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 Задубровский Н. Н.

Москва, 2022

# Содержание

Оглавление

[Содержание 2](#_Toc106363906)

[Введение 3](#_Toc106363907)

[1. Аналитическая часть 4](#_Toc106363908)

[1.1. Постановка задачи 4](#_Toc106363909)

[1.2. Описание представленного датасета 5](#_Toc106363910)

[1.3. Описание используемых методов 6](#_Toc106363911)

[1.4. Разведочный анализ данных 9](#_Toc106363912)

[2. Практическая часть 11](#_Toc106363913)

[2.1. Предобработка данных 11](#_Toc106363914)

[2.2. Разработка и обучение моделей 17](#_Toc106363915)

[2.3. Нейронная сеть, рекомендующая соотношение матрица-наполнитель 20](#_Toc106363916)

[2.4. Разработка приложения 22](#_Toc106363917)

[Было разработано приложение с web интерфейсом, позволяющее пользователю ввести значения параметров и после нажатии на кнопу, выдающее рекомендацию по соотношению Матрица-наполнитель: 22](#_Toc106363918)

[Интерфейс приложения приведен на рисунке: 22](#_Toc106363919)

[2.5. Создание удаленного репозитория и загрузка результатов работы на него. 22](#_Toc106363920)

[3. Заключение 23](#_Toc106363921)

[4. Библиографический список 24](#_Toc106363922)

Введение

Достаточно известно определение, согласно которому: композиты – это материалы, состоящие из двух или более компонентов (армирующих элементов и скрепляющей их матрицы) и обладающие свойствами, отличными от суммарных свойств компонентов. При этом предполагается, что компоненты, входящие в состав композита, должны быть хорошо совместимыми и не растворяться или иным способом поглощать друг друга.

Композиционный материал – это любой материал с гетерогенной структурой, т. е. со структурой, состоящей минимум из двух фаз. к композитам относятся материалы, обладающие рядом признаков:

1. состав, форма и распределение компонентов материала «запроектированы заранее»;

2. материал не встречается в природе, а создан человеком;

3. материал состоит из двух или более компонентов, различающихся по химическому составу и разделенных выраженной границей;

4. свойства материала определяются каждым из его компонентов, которые должны присутствовать в материале в достаточно больших количествах (больше некоторого критического содержания);

5. материал обладает такими свойствами, которых не имеют его компоненты, взятые в отдельности;

6. материал неоднороден в микромасштабе и однороден в макромасштабе.

Классификация КМ по материалу матрицы (материаловедческий принцип) Наиболее важным признаков классификации КМ является материал матрицы. КМ с металлической матрицей называют металлическими композиционными материалами (МКМ), с полимерной матрицей – полимерными композиционными материалами (ПКМ), с керамической – керамическими композиционными материалами (ККМ). КМ, содержащие два и более раз

1. Аналитическая часть
   1. Постановка задачи

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

<https://drive.google.com/file/d/1B1s5gBlvgU81H9GGolLQVw_SOi-vyNf2/view?usp=sharing>

* 1. Описание представленного датасета

Датасет представлен в виде двух файлов в формате xlsx:

* X\_bp.xlsx

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |
| 0 | 1,857142857 | 2030 | 738,7368421 | 30 | 22,26785714 | 100 | 210 | 70 | 3000 | 220 |
| 1 | 1,857142857 | 2030 | 738,7368421 | 50 | 23,75 | 284,6153846 | 210 | 70 | 3000 | 220 |
| 2 | 1,857142857 | 2030 | 738,7368421 | 49,9 | 33 | 284,6153846 | 210 | 70 | 3000 | 220 |
| 3 | 1,857142857 | 2030 | 738,7368421 | 129 | 21,25 | 300 | 210 | 70 | 3000 | 220 |

Файл содержит 11 колонок и 1024 строки. Первая колонка содержит индексные значения.

* X\_nup.xlsx

|  |  |  |  |
| --- | --- | --- | --- |
|  | Угол нашивки, град | Шаг нашивки | Плотность нашивки |
| 0 | 0 | 4 | 57 |
| 1 | 0 | 4 | 60 |
| 2 | 0 | 4 | 70 |
| 3 | 0 | 5 | 47 |

Файл содержит 4 колонки и 1041 строку

Количество строк в файлах различается. По условию задачи данные из файлов нужно объединить методом INNER, по индексным значениям, расположенным в первом столбце. При таком объединении количество строк в итоговом датасете будет равна количеству строк с одинаковым индексом в обоих фалах. Остальные строки будут исключены – Рис.1

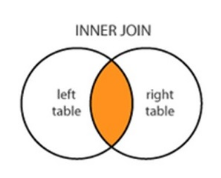


Рис.1 Объединение данных методом INNER

Сразу можно отметить о наличии высокой точности данных в датасете, например параметр «Модуль упругости», имеющем значения в сотни Гпа, имеет точность 12 знаков после запятой. Возникает вопрос – это точность достигается путем измерения величины, или, вследствие математических вычислений/преобразований над данными?

