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»

### НА ТЕМУ:

### «Прогнозирование конечных свойств новых материалов (композиционных материалов)»

### Выполнила:

### Колесникова Ирина Игоревна

**Содержание**

Введение………………………………………………………………………...2

1. Аналитика…………………………………………………………………….5

1.1. Постановка задачи…………………………………………………………5

1.2. Описание используемых методов………………………………………...9

1.3. Разведочный анализ данных……………………………………………...13

2. Практическая часть………………………………………………………….17

2.1. Предобработка данных……………………………………………………17

2.2. Разработка и обучение модели…………………………………………....21

2.3. Тестирование модели……………………………………………………...24

2.4. Нейросеть…………………………………………………………………..26

2.5. Приложение………………………………………………………………...29

3. Заключение…………………………………………………………………...30

4. Библиографический список………………………………………………..

**Введение**

Непрерывное развитие науки и техники, возникновение новых технологий, продвижение исследований вглубь океанов и космоса влекут за собой разработку современных материалов, позволяющих повышать эффективность и надёжность конструкций. Традиционные конструкционные материалы достигли максимальных пределов своих механических и химических характеристик. Они отступают под натиском более совершенных композиционных материалов, разработке которых придаётся огромное значение во всем мире. Улучшение свойств современных композитов — это снижение потребления энергетических ресурсов, минимизация влияния на экологию, действенный метод борьбы с коррозией, рост конструкторского и технологического потенциала.

Композитный материал или композит— многокомпонентный материал, изготовленный (человеком или природой) из двух или более компонентов с существенно различными физическими и/или химическими свойствами, которые, в сочетании, приводят к появлению нового материала с характеристиками, отличными от характеристик отдельных компонентов и не являющимися простой их суперпозицией. В составе композита принято выделять матрицу/матрицы и наполнитель/наполнители, последние выполняют функцию армирования (как, например, арматура в железобетоне). В качестве наполнителей композитов как правило выступают углеродные или стеклянные волокна, а роль матрицы играет полимер. Сочетание разных компонентов позволяет улучшить характеристики материала и делает его одновременно лёгким и прочным. При этом отдельные компоненты остаются таковыми в структуре композитов, что отличает их от смесей и затвердевших растворов. Варьируя состав матрицы и наполнителя, их соотношение, ориентацию наполнителя, получают широкий спектр материалов с требуемым набором свойств. Многие композиты превосходят традиционные материалы и сплавы по своим механическим свойствам и в то же время они легче. Использование композитов обычно позволяет уменьшить массу конструкции при сохранении или улучшении её механических характеристик.

В настоящее время практически ни одна из солидных международных выставок не проходит без присутствия решений для базальтокомпозитов. Базальтокомпозит – это композит, в основу которого входит базальтовое волокно. Базальтовое волокно в свою очередь является продуктом вулканической деятельности. За счет первичного плавления базальтовых пород и их гомогенизация уже произведены самой природой, а это ощутимо снижает стоимость получаемого продукта. Немаловажной характеристикой базальтового волокна является его прекрасная совместимость со всевозможными матричными материалами: полимерными, керамическими, металлическими, цементными, углеродными — и различными волокнами для получения гибридных композиционных материалов. Для изготовления базальтокомпозитов применимы практически все известные методы: намотка, формование, напыление, литьё и прессование.

Заинтересованность такими гибридными композитами растет с каждым днем, но возникает одна «проблема»: даже если мы знаем свойства каждого материала, каким получится композит из этих материалов – сложно сказать. Для решения этого вопроса есть два пути: физические испытания образцов материалов (что является трудно затратным занятием) или прогнозирование характеристик.

Суть прогнозирования заключается в симуляции представительного элемента объема композита, на основе данных о характеристиках входящих компонентов (связующего и армирующего компонента). Наша задача состоит как раз в этом. Необходимо изучить данные по базальтопластику, а также данные по углепластиковым нашивкам. Исходя из анализа данных, определить свойства гибридного композита. В ходе работы были разработаны и протестированы несколько моделей для свойств «Модуль упругости при растяжении» и «Прочность при растяжении», а также написана нейронная сеть, способная определить соотношение «матрица – наполнитель». На основе нейронной сети было написано пользовательское приложение.

**Аналитика**

* 1. **Постановка задачи**

На входе есть два датасета со своими признаками (свойствами). Первый набор данных data\_bp (для базальтопластика) имеет в себе 1023 значения, 10 признаков и один столбец с нумерацией. Второй набор данных data\_nup (углепластик) имеет 1040 значений, 3 признака и один столбец с нумерацией. Для того, чтобы получить общую сводную таблицу, объединяем два датасета по inner, но так как изначально количество значений в обоих наборах разное, то полученный датасет не будет иметь все значения из второго набора.

