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 Олейник Владимир Александрович

Москва, 2023

Оглавление

Введение

Data Science (наука о данных) – раздел информатики, изучающий проблемы анализа, обработки и представления данных в цифровой форме. Объединяет методы по обработке данных в условиях больших объёмов и высокого уровня параллелизма, статистические методы, методы интеллектуального анализа данных и приложения искусственного интеллекта для работы с данными, а также методы проектирования и разработки баз данных.

Основная практическая цель профессиональной деятельности в науке о данных - обнаружение закономерностей в данных, извлечение знаний из данных в обобщённой форме для дальнейшего использования в различных целях.

Результаты, полученные с помощью науки о данных, используются для решения задач в разных отраслях. Одной из таких задач, рассмотренной в данной работе, является прогнозирование конечных свойств новых материалов (композиционных материалов).

Композиционный материал - многокомпонентный материал, изготовленный (человеком или природой) из двух или более компонентов с существенно различными 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 и наполнитель, который выполняет функцию армирования. Варьируя состав матрицы и наполнителя, их соотношение, ориентацию наполнителя, получают широкий спектр материалов с требуемым набором свойств.

При изготовлении композиционных материалов имеется недостаток - при наличии, сведений об исходных характеристиках компонентов, достаточно проблематично определить конечные характеристики нового материала, состоящего из этих компонентов. Для решения этой проблемы есть два пути: физические испытания образцов материалов, или прогнозирование характеристик на основе данных о характеристиках входящих компонентов.

1 Аналитическая часть

1.1 Постановка задачи

Для решения задачи прогнозирования конечных свойств новых материалов предоставлены два файла Microsoft Excel, содержащие данные о некоторых исходных характеристиках компонентов и конечных характеристиках композиционного материала.

Краткое описание файлов:

1) X_bp, состоящий из 11 столбцов и 1024 строк, включая столбец индекса и строки наименования столбцов;

2) X_nup, состоящий из 4 столбцов и 1041 строк, включая столбец индекса и строки наименования столбцов.

Требуется:

1) обучить нескольких моделей для прогноза модуля упругости при растяжении и прочности при растяжении композиционных материалов;

2) написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель.

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

В результате объединения файлов по индексу с использованием INNER JOIN (внутреннее соединение - объединяются только строки, имеющие одинаковый индекс) получен датасет размерностью 1023 строки и 13 колонок (признаков).

Пропуски в данных отсутствуют. Все признаки имеют числовой тип данных. Исходя из задачи, признаки разделены на десять входных и три выходных переменных. Сводная информация о датасете указана в таблице 1.

Таблица 1 – сводная информация о датасете

Наименование колонки (признака)	Тип переменной	Количество непустых значений	Тип данных	Количество уникальных значений
Соотношение матрица-наполнитель	выходной	1023	float64	1014
Плотность, кг/м3	входной	1023	float64	1013
Модуль упругости, ГПа	входной	1023	float64	1020
Количество отвердителя, м.%	входной	1023	float64	1005
Содержание эпоксидных групп,%_2	входной	1023	float64	1004
Температура вспышки, С_2	входной	1023	float64	1003
Поверхностная плотность, г/м2	входной	1023	float64	1004
Модуль упругости при растяжении, ГПа	выходной	1023	float64	1004
Прочность при растяжении, МПа	выходной	1023	float64	1004
Потребление смолы, г/м2	входной	1023	float64	1003
Угол нашивки, град	входной	1023	int64	2
Шаг нашивки	входной	1023	float64	989
Плотность нашивки	входной	1023	float64	988

1.2 Описание используемых методов

Предсказание численной характеристики объекта предметной области по определенному набору признаков – это задача регрессии, которая относится к категории задач обучения с учителем.

Регрессионная модель – это уравнение, в котором объясняемая переменная (целевой признак) представляется в виде функций от объясняющих переменных признаков.

Для решения поставленной задачи будут рассмотрены следующие методы построения регрессионной модели:

- линейная регрессия методом наименьших квадратов;
- дерево принятия решений;
- метод случайного леса;
- метод опорных векторов;

- метод k-ближайших соседей;
- нейронная сеть.

1.2.1. Линейная регрессия методом наименьших квадратов

Линейная регрессия - это регрессионная модель зависимости одной (целевой) переменной от другой или нескольких других переменных (признаков) с линейной функцией зависимости, которая, согласно определённым математическим критериям, наиболее соответствует данным.

При наличии одной входной переменной линейная регрессия называется простой, когда входных переменных несколько – множественной.

Из множества методов обучения линейных регрессионных моделей наиболее распространенным является метод наименьших квадратов. С помощью данного метода вычисляется прямая (гиперплоскость), сумма квадратов между которой и данными минимальна.

Линейная регрессия является линейной моделью, которая предполагает линейную связь между входными переменными и выходной переменной. Эта связь измеряется путем выявления изменений выходной переменной при изменении входных переменных.

Достоинства метода:

- простота реализации;
- эффективность при линейных зависимостях;
- простота интерпретации.

Недостатки метода:

- ограниченная эффективность при нелинейных зависимостях;
- чувствительность к выбросам.

Линейная регрессия наименьших квадратов реализована в библиотеке scikit-learn - sklearn.linear_model. LinearRegression.

1.2.2. Дерево принятия решений

Дерево принятия решений - метод автоматического анализа больших массивов данных. Дерево решений представляет собой иерархическую древовидную структуру. Структура дерева представляет собой листья и ветки. На рёбрах (ветках) дерева решения записаны признаки, от которых зависит целевая переменная, в листьях записаны значения целевой переменной, а в остальных узлах - признаки, по которым различаются случаи. Каждый лист представляет собой значение целевой переменной, изменённой в ходе движения от корня по рёбрам дерева до листа. Каждый внутренний узел сопоставляется с одним из входных признаков.

Основная задача при построении дерева решений - последовательно и рекурсивно разбить обучающее множество на подмножества с применением решающих правил в узлах.

В основе построения лежат «жадные» алгоритмы, допускающие локально-оптимальные решения на каждом шаге разбиения в узлах, которые приводят к оптимальному итоговому решению. При выборе одного признака и произведении разбиения по нему на подмножества, алгоритм не может вернуться назад и выбрать другой признак.

Разбиение должно осуществляться по определенному правилу, для которого выбирают признак. Критериев выбора признака существует много, но наибольшей популярностью пользуется информационная энтропия.

Энтропия рассматривается, как мера неоднородности подмножества. Если выбранный признак разбиения обеспечивает максимальное снижение энтропии результирующего подмножества относительно родительского, его можно считать наилучшим. В таком случае лучшим признаком будет тот, который обеспечит максимальный прирост информации результирующего узла относительно исходного

Достоинства метода:

- простота интерпретации;

• не требует специальной подготовки данных;

• способен работать с разными типами входных признаков.

Недостатки метода:

• проблема получения оптимального дерева решений;

• возможно переобучение.

Дерево решений для задач регрессии реализовано в библиотеке scikit-learn - sklearn.tree.DecisionTreeRegressor.

1.2.3. Метод случайного леса

Метод случайного леса - алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга и метод случайных подпространств

Все деревья строятся независимо по следующей схеме:

1) выбирается подвыборка обучающей выборки - по ней строится дерево (для каждого дерева - своя подвыборка);

2) выбираются наилучшие признаки для построения расщепление в дереве, Признаки выбираются из некоторого случайного подмножества признаков (для каждого нового расщепления — свои случайные признаки).

В задаче регрессии результат выводится путём усреднения прогнозов, полученных от каждого дерева. Чем больше деревьев, тем лучше качество, но время настройки и работы также пропорционально увеличиваются.

Достоинства метода:

• способность эффективно обрабатывать данные с большим числом признаков;

• нечувствительность к масштабированию значений признаков.

Недостатки метода:

• большой размер получающихся моделей;

• сложность интерпретации;

• требует больше вычислительных ресурсов.

Случайный лес для задач регрессии реализован в библиотеки scikit-learn - sklearn.ensemble.RandomForestRegressor.

1.2.4. Метод опорных векторов

Метод опорных векторов - это набор контролируемых методов обучения, используемых для классификации, регрессии и обнаружения выбросов.

Метод опорных векторов конструирует гиперплоскость или набор гиперплоскостей в пространстве большой или бесконечной размерности, которые можно использовать для классификации, регрессии или других задач.

В основе метода опорных векторов для задач регрессии лежит поиск гиперплоскости, при которой риск в многомерном пространстве будет минимальным. Регрессии опорных векторов 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

Достоинства метода:

- эффективен в пространствах больших размеров;
- эффективен в случаях, когда количество измерений больше, чем количество выборок;
- эффективен с точки зрения использования памяти, потому использует подмножество обучающих точек в функции принятия решений.

Недостатки метода:

- чувствительность к выбросам;
- при большом объеме данных увеличивается время обучения и требует больше вычислительных ресурсов.

Метод опорных векторов для задач регрессии реализован в библиотеке scikit-learn - sklearn.svm.LinearSVR.

1.2.5. Метод k-ближайших соседей

Метод k-ближайших соседей - метрический алгоритм. В случае использования метода для регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны.

Регрессия на основе данного метода может использоваться в случаях, когда метки данных являются непрерывными, а не дискретными переменными.

Алгоритм может быть применим к выборкам с большим количеством признаков (многомерным). Задача поиска ближайшего соседа заключается в отыскании среди множества объектов, расположенных в многомерном метрическом пространстве, объектов близких к заданному, согласно некоторой функции расстояния (близости). классический вариант такой функции - евклидова метрика

Алгоритм выполнения алгоритма k-ближайших соседей состоит в:

1) определении параметра k - число ближайших соседей

2) расчет расстояния;

3) сортировка расстояний и определение ближайшего соседа.

Преимущества метода:

- простота интерпретации.

Недостатки метода:

- по мере увеличения объема данных алгоритм замедляется и требует больше вычислительных ресурсов;
- чувствительность к выбросам;
- сложность определения оптимального параметра k.

Метод k-ближайших соседей для задач регрессии реализован в библиотеке scikit-learn - sklearn.neighbors.KNeighborsRegressor.

1.2.6. Нейронная сеть

Нейронная сеть (искусственная нейронная сеть) - математическая модель, а также её программное или аппаратное воплощение, построенная по принципу

организации и функционирования биологических нейронных сетей - сетей нервных клеток живого организма.

Искусственная нейронная сеть представляет собой систему соединенных и взаимодействующих между собой простых процессоров (искусственных нейронов).

Основными элементами структуры нейронной сети являются:

1) искусственные нейроны, представляющие собой элементарные, связанные между собой единицы;

2) синапс - это соединение, которые используется для отправки - получения информации между нейронами;

3) сигнал - информация, подлежащая передаче.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения - одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами.

