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»

Тема: «Прогнозирование конечных свойств новых материалов (композиционных материалов)»

Слушатель Бойко Татьяна Сергеевна

Москва, 2023

Содержание

Введение

Композиционные материалы — это искусственно созданные материалы, состоящие из нескольких других с четкой границей между ними. Композиты обладают теми свойствами, которые не наблюдаются у компонентов по

отдельности. При этом композиты являются монолитным материалом, т. е. компоненты материала неотделимы друг от друга без разрушения конструкции в целом. Они могут быть созданы из различных типов материалов, таких как полимеры, металлы, керамика и углеродные материалы, и могут иметь различные формы, включая листы, волокна, маты и пены. Основным преимуществом композиционных материалов является их высокая прочность и легкость, что делает их особенно полезными в авиационной, автомобильной и аэрокосмической промышленности. Они также обладают высокой коррозионной стойкостью, устойчивостью к высоким температурам и могут иметь высокую степень гибкости и устойчивости к ударам. 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, но данный принцип сохраняется. Ниже перечислены некоторые примеры композиционных материалов:

1. Стеклопластик - это композитный материал, который состоит из стекловолокна, пропитанного эпоксидной смолой. Он обладает высокой прочностью, легкостью и стойкостью к коррозии и используется в автомобильной, судостроительной и аэрокосмической промышленности.

2. Углеродные волокна - это материалы, состоящие из тонких волокон углерода, которые связаны вместе с помощью эпоксидной смолы. Они имеют высокую прочность и легкость и широко используются в производстве автомобилей, самолетов, велосипедов и спортивных товаров.

3. Композитные пены - это материалы, состоящие из полимерных материалов, таких как полиуретан, вспененные с добавлением различных добавок, таких как алюминий или карбонат кальция. Они широко используются в производстве лодок,

плотов, ветряных турбин и других изделий, которые требуют высокой прочности и легкости.

4. Композитные материалы на основе арамида - это материалы, состоящие из волокон арамида, таких как Кевлар, связанных вместе с помощью эпоксидной смолы. Они обладают высокой прочностью и стойкостью к ударам и используются в производстве защитной одежды, бронежилетов и других изделий.

Это только несколько примеров композитных материалов, которые могут использоваться в различных отраслях промышленности.

Однако, производство композитных материалов является сложным и требует специального оборудования и навыков. 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 (включая различные методы анализа, такие как рентгеновская дифракция, электронная микроскопия и термический анализ) и прогнозирование характеристик. Суть прогнозирования заключается в симуляции представительного элемента объема композита, на основе данных о характеристиках входящих компонентов (связующего и армирующего компонента).

В прогнозировании могут использоваться компьютерное моделирование и симуляции для оценки свойств материалов на основе их структуры и компонентов. Это позволяет исследовать свойства материалов на молекулярном уровне и улучшить их характеристики.

Также может быть использовано машинное обучение и анализ данных для поиска связей между структурой материала и его свойствами, что позволяет оптимизировать процесс разработки и производства новых материалов.

Независимо от метода, прогнозирование конечных свойств новых материалов является ключевым фактором при их разработке и может привести к созданию материалов с улучшенными свойствами и более широким спектром применения.

Машинное обучение и анализ данных могут быть использованы для поиска связей между структурой композиционных материалов и их свойствами. Этот подход может помочь в определении оптимальных параметров производства и формирования структуры материала для достижения желаемых свойств.

Для этого необходимо создать модель, которая может обучиться на данных о структуре и свойствах материалов, и затем использовать эту модель для прогнозирования свойств новых материалов на основе их структуры.

Примерами методов машинного обучения, которые могут использоваться для поиска связей между структурой композиционных материалов и их свойствами, являются регрессионный анализ, глубокое обучение, методы кластеризации и методы обработки изображений.

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. Традиционно разработка композитных материалов является долгосрочным процессом, так как из свойств отдельных компонентов невозможно рассчитать конечные свойства композита. Для достижения 4 определенных

характеристик требуется большое количество различных комбинированных тестов, что делает насущной задачу прогнозирования успешного решения, снижающего затраты на разработку новых материалов.