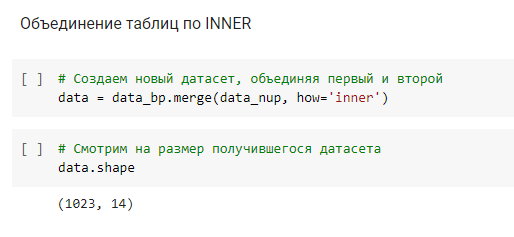


Рисунок 1 – Датасет, полученный после объединения по inner

В результате получаем набор данных с 1023 наблюдениями, 13 признаками и одним столбцом нумерации. Тип данных во всех колонках float64 (число с плавающей точкой), кроме столбца «Угол нашивки» - здесь у нас тип данных int (целочисленное значение). В данном наборе наблюдаются наши целевые признаки: модуль упругости при растяжении, прочность при растяжении и соотношение матрица-наполнитель. Так как нам есть на что опираться, можно сделать вывод, что это будет обучение с учителем.

Прежде чем преступить к решению нашей задачи, необходимо рассмотреть данные подробнее: проверить на пропущенные значения, дубликаты, статистические показатели (mean, std, median).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2 – поиск пропущенных значений

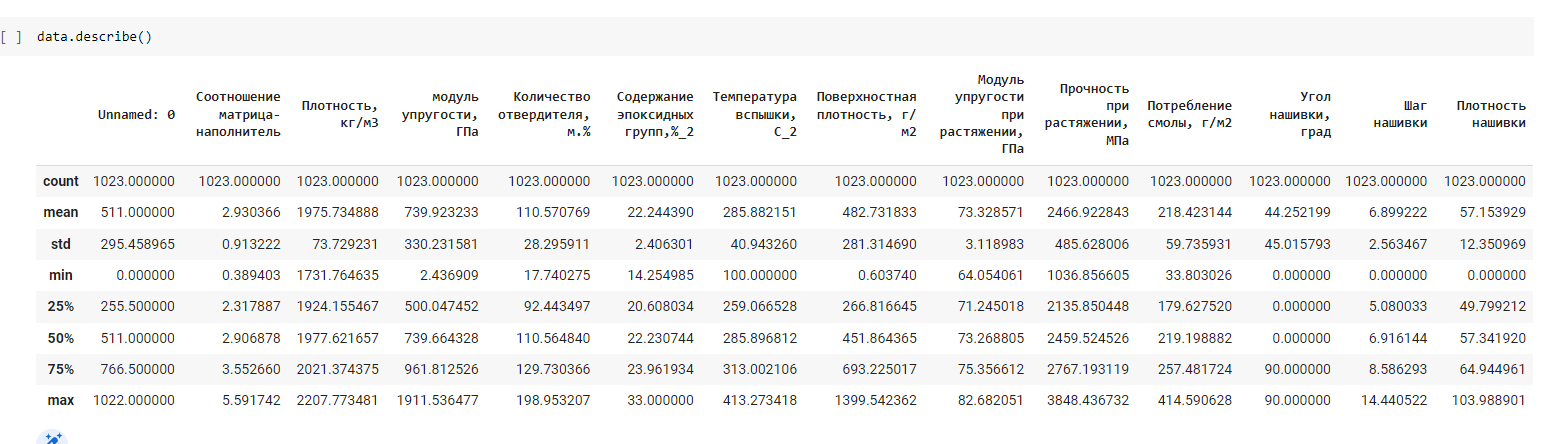


Рисунок 3 – статистические показатели по всем признакам датасета

Также необходимо нарисовать гистограммы распределения каждой из переменной, диаграммы ящика с усами, попарные графики рассеяния точек.

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 4 – боксы с усами для полученного датасета

После небольшого анализа можно описать наши данные: пропусков и дубликатов не обнаружено, но почти в каждом признаке видны выбросы (см. рисунок 4). В дальнейшем выбросы нужно будет удалить, так как есть модели машинного обучения, которые очень «чувствительны» к выбросам. Если присмотреться к статистическим показателям, то можно увидеть и разнородность данных (см. рисунок 3). Подобное можно увидеть на графике оценка плотности ядра».

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 5 – оценка плотности ядра для композитных данных

Так как у нас решается задача регрессии, то нам важны взаимосвязи между признаками. Поэтому посмотрим на корреляционную матрицу датасета по всем признакам.Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 6 – корреляционная матрица для композитных данных.

Корреляционная матрица показывает нам то, что взаимосвязи между признаками плохо выражены, корреляция везде стремится к нулю, что говорит об отсутствии или очень слабой линейной связи, возможно даже, что взаимосвязи нелинейны.

Теперь можно сказать, какой пласт работы нас ждет: необходимо очистить датасет от выбросов, провести нормализацию и стандартизацию данных, после чего посмотреть, появятся ли какие-либо связи между признаками.

.

**1.2 Описание используемых методов**

Как излагалось выше, мы имеем дело с регрессионной задачей. Это пример контролируемого (supervised) машинного обучения, когда на основе истории предыдущих данных мы получаем предсказание. Для решения такой задачи были применены следующие модели:

а) линейная регрессия;

б) LASSO;

в) SVR;

г) Random Forest Regressor.