Преимущества нейронных сетей:

- адаптивность и эффективность;
- в процессе обучения способны выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение;
- могут аппроксимировать непрерывные функции.

Недостатки нейронных сетей:

- возможно переобучение;
- результат работы зависит от выбора исходных данных для обучения;
- выбор топологии сети, подбор характеристик сети и параметров обучения;
- требует больше вычислительных ресурсов;
- в процессе обучения могут проявиться проблемы паралича или попадания сети в локальный минимум поверхности ошибок.

1.3 Разведочный анализ данных

Разведочный анализ данных - предварительное исследование датасета с целью определения его основных характеристик, взаимосвязей между признаками.

Описание колонок (признаков) датасета:

- «Соотношение матрица-наполнитель» - характеристика композита;
- «Плотность, кг/м3» - характеристика матрицы;
- «Модуль упругости, ГПа» - характеристика матрицы;
- «Количество отвердителя, м.%» - характеристика матрицы;
- «Содержание эпоксидных групп,%_2» - характеристика матрицы;
- «Температура вспышки, С_2» - характеристика матрицы;
- «Поверхностная плотность, г/м2» - характеристика матрицы;
- «Модуль упругости при растяжении, ГПа» - характеристика композита;
- «Прочность при растяжении, МПа» - характеристика композита;
- «Потребление смолы, г/м2» - характеристика наполнителя;
- «Угол нашивки, град» - характеристика наполнителя;
- «Шаг нашивки» - характеристика наполнителя;
- «Плотность нашивки» - характеристика наполнителя.

Дублирующие записи искажают статистические данные датасета и снижают качество обучения модели, потому необходимо удалить такие строки. После удаления дублирующих записей с помощью pandas.drop_duplicates(), размерность датасета не изменилась, значит дубликаты отсутствовали.

Проверим датасет на пропуски pandas.DataFrame.isnull().any() и уникальные значения pandas.DataFrame.nunique(). Пропуски отсутствуют. Признак «Угол нашивки, град» имеет только два уникальных значения, поэтому в дальнейшем изменим тип данных на категориальный.

Выведем описательные статистические данные, воспользовавшись pandas.DataFrame.describe(), добавив дополнительно информацию о медиане и моде. Полученные данные отображены на рисунке 1.

	count	mean	std	min	25%	50%	75%	max	median	mode
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660	5.591742	2.906878	1.857143
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375	2207.773481	1977.621657	2030.000000
Модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	500.047452	739.664328	961.812526	1911.536477	739.664328	738.736842
Количество отвердителя, м.%	1023.0	110.570769	28.295911	17.740275	92.443497	110.564840	129.730366	198.953207	110.564840	129.000000
Содержание эпоксидных групп,%_2	1023.0	22.244390	2.406301	14.254985	20.608034	22.230744	23.961934	33.000000	22.230744	21.250000
Температура вспышки, С_2	1023.0	285.882151	40.943260	100.000000	259.066528	285.896812	313.002106	413.273418	285.896812	300.000000
Поверхностная плотность, г/м2	1023.0	482.731833	281.314690	0.603740	266.816645	451.864365	693.225017	1399.542362	451.864365	210.000000
Модуль упругости при растяжении, ГПа	1023.0	73.328571	3.118983	64.054061	71.245018	73.268805	75.356612	82.682051	73.268805	70.000000
Прочность при растяжении, МПа	1023.0	2466.922843	485.628006	1036.856605	2135.850448	2459.524526	2767.193119	3848.436732	2459.524526	3000.000000
Потребление смолы, г/м2	1023.0	218.423144	59.735931	33.803026	179.627520	219.198882	257.481724	414.590628	219.198882	220.000000
Угол нашивки, град	1023.0	44.252199	45.015793	0.000000	0.000000	0.000000	90.000000	90.000000	0.000000	0.000000
Шаг нашивки	1023.0	6.899222	2.563467	0.000000	5.080033	6.916144	8.586293	14.440522	6.916144	5.000000
Плотность нашивки	1023.0	57.153929	12.350969	0.000000	49.799212	57.341920	64.944961	103.988901	57.341920	57.000000

Рисунок 1 - Описательные статистические данные

Построим и визуализируем корреляционную матрицу (рисунок 2).

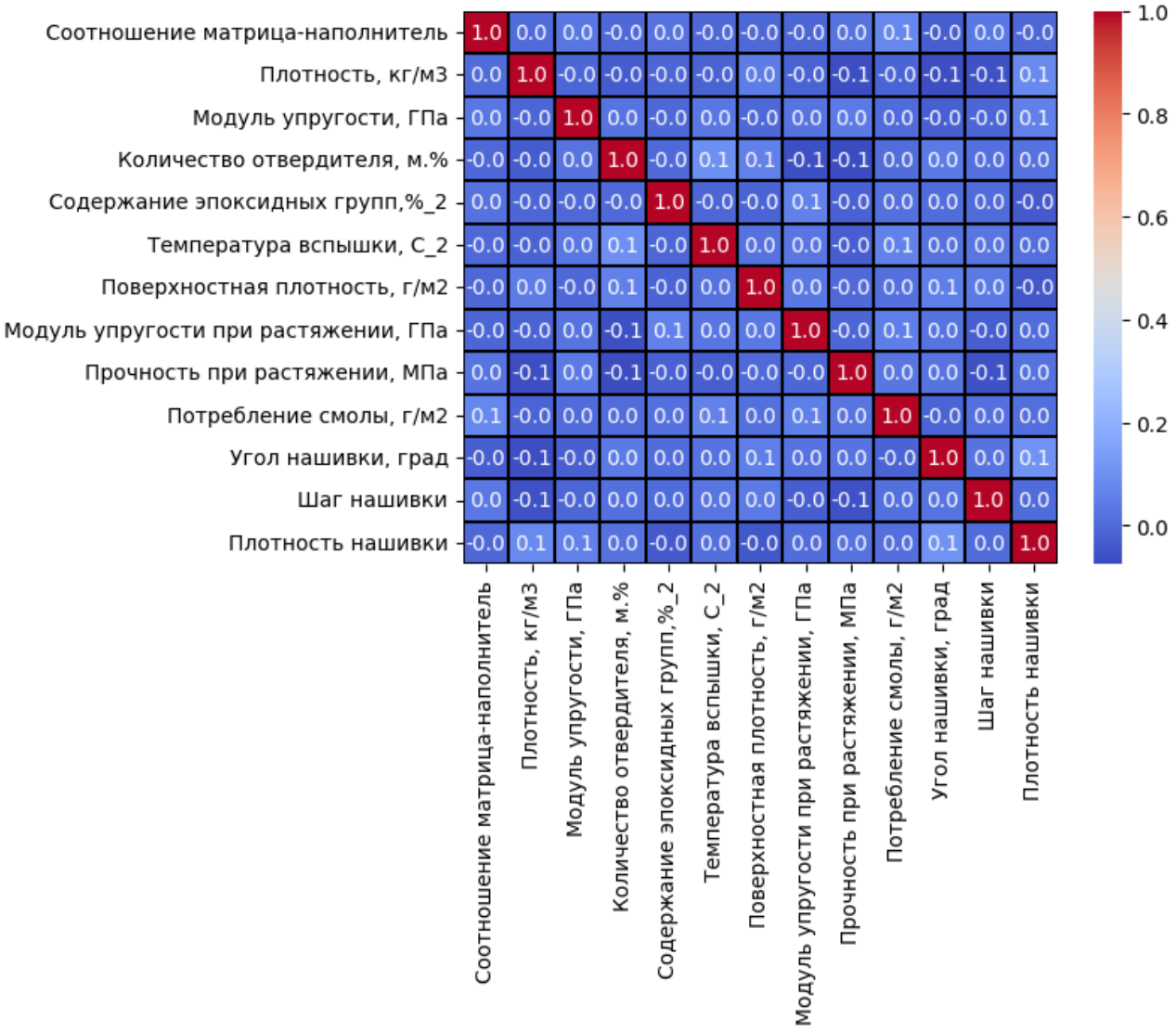


Рисунок 2 - Матрица корреляции

Выведем графики распределения, попарные графики рассеяния точек (рисунок 3).

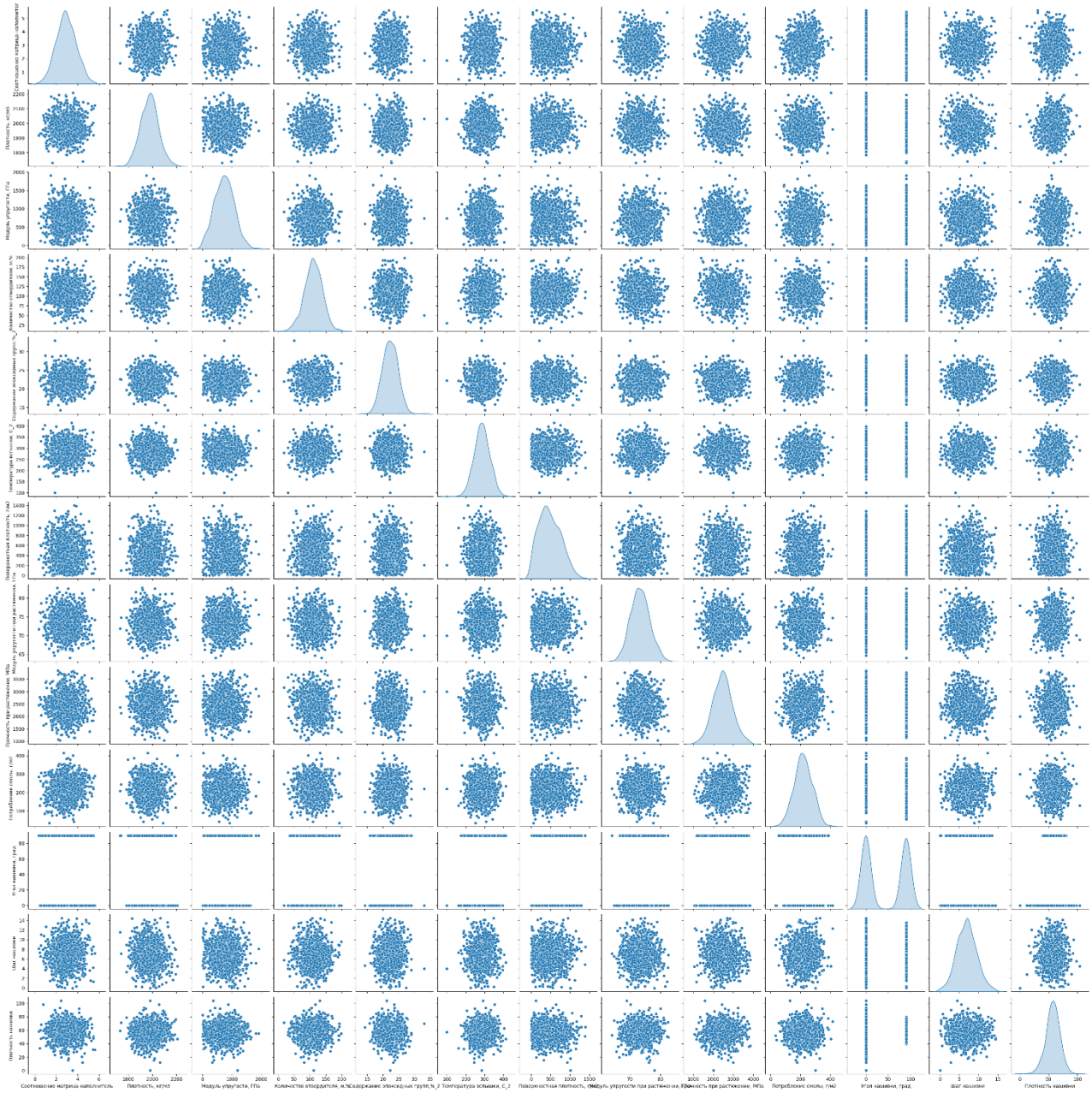
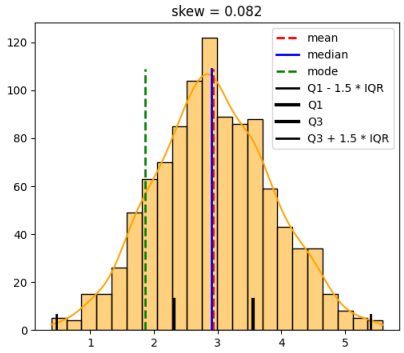
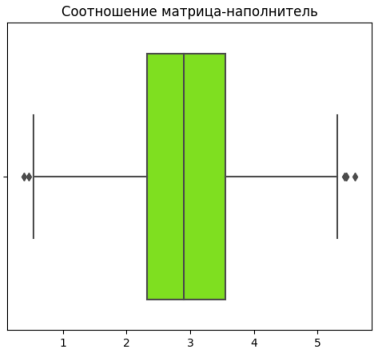
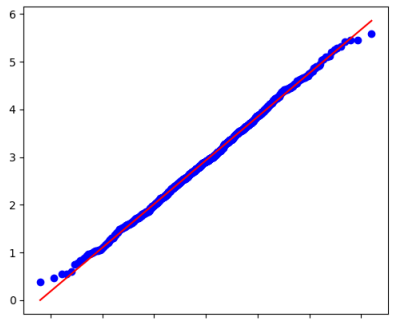