После объединения данных из представленных файлов, получаем выборку размерностью 13колонок \* 1023 строки. Строки, не совпадающие по значению индекса, отсеялись на этапе объединения данных. Пропуски в данных отсутствуют.

* 1. Описание используемых методов

Для прогнозирования модуля упругости при растяжении и прочности при растяжении мы будем решать задачу регрессии. Создание регрессионной модели представляет собой итерационный процесс, направленный на поиск эффективных независимых переменных, чтобы объяснить зависимые переменные, которые мы пытаемся смоделировать. Затем пошаговое удаление и/или добавление переменных до тех пор, пока не будет найдена наилучшим образом подходящую регрессионную модель.

В рамках этой задачи мы будем использовать следующие модели:

* Линейная регрессия

Линейную регрессию можно представить в виде уравнения, которое описывает прямую, наиболее точно показывающую взаимосвязь между входными переменными X и выходными переменными Y (Рис2). Для составления этого уравнения нужно найти определённые коэффициенты B для входных переменных. Например: Y = B0 + B1 \* X. Зная X, мы должны найти Y, и цель линейной регрессии заключается в поиске значений коэффициентов B0 и B1.

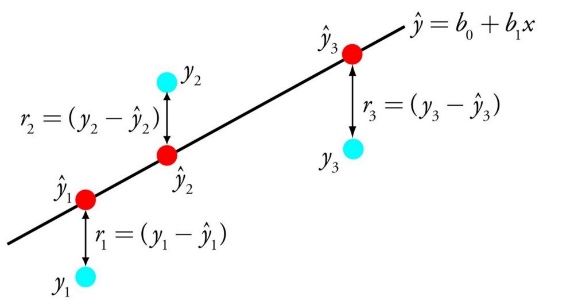


Рис.2

* Бутстрэп-агрегирование или бэггинг

Бэггинг – это простой и очень мощный ансамблевый метод, который объединяет прогнозы из нескольких методов Машинного обучения вместе, чтобы предсказывать более точно, чем любая отдельная модель.

Бутстрэп-агрегирование – это процедура, которая используется для сокращения чрезмерной дисперсии (Variance) алгоритмов – Деревьев решений (Decision Tree), таких как Алгоритм классификации и регрессии (CART).

Метод бутстрэпа заключается в следующем: Пусть имеется выборка размера  n\*m. Равномерно возьмем из выборки k объектов с возвращением. Это означает, что мы будем k раз выбирать произвольный объект выборки (считаем, что каждый объект «достается» с одинаковой вероятностью), причем каждый раз мы выбираем из всех исходных объектов.

* Случайный решающий лес

[Случайный лес](https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm) — модель, состоящая из ансамбля деревьев решений. Вместо того, чтобы просто усреднять прогнозы разных деревьев (такая концепция называется просто «лес»), эта модель использует [две ключевые концепции](https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf), которые и делают этот лес случайным:

* Случайная выборка образцов из набора данных при построении деревьев.
* При разделении узлов выбираются случайные наборы параметров.
* Случайная выборка тренировочных образцов

В процессе тренировки каждое дерево случайного леса учится на случайном образце из набора данных. Выборка образцов происходит с возмещением (в статистике этот метод называется бутстреппинг, bootstrapping). Это даёт возможность повторно использовать образцы одним и тем же деревом. Хотя каждое дерево может быть высоковариативным по отношению к определённому набору тренировочных данных, обучение деревьев на разных наборах образцов позволяет понизить общую вариативность леса, не жертвуя точностью.

При тестировании результат выводится путём усреднения прогнозов, полученных от каждого дерева. Подход, при котором каждый обучающийся элемент получает собственный набор обучающих данных (с помощью бутстреппинга), после чего результат усредняется, называется бэггинг (bagging, от [bootstrap aggregating](https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/" \t "_blank)).

По условию задачи, необходимо провести поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10. Кросс-валидация (перекрестная проверка) – это метод оценки Моделей (Model) Машинного обучения (ML) путем обучения нескольких из них на подмножествах доступных входных данных и их оценки на другом дополнительном подмножестве. Такая проверка используется для обнаружения Переобучения (Overfitting), т.е. неспособности распознать паттерн.

Для оценки качества работы моделей будем использовать следующие метрики:

Mean Absolute Error (MAE)

Метрика измеряет среднюю сумму абсолютной разницы между фактическим значением и прогнозируемым значением.

Mean Absolute 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. А также в силу свойств этой метрики, усиливается влияние ошибок, по квадратуре от исходного значения. Это значит, что если в исходных измерениях мы ошиблись на 1, то метрика покажет 1, 2-4, 3-9 и так далее. Чем меньше MSE, тем точнее наше предсказание. Оптимум достигается в точке 0, то есть предсказание произведено без погрешности.