Рассмотрим первую модель – Линейная регрессия.

Линейная регрессия - это алгоритм машинного обучения, основанный на обучении под наблюдением. Он выполняет задачу регрессии. Регрессионные модели представляют собой целевое значение прогноза, основанное на независимых переменных. Он в основном используется для выяснения взаимосвязи между переменными и прогнозирования.

Преимущества линейной регрессии:

1) линейная регрессия проста в реализации и легче интерпретировать выходные коэффициенты;

2) когда известно, что связь между независимой и зависимой переменными имеет линейную зависимость, этот алгоритм лучше всего использовать из-за его меньшей сложности по сравнению с другими алгоритмами;

3) линейная регрессия подвержена чрезмерной подгонке, но ее можно избежать, используя некоторые методы уменьшения размерности, методы регуляризации (L1 и L2) и перекрестной проверки.

Недостатки линейной регрессии:

1) выбросы могут сильно влиять на регрессию;

2) линейная регрессия рассматривает взаимосвязь между средним значением зависимых переменных и независимыми переменными. Так же, как среднее значение не является полным описанием одной переменной, линейная регрессия не является полным описанием взаимосвязей между переменными.

LASSO - это линейная модель, которая оценивает разреженный коэффициент. Это полезно в некоторых ситуациях, потому что обычно используются варианты с меньшим количеством значений параметров, эффективно уменьшая количество переменных, от которых зависит данное решение. Следовательно, лассо и его варианты являются основой сжатого поля восприятия. При определенных условиях он может восстановить точный набор ненулевых весов.

SVR - в основе метода опорных векторов для задач регрессии или регрессии опорных векторов (SVR) лежит поиск гиперплоскости, при которой риск в многомерном пространстве будет минимальным. По сравнению с традиционной регрессионной моделью SVR оценивает коэффициенты путем минимизации квадратичных потерь. Так, если прогнозное значение попадает в область гиперплоскости, то потери равны нулю. В противном случае разности прогнозного и фактического значений. Модель, созданная с помощью регрессии опорных векторов, зависит только от подмножества обучающих данных, поскольку функция стоимости игнорирует образцы, прогноз которых близок к их целевому значению.

Плюсы SVR:

1) эффективен в пространствах больших размеров;

2) эффективен в случаях, когда количество измерений превышает количество образцов;

3) использует подмножество обучающих точек в функции принятия решений (называемых опорными векторами), поэтому это также эффективно с точки зрения памяти;

4) универсальность: для функции принятия решения могут быть указаны различные функции ядра.

Минусы SVR:

1) если количество функций намного превышает количество выборок, избегайте чрезмерной подгонки при выборе функции ядра и термин регуляризации имеет решающее значение;

2) SVM не предоставляют напрямую оценки вероятностей, они рассчитываются с использованием дорогостоящей пятикратной перекрестной проверки.

RandomForestRegressor (RFR) - каждое дерево в ансамбле строится из выборки, взятой с заменой (то есть выборкой начальной загрузки) из обучающего набора.

Кроме того, при разбиении каждого узла во время построения дерева наилучшее разбиение находится либо по всем входным характеристикам, либо по случайному подмножеству размера max\_features.

Назначение этих двух источников случайности — уменьшить дисперсию оценки леса. В самом деле, отдельные деревья решений обычно демонстрируют высокую дисперсию и имеют тенденцию переоснащаться. Внедренная случайность в лесах дает деревья решений с несколько несвязанными ошибками прогнозирования. Если взять среднее значение этих прогнозов, некоторые ошибки могут быть устранены. Случайные леса уменьшают дисперсию за счет комбинирования разных деревьев, иногда за счет небольшого увеличения смещения. На практике уменьшение дисперсии часто бывает значительным, что дает в целом лучшую модель.

Плюсы RFR:

1) алгоритм случайного леса не является предвзятым, поскольку существует несколько деревьев, и каждое дерево обучается на подмножестве данных. В принципе, алгоритм случайного леса опирается на силу «толпы», поэтому общая предвзятость алгоритма уменьшается;

2) этот алгоритм очень стабилен. Даже если новая точка данных введена в набор данных, общий алгоритм не сильно пострадает, так как новые данные могут повлиять на одно дерево, но ему очень трудно повлиять на все деревья;

3) алгоритм случайного леса хорошо работает, когда есть как категориальные, так и числовые признаки;

4) алгоритм случайного леса также хорошо работает, когда данные имеют пропущенные значения или они не были хорошо масштабированы.

Минусы RFR:

1) основным недостатком является сложность алгоритма, что требует гораздо больше вычислительных ресурсов из-за большого количества деревьев решений, соединенных вместе;

2) из-за своей сложности они требуют гораздо больше времени для обучения, чем другие сопоставимые алгоритмы.

Для оценки работы моделей были использованы метрики качества r2-score, mean\_absolute\_error, mean\_squared\_error.