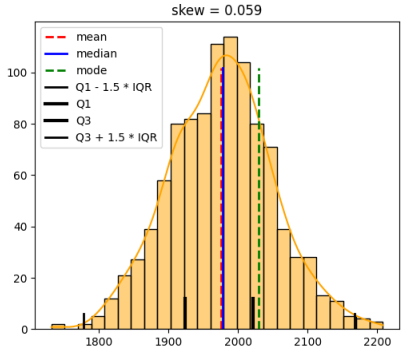
Рисунок 3 - Попарные графики рассеяния точек

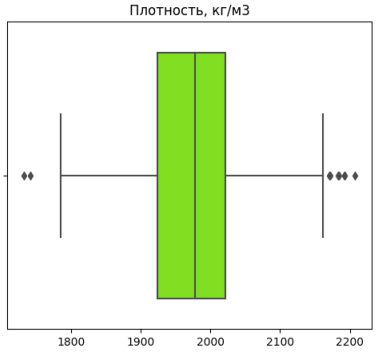
С помощью пользовательской функции выведем графики распределения, диаграммы «ящик с усами» (boxplot) и график «квантиль – квантиль» (Q-Q plot) для каждого признака (рисунок 4). На графиках распределения дополнительно отобразим среднее значение, медиану, моду.

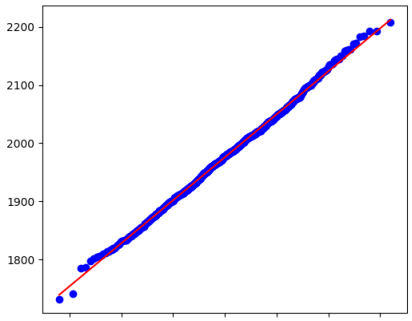

skew = 0.082
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

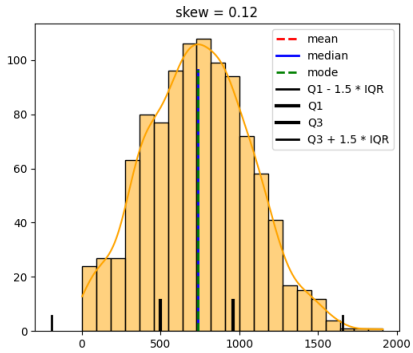

Соотношение матрица-наполнитель

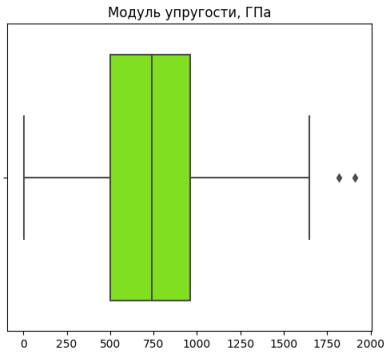


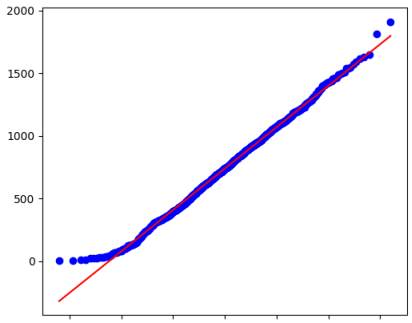

skew = 0.059
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

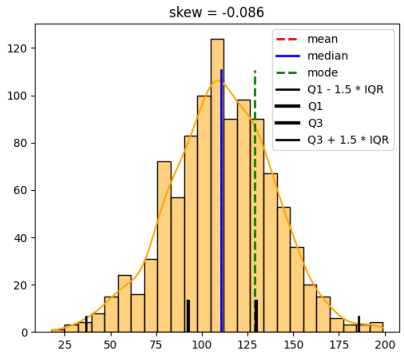

Плотность, кг/м3

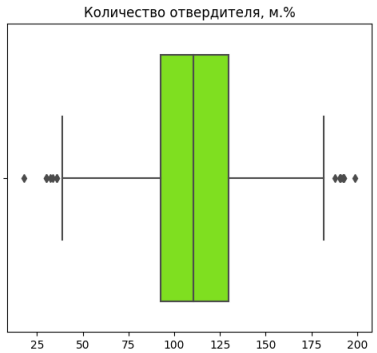


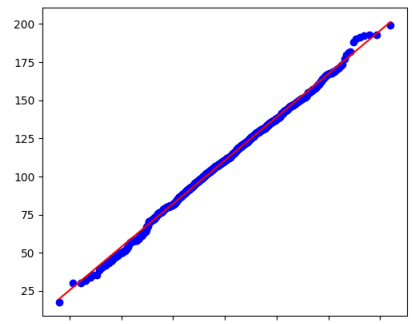

skew = 0.12
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

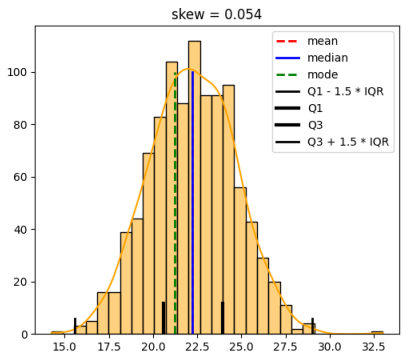

Модуль упругости, ГПа

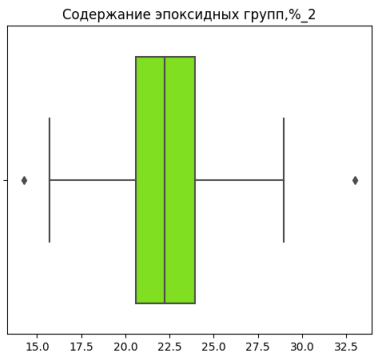


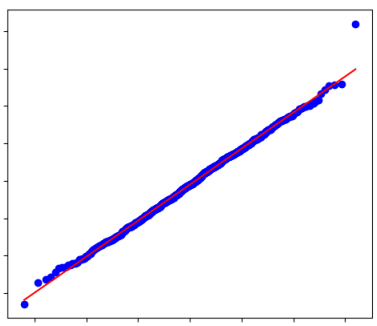

skew = -0.086
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

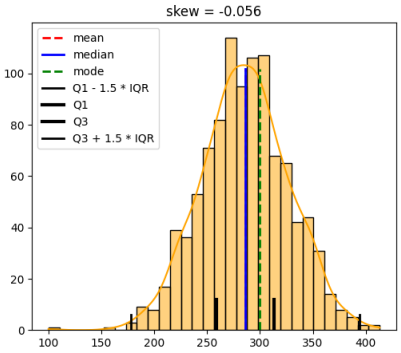

Количество отвердителя, м.%

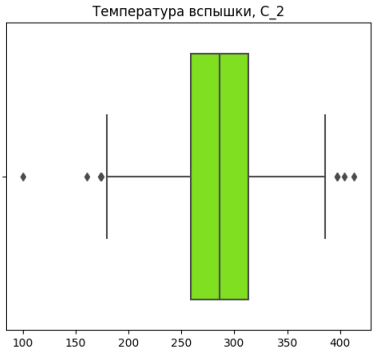


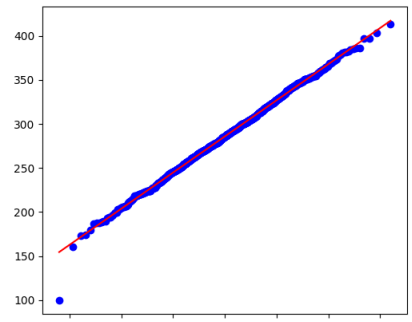

skew = 0.054
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

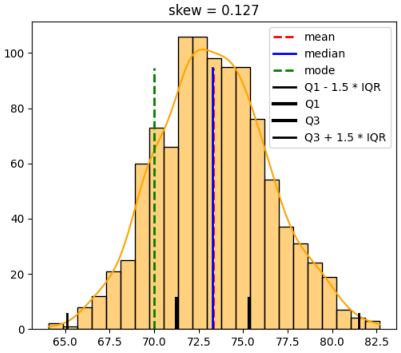

Содержание эпоксидных групп,%_2

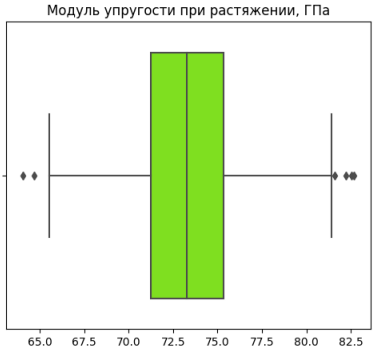


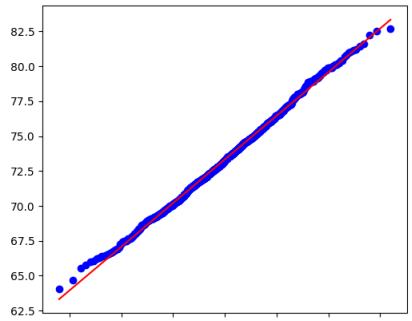

skew = -0.056
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

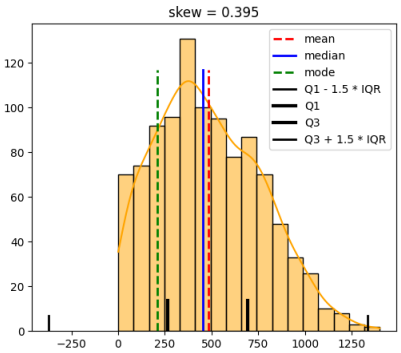

Температура вспышки, С_2

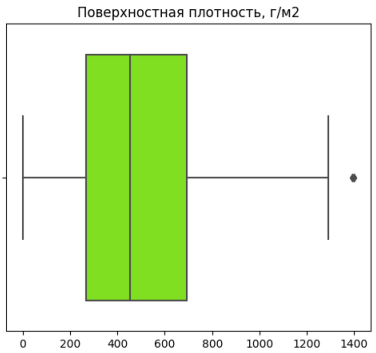


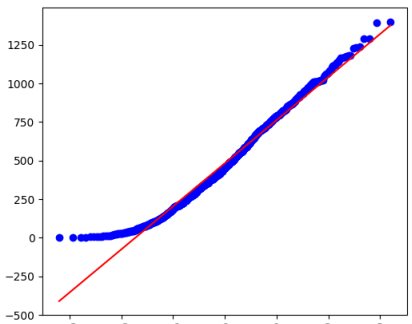

skew = 0.127
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

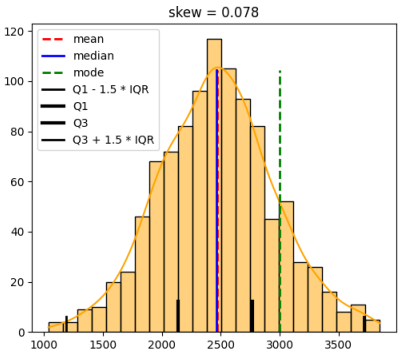

Модуль упругости при растяжении, ГПа

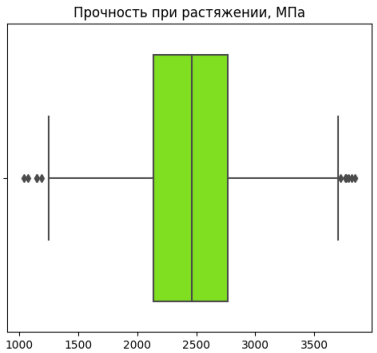


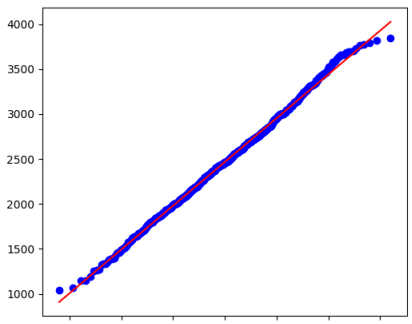

skew = 0.395
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

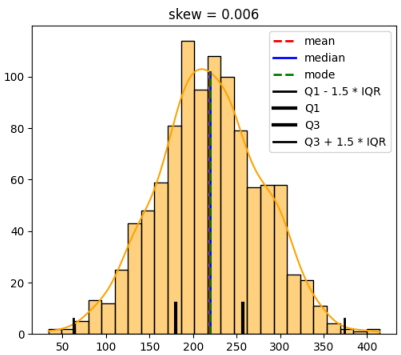

Поверхностная плотность, г/м2

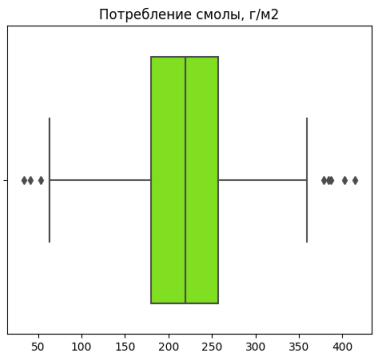


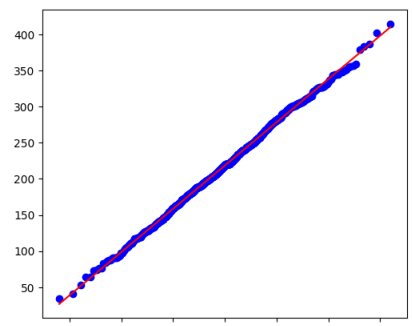

skew = 0.078