1. Аналитическая часть

1.1. Постановка задачи

Для исследовательской работы были даны 2 файла: X_bp.xlsx (с данными о параметрах, состоящий из 1023 строк и 10 столбцов данных) и X_nup.xlsx (данными нашивок, состоящий из 1040 строк и 3 столбцов данных). Для разработки моделей по прогнозу модуля упругости при растяжении, прочности при растяжении и соотношения матрица-наполнитель нужно объединить 2 файла. Объединение по типу INNER, поэтому часть информации (17 строк таблицы X_nup.xlsx), не имеющая соответствующих строк в таблице X_bp.xlsx, будет удалена.

```
# Объединить таблицу по индексу
df = pd.merge(df1, df2, on='Unnamed: 0', how='inner')
df.head()
```

	Unnamed: 0	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	0	1.857143	2030.0	738.736842	30.00	22.267857	100.000000	210.0	70.0	3000.0	220.0
1	1	1.857143	2030.0	738.736842	50.00	23.750000	284.615385	210.0	70.0	3000.0	220.0
2	2	1.857143	2030.0	738.736842	49.90	33.000000	284.615385	210.0	70.0	3000.0	220.0
3	3	1.857143	2030.0	738.736842	129.00	21.250000	300.000000	210.0	70.0	3000.0	220.0

Рисунок 1 – пример начала работы с датасетами

Также необходимо провести разведочный анализ данных, нарисовать гистограммы распределения каждой из переменной, диаграммы boxplot (ящик с усами), попарные графики рассеяния точек.

```
df.hist(figsize = (20,20), color = "Pink")
plt.show()
```

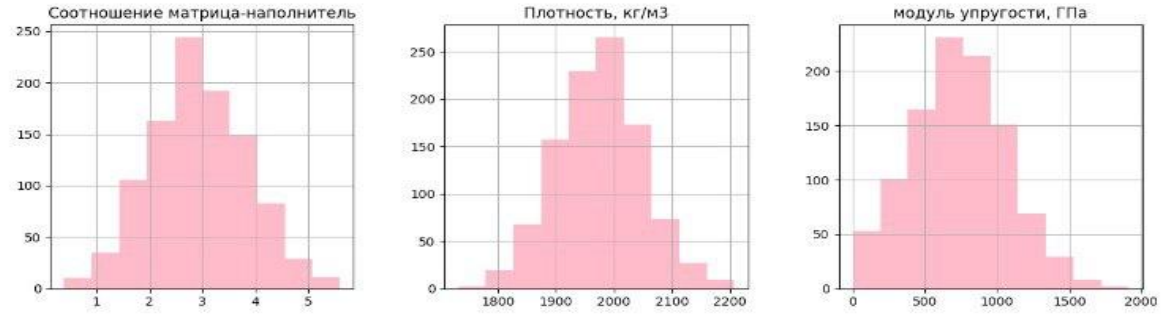

Рисунок 2 – гистограммы распределения переменных

Для каждой колонки получить среднее, медианное значение, провести анализ и исключение выбросов, проверить наличие пропусков; сделать предобработку: удалить шумы и выбросы, сделать нормализацию и стандартизацию. Обучить несколько моделей для прогноза модуля упругости при растяжении и прочности при растяжении. Написать нейронную сеть, которая будет рекомендовать соотношение матрица-наполнитель. Разработать приложение с графическим интерфейсом, которое будет выдавать прогноз соотношения «матрица-наполнитель». Оценить точность модели на тренировочном и тестовом датасете. Создать репозиторий в GitHub и разместить код исследования. Оформить файл README.

```
ax = fig2.add_subplot(133)
sns.boxplot(y = df['Плотность нашивки'], color='#F5B9FC')
```

```
<AxesSubplot: ylabel='Плотность нашивки'>
```

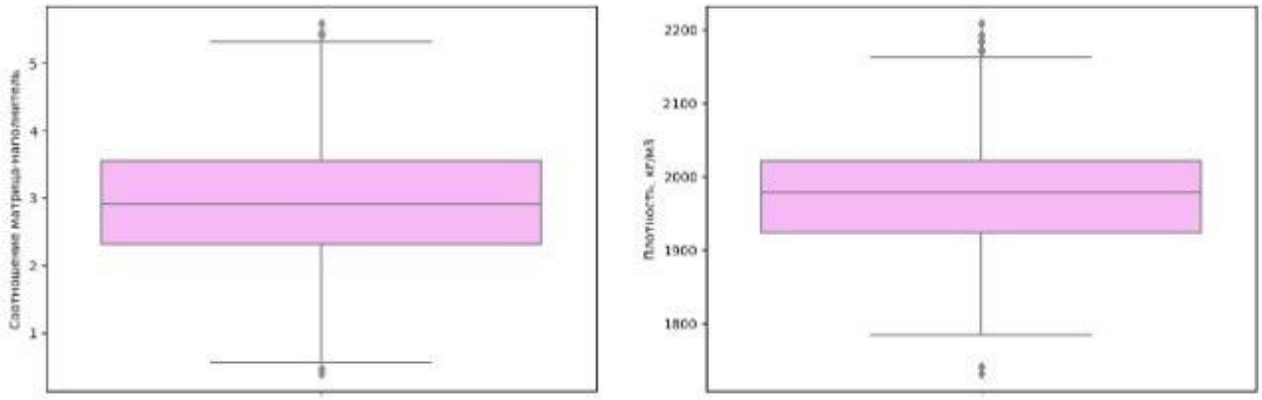

Рисунок 3 – диаграммы переменных – “ящики с усами” (Boxplot)