R2-score - один из показателей оценки эффективности моделей машинного обучения на основе регрессии. Он также известен как коэффициент детерминации. Суть его работы заключается в измерении количества отклонений в прогнозах, объясненных набором данных. Показывает разницу между выборками в наборе данных и прогнозами, сделанными моделью. Когда коэффициент детерминации близок к нулю, тогда модель хорошо описывает данные; стремится к нулю, то качество прогноза идентично средней величине целевой переменной. Отрицательные значение коэффициента детерминации означают плохую объясняющую способность модели.

MAE (mean\_absolute\_error) - это усреднённая сумма модулей разницы между реальным и предсказанным значениями. Чем ниже MAE, тем лучше модель соответствует набору данных.

MSE (mean\_squared\_error) – это среднее арифметическое квадратов разностей между предсказанными и реальными значениями модели машинного обучения. MAE во многом похожа на MSE, но она отличается меньшей чувствительностью к выбросам значений (так как не берётся квадрат отклонения). Наилучшие значения при MSE, стремящимся к нулю.

**1.3 Разведочный анализ**

В данном разделе приведем краткое описание методов разведочного анализа данных, которые используются для первоначального анализа.

Чтобы получить всю информацию о датасете в одном месте, можно воспользоваться pandas\_profiling. Применяя метод profile\_report() к датасету, получаем разбор всех признаков, по ним видим пропущенные значения, графики распределения признаков, описательную статистику, корреляцию и много другого.

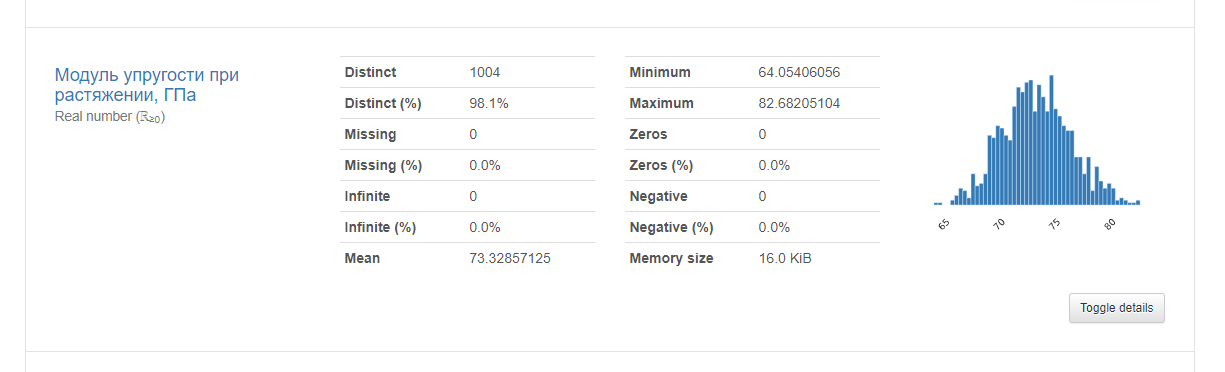


Рисунок 7 – пример profile\_report()

Но «старыми» методами тоже можно воспользоваться. На рисунке 2 с помощью data.info() посмотрели на наличие пропущенных значений. На рисунке 3 построили статистические показатели по всем признакам, используя метод .describe(). Очень важно проверить данные на наличие дубликатов:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8 – проверка на дубликаты

Также можно посмотреть на уникальные значения в наших признаках:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 – количество уникальных значений

Угол нашивки выдает нам два уникальных значения – это ноль (ноль градусов) и девяносто (девяносто градусов). Такие значения мы можем кодировать, например, с помощью LabelEncoder().

Также очень важно знать, какому закону распределения подчиняются данные. Чтобы это увидеть, строим гистограммы распределения.

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 10 – примеры гистограмм распределения

На графиках видно, что все признаки (не считая «Угол нашивки»), имеют нормальное распределение.

Попарные графики рассеяния частот показывают тесноту связи между двумя признаками, а также на них можно увидеть выбросы.

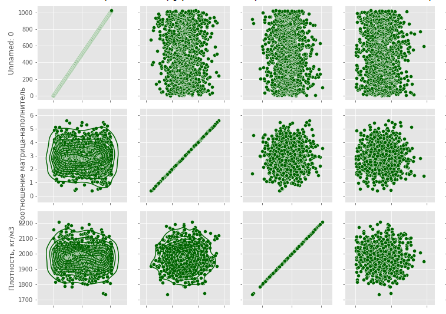


Рисунок 11 – часть графиков рассеяния частот

Для того, чтобы немного оптимизировать свой код, сделала функции визуализации боксов с усами, оценки плотности ядра и построения корреляционной карты (тепловая, более наглядная).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 – функция отрисовки боксов с усами

Изображение выглядит как текст, письмо

Автоматически созданное описание

Рисунок 13 – функция отрисовки оценки плотности ядра

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 14 – функция отрисовки корреляционной матрицы

Результаты выполнения данных функций представлены выше (см. Рисунок 4, Рисунок 5, Рисунок 6)