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR


Прочность при растяжении, МПа




skew = 0.006
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR


Потребление смолы, г/м2



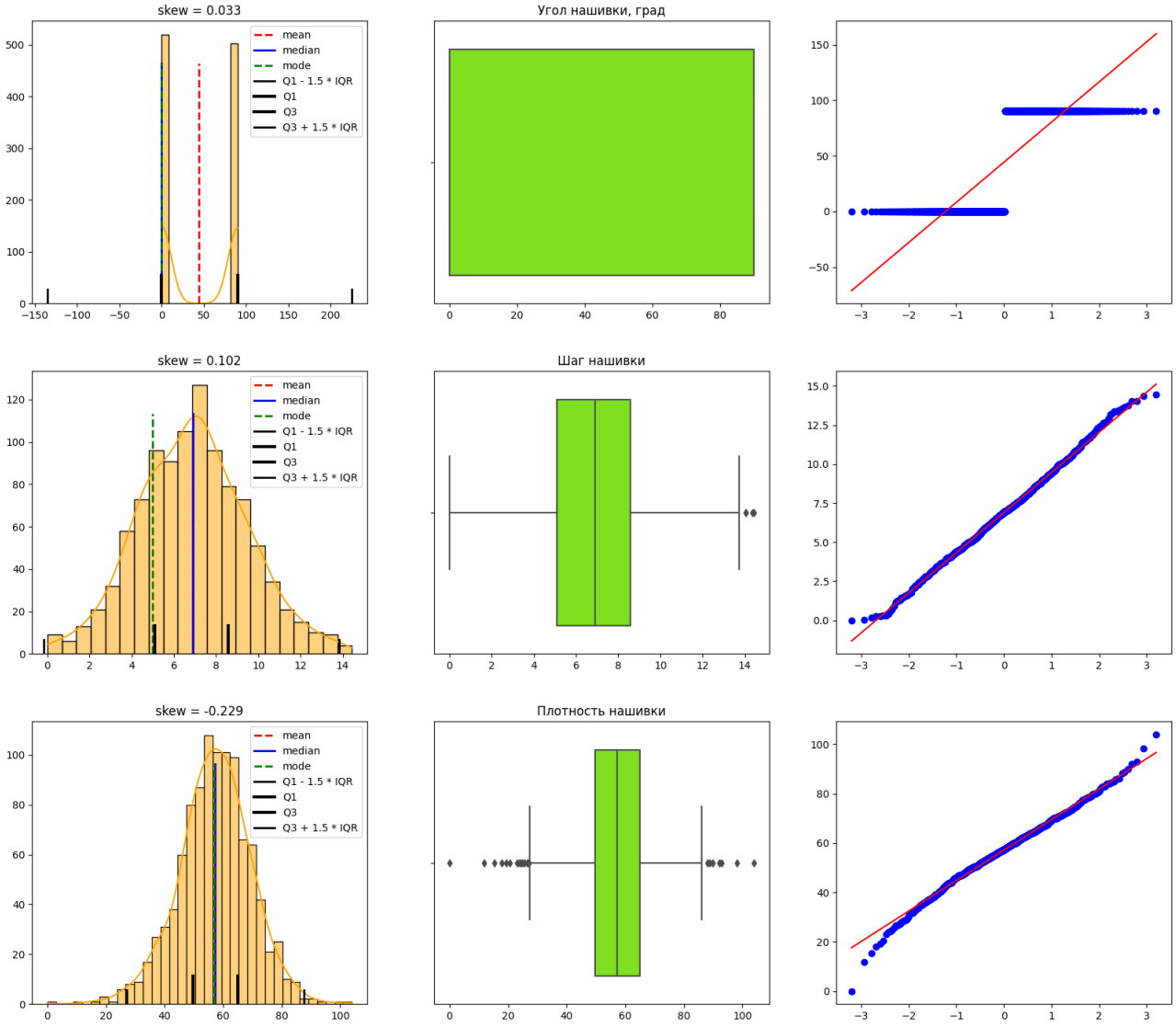


Рисунок 4 – Графики распределения, диаграммы «ящик с усами», графики «квантиль-квантиль»

По графикам можно сделать выводы:

- наличие выбросов;
- явная корреляция между признаками отсутствует;
- распределение близкое к нормальному по всем признакам, кроме «Поверхностная плотность, г/м2», признак имеет положительную асимметрию.

Все статистические данные и другую полезную информацию можно получить также с помощью метода ProfileReport библиотеки ydata_profiling.

Выброс - это элемент маломощного подмножества выборки, существенно отличающийся от остальных элементов выборки. Причинами появления выбросов могут быть ошибки в измерениях или технические ошибки форматирования данных.

Выбросы могут негативно повлиять на построение модели, поэтому лучше выполнить обнаружение и устранение (удаление или замена) выбросов.

Существует несколько подходов для обнаружения выбросов:

1) использование трех сигм, которые следуют правилу 68-95-99,7, при этом 68% данных находятся в пределах одного стандартного отклонения от среднего, 95% данных находятся в пределах двух стандартных отклонений от среднего и 99,7% данных находятся в пределах трех стандартных отклонений от среднего;

2) использование 5-95 квантилей - значения ниже 5% квантиля и выше 95% квантиля можно считать выбросами;

3) использование межквартильного размаха (interquartile range, IQR), который является мерой статистической дисперсии. Для вычисления IQR набор данных делится на квартили, или четыре упорядоченные по рангу четные части, с помощью линейной интерполяции. Эти квартили обозначаются Q1 (нижний квартиль, соответствует 25-му процентилю), Q2 (медиана) и Q3 (верхний квартиль, соответствует 75-му процентилю). IQR определяется как разность третьего квартиля и первого квартиля.

Использую подходы, указанные выше получили следующее:

- правила трех сигм - удалено 23 строки, при повторных итерациях еще 4;
- межквартильный размах - удалено 87 строк, при повторных итерациях еще 14;
- 5- 95 квантилей - удалено 727 строк (большая потеря данных).