1.2 Описание используемых методов

Данная задача в рамках классификации категорий машинного обучения относится к машинному обучению с учителем и традиционно это задача регрессии. Цель любого алгоритма обучения с учителем — определить функцию потерь и минимизировать её, поэтому для наилучшего решения были исследованы (и некоторые из них применены) следующие методы:

− К-ближайших соседей (KNeighborsRegressor);

− дерево решений (DecisionTreeRegressor);

− линейная регрессия (Linear regression);

− стохастический градиентный спуск (SGDRegressor);

− лассо регрессия (Lasso);

− гребневая регрессия (Ridge);

− случайный лес (RandomForest);
− эластичная регрессия (ElasticNet) ;
− градиентный бустинг (GradientBoostingRegressor);
− градиентный бустинг (AdaBoostRegressor);
− метод опорных векторов (Support Vector Regression);
− многослойный перцептрон (MLPRegressor)

Линейная регрессия (Linear regression) — это алгоритм машинного обучения, основанный на контролируемом обучении, рассматривающий зависимость между одной входной и выходными переменными. Это один из самых простых и эффективных инструментов статистического моделирования. Она определяет зависимость переменных с помощью линии наилучшего соответствия. Модель регрессии создаёт несколько метрик. R2, или коэффициент детерминации, позволяет измерить, насколько модель может объяснить дисперсию данных. Если R-квадрат равен 1, это значит, что модель описывает все данные. Если же R-квадрат равен 0,5, модель объясняет лишь 50 процентов дисперсии данных. Оставшиеся отклонения не имеют объяснения. Чем ближе R2 к единице, тем лучше.

Достоинства метода: быстр и прост в реализации; легко интерпретируем, имеет меньшую сложность по сравнению с другими алгоритмами.

Недостатки метода: моделирует только прямые линейные зависимости; требует прямую связь между зависимыми и независимыми переменными; выбросы оказывают огромное влияние, а границы линейны.

Лассо регрессия (Lasso) – это линейная модель, которая оценивает разреженные коэффициенты. Это простой метод, позволяющий уменьшить сложность модели и предотвратить переопределение, которое может возникнуть в

результате простой линейной регрессии. Данный метод вводит дополнительное слагаемое регуляризации в оптимизацию модели. Это обеспечивает более устойчивое решение. В регрессии лассо добавляется условие смещения в функцию оптимизации для того, чтобы уменьшить коллинеарность и, следовательно, дисперсию модели. Но вместо квадратичного смещения, используется смещение абсолютного значения.

Достоинства метода: легко полностью избавляется от шумов в данных; быстро работает; не очень энергоемко; способно полностью убрать признак из датасета; доступно обнуляет значения коэффициентов.

Недостатки метода: часто страдает качество прогнозирования; выдает ложное срабатывание результата; случайным образом выбирает одну из коллинеарных переменных; не оценивает правильность формы взаимосвязи между независимой и зависимой переменными; не всегда лучше, чем пошаговая регрессия. Лассо-регрессию следует использовать, когда есть несколько характеристик с высокой предсказательной способностью, а остальные бесполезны. Она обнуляет бесполезные характеристики и оставляет только подмножество переменных.

Гребневая регрессия (Ridge) – это регрессия, которая добавляет дополнительный штраф к функции стоимости, но вместо этого суммирует квадраты значений коэффициентов (норма L-2) и умножает их на некоторую постоянную лямбду. По сравнению с Лассо этот штраф регуляризации уменьшит значения коэффициентов, но не сможет принудительно установить коэффициент равным 0. Это ограничивает использование регрессии гребня в отношении выбора признаков. Однако, когда $p > n$, он способен выбрать более n релевантных предикторов, если необходимо, в отличие от Лассо. Он также выберет группы коллинеарных элементов, которые его изобретатели назвали «эффектом группировки». Как и в случае с Лассо, мы можем варьировать лямбду, чтобы

получить модели с различными уровнями регуляризации, где лямбда = 0 соответствует OLS, а лямбда приближается к бесконечности, что соответствует постоянной функции. Анализ регрессии Лассо, так и Риджа показывает, что один метод не всегда лучше, чем другой; нужно попробовать оба метода, чтобы определить, какой использовать. Ридж регрессию лучше применять, когда предсказательная способность набора данных распределена между различными характеристиками. Ридж регрессия не обнуляет характеристики, которые могут быть полезны при составлении прогнозов, а просто уменьшает вес большинства переменных в модели.