При построении моделей руководствуемся следующим Pipeline машинного обучения

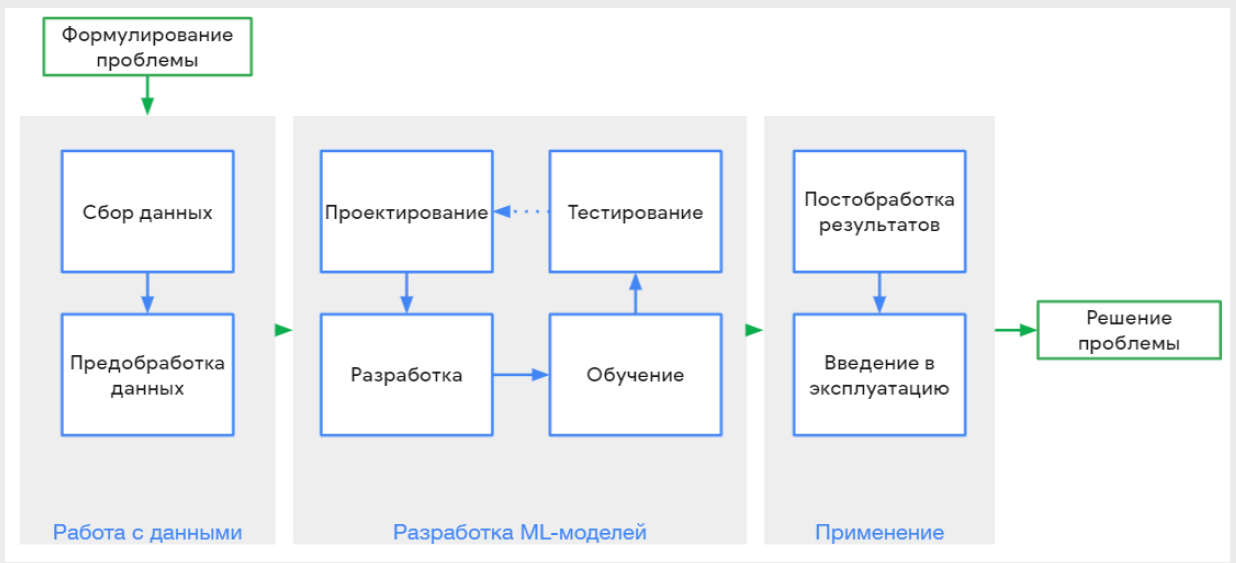


Рис. 4

* 1. Разведочный анализ данных

Разведочный анализ данных (англ. *exploratory data analysis, EDA*) — анализ основных свойств данных, нахождение в них общих закономерностей, распределений и аномалий, построение начальных моделей, зачастую с использованием инструментов визуализации.

Для разведочного анализа данных будем использовать следующие методы:

* Выведем среднее значение признаков
* Выведем медианное значение признаков (это значение элемента в центре ранжированного ряда)
* Проверим данные на наличие пропусков и выбросов
* Выведем описательную статистику по датафрейму
* Построим гистограммы распределения параметров - график частотной таблицы.
* Построим boxplot(ящик с усами по каждому признаку)

Вот, что показывает боксплот:

Медиана – это значение элемента в центре ранжированного ряда.  
Например, если всех осьминогов выставить в порядке возрастания их оценок, то медианой будет та оценка, которую поставил осьминог в середине. А это значит, что половина осьминогов справа оценили вероятность покупки ниже, а другая половина (слева) выше, чем медианный.  
Медиана меньше подвержена влиянию выбросов, поэтому в центре отображается именно она, а не среднеарифметическое.

Верхний квартиль – это такая оценка, выше которой только 25% оценок.

Нижний квартиль – это такое значение, ниже которого только 25% оценок.

Межквартильный размах (МКР) – это разница между 75% и 25% квартилем. Внутри этого диапазона лежит 50% наблюдений. Если диапазон узкий (как в случае с осьминогами), значит члены подгруппы единогласны в своих оценках. Если широкий – значит однородного мнения нет (как у цыплят).

Выбросы – это нетипичные наблюдения.

Выбросы – это значения за пределами:

25% перцентили минус 1.5 х МКР

75% перцентили плюс 1.5 х МКР

* Проверим на наличие дубликатов данных
* Получим дисперсию по признакам
* Построим попарные диаграммы рассеяния
* Выведем тепловую карту ковариационной матрицы. Ковариация — это мера, которая количественно определяет силу и направление взаимосвязи между парой переменных. Соответственно, ковариационная матрица – табличное представление взаимосвязи между параметрами датасета. Тепловая карта может быть использована для визуального отображения матрицы. Цвета представляют числа или элементы матрицы.

После анализа данных выполним предобработку датасета:

* Поиск и устранение пропусков данных. Для эффективного обучения и предсказания не должно быть отсутствующих значений в датасете. В случае, их присутствия, есть несколько методов борьбы с ними: Удаление/заполнение новыми значениями
* Поиск и устранение полных дублей данных. Полные дубли могут так же негативно сказаться на процессе обучения.
* Поиск и устранение выбросов (Выброс - это элемент данных / объект, который значительно отличается от остальных (так называемых нормальных) объектов.).