2 Практическая часть

2.1 Предобработка данных

Предварительная обработка данных в машинном обучении – это важный шаг, который помогает повысить качество данных. Предобработка данных в машинном обучении относится к технике подготовки необработанных данных с целью сделать их пригодными для построения и обучения моделей машинного обучения. Иными словами, это метод интеллектуального анализа данных, который преобразует необработанные данные в понятный и читаемый формат.

Для предварительной обработки данных воспользуемся следующими трансформаторными классами пакета sklearn.preprocessing библиотеки scikit-learn:

1) StandardScaler – стандартизация путем удаления среднего значения и масштабирования до единичной дисперсии (центрирование и масштабирование происходят независимо для каждого объекта путем вычисления соответствующей статистики по выборкам в обучающем наборе);

2) MinMaxScaler - преобразование объектов, масштабируя каждый объект до заданного диапазона (масштабируется и преобразуется каждый признак по отдельности таким образом, чтобы он находился в заданном диапазоне в обучающем наборе, чаще всего, между нулем и единицей);

3) PowerTransformer - функциональное преобразование мощности, чтобы сделать данные более похожими на гауссовы (семейство параметрических монотонных преобразований, которые применяются для придания данным более гауссовского типа; поддерживает преобразование Бокса-Кокса и преобразование Йео-Джонсона);

4) QuantileTransformer - преобразование объектов, используя информацию о квантилях (применяется к каждому элементу независимо; этот метод преобразования имеет тенденцию распространять наиболее часто встречающиеся значения, что уменьшает влияние незначительных выбросов);

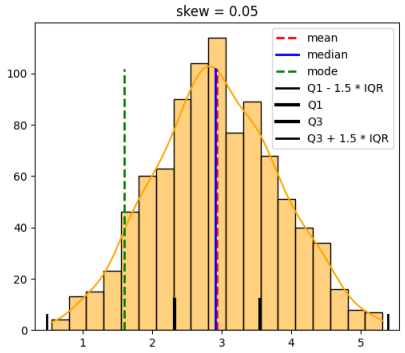
5) RobustScaler - масштабирование объектов, используя статистику, устойчивую к выбросам (удаляет медиану и масштабирует данные в соответствии с диапазоном квантилей, центрирование и масштабирование происходят независимо для каждого объекта путем вычисления соответствующей статистики по выборкам в обучающем наборе).

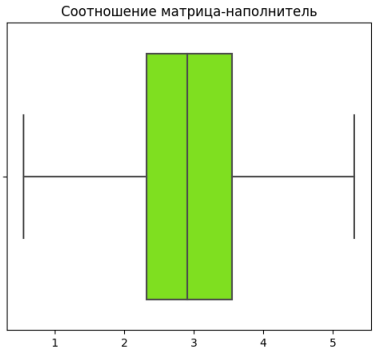
Предобработку данных сделаем для всех полученных датасетов после удаления выбросов с помощью StandardScaler (для признаков с нормальным распределением) совместно с PowerTransformer (для признаков с явной асимметрией, преобразованием Йео-Джонсона) и MinMaxScaler. Дополнительно выполним предобработку датасета без удаления выбросов с помощью RobustScaler и QuantileTransformer.

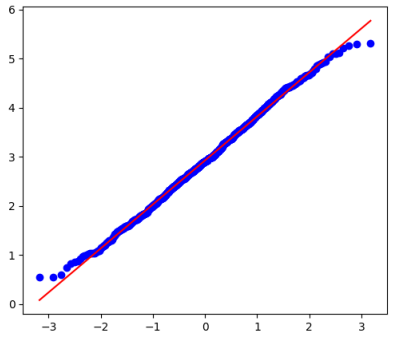
Целевые переменные преобразовывать не будем.

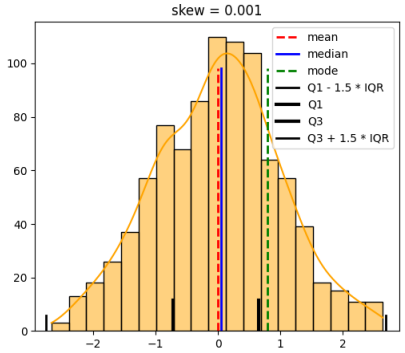
После предобработки данных получили восемь датасетов:

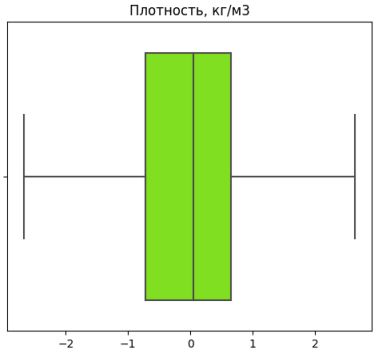
- 3sig_SC (выбросы – правило трех сигм, преобразование - StandardScaler + PowerTransformer);
- 3sig_MMS (выбросы – правило трех сигм, преобразование - MinMaxScaler);
- IQR_SC (выбросы – межквартильный размах, преобразование - StandardScaler + PowerTransformer) - графики распределения, диаграммы «ящик с усами», графики «квантиль-квантиль» каждого признака отображены на рисунке 5;
- IQR_MMS (выбросы – межквартильный размах, преобразование - MinMaxScaler);
- 5Q95_SC (выбросы – 5-95 квантилей, преобразование - StandardScaler + PowerTransformer);
- 5Q95_MMS (выбросы – 5-95 квантилей, преобразование - MinMaxScaler);
- all_RS (выбросы – не устранены, преобразование - RobustScaler);
- all_QT (выбросы – не устранены, преобразование - QuantileTransformer).

