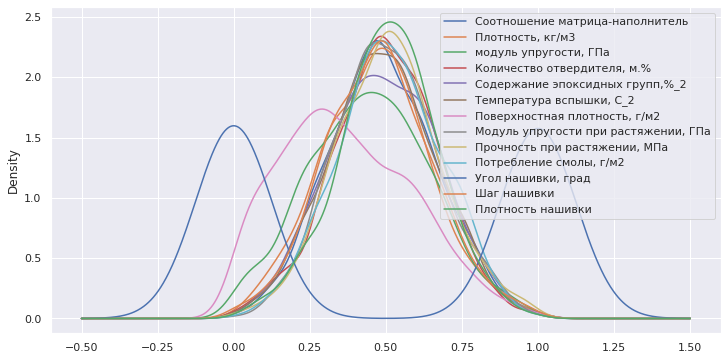
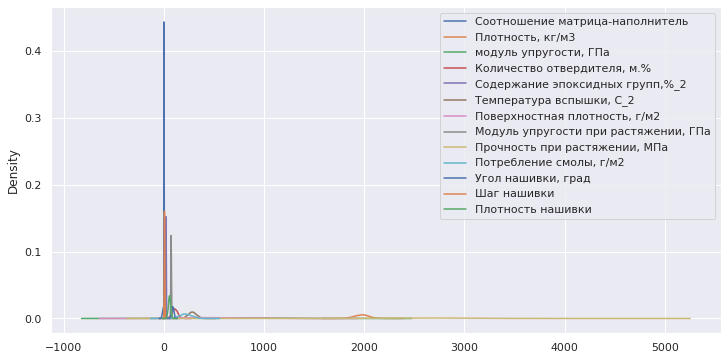


Рисунок 9 – Разброс данных до и после нормализации

С нормализованными данными переходим к построению моделей для прогноза модуля упругости при растяжении и прогноза прочности при растяжении.

* 1. **Разработка и обучение модели**

Для решения задачи предсказания модуля упругости при растяжении и прочности при растяжении были использованы следующие методы:

* Линейная регрессия;
* Случайный лес;
* Градиентный бустинг;
* Метод k ближайших соседей.

В работе с каждой моделью придерживались следующего алгоритма:

1. Вызвать модель регрессии, передать параметры;
2. Вызвать метод GridSearchCV – это инструмент для автоматического подбора гиперпараметров для моделей машинного обучения путем поиска по сетке с перекрестной проверкой, т.е. он создает модель для каждой возможной комбинации параметров. Передаем в GridSearchCV модель регрессии и возможные параметры. Количество блоков по условию задачи равняется 10. Выводим результат лучшего параметра;
3. Обучаем модель с учетом лучших параметров;
4. Считаем ошибки модели, вносим в сводную таблицу по всем моделям.

В качестве оценки работы моделей были использованы следующие метрики:

* MSE (Mean Squared Error) - средняя квадратическая ошибка. Суть метода заключается по сути в том, чтобы минимизировать сумму квадратов отклонений фактических значений от расчётных. Если полученную сумму разделить на число наблюдений, то получим MSE. Формула (6) целевой функции в этом случае выглядит следующим образом:

; (6)

* R2 (Коэффициент детерминации) - это доля [дисперсии](http://www.machinelearning.ru/wiki/index.php?title=%D0%94%D0%B8%D1%81%D0%BF%D0%B5%D1%80%D1%81%D0%B8%D1%8F_%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D0%BE%D0%B9_%D0%B2%D0%B5%D0%BB%D0%B8%D1%87%D0%B8%D0%BD%D1%8B&action=edit) зависимой переменной, объясняемая рассматриваемой моделью

; (7)

* MAE (Mean Absolute Error) - средняя абсолютная ошибка. Формула (8)

. (8)

* 1. **Тестирование модели**

Результаты оценок каждой модели были оформлены в виде таблицы, с показателями которых можно ознакомиться ниже.

Для моделей предсказания модуля упругости при растяжении результаты ошибок на рисунке 10.

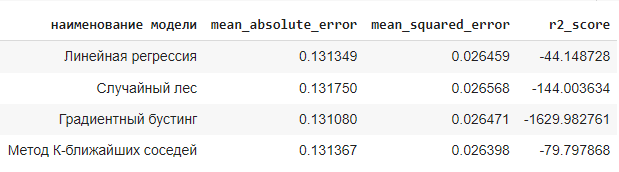


Рисунок 10 – Ошибки модели предсказания модуля упругости

Для моделей предсказания прочности при растяжении результаты ошибок на рисунке 11.