1. Практическая часть
   1. Предобработка данных

Ниже будут приведены примеры кода, используемого при предобработке данных:

1. Проверяем количество пропусков в данных:

for p in Df.columns:

  sum\_isna=Df[p].isna().sum()

  if sum\_isna>0:

    print("!!!  Количество пропусков в столбце " + p + "=" + str(sum\_isna))

  else:

    print("     Пропуски отсутствуют в столбце " + p)

В результате выполнения не обнаруживаем пропусков. Количество заполненных значений по каждому параметру соответствует общему количеству строк первого файла. В исходных тпблицах количество строк не совпадает, соответвтенно при объединении по inner по индексу сразу отсеиваются пропуски в одной из таблиц.

1. Выводим среднее значения по колонкам датасета:

np.mean(Df)

Соотношение матрица-наполнитель 2.9304

Плотность, кг/м3 1975.7349

модуль упругости, ГПа 739.9232

Количество отвердителя, м.% 110.5708

Содержание эпоксидных групп,%\_2 22.2444

Температура вспышки, С\_2 285.8822

Поверхностная плотность, г/м2 482.7318

Модуль упругости при растяжении, ГПа 73.3286

Прочность при растяжении, МПа 2466.9228

Потребление смолы, г/м2 218.4231

Угол нашивки, град 44.2522

Шаг нашивки 6.8992

Плотность нашивки 57.1539

dtype: float64

1. Выводим медианные значения по столбцам:

for p in Df.columns:

  median\_col = np.median(Df[p])

  print(str(p)+" ["+str(median\_col)+ "]")

Соотношение матрица-наполнитель [2.90687765033521]

Плотность, кг/м3 [1977.62165679058]

модуль упругости, ГПа [739.664327697792]

Количество отвердителя, м.% [110.564839894065]

Содержание эпоксидных групп,%\_2 [22.2307437560244]

Температура вспышки, С\_2 [285.896812331237]

Поверхностная плотность, г/м2 [451.86436518306]

Модуль упругости при растяжении, ГПа [73.2688045943481]

Прочность при растяжении, МПа [2459.52452600309]

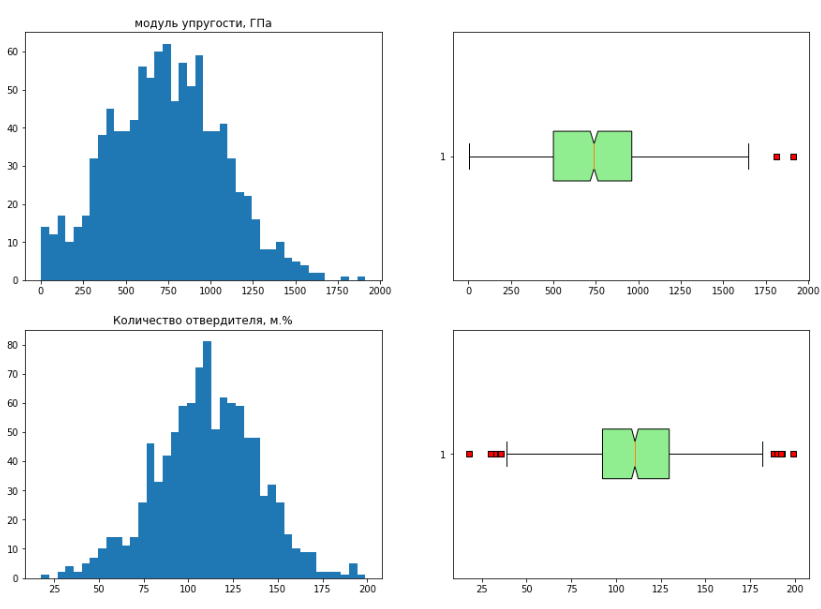
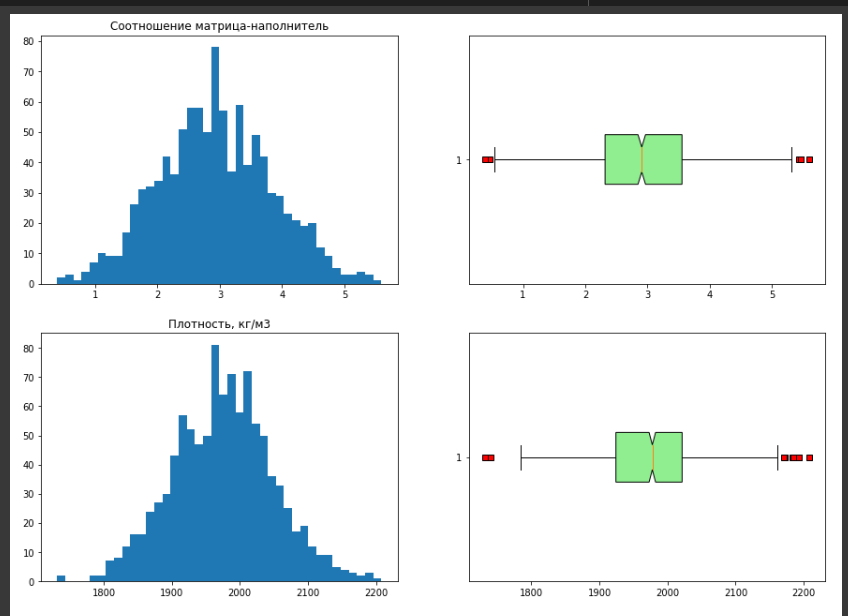
Потребление смолы, г/м2 [219.198882195134]