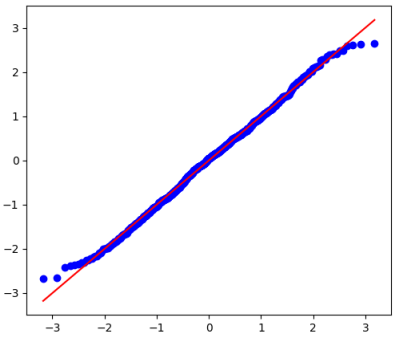

skew = 0.05
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

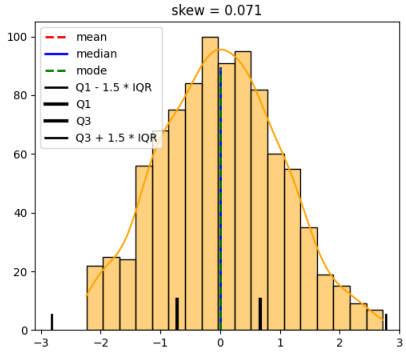

Соотношение матрица-наполнитель

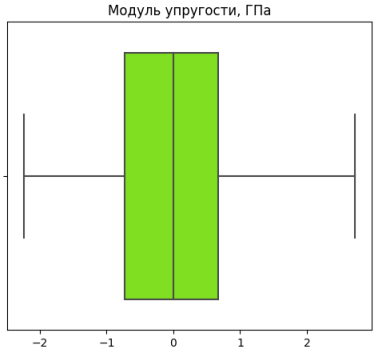


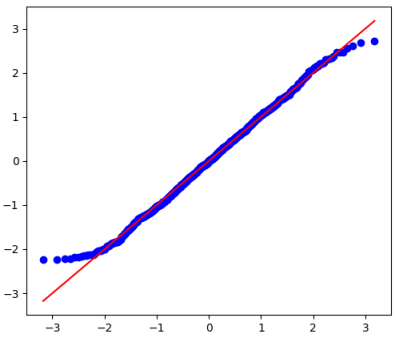

skew = 0.001
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

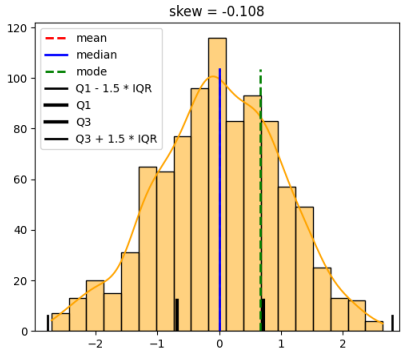

Плотность, кг/м3

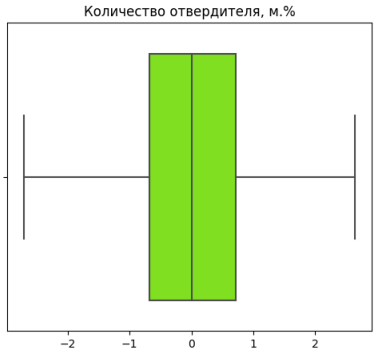


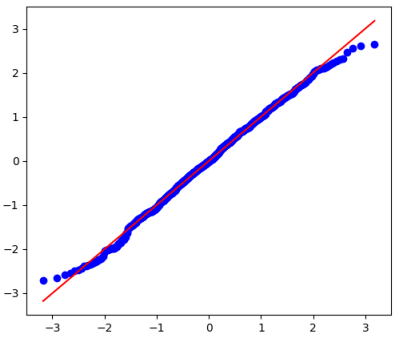

skew = 0.071
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

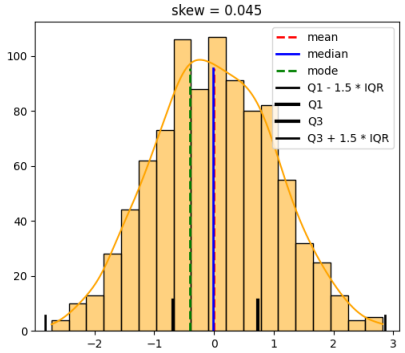

Модуль упругости, ГПа

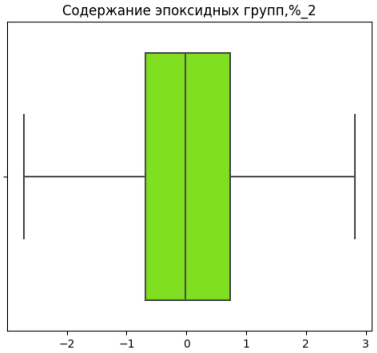


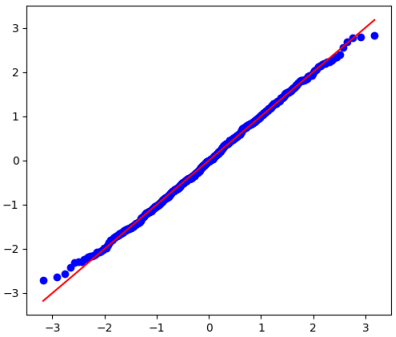

skew = -0.108
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

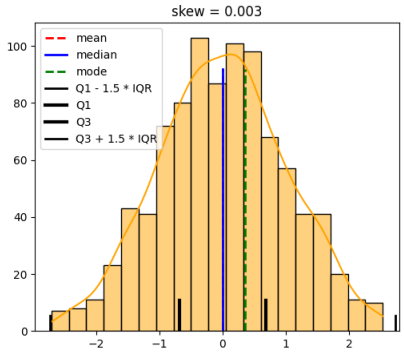

Количество отвердителя, м.%

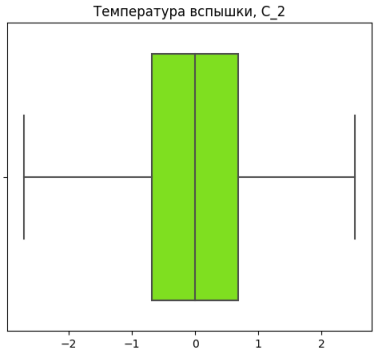


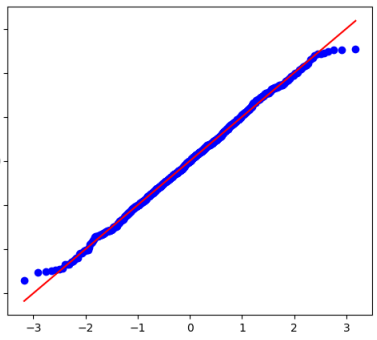

skew = 0.045
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

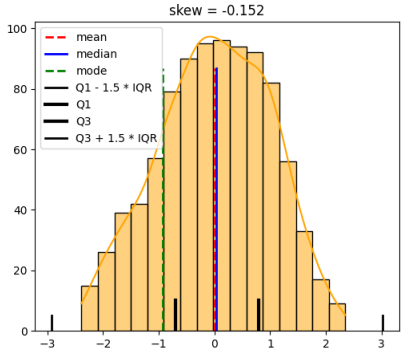

Содержание эпоксидных групп,%_2

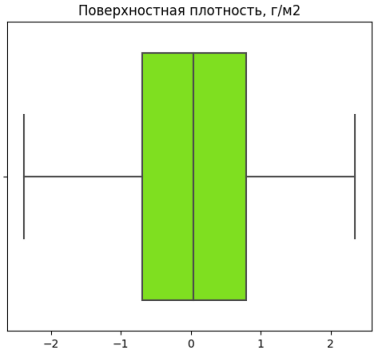


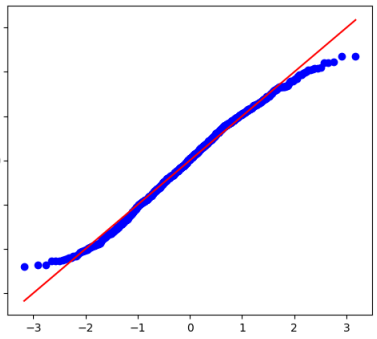

skew = 0.003
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

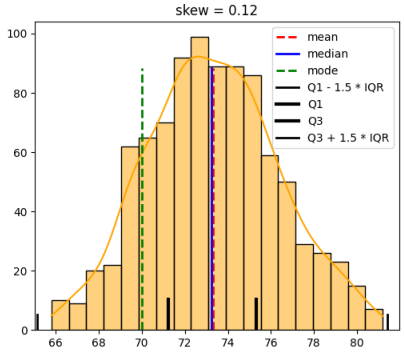

Температура вспышки, С_2

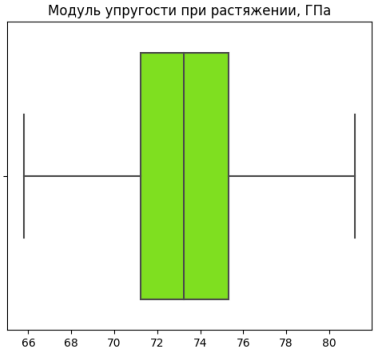


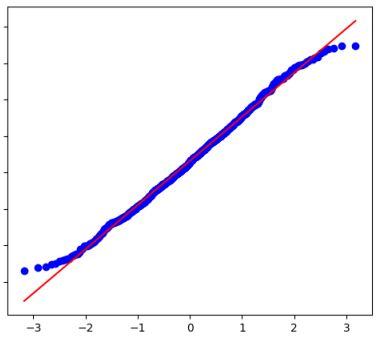

skew = -0.152
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

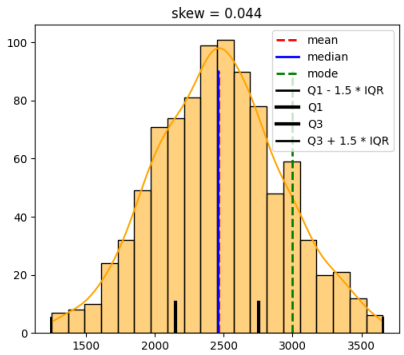

Поверхностная плотность, г/м2

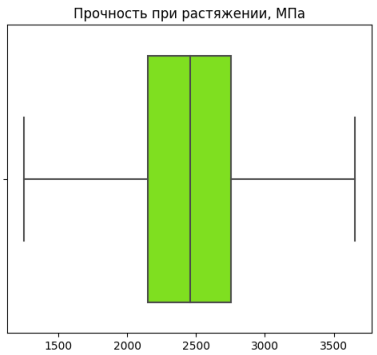


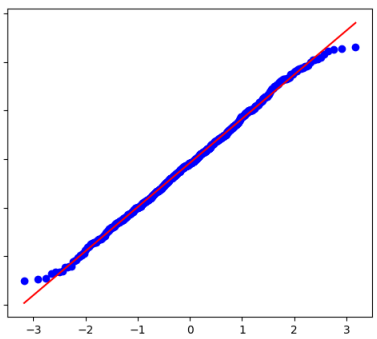

skew = 0.12
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR

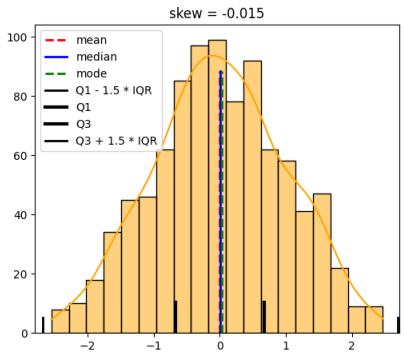

Модуль упругости при растяжении, ГПа

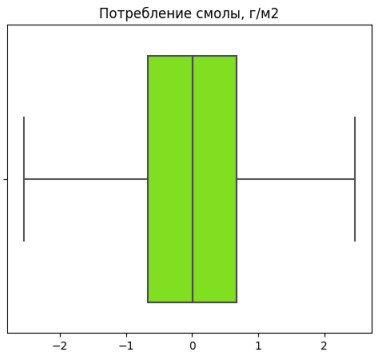


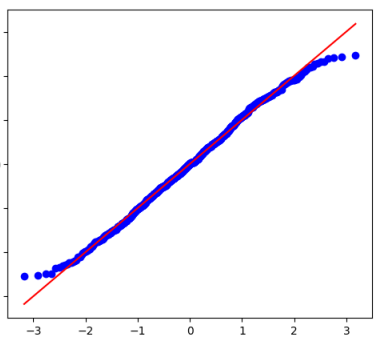

skew = 0.044
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR


Прочность при растяжении, МПа




skew = -0.015
mean
median
mode
Q1 - 1.5 * IQR
Q1
Q3
Q3 + 1.5 * IQR


Потребление смолы, г/м2



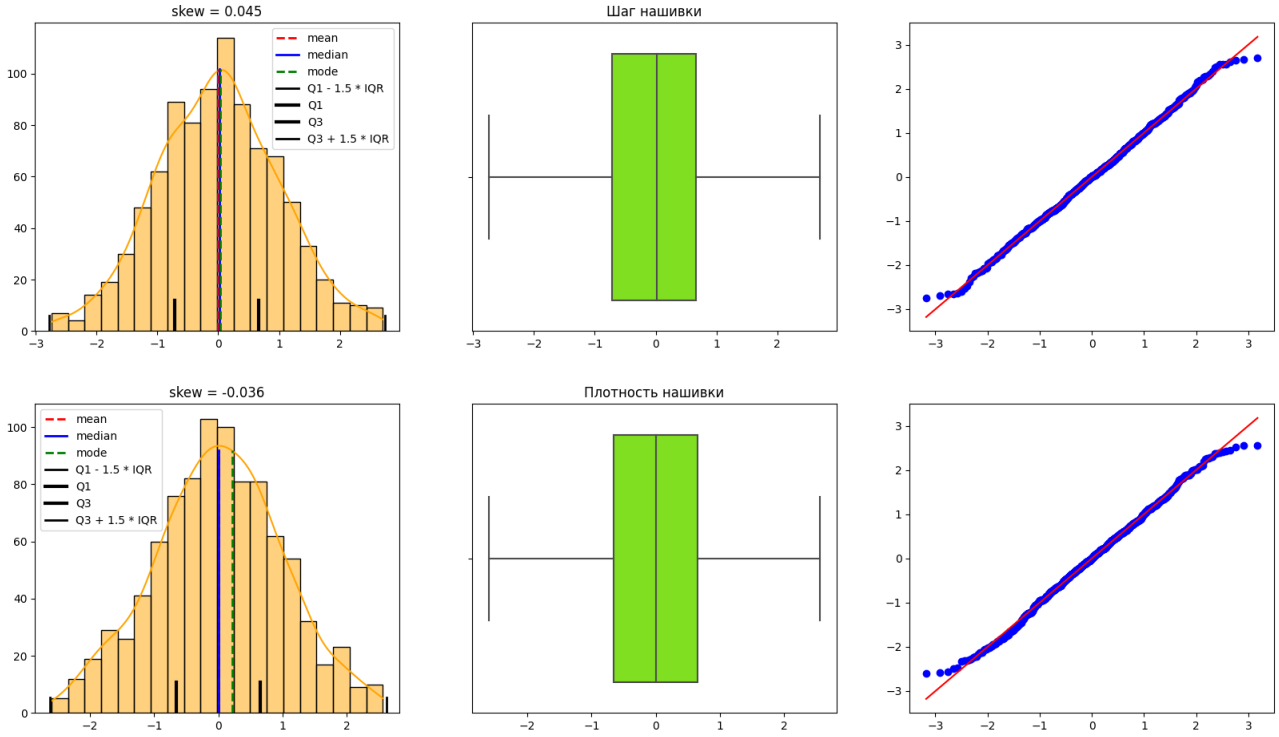


Рисунок 5 - Графики распределения, диаграммы «ящик с усами», графики «квантиль-квантиль» после предобработки данных

2.2 Выбор стратегии

Для выбора стратегии устранения выбросов и предобработки данных было принято решений прогнать все полученные датасеты через обучение моделей с параметрами по умолчанию, и выбрать наилучший вариант на основе метрик полученных моделей.