**Практическая часть**

**2.1 Предобработка данных**

Для начала закодируем признак «Угол нашивки» с помощью LabelEncoder() и сохраним модель кодировки для приложения: Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 – LabelEncoder() для угла нашивки

Далее удалим неинформативный столбец с нумерацией наблюдений:



Рисунок 13 – удаление первого столбца в датафрейме

Изучая боксы с усами, мы нашли в данных большое количество выбросов. Они были почти во всех столбцах, кроме «Угол нашивки». Для итеративного удаления выбросов, напишем функцию, которая будет находить выбросы с помощью IQR и удалять их.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 14 – функция очистки от выбросов

После первого удаления с помощью функции очистки проверяем, остались ли еще выбросы. Смотрим боксы с усами:

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 15 – боксы с усами после первого удаления выбросов

На графиках видим еще выбросы. Выделяем те признаки, в которых остались выбросы, и опять прогоняем функцию очистки. Делаем это до тех пор, пока не избавимся от всех выбросов. В конечном итоге делали мы это четыре раза, пока наш датасет наконец то не очистился. Так датасет сократился до размерности (922, 13), где 922 – это наблюдения, 13 – признаки.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 16 – размер датасета после очистки от выбросов

Поскольку признаки в датасете сильно разбросаны, нам необходимо провести нормализацию и стандартизацию. Методы, которые я использовала: StandardScaler, MinMaxScaler, MaxABSScaler, RobustScaler, Normalizer (стандартный, для строк), Normalizer (для столбцов), а также стандартизировала нормализированный датасет. Для этого, написала функцию трансформирования признаков:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 17 – функция преобразования датасета

Во всех случаях после преобразования необходимо посмотреть на матрицу корреляции, так как связи между признаками очень важны. Все трансформаторы, кроме Normalizer (стандартный, для строк), не показали заметных улучшений в корреляции. Взаимосвязи увидели только с нормализацией:

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 18 – корреляционная карта при Normalizer

Для начала проявилась взаимосвязь между целевыми признаками: «Модуль упругости при растяжении» и «Прочность при растяжении». Корреляция составляет -0,75. Это означает, что эти признаки находятся в обратной зависимости: при увеличении показателя одного признака идет уменьшение значений второго признака. И есть другие признаки, которые коррелируют с целевыми признаками: плотность, содержание эпоксидных групп, температура вспышки, плотность нашивки.

**2.2 Разработка и обучение модели**

В данной части задания нужно разработать и обучить модели для предсказания признаков «Модуль упругости при растяжении» и «Прочность при растяжении». Модели для обучения описаны в части 1.2 Описание используемых моделей.

Прежде, чем вводить данные в модели, нужно выделить целевые признаки, то есть разделить данные на X и y. После этого делим данные на тренировочные выборки и тестовые, на тест уходит 30% всех данных.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 19 – размеры тестовых и тренировочных выборок

Для сохранения оценок модели создадим словари скоров, для оптимизации процесса создадим две функции: первая функция обучает модели и сохраняет в словари ее оценки (r2, mae, mse), вторая функция ведет поиск гиперпараметров по сетке для SVR и RandomForestRegressor

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 20 – словари для хранения оценок моделей

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 21 – функция обучения модели и вывода оценок

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 22 – функция поиска по сетке

Прогнозировать решила два датасета. Первый – это сначала нормализованный датасет, потом еще и стандартизованный. Второй датасет просто стандартизованный. Обучаем модели Линейной регрессии и LassoCV по каждому прогнозируемому признаку отдельно. Для моделей SVR и RandomForestRegressor запускаем поиск оптимальных параметров по сетке (используется GridSearchCV c перекрестной проверкой  **cv=10**) и с новыми параметрами для каждого признака отдельно запускается обучение модели, а потом и предсказание. В функции обучения модели и вывода скора присутствует отрисовка тестовых данных целевой функции и данных, которые смогла предсказать модель.