Эластичная сеть (ElasticNet) – это регрессия, которая включает в себя термины регуляризации как L-1, так и L-2. Это дает преимущества регрессии Лассо и Риджа. Было установлено, что она обладает предсказательной способностью лучше, чем у Лассо, хотя все еще выполняет выбор функций. Поэтому получается лучшее из обоих методов, выполняя выбор функции Лассо с выбором группы объектов Ridge. Elastic Net поставляется с дополнительными издержками на определение двух лямбда-значений для оптимальных решений. Компромисс смещения дисперсии - это компромисс между сложной и 8 простой моделью, в которой промежуточная сложность, вероятно, является наилучшей. Лассо, Ридж-регрессия и Эластичная сеть - это модификации обычной линейной регрессии наименьших квадратов, которые используют дополнительные штрафные члены в функции стоимости, чтобы сохранить значения коэффициента небольшими и упростить модель. Лассо полезно для выбора функций, когда наш набор данных имеет функции с плохой предсказательной силой. Регрессия гребня полезна для группового эффекта, при котором коллинеарные элементы могут быть выбраны вместе. Elastic Net сочетает в себе регрессию Лассо и Риджа, что потенциально приводит к модели, которая является простой и прогнозирующей.

Градиентный бустинг (Gradient Boosting) — это ансамбль деревьев решений, обученный с использованием градиентного бустинга. В основе данного алгоритма лежит итеративное обучение деревьев решений с целью минимизировать функцию потерь. Основная идея градиентного бустинга: строятся последовательно несколько базовых классификаторов, каждый из которых как можно лучше компенсирует недостатки предыдущих. Финальный классификатор является линейной композицией этих базовых классификаторов.

Достоинства метода: новые алгоритмы учатся на ошибках предыдущих; требуется меньше итераций, чтобы приблизиться к фактическим прогнозам; наблюдения выбираются на основе ошибки; прост в настройке темпа обучения и применения; легко интерпретируем.

Недостатки метода: необходимо тщательно выбирать критерии остановки, или это может привести к переобучению, наблюдения с наибольшей ошибкой появляются чаще; слабее и менее гибок чем нейронные сети.

Метод ближайших соседей - К-ближайших соседей (kNN - k Nearest Neighbours) ищет ближайшие объекты с известными значения целевой переменной и основывается на хранении данных в памяти для сравнения с новыми элементами. Алгоритм находит расстояния между запросом и всеми 9 примерами в данных, выбирая определенное количество примеров (k), наиболее близких к запросу, затем голосует за наиболее часто встречающуюся метку (в случае задачи классификации) или усредняет метки (в случае задачи регрессии).

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

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

Дерево решений (DecisionTreeRegressor) – метод автоматического анализа больших массивов данных. Это инструмент принятия решений, в котором используется древовидная структура, подобная блок-схеме, или модель решений и всех их возможных результатов, включая результаты, затраты и полезность. Дерево принятия решений – эффективный инструмент интеллектуального анализа данных и предсказательной аналитики. Алгоритм дерева решений подпадает под категорию контролируемых алгоритмов обучения. Он работает как для непрерывных, так и для категориальных выходных переменных. Правила генерируются за счёт обобщения множества отдельных наблюдений (обучающих примеров), описывающих предметную область. Регрессия дерева решений отслеживает особенности объекта и обучает модель в структуре дерева прогнозированию данных в будущем для получения значимого непрерывного вывода. Дерево решений один из вариантов решения регрессионной задачи, в случае если зависимость в данных не имеет очевидной корреляции.

Достоинства метода: помогают визуализировать процесс принятия решения и сделать правильный выбор в ситуациях, когда результаты одного 10 решения влияют на результаты следующих решений, создаются по понятным правилам; просты в применении и интерпретации; заполняют пропуски в данных наиболее вероятным решением; работают с разными переменными; выделяют наиболее важные поля для прогнозирования.

Недостатки метода: ошибаются при классификации с большим количеством классов и небольшой обучающей выборкой; имеют нестабильный процесс (изменение в одном узле может привести к построению совсем другого дерева); имеет затратные вычисления; необходимо обращать внимание на размер; ограниченное число вариантов решения проблемы.

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

Достоинства метода: не переобучается; не требует предобработки входных данных; эффективно обрабатывает пропущенные данные, данные с большим числом классов и признаков; имеет высокую точность предсказания и внутреннюю оценку обобщающей способности модели, а также высокую параллелизуемость и масштабируемость.

Недостатки метода: построение занимает много времени; сложно интерпретируемый; не обладает возможностью экстраполяции; может недообучаться; трудоёмко прогнозируемый; иногда работает хуже, чем линейные методы.