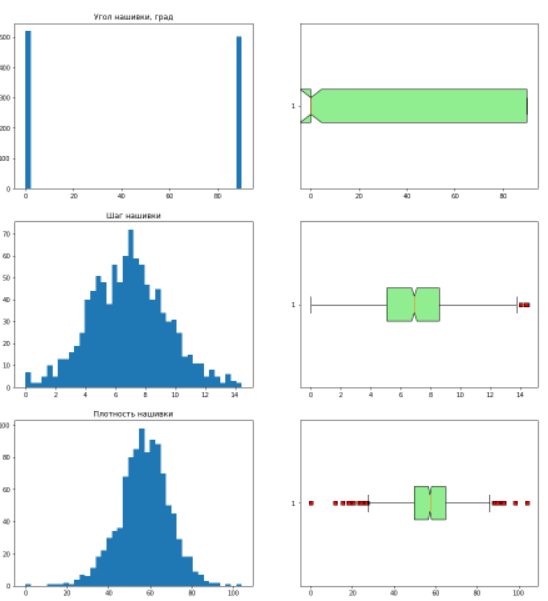
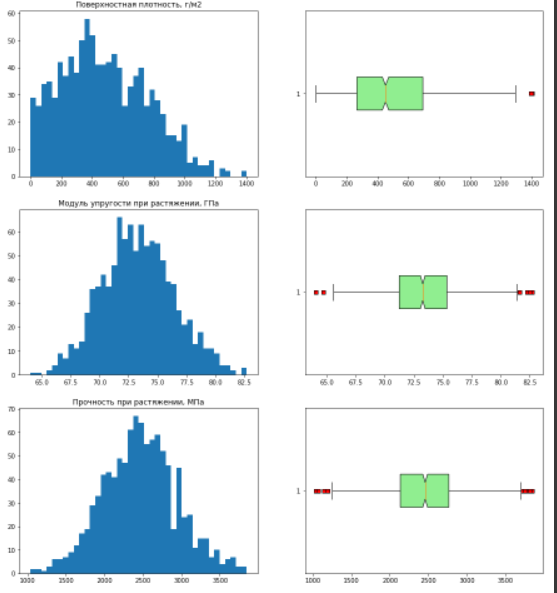
Угол нашивки, град [0.0]

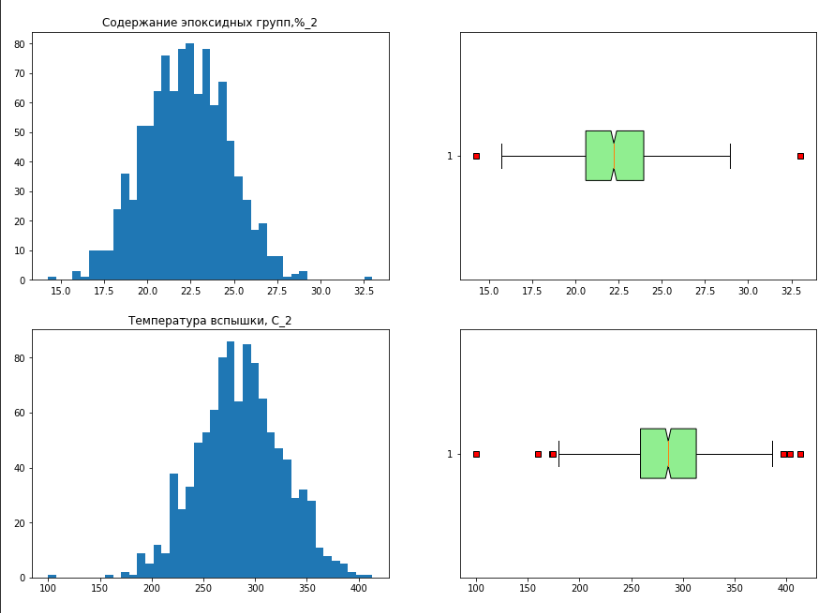
Шаг нашивки [6.9161438559491]

Плотность нашивки [57.3419198469929]

1. Строим гистограммы распределения признаков и boxplot(ящик с усами):







1. Найдем дисперсию:

Df.var()