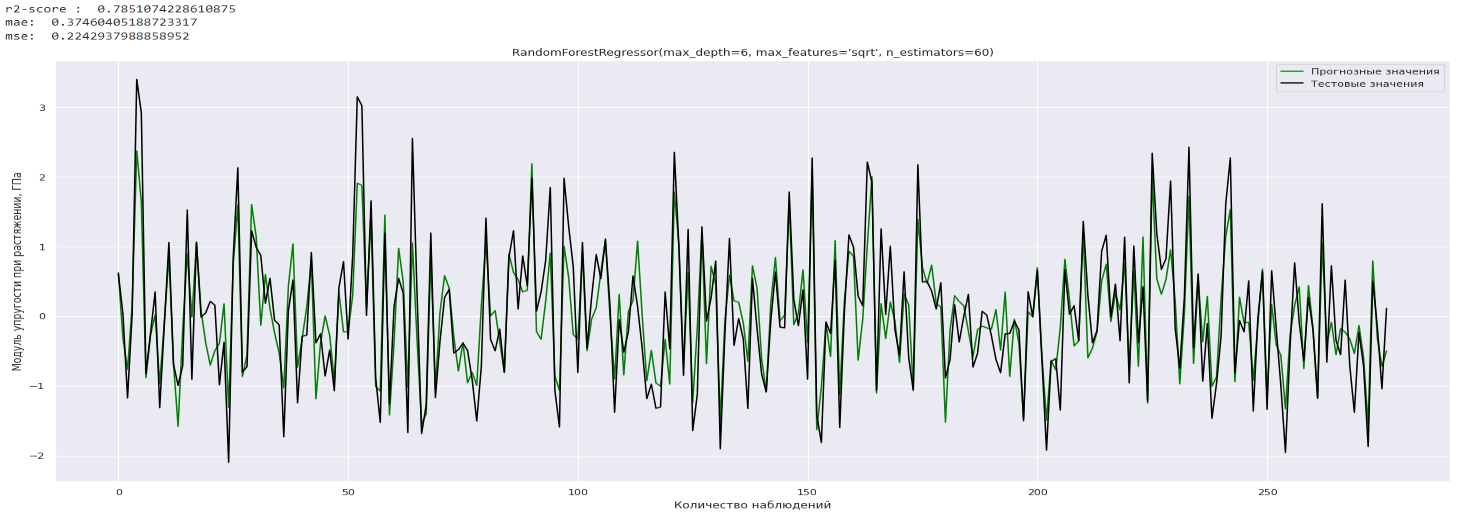


Рисунок 23 – пример отрисовки предикта и тестовых значений для первого датасета

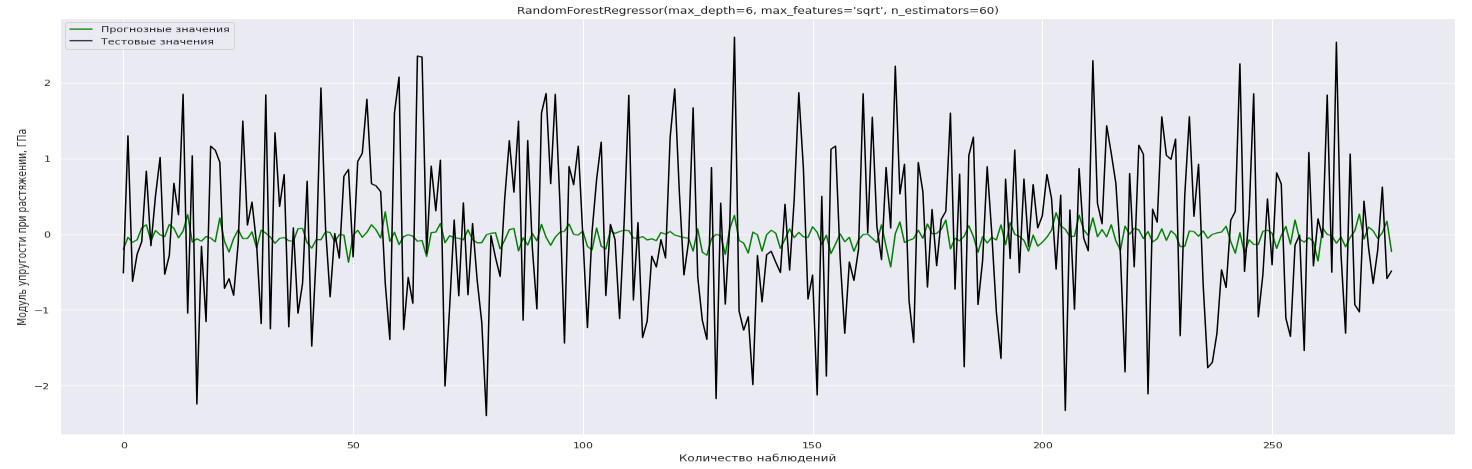


Рисунок 24 - пример отрисовки предикта и тестовых значений для второго датасета

**2.3 Тестирование модели**

После обучения моделей по каждому целевому признаку получили оценки.

Изображение выглядит как текст, письмо

Автоматически созданное описание

Рисунок 25 – r2 по первой выборке

Изображение выглядит как текст, письмо

Автоматически созданное описание

Рисунок 26 – mae по первой выборке

Изображение выглядит как текст, письмо

Автоматически созданное описание

Рисунок 27 - mse по первой выборке

Теперь посмотрим на результаты работы моделей по второй выборке

Изображение выглядит как текст, письмо

Автоматически созданное описание

Рисунок 28 - r2 по второй выборке

Изображение выглядит как текст, письмо

Автоматически созданное описание

Рисунок 29 – mae по второй выборке

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 30 – mse по второй выборке.

Результаты первой выборки больше порадовали, r2 почти по всем моделям стремится к единице, mae и mse наоборот стремятся к нулю. А вот вторая выборка, которую просто стандартизировали, показала не очень хорошие результаты.