С помощью цикла все датасеты были разделены на входные и целевые переменные, потом на обучающую и тестовую выборки в соотношении 70% обучающая и 30% тестовая, далее произведено обучение моделей и снятие метрик.

Полученные метрики моделей показывают, что в большинстве случаев наилучшие результаты выдает модель обычной линейной регрессии.

Далее выполнено сравнение датасетов между собой по модели линейной регрессии, чтоб определится с каким датасетом будем работать дальше. Данные для сравнения изображены на рисунке 6.

	Dataset	Target	Model	MAE	MSE	R2
16	5Q95_MMS	Модуль упругости	LinReg	1.977394	6.030400	-0.033687
10	IQR_MMS	Модуль упругости	LinReg	2.524100	9.524649	-0.031842
13	5Q95_SC	Модуль упругости	LinReg	1.978322	6.013756	-0.030834
7	IQR_SC	Модуль упругости	LinReg	2.520539	9.492982	-0.028411
4	3sig_MMS	Модуль упругости	LinReg	2.506775	9.296600	-0.006135
19	all_RS	Модуль упругости	LinReg	2.529999	9.638852	-0.005507
1	3sig_SC	Модуль упругости	LinReg	2.506372	9.287463	-0.005146
22	all_QT	Модуль упругости	LinReg	2.529440	9.628619	-0.004440
14	5Q95_SC	Прочность	LinReg	309.810110	145337.831903	-0.098522
17	5Q95_MMS	Прочность	LinReg	309.309035	145176.284230	-0.097301
5	3sig_MMS	Прочность	LinReg	408.727886	267297.745566	-0.019884
23	all_QT	Прочность	LinReg	388.130652	247203.693781	-0.018003
2	3sig_SC	Прочность	LinReg	408.284904	266768.856351	-0.017866
20	all_RS	Прочность	LinReg	387.848720	247045.994937	-0.017353
11	IQR_MMS	Прочность	LinReg	368.585349	207109.075737	-0.011279
8	IQR_SC	Прочность	LinReg	368.224934	206942.952527	-0.010468
15	5Q95_MMS	Соотношение М-Н	LinReg	0.602618	0.524602	-0.025625
12	5Q95_SC	Соотношение М-Н	LinReg	0.602377	0.523843	-0.024139
18	all_RS	Соотношение М-Н	LinReg	0.820590	0.987593	-0.015588
21	all_QT	Соотношение М-Н	LinReg	0.820772	0.987572	-0.015567
3	3sig_MMS	Соотношение М-Н	LinReg	0.719608	0.782180	-0.000226
0	3sig_SC	Соотношение М-Н	LinReg	0.719606	0.782141	-0.000176
9	IQR_MMS	Соотношение М-Н	LinReg	0.725964	0.793614	0.012530
6	IQR_SC	Соотношение М-Н	LinReg	0.725879	0.793490	0.012684

Рисунок 6 – Метрики модели линейной регрессии на разных датасетах

Дальнейшее обучение моделей будет на датасете с устранением выбросов, используя межквартильный размах, и с преобразованием StandardScaler, совместно с PowerTransformer.

2.3 Разработка и обучение моделей

Для прогнозирования модуля упругости при растяжении и прочности при растяжении будут обучены модели с применением методов, указанных в подразделе 1.2.

Сначала разделим датасет на входные и целевую переменные, потом с помощью sklearn.model_selection.train_test_split библиотеки scikit-learn разделим датасет на тестовую (70%) и обучающую (30%) выборки.

Далее, используя sklearn.compose.ColumnTransformer, сделаем препроцессинг данных, который позволит автоматизировать предобработку данных перед обучением модели.

Препроцессинг будет состоять из следующих преобразований:

- StandardScaler для числовых признаков без асимметрии;
- PowerTransformer для числовых признаков с явной асимметрией;
- OneHotEncoder для категориальных признаков.

Воспользуемся функцией sklearn.model_selection.GridSearchCV библиотеки scikit-learn для автоматического подбирания параметров моделей машинного обучения. GridSearchCV находит наилучшие параметры, путем обычного перебора, создавая модель для каждой возможной комбинации параметров.

После получения наилучших параметров, инициализируем модели для обучения. Используя sklearn.pipeline.Pipeline библиотеки scikit-learn, объединим препроцессинг данных и модель в пайплайн, произведем обучение, снимем метрики и сохраним полученные модели в формате pkl для дальнейшего использования.

2.4 Оценка моделей

Оценку моделей будем производить с помощью коэффициента детерминации, дополнительно посмотрим результаты среднеквадратичной ошибки и средней абсолютной ошибки.

Коэффициент детерминации (R-квадрат) - это доля дисперсии зависимой переменной, объясняемая рассматриваемой моделью зависимости, то есть объясняющими переменными.

Среднеквадратичная ошибка (MSE) рассчитывается путем получения среднего квадрата разницы между исходным и прогнозируемым значениями данных.

Средняя абсолютная ошибка (MAE) представляет собой среднюю абсолютную разницу между исходным и прогнозируемым значениями данных.

На рисунках 7, 8 отображены полученные метрики работы моделей.

На рисунке 7

	Model	MAE	MSE	R2
0	LinReg	2.520664	9.494092	-0.028532
1	DecTreeReg	3.434590	18.378696	-0.991035
2	LinSVR	2.522822	9.529168	-0.032332
3	KNNReg	2.488656	9.199464	0.003386
4	RandForReg	2.526412	9.775038	-0.058968

Рисунок 7 – Метрики моделей для прогноза модуля упругости при растяжении

	Model	MAE	MSE	R2
0	LinReg	368.612465	210248.957038	-0.026610
1	DecTreeReg	546.510000	455256.167719	-1.222940
2	LinSVR	368.192588	206855.073445	-0.010039
3	KNNReg	366.052595	204508.051117	0.001421
4	RandForReg	383.447685	217757.993006	-0.063276

Рисунок 8 – Метрики моделей для прогноза прочности при растяжении

Модели, реализованные с помощью метода k-ближайших соседей, показывают наилучшие значения коэффициента детерминации для прогнозирования целевых переменных.

2.5 Разработка и обучение нейронной сети

Для прогнозирования соотношения матрицы-наполнитель разработаем и обучим нейронную сеть с помощью библиотеки tensorflow.

Так как мы уже обучили модели для прогнозирования двух других целевых переменных, то будем считать их входными переменными.

Сначала сделаем два препроцессинга данных: один аналогичен препроцессингу созданному ранее, но с добавлением двух новых входных переменных, второй будет использовать MinMaxScaler для преобразования числовых признаков и OneHotEncoder для категориальных признаков. Для дальнейшего использования сохраним полученные модели препроцессинга в формате pkl.

Инициализируем модель нейронной сети и попробуем вручную подбирать параметры сети для датасета с разной предобработкой данных, будем изменять количество слоев и нейронов, использовать разные функции активации и оптимизаторы.

При обучении нейронной сети воспользуемся ModelCheckpoint для сохранения лучшей модели в формате hdf5 и EarlyStopping для ранней остановки обучения, при отсутствии улучшения точности модели после 20 эпох обучения

На рисунке 9 изображены метрики моделей нейронной сети, полученных в результате ручного перебора параметров.

Наилучший показатель детерминации показала модель со следующими параметрами сети:

- входной слой (количество нейронов - 8, инициализация матрицы весов – 'normal', функция активации – 'relu');
- выходной слой (количество нейронов - 1, инициализация матрицы весов – 'normal', функция активации – 'linear');
- функция ошибки 'mean_squared_error';
- оптимизатор - Adam (скорость обучения - 0.0001);
- метрика - 'mean_squared_error'.

	Model	MAE	MSE	R2
30	Best_model31	1.963414	4.923813	-5.126551
29	Best_model30	1.740889	3.973470	-3.944068
28	Best_model29	1.621751	3.734108	-3.646237
31	Best_model32	1.088815	1.701520	-1.117150
10	Best_model11	0.749052	0.847264	-0.054225
11	Best_model12	0.745755	0.835289	-0.039324
9	Best_model10	0.747721	0.831469	-0.034572
32	Best_model33	0.736572	0.831357	-0.034432
5	Best_model6	0.736589	0.831284	-0.034341
1	Best_model2	0.743393	0.824330	-0.025688
15	Best_model16	0.732756	0.822527	-0.023445
14	Best_model15	0.737936	0.819443	-0.019608
0	Best_model1	0.738560	0.815447	-0.014636
3	Best_model4	0.730449	0.812804	-0.011347
19	Best_model20	0.725915	0.807424	-0.004653
21	Best_model22	0.724715	0.804841	-0.001439
13	Best_model14	0.734034	0.804738	-0.001312
23	Best_model24	0.725160	0.802765	0.001143
2	Best_model3	0.734693	0.802657	0.001278
8	Best_model9	0.729146	0.802572	0.001384
25	Best_model26	0.724465	0.802567	0.001391
22	Best_model23	0.725965	0.801658	0.002521
24	Best_model25	0.724630	0.801051	0.003276
20	Best_model21	0.724237	0.800501	0.003960
4	Best_model5	0.726718	0.800239	0.004287
18	Best_model19	0.727169	0.799116	0.005684
16	Best_model17	0.725051	0.796817	0.008545
6	Best_model7	0.726104	0.796035	0.009517
26	Best_model27	0.723470	0.795591	0.010070
12	Best_model13	0.727571	0.795451	0.010245
27	Best_model28	0.724351	0.792748	0.013608
17	Best_model18	0.724967	0.792472	0.013951
33	Best_model34	0.722363	0.790095	0.016909
7	Best_model8	0.715045	0.781157	0.028030

	Model	MAE	MSE	R2
10	Best_model11	0.746346	0.859275	-0.069170
11	Best_model12	0.737074	0.838771	-0.043657
5	Best_model6	0.732213	0.826989	-0.028997
30	Best_model31	0.730297	0.816529	-0.015983
26	Best_model27	0.737102	0.814473	-0.013424
17	Best_model18	0.731515	0.807395	-0.004617
27	Best_model28	0.731149	0.807105	-0.004256
13	Best_model14	0.725434	0.806107	-0.003014
19	Best_model20	0.725087	0.804923	-0.001541
21	Best_model22	0.724728	0.804803	-0.001392
25	Best_model26	0.724666	0.804310	-0.000779
0	Best_model1	0.725322	0.804035	-0.000437
3	Best_model4	0.724914	0.803808	-0.000154
23	Best_model24	0.724755	0.803729	-0.000055
29	Best_model30	0.727404	0.803535	0.000186
15	Best_model16	0.724841	0.803500	0.000229
24	Best_model25	0.724744	0.803040	0.000801
20	Best_model21	0.724952	0.802893	0.000985
8	Best_model9	0.725086	0.802721	0.001198
22	Best_model23	0.724838	0.802277	0.001751
7	Best_model8	0.725162	0.802036	0.002051
6	Best_model7	0.725288	0.801811	0.002331
9	Best_model10	0.724703	0.801580	0.002619
18	Best_model19	0.725639	0.801380	0.002867
2	Best_model3	0.726235	0.801257	0.003021
12	Best_model13	0.727441	0.801070	0.003252
16	Best_model17	0.725546	0.800843	0.003535
14	Best_model15	0.724953	0.800805	0.003582
4	Best_model5	0.725411	0.800184	0.004356
1	Best_model2	0.725233	0.799440	0.005281
32	Best_model33	0.724462	0.798938	0.005906
28	Best_model29	0.724881	0.798927	0.005920
33	Best_model34	0.722947	0.796763	0.008611
31	Best_model32	0.725978	0.796449	0.009003

Рисунок 9 – Метрики моделей нейронной сети (слева – датасет после StandardScaler, справа – датасет после MinMaxScaler)

Дополнительно было решено выполнить попытку параметров сети перебором через цикл. Параметры цикла:

- количество слоев – [1, 2, 3];
- количество нейронов - [8, 16, 32, 64];
- функции активации - ['tanh', 'linear', 'relu'];
- оптимизаторы - ['sgd', 'adam'].