Градиентный бустинг (AdaBoost) – это алгоритм, который работает по принципу перевзвешивания результатов. Есть деревья решений, а ансамбль из них это градиентный бустинг, задача решается с помощью градиентсного спуска. Алгоритм AdaBoost учится на ошибках, больше концентрируясь на сложных участках, с которыми от столкнулся в процессе предыдущей итерации обучения. На каждой итерации дается вес алгоритмам. Каждый новый алгоритм корректирует ошибки предыдущих до получения хорошего результата. Все

прогнозы объединяются с помощью голосования для получения окончательного прогноза.

Достоинства метода: AdaBoost легко реализовать, достаточно класса моделей и их количества. Он итеративно исправляет ошибки слабого классификатора и повышает точность путем объединения слабых учащихся. Можно использовать многие базовые классификаторы с AdaBoost. AdaBoost не склонен к переоснащению.

Недостатки метода: AdaBoost чувствителен к шумным данным. AdaBoost обучается дольше линейной регрессии, классификация дольше чем при использовании логистической регрессии. На AdaBoost сильно влияют отклонения, так как он пытается идеально подогнать каждую точку. AdaBoost работает медленнее и чуть хуже, чем XGBoost. Но легче в понимании.

Стохастический градиентный спуск (SGDRegressor) – это простой, но очень эффективный подход к подгонке линейных классификаторов и регрессоров под выпуклые функции потерь. Этот подход подразумевает корректировку весов нейронной сети, используя аппроксимацию градиента функционала, вычисленную только на одном случайном обучающем примере из выборки.

Достоинства метода: эффективен; прост в реализации; имеет множество возможностей для настройки кода; способен обучаться на избыточно больших выборках.

Недостатки метода: требует ряд гиперпараметров; чувствителен к масштабированию функций; может не сходиться или сходиться слишком медленно; функционал многоэкстремален; процесс может "застрять" в одном из локальных минимумов; возможно переобучение.

Метод опорных векторов (Support Vector Regression) – этот бинарный линейный классификатор был выбран, потому что он хорошо работает на

небольших датасетах. Данный алгоритм – это алгоритм обучения с учителем, использующихся для задач классификации и регрессионного анализа, это контролируемое обучение моделей с использованием схожих алгоритмов для анализа данных и распознавания шаблонов. Учитывая обучающую выборку, где 12 алгоритм помечает каждый объект, как принадлежащий к одной из двух категорий, строит модель, которая определяет новые наблюдения в одну из Категорий. Модель метода опорных векторов – отображение данных точками в пространстве, так что между наблюдениями отдельных категорий имеется разрыв. Каждый объект данных представляется как вектор (точка) в p-мерном пространстве. Он создаёт линию или гиперплоскость, которая разделяет данные на классы. *Достоинства метода:* для классификации достаточно небольшого набора данных. При правильной работе модели, построенной на тестовом множестве, вполне возможно применение данного метода на реальных данных. Эффективен при большом количестве гиперпараметров. Способен обрабатывать случаи, когда гиперпараметров больше, чем количество наблюдений. Существует возможность гибко настраивать разделяющую функцию. Алгоритм максимизирует разделяющую полосу, которая, как подушка безопасности, позволяет уменьшить количество ошибок классификации.

Недостатки метода: неустойчивость к шуму, поэтому в работе требуется тщательнейшая работа с выбросами, иначе в обучающих данных шумы становятся опорными объектами-нарушителями и напрямую влияют на построение разделяющей гиперплоскости; для больших наборов данных требуется долгое время обучения; достаточно сложно подбирать полезные преобразования данных; параметры модели сложно интерпретировать, поэтому были рассмотрены и другие методы.

Многослойный перцептрон (MLPRegressor) — это алгоритм обучения с учителем, который изучает функцию $f(\cdot):Rm \rightarrow Ro$ обучением на наборе данных, где m — количество измерений для ввода и o- количество размеров для вывода. Это искусственная нейронная сеть, имеющая 3 или более слоёв перцептронов. Эти слои – один входной слой, 1 или более скрытых слоев и один выходной слой перцептронов.

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

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

Используемые метрики качества моделей:

R2 (коэффициент детерминации) измеряет долю дисперсии, объясняемую моделью, в общей дисперсии целевой переменной. Если он близок к единице, то модель хорошо объясняет данные, если же он близок к нулю, то качество прогноза идентично средней величине целевой переменной (т.е. очень низкое). Отрицательные значение коэффициента детерминации означают плохую объясняющую способность модели.

MAE (Mean Absolute Error) показывает среднее абсолютное отклонение прогнозов модели от фактических значений.