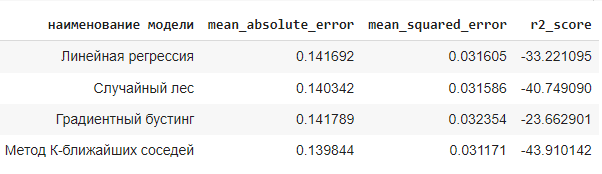


Рисунок 11 - Ошибки модели предсказания прочности

Результаты построения и обучения моделей, к сожалению, не дали значительного положительного результата. Наименьшая ошибка в предсказании модуля упругости при растяжении получилась у модели линейной регрессии, в предсказании прочности при растяжении у модели «случайный лес». Хотя, эти показатели не сильно отличаются от остальных моделей.

* 1. **Нейронная сеть**

Для построения рекомендательной системы соотношения матрица-наполнитель использовали многослойный персептрон.

Первым шагом необходимо разделить очищенный от выбросов датасет на выходные данные в виде столбца «Соотношение матрица-наполнитель» и входные данные, которые включают все остальные столбцы. Разделить входные и выходные данные на тренировочную и тестовую части в соотношении 80 и 20% с помощью train\_test\_split, после нормализовать данные используя TensorFlow.layers.Normalization.

После этого можно создавать нейронную сеть с помощью Sequential - модель в библиотеке Keras, позволяющая создать нейронную сеть прямого распространения путем последовательного добавления слоев. Результат на рисунке 12.

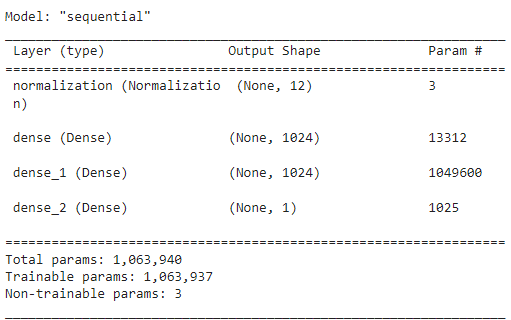


Рисунок 12 – Информация по модели нейронной сети

Модель нейронной сети имеет следующие настраиваемые гиперпараметры:

* входной слой нормализации 12 признаков;
* скрытых слоев: 2;
* активационная функция скрытых слоев: relu - выполняет простое нелинейное преобразование поданных на вход данных (x). Возвращает x, если x > 0 и 0 в противном случае. Отличается высокой скоростью вычисления;
* нейронов в каждом скрытом слое: по 1024;
* выходной слой с 1 нейроном (т.е. для одного признака), так как на выходе выводится одно значение для введенных данных;
* для оптимизации, как наиболее распространенный и дающий лучшие результаты, был применен метод Adam (adaptive moment estimation) - оптимизационный алгоритм, используемый для обучения сети, основная функция которого - изменение весов сети для уменьшения ошибки сети в процессе обучения. Для каждого нейрона алгоритм изменяет веса индивидуально;
* для оценки качества модели применена loss-функция: MeanSquaredError (MSE).

Далее было проведено обучение модели (рисунок 13) на тренировочных данных при помощи метода fit со следующими параметрами:

* аргумент validation\_split позволяет автоматически зарезервировать часть тренировочных данных для валидации. Это необходимо для того, чтобы иметь возможность обучить модель и оценить результаты работу с данными параметрами, не затрагивая тестовую выборку. Значением аргумента является доля данных, которые должны быть зарезервированы, в нашем случае это 20% тренировочных данных;
* verbose - режим вывода информации о процессе обучения нейронной сети;
* еpoch - количество повторений циклов обучения для всей выборки данных. В данном случае их 70, так как большее количество эпох не дало лучшего результата. Предоставленный заказчиком датасет имеет небольшие размеры и может обучаться на меньшем количестве эпох.

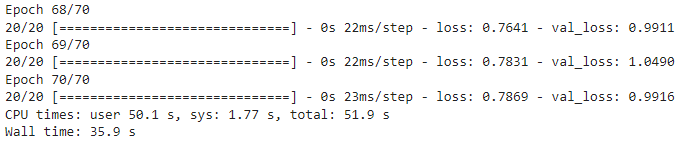
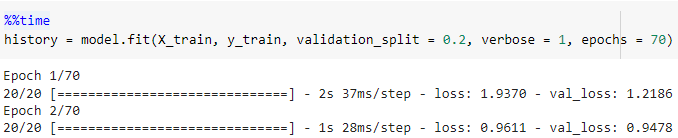


Рисунок 13 – Обучение модели

По результатам обучения строим график, на котором две кривых – отображение среднеквадратической ошибки модели на тестовых (голубая линия) и валидационных данных (красная линия) относительно числа итераций. На рисунке 14 видим, что линии идут рядом, ошибка постепенно снижается и выходит на плато, где остается приблизительно на одном уровне до конца обучения.