**2.4 Нейросеть**

Наша задача состоит в том, чтобы написать нейронную сеть, которая будет рекомендовать соотношение «матрица-наполнитель». Так как это уже другая задача, нам стоит переопределить признаки и целевой признак. Соответственно, исходя из задания, целевой признак – «Соотношение матрица-наполнитель». Возвращаемся к датасету без выбросов и с помощью MinMaxScaler скалируем данные.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 31 - преобразование данных для нейронной сети

Для данной задачи было построено 5 нейронных сетей, от простой линейной модели до многослойного перцептрона. Реализация моделей осуществлялась с помощью класса Sequential.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 32 – построение линейной модели

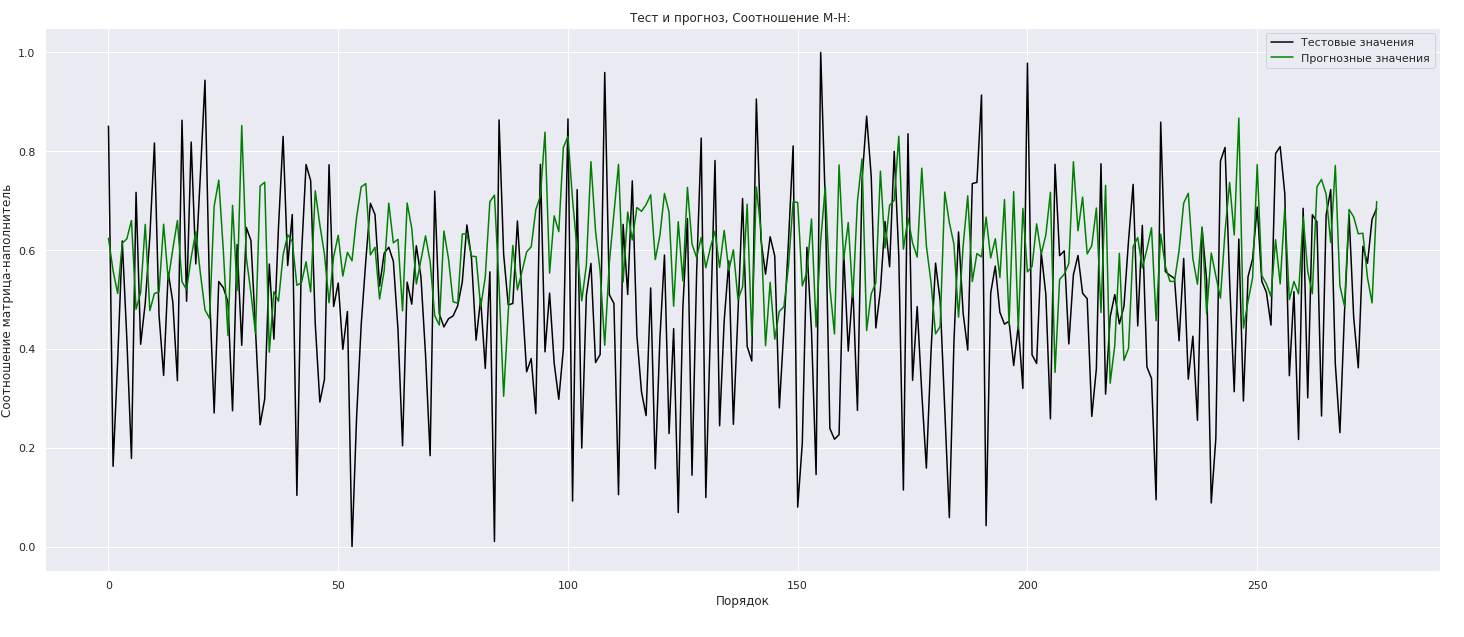


Рисунок 33 – тестовые и прогнозные значения линейной модели

Простая линейная нейронная сеть не сильно справилась с задачей. Попробуем построить более сложную архитектуру нейронной сети.

Добавили в модель слои Dence, убрали normalizer

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 34 – архитектура нейронной сети

После обучения и предикта, посмотрим на график тестовых и прогнозных значений:

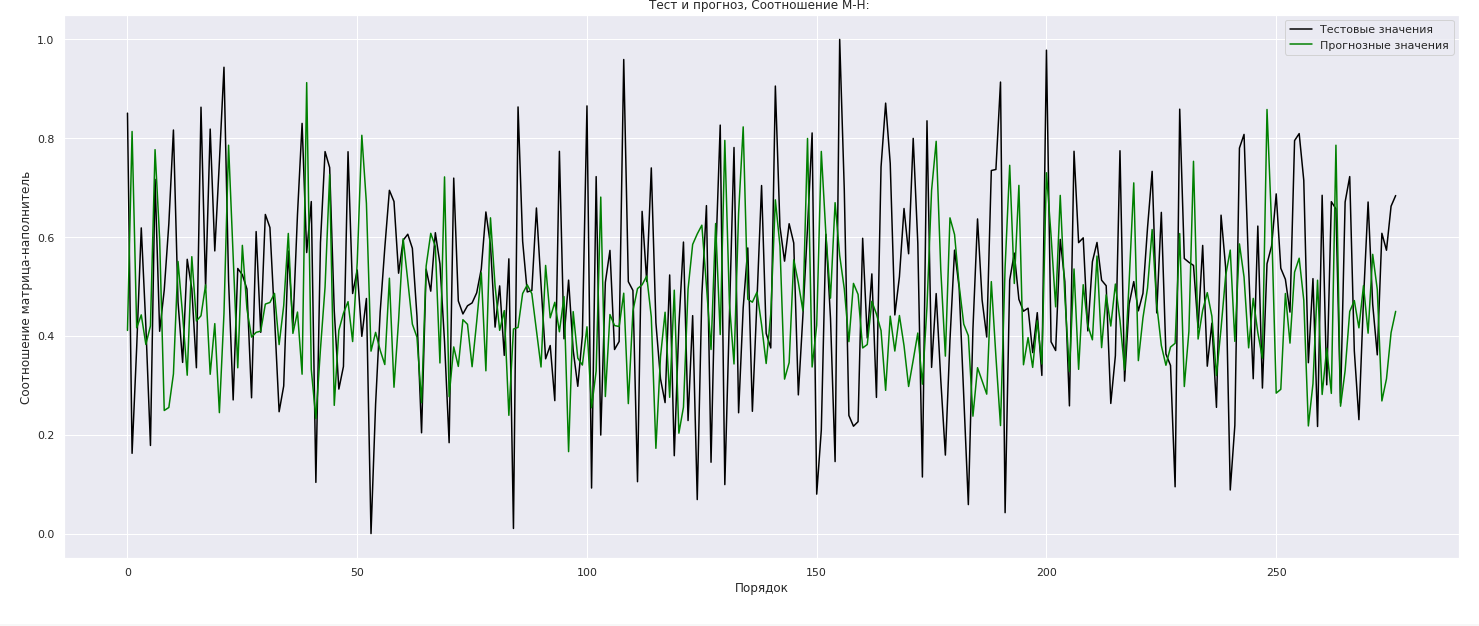


Рисунок 34 – тестовые и прогнозные значения многослойного перцептрона

Построив пять моделей нейросетей, можем оценить качество работы моделей, оценивать будем по метрике mse.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 35 – mse нейросетей

Лучше всего сработала model\_4. Сохраним эту модель для дальнейшего развертывания приложения.

**2.5 Приложение**

Приложение было написано для рекомендации соотношения «матрица-наполнитель».

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 36 – результат работы приложения

Для пользования приложением необходимо перейти по ссылке

<http://91.201.53.227:5000/> , заполнить поля и нажать кнопку «Предсказать»

**3. Заключение**

В данной исследовательской работе был проведен анализ данных композитных материалов для прогнозирования свойств полученного гибридного материала. Можно сказать, что датасет очень сложный, так как стандартными методами обработки данных почти не удалось выявить взаимосвязи между признаками. «Почти не удалось», так как проведенная нормализация с помощью Normalize() была совершена над строками, а не над признаками.

В дальнейшем возможно пересмотреть данный датасет, попробовать выбрать итоговые фичи через MIC, который как раз создавался для выявления нелинейных связей. В работе мы выяснили, что, преобразуя данные, нормируя их и скалируя, корреляция все равно остается около нулевой, что и говорит нам о наличии слабых нелинейных связях.

Также можно будет попробовать внести нелинейные фичи самостоятельно, провернуть что-то вроде Kernel Trick. У нас всего один категориальный признак – угол нашивки – по нему можно разбить датасет на две части и рассматривать их отдельно. Либо вложить это в xgboost, который хорошо работает с такими данными.

**Библиографический список**

1 КМ Редакция Базальтокомпозиты – универсальный материал для инновационных решений: - Режим доступа: <https://compositeworld.ru/articles/tech/id5f3daf50bb1e494c5d7a7a51> (дата обращения: 20.03.2023)

2 Грас Дж. Data Science. Наука о данных с нуля: Пер. с англ. – СПб.: БХВ-Петербург, 2020. – 336 с:

3 Мэтиз Эрик Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. – СПб.: Питер, 2020. – 512 с.: ил. – (Серия «Библиотека программиста»).

4 Траск Эндрю. Грокаем глубокое обучение. – СПб.: Питер, 2019. – 352 с.: ил.  — (Серия «Библиотека программиста»).

5 Документация библиотеки preprocessing: - Режим доступа: <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing> (дата обращения: 02.03.2023)

6 Документация библиотеки Linear regression: - Режим доступа: <https://scikit-learn.org/stable/modules/linear_model.html> (дата обращения: 02.03.2023)

6 Документация библиотеки LassoCV: - Режим доступа: <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoCV.html> (дата обращения: 03.03.2023)

7 Документация библиотеки SVR: - Режим доступа: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html> (дата обращения: 03.03.2023)

8 Документация библиотеки RandomForestRegressor: - Режим доступа: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html> (дата обращения: 03.03.2023)