Для расчета MAE сначала на каждом примере данных вычисляется абсолютная разница между прогнозом модели и фактическим значением. Затем все эти

разницы усредняются, чтобы получить единственное число, которое и будет показывать среднюю ошибку модели на данном наборе данных.

MAE определяется формулой:

*MAE = (1/n) * ∑|y_pred - y_true|,*

где y_pred - прогноз модели, y_true - фактическое значение, n - количество примеров в наборе данных.

Чем меньше значение MAE, тем лучше модель способна прогнозировать значения целевой переменной. MAE позволяет оценить точность модели в абсолютных единицах измерения целевой переменной.

MSE (Mean Squared Error) (средняя квадратичная ошибка) принимает значениях в тех же единицах, что и целевая переменная. Чем ближе к нулю MSE, тем лучше работают предсказательные качества модели.

1.3 Разведочный анализ данных

Прежде чем передать данные в работу моделей машинного обучения, необходимо обработать и очистить их. Необработанные данные могут содержать искажения и пропущенные значения и способны привести к неверным результатам.

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

```
df.describe().T
```

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

Рисунок 4 – описательная статистика датасета

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

```
# Проверим датасет на дубликаты
df.duplicated().sum()
#Дубликатов нет
```

```
0
```

Рисунок 5 - проверка датасета на наличие дубликатов

В качестве инструментов разведочного анализа используется: оценка статистических характеристик датасета; гистограммы распределения каждой из переменной; диаграммы ящика с усами; попарные графики рассеяния точек; тепловая карта; описательная статистика для каждой переменной; анализ и полное

исключение выбросов (3 повторных итерации); проверка наличия пропусков и дубликатов; ранговая корреляция Кендалла и Пирсона.

```
# сумма выбросов по каждому столбцу

df_clean = df.copy()
print(df_clean.shape)
for name in df_clean.columns:
    outlier = boxplot_stats(df_clean[name])
    High = outlier[0]['whishi']
    Low = outlier[0]['whislo']
    print ('Количество выбросов в столбце', name, ': ', len(outlier[0]['fliers']))
    df_clean = df_clean[~((df_clean[name] < Low) | (df_clean[name] > High))]
print(df_clean.shape)
```

```
(1023, 13)
Количество выбросов в столбце Соотношение матрица-наполнитель :  6
Количество выбросов в столбце Плотность, кг/м3 :  9
Количество выбросов в столбце модуль упругости, ГПа :  2
Количество выбросов в столбце Количество отвердителя, м.% :  14
Количество выбросов в столбце Содержание эпоксидных групп,%_2 :  2
Количество выбросов в столбце Температура вспышки, С_2 :  6
Количество выбросов в столбце Поверхностная плотность, г/м2 :  2
Количество выбросов в столбце Модуль упругости при растяжении, ГПа :  5
Количество выбросов в столбце Прочность при растяжении, МПа :  14
Количество выбросов в столбце Потребление смолы, г/м2 :  5
Количество выбросов в столбце Угол нашивки, град :  0
Количество выбросов в столбце Шаг нашивки :  4
Количество выбросов в столбце Плотность нашивки :  22
(932, 13)
```

Рисунок 6 – Начальное количество выбросов

Гистограммы используются для изучения распределений частот значений переменных. Мы видим очень слабую корреляцию между переменными.

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

```
# выбросы после очистки датасета
print(df_clean.shape)
plt.figure(figsize=(15, 30))
i=1
for name in df_clean.columns:
    plt.subplot(5,3,i)
    sns.boxplot(y=df_clean[name], color = 'pink')
    outlier = boxplot_stats(df_clean[name])
    print ('Количество выбросов в столбце', name, ': ', len(outlier[0]['fliers']))
    i +=1
print(df_clean.shape)
```