Соотношение матрица-наполнитель 0.833974852717230

Плотность, кг/м3 5435.999437591476635

модуль упругости, ГПа 109052.896799833994010

Количество отвердителя, м.% 800.658595663007304

Содержание эпоксидных групп,%\_2 5.790285905616200

Температура вспышки, С\_2 1676.350535572652689

Поверхностная плотность, г/м2 79137.954942948505050

Модуль упругости при растяжении, ГПа 9.728054264802276

Прочность при растяжении, МПа 235834.560482065746328

Потребление смолы, г/м2 3568.381437302487939

Угол нашивки, град 2026.421656116730901

Шаг нашивки 6.571363433503515

Плотность нашивки 152.546430265258238

1. Строим попарные диаграммы рассеяния

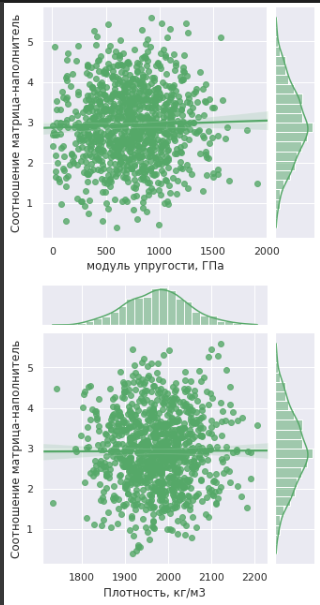
sns.pairplot(Df)

Изображение выглядит как текст

Автоматически созданное описание

А так же jointplot

sns.jointplot(x="модуль упругости, ГПа", y="Соотношение матрица-наполнитель", data=Df, kind="reg", truncate=False, color="g", height=5)

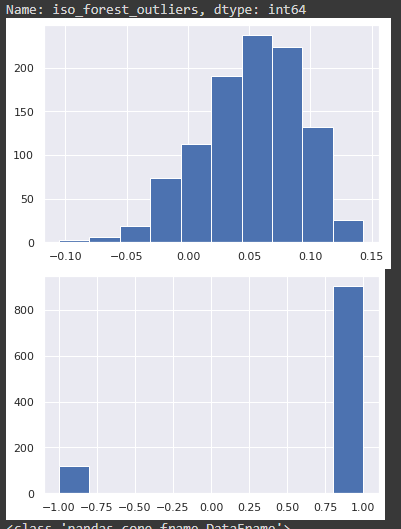


1. Строим Тепловую карта ковариоционной матрицы



По результатам анализа выявлено, что пропуски в данных отсутствуют, все параметры распределены по нормальному распределению, кроме Угла нашивки, который принимает всего два значения. В данных присутствуют выбросы. Параметры не имеют явной линейной корреляции. Все параметры кроме Угла нашивки имеют вещественный тип с очень высокой точностью. Например, величина Модуль упругости, измеряемый в сотнях ГПа имеет точность 12 знаков после запятой? Что это - невероятная точность измерения или погрешность вычисления? Такая точность, при условии отсутствия уникальных значений вряд ли будет способствовать эффективному прогнозированию. Требуется проверить данные на наличие шумов.

1. Устраним выбросы - применяем метод Случайного леса:



Найдено и устранено 120 выбросов в датасете

1. Проведем нормализацию признаков датафрейма

minmax\_sc = MinMaxScaler()

Df\_norm = minmax\_sc.fit\_transform(np.array(Df\_dd))

len(Df\_norm[0])

Изображение выглядит как текст

Автоматически созданное описание

* 1. Разработка и обучение моделей

Перед тем как начать обучать модели, разобьем наш датафрейм на обучающую и тестовую выборки в соотношении 70/30:

Изображение выглядит как текст

Автоматически созданное описание

* Линейная регрессия с кроссвалидацией

#Линейная регрессия

LR = LinearRegression()

results = cross\_val\_score(LR, x\_train\_data, y\_train\_data, cv=10, scoring='r2')

print(results)

LR.fit(x\_train\_data, y\_train\_data)

y\_pred = LR.predict(x\_test\_data)

score\_LR = LR.score(x\_test\_data, y\_test\_data)

MSE = mean\_squared\_error(y\_test\_data,y\_pred)

MAE = mean\_absolute\_error(y\_test\_data,y\_pred)

Результаты построенной модели приведены ниже:

score R2= -0.02337938859321631

MSE= 0.028086876779022496

MAE= 0.13976283155059643

На рисунке представлены результаты предсказания(линия красного цвета) и тестовая выборка(линия зеленого цвета):



* RandomForestRegressor

При построении этой модели использовали сетку гиперпараметров по сетке с количеством блоков равно 10:

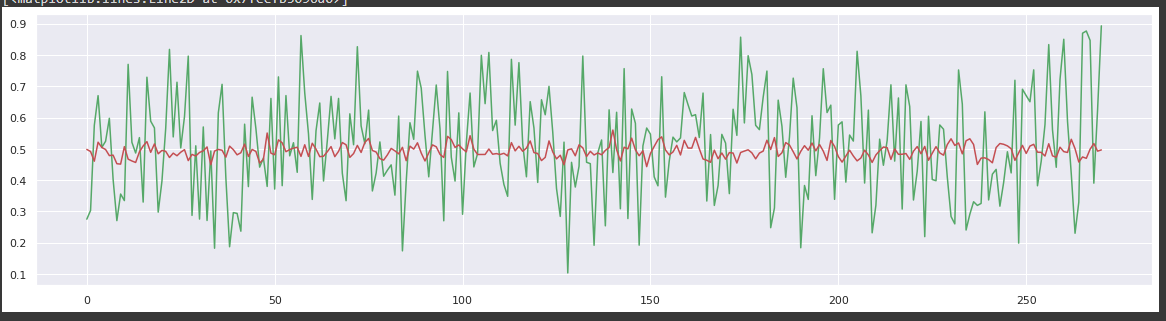
grid = {'n\_estimators': [10, 100, 800], 'bootstrap': [False, True], 'max\_features': [0.15, 10, "log2"],'min\_samples\_leaf':[1,10,20], 'min\_samples\_split': [2,5]}

score R2= -0.02337938859321631

MSE= 0.02815245882194944

MAE= 0.14062959239136083

На рисунке представлены результаты предсказания(линия красного цвета) и тестовая выборка(линия зеленого цвета):



* Ridge\_regression

odel = RidgeCV(alphas = [0.001, 0.1, 1], cv = 10, gcv\_mode='auto').fit(x\_train\_data, y\_train\_data)

score\_RR = model.score(x\_train\_data,y\_train\_data)

y\_pred= model.predict(x\_test\_data)

MSE\_RR = mean\_squared\_error(y\_test\_data,y\_pred)

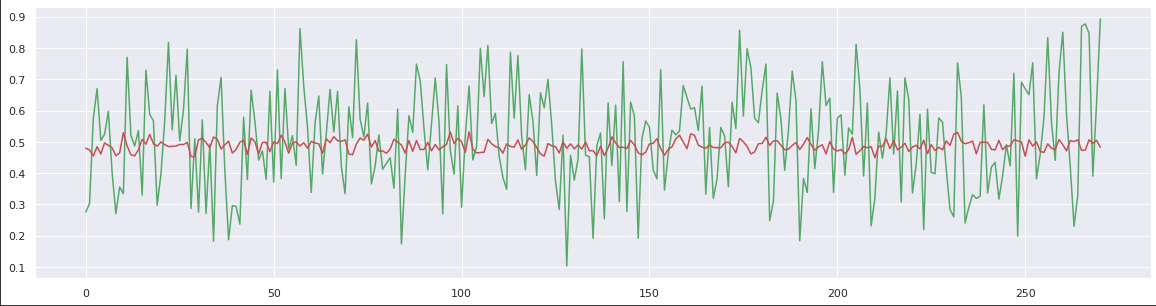
MAE\_RR = mean\_absolute\_error (y\_test\_data,y\_pred)

score R2= -0.0217530038466639

MSE= 0.02806899223820489

MAE= 0.13977598696938653

На рисунке представлены результаты предсказания (линия красного цвета) и тестовая выборка(линия зеленого цвета):



* BaggingRegressor

model=BaggingRegressor()

grid = {'n\_estimators': [10, 50, 100], 'random\_state': [5, 42, 80], 'max\_features': [1, 5, 10]}

GSCV\_BR=GSCV\_fit(model, grid, x\_train\_data, y\_train\_data)

Лучшими параметрами были выявлены BaggingRegressor(max\_features=1, n\_estimators=100, random\_state=5)