Полученные модели не дали коэффициент детерминации выше, чем модель, указанная выше. Метрики моделей изображены на рисунке 10.

	Model	MAE	MSE	R2
14	1_32_linear_sgd	0.726563	0.796562	0.008862
45	2_64_linear_adam	0.724769	0.796346	0.009131
15	1_32_linear_adam	0.727954	0.795575	0.010090
7	1_16_tanh_adam	0.724852	0.795409	0.010297
61	3_32_tanh_adam	0.725453	0.795114	0.010663
12	1_32_tanh_sgd	0.726094	0.794973	0.010839
13	1_32_tanh_adam	0.724726	0.794958	0.010858
3	1_8_linear_adam	0.724787	0.794945	0.010874
0	1_8_tanh_sgd	0.724009	0.794517	0.011407
33	2_16_linear_adam	0.725060	0.794266	0.011718
4	1_8_relu_sgd	0.726163	0.794230	0.011764
38	2_32_linear_sgd	0.724113	0.793861	0.012223
19	1_64_tanh_adam	0.725448	0.793434	0.012755
57	3_16_linear_adam	0.727378	0.793370	0.012834
69	3_64_linear_adam	0.727406	0.793238	0.012997
6	1_16_tanh_sgd	0.724135	0.792715	0.013649
51	3_8_linear_adam	0.725657	0.792572	0.013827
31	2_16_tanh_adam	0.723238	0.791520	0.015136
37	2_32_tanh_adam	0.723114	0.790683	0.016177
43	2_64_tanh_adam	0.725247	0.789880	0.017176

	Model	MAE	MSE	R2
70	3_64_relu_sgd	0.725000	0.798209	0.006813
22	1_64_relu_sgd	0.725677	0.798110	0.006935
38	2_32_linear_sgd	0.725473	0.797798	0.007324
13	1_32_tanh_adam	0.726329	0.797255	0.008000
41	2_32_relu_adam	0.729302	0.796966	0.008359
18	1_64_tanh_sgd	0.726350	0.796928	0.008407
6	1_16_tanh_sgd	0.726373	0.796768	0.008606
7	1_16_tanh_adam	0.725571	0.796539	0.008891
14	1_32_linear_sgd	0.726702	0.796411	0.009050
46	2_64_relu_sgd	0.725156	0.796231	0.009273
61	3_32_tanh_adam	0.725570	0.795737	0.009888
0	1_8_tanh_sgd	0.725738	0.795550	0.010121
36	2_32_tanh_sgd	0.724410	0.794819	0.011031
8	1_16_linear_sgd	0.725929	0.793905	0.012168
37	2_32_tanh_adam	0.725113	0.793248	0.012985
67	3_64_tanh_adam	0.724671	0.792947	0.013361
42	2_64_tanh_sgd	0.725185	0.792173	0.014323
4	1_8_relu_sgd	0.724775	0.791798	0.014790
10	1_16_relu_sgd	0.724342	0.790423	0.016500
44	2_64_linear_sgd	0.724840	0.790411	0.016516

Рисунок 10 – Метрики моделей нейронной сети, полученные циклом (слева – датасет после StandardScaler, справа – датасет после MinMaxScaler)

Для дальнейшего использования наилучшая модель и ее веса сохранены в формате h5.

2.6 Разработка приложения

Для разработки приложения будем использовать библиотеку Streamlit.

Streamlit - это фреймворк с открытым исходным кодом, специально используемый для задач, связанных с машинным обучением и наукой о данных. Он может создавать веб-приложения с гораздо меньшим количеством кода. Он широко используется, потому что поддерживает большинство научных библиотек данных.

Разработанное приложение позволяет пользователю:

1) внести характеристики матрицы и наполнителя (рисунок 11) - по умолчанию, заполнены значениями из тестовой выборки;

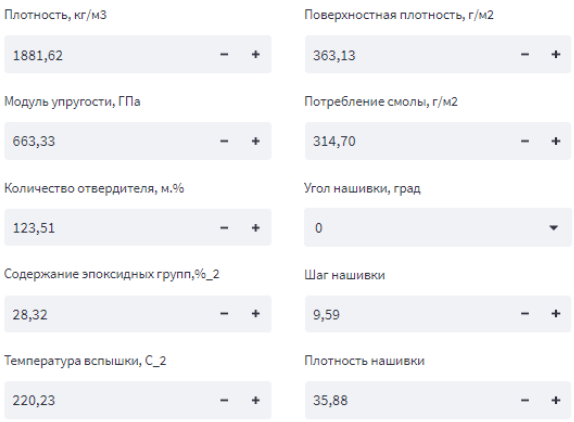


Рисунок 11 – Интерфейс ввода значений

2) выбрать модель машинного обучения (рисунок 12) – по умолчанию выбрана модель линейной регрессии;

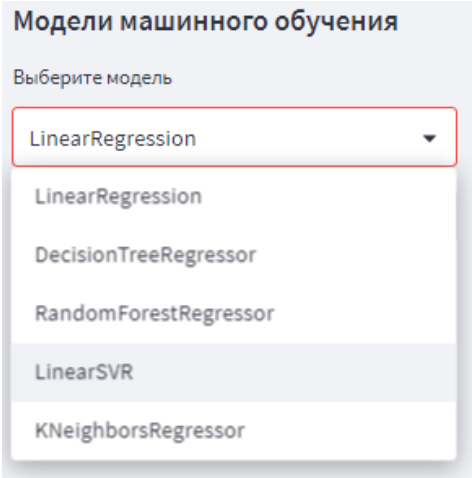


Рисунок 12 – Интерфейс выбора модели

3) получить прогнозы модуля упругости при растяжении, прочность при растяжении, соотношение матрица-наполнитель после нажатия кнопки «Получить прогнозы».

Принцип работы приложения после нажатия кнопки «Получить прогнозы» следующий:

1) формируется датасет из внесенных значений;

2) в зависимости от выбора пользователя загружаются модели для прогнозирования модуля упругости при растяжении и прочности при растяжении, и выполняется прогнозирование этих значений;

3) формируется датасет из внесенных и полученных значений из предыдущего пункта;

4) загружается модель препроцессинга данных и выполняется преобразования датасета из предыдущего пункта;

5) загружается модель нейронной сети и выполняется прогнозирование соотношения матрица-наполнитель;

6) вывод результатов работы (рисунок 13).

Рисунок 13 – Общий вид интерфейса приложения после вывода результатов

https://olvovchik-data-science-learn-applicationapp-tfgy2e.streamlit.app/ - ссылка на приложение в сети Интернет.

Заключение

Полученный коэффициент детерминации обученных моделей и нейронной сети практически нулевой – это означает, что связь между переменными регрессионной модели отсутствует и получаемые прогнозы ничем не отличаются от прогноза средним значением.

Итоговое решение поставленной задачи не достигнуто, требуется более детальный анализ данных (желательно с привлечением специалистов предметной области). Также можно попробовать использовать другие методы и модели прогнозирования, которые не были рассмотрены в данной работе.

Библиографический список

1 Библиотека Matplotlib документация [Электронный ресурс] : – Режим доступа: https://matplotlib.org/ (дата обращения 28.03.2023).

2 Библиотека Pandas документация [Электронный ресурс] : – Режим доступа: https://pandas.pydata.org/ (дата обращения 27.03.2023).

3 Библиотека Seaborn документация [Электронный ресурс] : – Режим доступа: https://seaborn.pydata.org/index.html (дата обращения 28.03.2023).

4 Библиотека Scikit-learn документация [Электронный ресурс] : – Режим доступа: https://scikit-learn.org/stable/index.html (дата обращения 01.04.2023).

5 Библиотека Scipy документация [Электронный ресурс] : – Режим доступа: https://scipy.org/ (дата обращения 31.03.2023).

6 Библиотека TensorFlow документация [Электронный ресурс] : – Режим доступа: https://www.tensorflow.org/ (дата обращения 07.04.2023).

7 Библиотека Streamlit документация [Электронный ресурс] : – Режим доступа: https://docs.streamlit.io/ (дата обращения 10.04.2023).

8 Википедия [Электронный ресурс] : – Режим доступа: https://ru.wikipedia.org/wiki/ (дата обращения 31.03.2023).

9 Гапанюк Ю. Е. Репозиторий курсов по машинному обучению [Электронный ресурс] : – Режим доступа: https://github.com/ugapanyuk/ml_course_2022 (дата обращения 28.03.2023).

10 Коротеев М. Машинное обучение обучению [Электронный ресурс] : – Режим доступа: https://koroteev.site/ml/ (дата обращения 10.04.2023).

11 Капаца Е. Машинное обучение доступным языком [Электронный ресурс] : – Режим доступа: https://www.helenkapatsa.ru/ (дата обращения 30.03.2023).

12 Метрики качества линейных регрессионных моделей [Электронный ресурс] : – Режим доступа: https://loginom.ru/blog/quality-metrics (дата обращения 02.04.2023).

13 Оптимизация гиперпараметров с помощью поиска по сетке и случайного поиска в Python [Электронный ресурс] : – Режим доступа: https://habr.com/ru/companies/otus/articles/698370/ (дата обращения 05.04.2023).

14 Умная нормализация данных [Электронный ресурс] : – Режим доступа: https://habr.com/ru/articles/527334/ (дата обращения 01.04.2023).

15 Streamlit. Поиск кратчайшего пути [Электронный ресурс] : – Режим доступа: https://habr.com/ru/articles/568836/ (дата обращения 10.04.2023).

16 Joseph Misiti Awesome Machine Learning [Электронный ресурс] : – Режим доступа: https://github.com/josephmisiti/awesome-machine-learning (дата обращения 07.04.2023).