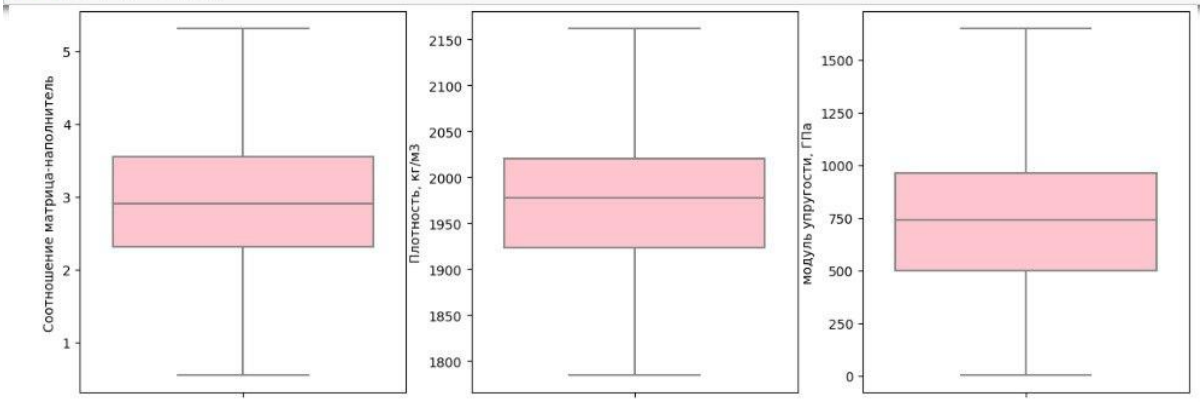

Рисунок 7 – Boxplot после очистки датасета

Данные объединенного датасета не имеют чётко выраженной зависимости, что подтверждает тепловая карта с матрицей корреляции.

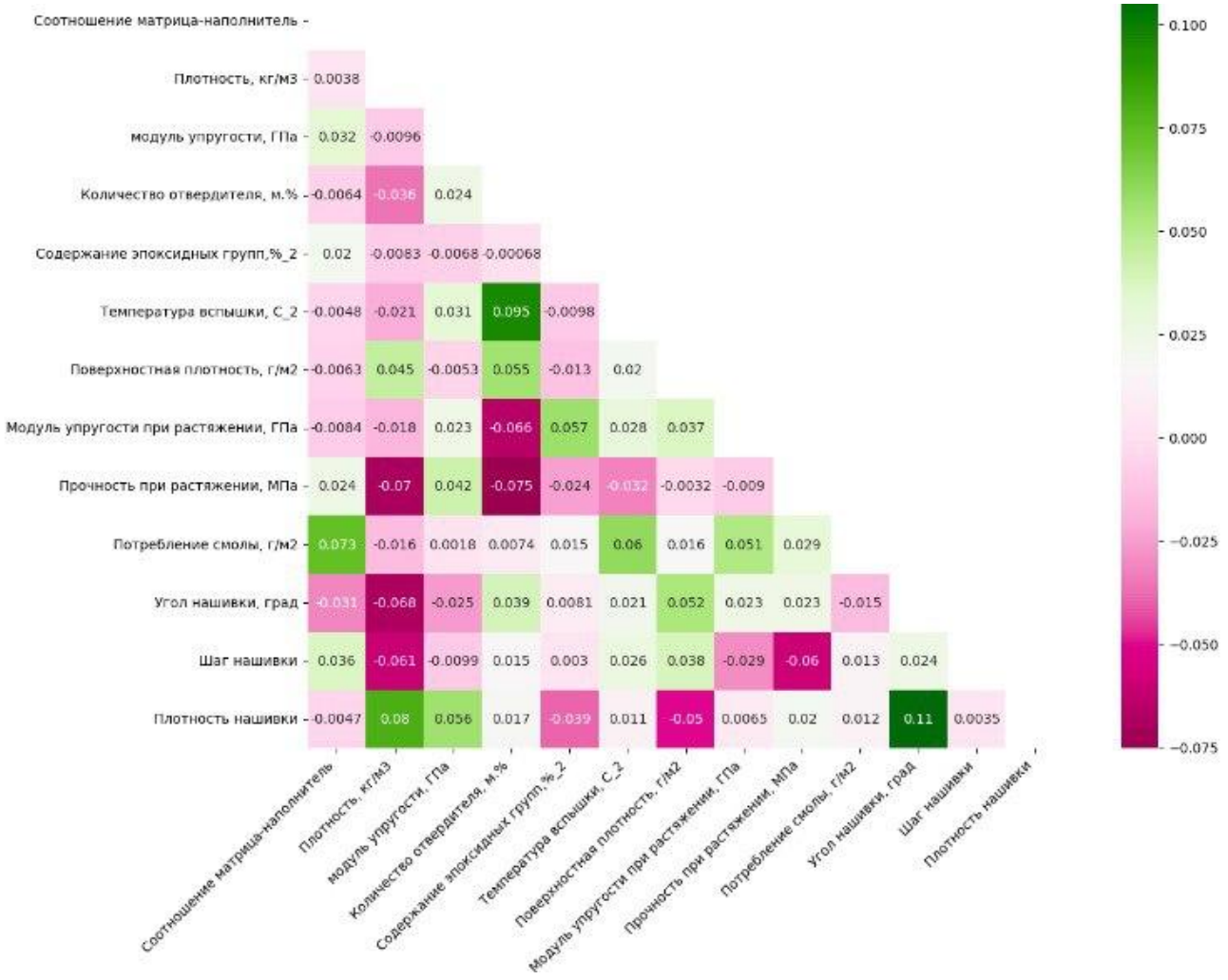

Рисунок 8 – тепловая карта с корреляцией данных

Максимальная корреляция между плотностью нашивки и углом нашивки 0.11, значит нет зависимости между этими данными. Корреляция между всеми параметрами очень близка к 0, корреляционные связи между переменными не наблюдаются.