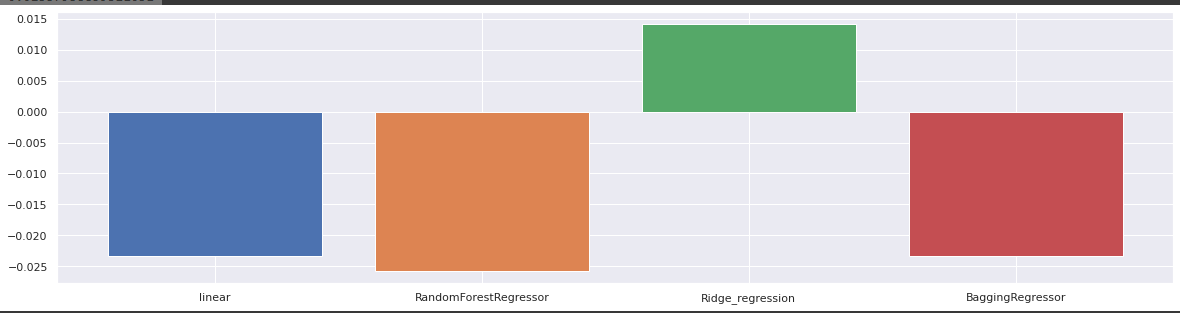
score R2= -0.0217530038466639

MSE\_BR= 0.027735202312479005

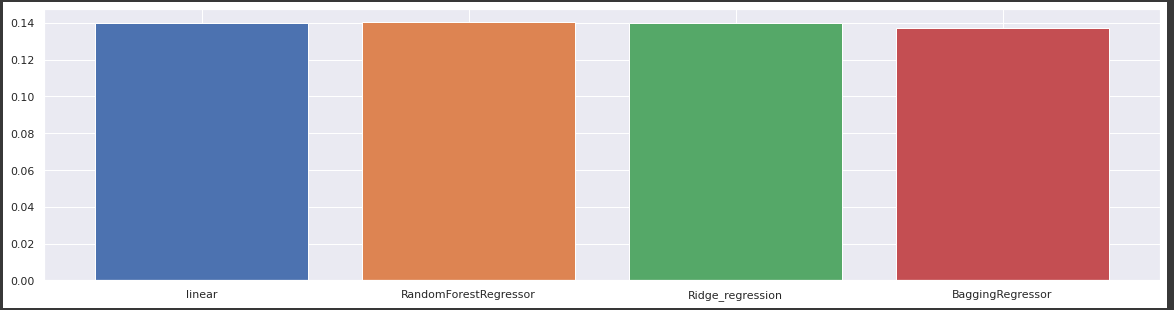
MAE\_BR= 0.13736473773675692

Сравним результаты работы моделей по метрикам

Метрика R\_2:



МАЕ:



MSE:

* 1. Нейронная сеть, рекомендующая соотношение матрица-наполнитель

Иключаем из признако «Угол нашивки, град», так как он имеет всего лишь два значения, что на мой взгляд будет негативно сказываться на обучении сети.

Нейронную сеть для прогнозирования отношения Матрица-наполнитель будем строить на базе библиотеки keras. Конфигурация модели была определена опытным путем, с условием что количество нейроной первого скрытого слоя должно соответствоавать количеству переменных. В нашем случае – 11. Количество слоев, для прогнозирования нелинейных зависимостей, должно быть более 2. Так же производим Dropout после скрытых слоев

Код нейронной сети:

model = tf.keras.models.Sequential()

model.add(Dense(11, activation ='relu'))

model.add(Dropout(0.1))

model.add(Dense(6, activation ='relu'))

model.add(Dropout(0.2))

model.add(Dense(1, activation ='relu'))

model.compile(loss = 'mse', optimizer = 'adam', metrics=['mae'])

early\_stop = EarlyStopping(monitor='val\_loss', mode='min', verbose=1, patience=3, restore\_best\_weights=True)

result = model.fit(x\_train\_n,

                    y\_train\_n,

                    batch\_size = 64,

                    epochs = 100,

                    validation\_data = (x\_test\_n, y\_test\_n),

                    verbose = 1,

                    callbacks=early\_stop)

* 1. Разработка приложения

Было разработано приложение с web интерфейсом, позволяющее пользователю ввести значения параметров и после нажатии на кнопу, выдающее рекомендацию по соотношению Матрица-наполнитель:

Интерфейс приложения приведен на рисунке:

Изображение выглядит как текст

Автоматически созданное описание

* 1. Создание удаленного репозитория и загрузка результатов работы на него.

Был создан репозиторий на GitHub, который можно просмотреть по ссылке:

[https://github.com/](https://github.com/EvgeniyBorsch/ML-VKR)ZadubrovskiyN/VKR

В нем находятся коды, презентации, а также создан Readme.txt файл с описанием содержания. На рисунке 17 представлен скриншот страницы.

1. Заключение

В ходе выпускной квалификационной работы был выполнен анализ предоставленного датасета. Данные в датасете сильно зашумлены, вследствии чего построить модель с высокой точностью прогнозирования не удалось. Все модели выдают результат близкий к среднему значения искомой переменной.

1. Библиографический список

* «Полимерные Композиционные Материалы», Л.И. Бондалетова, В.Г. Бондалетов, Издательство Томского политехнического университета 2013
* Конспект Лекций По Дисциплине «Механика композиционных материалов» Ю.В. Скворцов, Самара 2013
* «Классификация, регрессия и другие алгоритмы Data Mining с использованием R»В.К. Шитиков, С.Э. Мастицкий, Тольятти, Лондон – 2017
* «Начинаем программировать на Python», Геддис Т., Санкт-Петербург 2019, 4 изд.
* «Чистый Python. Тонкости программирования для Профи», Ден Байдер, Питер 2018
* «Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными.» Вильямс, 2017
* «Крупномасштабное машинное обучение вместе с Python», [Бастиан Шарден](https://eg1lib.org/g/%D0%91%D0%B0%D1%81%D1%82%D0%B8%D0%B0%D0%BD%20%D0%A8%D0%B0%D1%80%D0%B4%D0%B5%D0%BD), [Лука Массарон](https://eg1lib.org/g/%D0%9B%D1%83%D0%BA%D0%B0%20%D0%9C%D0%B0%D1%81%D1%81%D0%B0%D1%80%D0%BE%D0%BD), [Альберто Боскетти](https://eg1lib.org/g/%D0%90%D0%BB%D1%8C%D0%B1%D0%B5%D1%80%D1%82%D0%BE%20%D0%91%D0%BE%D1%81%D0%BA%D0%B5%D1%82%D1%82%D0%B8)
* "Грас, Джоэл. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербурr, 2021. - 416 с.: ил