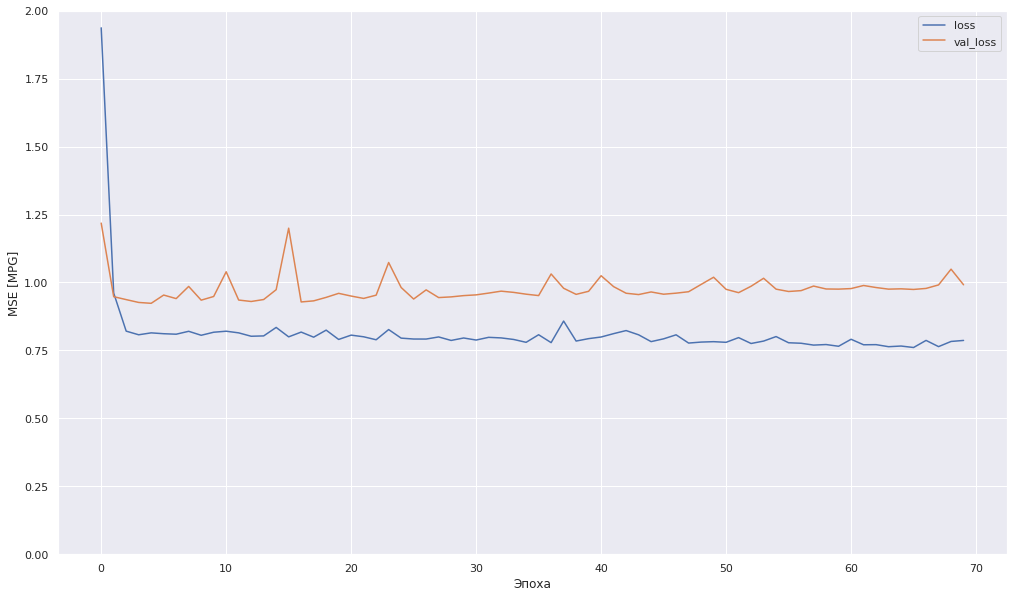


Рисунок 14 – Визуализация ошибки модели

Запускаем модель на тестовых данных. По результатам строим график (рисунок 15), для сравнения оригинальных значений выборки и значениями, предсказанными моделью.



Рисунок 15 - Визуализация результата работы модели на тестовых данных

* 1. **Разработка приложения**
  2. **Создание репозитория**

Создан репозиторий на GitHub, где размещен код исследования, датасет заказчика и оформлен файл README.

Страница слушателя: [https://github.com/ElenSergevna](https://github.com/ElenSergevna/VKR-Data-Science)

Репозиторий: <https://github.com/ElenSergevna/VKR-Data-Science>

Коммиты:

**Заключение**

В результате проделанной выпускной квалификационной работы по курсу «Data Science» были изучены теоретические основы методов машинного обучения и основные библиотеки высокоуровневого языка программирования Python, как один из основных инструментов для анализа данных. В процессе выполнения практической части была возможность использовать изученные на курсе методы машинного обучения и построения моделей на реальных данных.

Результаты построения и обучения моделей, к сожалению, не дали значительного положительного результата, но позволили приобрести опыт работы с данными инструментами.

**Литература**

1. ГОСТ 31938-2011 Арматура композитная полимерная для армирования бетонных конструкций. Общие технические условия. - Введ. 2014–01–01. – М. : Изд-во стандартов, 2014.
2. СП 28.13330.2012 Защита строительных конструкций от коррозии. – Введ. 2013-01-01. Актуализированная редакция СНиП 2.03.11-85.
3. Барабанщиков Ю., Беляева С., Авдеева А., Перес М. Стеклопластиковая арматура для бетона. Прикладная механика и материалы. 2015. – 475 с.
4. Bengio Y., Courville A. Pascal Vincent [Representation Learning: A Review and New Perspectives](http://arxiv.org/abs/1206.5538) / 2014. – 67 с.
5. Lutz М. [A Python Q&A Session](https://www.oreilly.com/library/view/learning-python-3rd/9780596513986/ch01.html) (англ.). Learning Python, 3rd Edition / O'Reilly Media, Inc. 2007.
6. Миронов А. М. Машинное обучение часть 1 ст. 9 – Режим доступа <http://www.intsys.msu.ru/staff/mironov/machine_learning_vol1.pdf> (дата обращения 02.06.2022)
7. Орельен Ж. Прикладное машинное обучение с помощью scikit-learn и tensorflow / Вильямс, 2018. – 134 с.
8. Чуракова И. Я. Направления использования методик выявления аномальных наблюдений при решении задач операционного менеджмента. Научный доклад № 13 (R)–2010. СПб.: ВШМ СПбГУ, 2010.