Гистограммы показывают нормальное распределение, за исключением признака Угол нашивки, который имеет всего два значения 0 и 90 градусов. Данный столбец мы преобразуем в числа 0 и 1 с помощью кодировщика LabelEncoder.

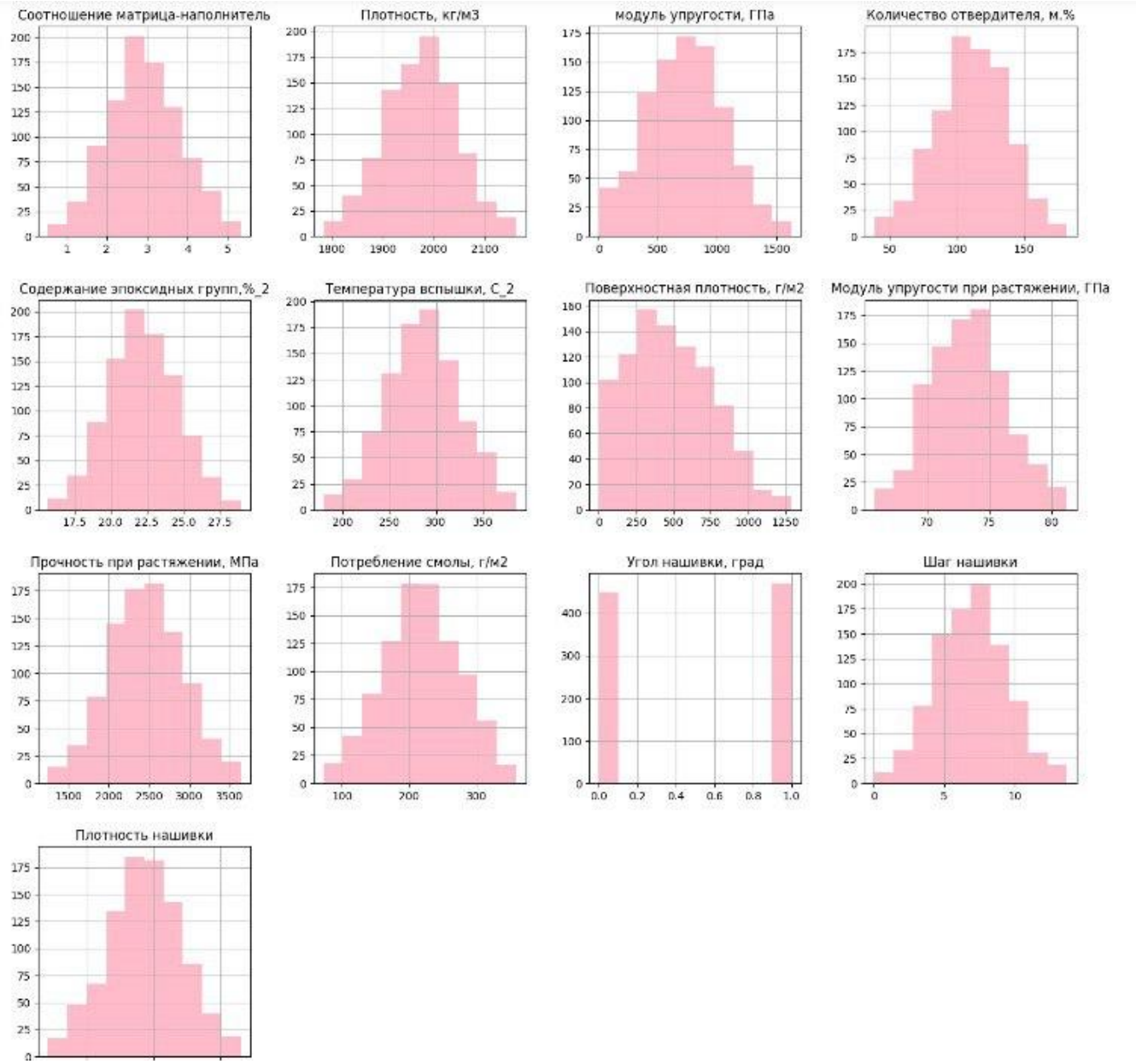

Рисунок 7 – Гистограммы распределения очищенного датасета

2. Практическая часть

2.1 Предобработка данных

По условиям задания нормализуем значения. Для этого применим MinMaxScaler().